발간등록번호: 11-B552989-000445-01 ISBN: 979-11-87770-29-9 95530

최종보고서(일반과제용)

보안등급: 일반[✔], 보안[ ]

중앙행정기관명	국토교통부	사업명	사업명	도로기술연구사업	
전문기관명	국토교통과학기술진흥원		내역사업명	도로기술연구사업	
공고번호	2018년 공고-제11호	총괄연구개발 식별번호		20POQW-C145770-03	
		연구개발과제번호		20POQW-C145770-03	
국가과학기술표준분류	1순위 EI0505	50%	2순위 EE1203	30%	3순위 ED0705	20%
총괄연구개발명 (해당 시 작성)	국문	드론을 활용한 비탈면 및 도로포장 관리 효율화 기술 개발			
	영문	Development of Drone-Based Technology for Efficient Maintenance of Cutting Slope and Pavement			
연구개발과제명	국문	드론을 활용한 비탈면 및 도로포장 관리 효율화 기술 개발			
	영문	Development of Drone-Based Technology for Efficient Maintenance of Cutting Slope and Pavement			
주관연구개발기관	기관명	한국건설기술연구원	사업자등록번호	229-82-01135	
	주소	(우) 10223 경기도 고양시 일산서구 고양대로 283	법인등록번호	111241-0002284	
연구책임자	성명	장진환	직위	연구위원	
	연락처 직장전화	031-910-0684	휴대전화	010-9816-1720	
	연락처 전자우편	jhjang@kict.re.kr	국가연구자번호	1083-3995	
연구개발기간	전 체	2018. 04. 13 - 2021. 06. 30 (39개월)			
	1년차	2018. 04. 13 - 2018. 12. 31 (9개월)			
	2년차	2019. 01. 01 - 2019. 12. 31 (12개월)			
	3년차	2020. 01. 01 - 2021. 06. 30 (18개월)			

연구개발비 (단위: 천원)	정부지원 연구개발비	기관부담 연구개발비		그 외 기관 등의 지원금				합계			연구개발비 외 지원금
				지방자치단체		기타( )					
	현금	현금	현물	현금	현물	현금	현물	현금	현물	합계	
총 계	4,309,000	153,800	1,284,200					4,462,800	1,284,200	5,747,000	
1년차	961,000	34,600	286,400					995,600	286,400	1,282,000	
2년차	1,435,000	50,400	428,600					1,485,400	428,600	1,914,000	
3년차	1,913,000	68,800	569,200					1,981,800	569,200	2,551,000	

공동연구개발기관 등	기관명	책임자	직위	휴대전화	전자우편	비고	
						역할	기관유형
공동연구개발기관	부산대	홍상훈	조교수	010-6250-5207	geoshong@pusan.ac.kr	공동	대학
	공간정보	허정주	부장	010-2619-8252	gpskorea@hanmail.net	공동	중소기업
	신한항업	배경호	연구소장	010-3564-4751	qpandora@daum.net	공동	중소기업
	메타빌드	박상율	수석연구원	010-8947-0388	yullysis@metabuild.co.kr	공동	중소기업
	아와소프트	장재호	부사장	010-6426-8750	cjhaws@awasoft.co.kr	공동	중소기업
	롯데건설	이석진	수석부장	010-5276-9127	stjiny@lotte.net	공동	대기업
	대구가톨릭대	서동만	교수	010-4053-5447	sarum@cu.ac.kr	공동	대학
위탁연구개발기관	케이알엠	이상국	이사	010-4707-0215	kroadm@kroadm.kr	위탁	중소기업

연구개발담당자 실무담당자	성명	박근형	직위	수석연구원
	연락처 직장전화	0312-910-0439	휴대전화	010-2988-6504
	연락처 전자우편	godpark@kict.re.kr	국가연구자번호	1102-7187

이 최종보고서에 기재된 내용이 사실임을 확인하며, 만약 사실이 아닌 경우 관련 법령 및 규정에 따라 제재처분 등의 불이익도 감수하겠습니다.

2021년 10월 12일

연구책임자: 장 진 환 (인)

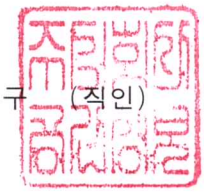
주관연구개발기관의 장: 김 병 석 (직인)

국토교통부장관 귀하

공동연구개발기관의 장: 최 경 민 (직인)



공동연구개발기관의 장: 김 석 구 (직인)



공동연구개발기관의 장: 배 상 태 (직인)



공동연구개발기관의 장: 조 풍 연 (직인)



공동연구개발기관의 장: 김 양 유 (직인)



공동연구개발기관의 장: 하 석 주 (직인)



공동연구개발기관의 장: 김 기 홍 (직인)



위탁연구개발기관의 장: 김 부 일 (직인)



[요 약 문]

<table>
<tr><td colspan="2">사업명</td><td colspan="3">도로기술연구사업</td><td colspan="2">총괄연구개발 식별번호</td><td colspan="2">-</td></tr>
<tr><td colspan="2">내역사업명</td><td colspan="3"></td><td colspan="2">연구개발과제번호</td><td colspan="2"></td></tr>
<tr><td colspan="2">국가과학기술
표준분류</td><td>1순위 EI0505</td><td>50%</td><td>2순위 EE1203</td><td>30%</td><td>3순위 ED0705</td><td>20%</td></tr>
<tr><td colspan="2">총괄연구개발명</td><td colspan="6">-</td></tr>
<tr><td colspan="2">연구개발과제명</td><td colspan="6">드론을 활용한 비탈면 및 도로포장 관리 효율화 기술 개발</td></tr>
<tr><td colspan="2">전체 연구개발기간</td><td colspan="6">2018. 04. 13 - 2021. 06. 30 (39개월)</td></tr>
<tr><td colspan="2">총 연구개발비</td><td colspan="6">총 5,747,000천원
(정부지원연구개발비: 4,309,000천원, 기관부담연구개발비: 1,438,000천원)</td></tr>
<tr><td colspan="2">연구개발단계</td><td colspan="3">기초[ ] 응용[ ] 개발[✔]
기타(위 3가지에 해당되지 않는 경우)[ ]</td><td>기술성숙도</td><td colspan="2">착수시점 기준(○)
종료시점 목표( )</td></tr>
<tr><td colspan="2">연구개발과제 유형</td><td colspan="6">지정공모 [✔] 자유공모 [ ]</td></tr>
<tr><td colspan="2">연구개발과제 특성</td><td colspan="6">실용화</td></tr>
<tr><td rowspan="2">연구
개발
목표
및
내용</td><td>최종 목표</td><td colspan="6">비탈면 및 포장관리 효율성 극대화를 위한 드론 장착 센서(SAR-L, 영상, 초분광) 데이터 분석 기반의 비탈면 변위(變位), 포트홀(Pothole), 포장상태지수(PCI) 측정 기술 개발</td></tr>
<tr><td>전체 내용</td><td colspan="6">○ 드론을 활용한 비탈면 변위 측정 기술 개발
- 드론 탑재 SAR 센서 기반 비탈면 변위 측정 기술 개발
· 드론 촬영 비탈면 SAR 영상 생성 기술 개발
· 드론 촬영 SAR 영상 기반 비탈면 변위 측정 기술 개발
- 시계열 자료 해석 및 GIS 맵 기반 위험 비탈면 가시화 기술 개발
· 시계열 SAR 자료 수집/처리분석 기술(알고리즘) 및 정밀변위 해석기술 개발
· DSM 기반 위험 비탈면 가시화 기술 개발
· 드론 SAR 영상을 이용한 비탈면 관리 효율화 방법론 개발
- 드론 탑재 SAR 센서 기반 비탈면 변위측정 및 가시화 평가/검증 기술 개발

○ 드론을 활용한 포트홀 탐지 기술 개발
- 드론 탑재 영상 센서 기반 포트홀 탐지 기술 개발
· 드론 촬영 영상 분석 기반 포트홀 객체 인지 · 추출 기술 개발
· 드론 촬영 영상 분석 기반 포트홀 면적 산출 기술 개발
- GIS 맵 기반 포트홀 가시화 기술 개발
· GIS 맵 기반 드론 측정 포트홀 데이터 가시화 기술 개발
· 드론 측정 데이터 기반 포트홀 긴급보수 효율화 방법론 개발
- 드론 탑재 영상 센서 포트홀 탐지 및 가시화 결과 평가/검증 기술 개발
※ 포트홀 : 직경 15㎝ 이상(FHWA 기준)

○ 드론을 활용한 포장 노후화 상태(PCI) 측정 기술 개발
- 드론 수집 초분광 센서 데이터 분석 기반 PCI 추정 기술 개발
· 드론 수집 초분광 데이터 처리 및 분석 기술 개발
· 드론 수집 초분광 데이터와 PCI 기준값 매칭 라이브러리 구축
· 드론 수집 초분광 데이터 분석 기반 PCI 정보 수집 기술 개발
- PMS 맵 기반 PCI 측정 데이터 가시화 기술 개발
· 포장관리시스템(PMS) 및 이동형매핑시스템(MMS) 맵 기반 드론측정 PCI 정보 가시화 기술 개발
· 드론 수집 PCI 정보를 활용한 포장조사 효율화 방법론 개발</td></tr>
</table>

			- 포장 노후화 상태(PCI) 측정 및 가시화 결과 평가/검증 기술 개발
	1년차	목표	○ 비탈면 변위 측정 알고리즘 개발 ○ 포트홀 탐지 기계학습 알고리즘 개발 ○ PCI 추정 방법론 정립
		내용	○ 드론 탑재 SAR 센서 기반 비탈면 변위 측정 기술 개발 - 드론 촬영 비탈면 SAR 영상 생성 기술 개발 - 드론 촬영 SAR 영상 기반 비탈면 변위 측정 기술 개발 ○ 드론 탑재 영상 센서 기반 포트홀 탐지 기술 개발 - 드론 촬영 영상 분석 기반 포트홀 및 균열 객체 인지·추출 기술 개발(학습 알고리즘 개발) ○ 드론 수집 초분광 센서 데이터 분석 기반 PCI 추정 기술 개발 - 드론으로부터 수집된 초분광 데이터 처리 및 분석 기술 개발 - 드론으로부터 수집된 초분광 데이터로부터 최적의 밴드선정 알고리즘 개발 - 드론으로부터 수집된 초분광 데이터 분석 기반 PCI 정보 수집 기술 개발
	2년차	목표	○ 비탈면 변위측정 알고리즘 구현 ○ 포트홀 탐지 알고리즘 구현 ○ PCI 수집장치 제작 및 SW 개발
		내용	○ 시계열 자료 해석기술 개발 - 시계열 SAR 자료 수집/ 처리분석 기술(알고리즘) 및 정밀 변위 해석 기술 개발 ○ GIS 맵 기반 위험 비탈면 가시화 기술 개발 ○ 드론 탑재 영상 센서 기반 포트홀 탐지 기술 개발 - 드론으로 촬영된 영상 분석 기반 포트홀 탐지 및 면적 산출 기술 개발 ○ 드론 수집 초분광 센서 데이터 분석 기반 PCI 추정 기술 개발 - 드론으로부터 수집된 초분광 데이터와 PCI 기준값 매칭 라이브러리 구축
	3년차	목표	○ 실 현장 테스트 ○ 개발기술 정량적 평가 ○ 개발기술 평가 및 활용지침 마련
		내용	○ 드론 탑재 SAR 센서 기반 비탈면 변위 측정 및 가시화 결과 평가/ 검증 기술 개발 ○ 드론 SAR 영상 이용한 비탈면 관리 효율화 매뉴얼 마련 ○ GIS 맵기반 포트홀 가시화 기술 개발 - GIS 맵기반 드론 측정 포트홀 데이터 가시화 기술 개발 - 드론 측정 데이터 기반 포트홀 긴급 보수 효율화 매뉴얼 마련 ○ 드론 탑재 영상 센서 포트홀 탐지 및 가시화 결과 평가/검증 기술 개발 ○ PMS 및 MMS 맵 기반 PCI 측정 데이터 가시화 기술 개발 - 포장관리시스템(PMS) 및 이동형매핑시스템(MMS) 맵 기반 드론 측정 PCI 정보 가시화 기술 개발 - 드론 수집 PCI 정보를 활용한 포장조사 효율화 매뉴얼 개발 ○ 포장 노후화 상태(PCI) 측정 및 가시화 결과 평가/검증 기술
연구개발성과			○ SAR 센서 기반 비탈면 변위 측정 드론시스템 - 비탈면 변위 측정을 위한 SAR(L 밴드) 센서 장착 드론 - 드론 SAR 데이터 기반 비탈면 변위 측정 소프트웨어 - 비탈면 변위 가시화 시스템 및 활용 지침 - 비탈면 변위 측정 및 가시화 결과 평가/검증 보고서 - 성능목표 : 변위탐지 정확도 5㎝ 이하, 변위위치 가시화 정확도 1m 이내 ○ 영상 센서 기반 포트홀 사전/사후 탐지 드론시스템

	- 포트홀 탐지를 위한 영상 센서 장착 드론 - 드론 영상 데이터 기반 포트홀 측정 소프트웨어 - 포트홀 위치 가시화 시스템 및 활용 지침 - 포트홀 탐지 및 가시화 결과 평가/검증 보고서 - 성능목표 : 검지율 80% 이상, 오검지율 10% 이하, 포트홀 위치 가시화 정확도 1m 이내 ○ 초분광 센서 기반 포장상태지수(PCI) 측정 드론시스템 - PCI 측정을 위한 초분광 센서 장착 드론 - 초분광 데이터 기반 PCI 추정 소프트웨어 - PCI 가시화 시스템 및 활용 지침 - 포장상태지수(PCI) 측정 및 가시화 결과 평가/검증 보고서 - 성능목표 : 기준값 대비 PCI 측정 상관계수 0.7, PCI 측정위치 가시화 정확도 1m 이내
연구개발성과 활용방안 및 기대효과	○ 활용 계획 - 본 연구는 광범위한 지역에 산재한 도로 비탈면 및 도로포장 조사를 드론을 활용하여 효율적으로 수행하는 기술을 개발하는 것으로써, 개발기술은 고속도로, 국도 등 비탈면(CSMS) 및 포장관리시스템(PMS)에 즉시 적용이 가능함 - 도로건설에 따른 비탈면 공사시 공정 단계별로 비탈면 변위를 수시로 점검해야 하지만, 고정식 장비 설치의 어려움 등으로 정밀 변위 측정의 어려움이 발생함. 본 연구 개발 드론 기반 비탈면 변위 측정시스템은 이러한 비탈면 공사시 공정 단계별 정밀변위를 측정하는 용도로 활용이 가능함 - 비탈면 변위 측정 기술은 CSMS뿐만 아니라 산사태 위험지역, 주택가 인근 급경사지 등에도 적용 가능하여 우면산 산사태 등을 예방할 수 있는 기술이고, 포장조사 기술은 도로뿐만 아니라 공항, 항만 등 포장조사에도 활용 가치가 높은 기술임 - 현행 조사차량을 이용한 포장조사는 많은 비용이 소요됨에 따라 지방도 이하 도로에는 조사가 현실적으로 어려운 실정이지만, 드론 기반 조사기술 개발시 저비용으로 조사가 가능함에 따라 기존에 할 수 없었던 지방도 이하 도로의 포장조사가 가능함 ○ 기대효과 - (비용 절감) 비탈면 변위측정 기술은 기존 비탈면 변위측정을 위한 상시계측시스템(1.5억원/개소) 대체기술로 적용 가능하고, 포장 노후화 상태 조사 기술은 기존 포장조사 대비 1/4 수준으로 조사시간 감소가 가능함에 따라 SOC 유지관리예산 절감 가능(편익/비용 = 2.25 : 경제성 확보) - (성능 개선) 기존 비탈면 상시계측시스템은 짐승, 바람 등 영향으로 오경보 발생이 많았지만, 드론 기반 기술의 경우 이러한 문제점 해결이 가능하고, 기존에는 포트홀 조사체계 부재로 인해 포트홀에 따른 교통사고, 민원 등이 잦았지만, 본 연구 개발기술을 적용할 경우 위험한 포트홀은 우선적으로 복구할 수 있기 때문에 이러한 문제점 완화 가능 - (첨단 건설산업 활성화) 본 연구에서 개발하고자 하는 기술은 효율적 SOC 유지관리를 위한 드론 및 드론센서 데이터 처리·해석 분야 기술 분야임에 따라 전·후방 연관 산업 파급효과가 막대하고, 그에 따른 첨단 건설 산업 분야 양질의 일자리 창출 가능
연구개발성과의 비공개여부 및 사유	해당없음

연구개발성과의 등록·기탁 건수	논문	특허	보고서 원문	연구시설·장비	기술요약정보	소프트웨어	표준	생명자원		화합물	신품종	
								생명정보	생물자원		정보	실물
	7	16				33						

연구시설·장비 종합정보시스템 등록 현황	구입기관	연구시설·장비명	규격 (모델명)	수량	구입 연월일	구입가격 (천원)	구입처 (전화)	비고 (설치장소)	ZEUS 등록번호

국문핵심어 (5개 이내)	드론	비탈면 변위	포트홀	포장상태지수
영문핵심어	Drone	Cutting slope displacement	Pothole	Pavement condition index

(5개 이내)				

〈 목 차 〉

1. 연구개발과제의 개요 ····· 1

2. 연구개발과제의 수행 과정 및 수행 내용 ····· 17

3. 연구개발과제의 수행 결과 및 목표 달성 정도 ····· 254

4. 연구개발성과의 관련 분야에 대한 기여 정도 ····· 329

5. 연구개발성과의 관리 및 활용 계획 ····· 333

부록 ····· 346

1. 연구개발과제의 개요

가. 최종목표 및 연차별 목표

(1) 연구개발과제 최종목표

<table>
<tr><th>구분</th><th>내용</th></tr>
<tr><td>최종목표</td><td>
<table>
<tr><td>비전</td><td>드론을 활용한 비탈면 및 포장관리 효율성 극대화</td></tr>
<tr><td>목표</td><td>드론 장착 센서(SAR-L, 영상, 초분광) 데이터 처리기술 개발을 통한 활용한 비탈면 변위(變位), 포트홀(Pothole), 포장상태지수(PCI) 측정 기술 개발</td></tr>
<tr><td>추진전략</td><td>기관별 업무특성을 고려한 최적의 컨소시엄 구성
* 출연연(설계, 평가)+학(알고리즘)+산(제작, 구현)
* 기관 보유 연구인프라 적극 활용
⊕
위험요인 파악 및 효율적 연구개발 추진전략 수립
* 국내현황 고려한 적정센서 선정 및 연구데이터 확보전략 수립
* 연구성과 세부 활용지침 마련</td></tr>
</table>
</td></tr>
<tr><td>성과물</td><td>
<table>
<tr><td>1. SAR(L밴드) 센서 기반 비탈면 변위측정 드론 시스템</td><td>- 비탈면 변위 측정을 위한 SAR(L 밴드) 센서 장착 드론
- 드론 SAR 데이터 기반 비탈면 변위 측정 소프트웨어
- 비탈면 변위 가시화 시스템 및 활용 지침
- 비탈면 변위 측정 및 가시화 결과 평가/검증 보고서</td></tr>
<tr><td>2. 영상센서 기반 포트홀 사전/사후 탐지 드론 시스템</td><td>- 포트홀 탐지를 위한 영상 센서 장착 드론
- 드론 영상 데이터 기반 포트홀 측정 소프트웨어
- 포트홀 위치 가시화 시스템 및 활용 지침
- 포트홀 탐지 및 가시화 결과 평가/검증 보고서</td></tr>
<tr><td>3. 초분광 센서 기반 포장상태지수(PCI) 측정 드론시스템</td><td>- PCI 측정을 위한 초분광 센서 장착 드론
- 초분광 데이터 기반 PCI 추정 소프트웨어
- PCI 가시화 시스템 및 활용 지침
- 포장상태지수(PCI) 측정 및 가시화 결과 평가/검증 보고서</td></tr>
</table>
</td></tr>
</table>



성능목표

성과물	성능항목	단위	현재 성능수준		개발 목표치				평가방법
			국외	국내	RFP 목표치	제안 목표치			
						'18	'19	'21	
SAR(L밴드) 센서 기반 비탈면 변위측정 드론시스템	변위 측정 정확도 (Precision)	cm	-	-	5	15	10	5	광파, 지상라이다 등 측량 데이터와 비교 분석을 통한 평가 수행
	비탈면 변위위치 가시화 정확도 (Accuracy)	m	-	-	2	5	3	1	
영상센서 기반 포트홀 사전/사후 탐지 드론시스템	포트홀 검지율 (Detection Rate)	%	60 (ATECH)	-	80	60	70	80	전문가 비디오 판독결과와 비교 분석을 통한 평가 수행
	포트홀 오검지율 (False Alarm Rate)	%	-	-	10	30	20	10	
	포트홀 위치 가시화 위치 정확도 (Accuracy)	m	-	-	2	2	1	0.5	정밀 측위(DGPS) 데이터와 비교 분석을 통한 평가 수행
초분광 센서 기반 포장상태지수(PCI) 측정 드론시스템	기준 값 대비 PCI 측정 상관계수 (Accuracy)	r	0.6 (Iowa State Univ.)	-	0.7	0.5	0.6	0.7	포장조사차량(PES)을 이용하여 수집한 기준값과 비교 분석을 통한 평가 수행
	PCI 측정위치 가시화 정확도 (Accuracy)	m	-	-	2	2	1	0.5	정밀 측위(DGPS) 데이터와 비교 분석을 통한 평가 수행

연차별 추진내용

성과물		SAR(L밴드) 센서 기반 비탈면 변위측정 드론시스템	영상센서 기반 포트홀 사전/사후 탐지 드론시스템	초분광 센서 기반 PCI 측정 드론시스템	비고
연차별 연구내용	1차 년도	○ 드론 탑재 SAR 센서 기반 비탈면 변위 측정 기술 개발 - 드론 촬영 비탈면 SAR 영상 생성 기술 개발 - 드론 촬영 SAR 영상 기반 비탈면 변위 측정 기술 개발	○ 드론 탑재 영상 센서 기반 포트홀 탐지 기술 개발 - 드론 촬영 영상 분석 기반 포트홀 및 균열 객체 인지·추출 기술 개발(학습 알고리즘 개발)	○ 드론 수집 초분광 센서 데이터 분석 기반 PCI 추정 기술 개발 - 드론으로부터 수집된 초분광 데이터 처리 및 분석 기술 개발 - 드론으로부터 수집된 초분광 데이터로부터 최적의 밴드선정을 위한 알고리즘 개발 - 드론으로부터 수집된 초분광 데이터 분석 기반 PCI 정보 수집 기술 개발	요소 기술
	2차 년도	○ 시계열 자료 해석기술 개발 - 시계열 SAR 자료 수집/처리분석 기술(알고리즘) 및 정밀 변위 해석 기술 개발 ○ GIS 맵 기반 위험 비탈면 가시화 기술 개발	○ 드론 탑재 영상 센서 기반 포트홀 탐지 기술 개발 - 드론으로 촬영된 영상 분석 기반 포트홀 탐지 및 면적 산출 기술 개발	○ 드론 수집 초분광 센서 데이터 분석 기반 PCI 추정 기술 개발 - 드론으로부터 수집된 초분광 데이터와 PCI 기준값 매칭 라이브러리 구축	통합 기술

	3차 년도	○ 드론 탑재 SAR 센서 기반 비탈면 변위 측정 및 가시화 결과 평가/검증 기술 개발 ○ 드론 SAR 영상 이용한 비탈면 관리 효율화 매뉴얼 마련	○ GIS 맵기반 포트홀 가시화 기술 개발 – GIS 맵기반 드론 측정 포트홀 데이터 가시화 기술 개발 – 드론 측정 데이터 기반 포트홀 긴급 보수 효율화 매뉴얼 마련 ○ 드론 탑재 영상 센서 포트홀 탐지 및 가시화 결과 평가/검증 기술 개발	○ PMS 및 MMS 맵 기반 PCI 측정 데이터 가시화 기술 개발 – 포장관리시스템(PMS) 및 이동형매핑시스템(MMS) 맵 기반 드론 측정 PCI 정보 가시화 기술 개발 – 드론 수집 PCI 정보를 활용한 포장조사 효율화 매뉴얼 개발 ○ 포장 노후화 상태(PCI) 측정 및 가시화 결과 평가/검증 기술	실용화 기술
테스트베드 구축목표 (3차년도)		– 실 도로 비탈면(경기 남양주 조안면 삼봉리,롯데건설 시공 중) 현장 테스트	– 연천 SOC 테스트베드 내 실증시험주로 포트홀 탐지/PCI 계측 현장 실험 – 실 도로 현장(수원국토관리사무소 관내 일반국도 안성~감곡 IC, 40km) 테스트		테스트 베드



(2) 연구개발과제 연차별 목표

구분	연구개발목표	핵심 연구개발성과 (연구개발성과물)	현재 국내・외 기술수준	연구개발 목표치	목표치 설정 근거	가중치 (%)
1년차 (2018)	○ 드론 탑재 SAR 센서 기반 비탈면 변위 측정 기술 개발	- 드론 촬영 비탈면 SAR 영상 생성 기술 - 드론 촬영 SAR 영상 기반 비탈면 변위 측정 기술	- 국내외 기술개발 사례 없음	- 변위측정 정확도 15cm	- 급경사지 관리기준(행안부) 상 주민대피 기준 적용	20
	○ 드론 탑재 영상 센서 기반 포트홀 탐지 기술 개발	- 드론 촬영 영상 분석 기반 포트홀 및 균열 객체 인지·추출 기술 (학습 알고리즘 개발)	- 포트홀 검지율 60%(미국)	- 포트홀 검지율 60%	- 국내외 최고 기술수준 기준	10
	○ 드론 수집 초분광 센서 데이터 분석 기반 PCI 추정 기술 개발	- 드론으로부터 수집된 초분광 데이터 처리 및 분석 기술 - 드론으로부터 수집된 초분광 데이터로부터 최적의 밴드선정을 위한 알고리즘 - 드론으로부터 수집된 초분광 데이터 분석 기반 PCI 정보 수집 기술	- 기준 값 대비 PCI 측정 상관계수 0.7	- 기준 값 대비 PCI 측정 상관계수 0.5	- 국내외 최고 기술수준 기준	10
2년차 (2019)	○ 비탈면 변위측정을 위한 SAR-L 시계열 자료 해석기술 개발	- 시계열 SAR 자료 수집/처리분석 기술(알고리즘) 및 정밀 변위 해석 기술	- 국내외 기술개발 사례 없음	- 변위측정 정확도 15cm	- 급경사지 관리기준(행안부) 상 주민대피 기준 적용	20
	○ 드론 탑재 영상 센서 기반 포트홀 탐지 기술 개발	- 드론으로 촬영된 영상 분석 기반 포트홀 탐지 및 면적 산출 기술	- 포트홀 검지율 60%(미국)	- 포트홀 검지율 70%	- 국내외 최고 기술수준 기준	10
	○ 드론 수집 초분광 센서 데이터 분석 기반 PCI 추정 기술 개발	- 드론으로부터 수집된 초분광 데이터와 PCI 기준값 매칭 라이브러리 구축	- 기준 값 대비 PCI 측정 상관계수 0.7	- 기준 값 대비 PCI 측정 상관계수 0.6	- 국내외 최고 기술수준 기준	10
3년차 (2020)	○ 드론 탑재 SAR 센서 기반 비탈면 변위 측정 및 가시화 결과 평가/검증 기술 개발	- 드론 SAR 영상 이용한 비탈면 관리 효율화 매뉴얼	- 국내외 기술개발 사례 없음	- 변위측정 정확도 15cm	- 급경사지 관리기준(행안부) 상 주민대피 기준 적용	30
	○ 드론 탑재 영상 센서 포트홀 탐지 및 가시화 결과 평가/검증 기술 개발	- GIS 맵기반 드론 측정 포트홀 데이터 가시화 기술 - 드론 측정 데이터 기반 포트홀 긴급 보수 효율화 매뉴얼	- 포트홀 검지율 60%(미국)	- 포트홀 검지율 80%	- 국내외 최고 기술수준 기준	10
	○ 포장 노후화 상태(PCI) 측정 및 가시화 결과 평가/검증 기술	- 포장관리시스템(PMS) 및 이동형 매핑시스템(MMS) 맵 기반 드론 측정 PCI 정보 가시화 기술 - 드론 수집 PCI 정보를 활용한 포장조사 효율화 매뉴얼	- 기준 값 대비 PCI 측정 상관계수 0.7	- 기준 값 대비 PCI 측정 상관계수 0.7	- 국내외 최고 기술수준 기준	10

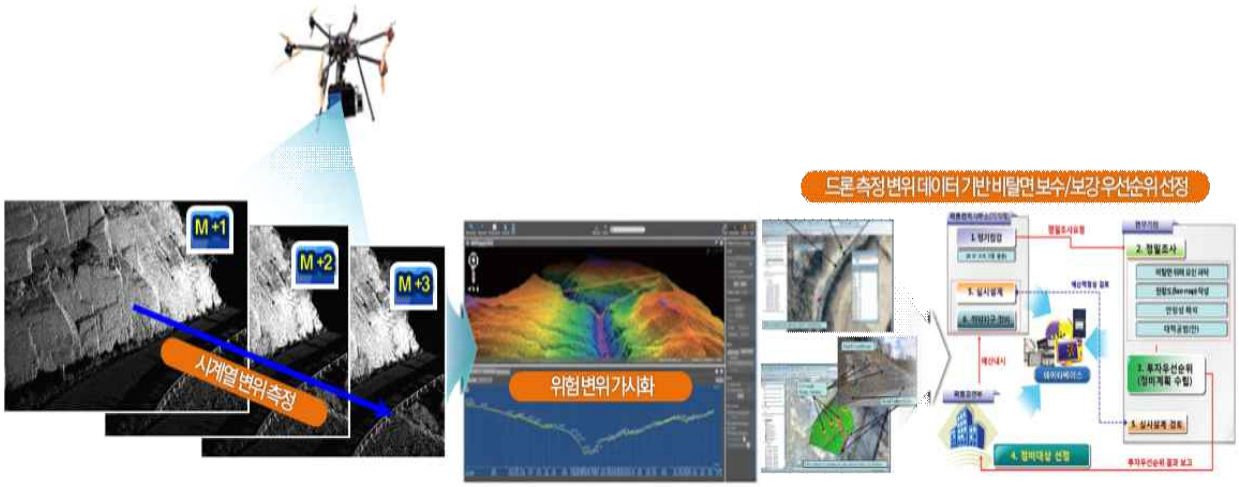
나. 연구개발 개요

(1) 기술 개요

○ 본 연구는 광범위한 지역에 분포해 있어 매년 전국적으로 수백억 원의 비용이 소요되는 도로 비탈면 및 포장 조사 효율성 향상을 위해 드론 기반의 측정기술을 개발하는 것으로 세부 연구내용은 ① 드론 센서(SAR-L) 데이터 분석 기반 비탈면 정밀변위 측정기술, ② 드론 영상 데이터 분석 기반 포트홀 자동탐지기술, ③ 드론 초분광 데이터 분석 기반 포장 노후화 상태(PCI) 측정기술로 구성됨

○ 드론 센서 데이터 분석 기반 비탈면 정밀변위 측정기술은 드론으로 촬영한 SAR(L 밴드) 영상의 시계열 프로파일링을 통해 변위를 측정하고, 측정된 변위 데이터를 해석하여 붕괴 위험성이 있는 비탈면을 가시화한 후, 이를 비탈면 보수·보강 우선순위 선정에 활용하는 기술을 의미함

- 드론 촬영 비탈면 SAR 영상 생성 및 비탈면 변위측정을 위한 드론 자동운항 기술 개발
- 드론 촬영 SAR 영상 기반 비탈면(지표면) 변위 측정 기술 개발
- 시계열 SAR 자료 수집·처리분석 기술(알고리즘) 및 정밀 변위 해석 기술 개발
- GIS 맵 기반 위험 비탈면 가시화 기술 개발
- 드론 탑재 SAR 센서 기반 비탈면 변위 측정 및 가시화 결과 평가/검증 기술 개발



[드론 SAR-L 데이터 분석 기반 비탈면 정밀변위 측정기술]

○ 드론 수집 영상 데이터 분석 기반 포트홀 자동탐지기술은 영상처리 및 CNN(Convolution Neural Network) 알고리즘을 활용하여 포트홀 위치 및 크기를 측정하고, 이를 활용하여 포트홀을 신속보수하기 위한 가시화 기술로 구성됨

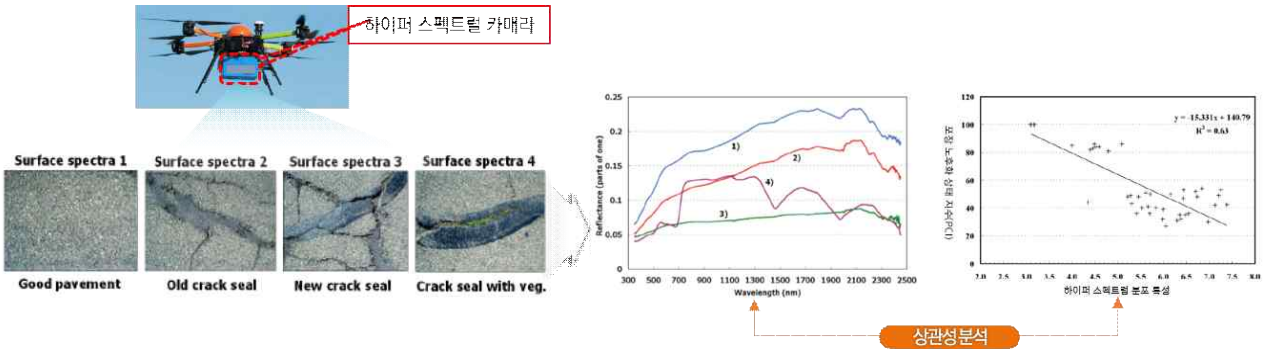
- 드론 촬영 영상 분석 기반 포트홀 및 균열 객체 인지·추출 기술 개발
- 드론 촬영 영상 분석 기반 포트홀 면적 산출 기술 개발
- GIS 맵 기반 드론 측정 포트홀 데이터 가시화 기술 개발
- 드론 탑재 영상 센서 포트홀 탐지 및 가시화 결과 평가/검증 기술 개발



[드론 수집 영상 데이터 분석 기반 포트홀 자동탐지기술]

○ 드론 수집 초분광 데이터 분석 기반 포장 노후화 상태 측정기술은 포장상태 측정용 하이퍼 스펙트럼(초분광) 카메라를 개발하여 도로포장 하이퍼 스펙트럼 데이터를 수집 후 수집된 하이퍼 스펙트럼 분포 특성 분석을 통한 포장상태지수(PCI)를 추정하는 기술을 의미함

- 드론 수집 초분광 데이터 처리 및 분석 기술 개발
- 드론 수집 초분광 데이터와 PCI 기준값 매칭 라이브러리 구축
- 드론 수집 초분광 데이터로부터 최적의 밴드선정을 위한 분석기술 개발
- 드론 수집 초분광 데이터 분석 기반 PCI 정보 수집 기술 개발
- 포장관리시스템(PMS) 맵 기반 드론 측정 PCI 정보 가시화 기술 개발
- 포장 노후화 상태(PCI) 측정 및 가시화 결과 평가/검증 기술 개발

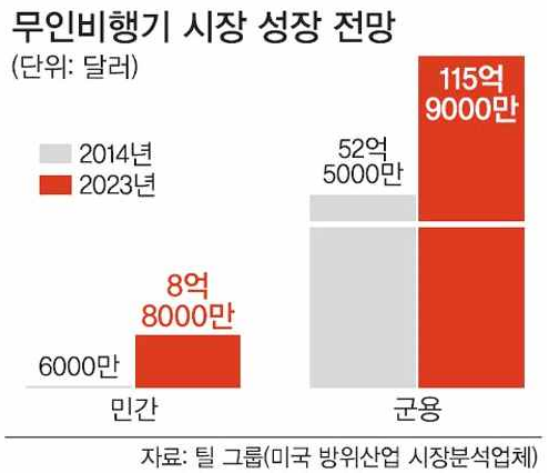


[드론 수집 초분광 데이터 분석 기반 포장 노후화 상태(PCI) 측정기술]

(2) 연구개발 배경

○ 4차 산업혁명 시대 도래에 따른 SOC 유지관리 분야 사회적 변화

- 최근 들어 드론과 관련한 항법, 통신, 센서 기술 등이 비약적으로 발전하면서 드론을 활용한 재난 안전, 방재, 물류 수송 등 다양한 분야 융합 시장이 급성장하고 있음
- 레저용 · 상업용 시장 성장 등으로 민수 제작 시장 규모는 2016년 26억 달러에서 2025년 109억 달러까지 4배 이상 성장할 것으로 전망(미국 Teal Group)
- 건설, 에너지, 농업, 통신, 보험 등 상업용 시장은 10년간 약 17배 성장이 전망되고, 용도별 시장은 취미 · 레저용이 22억 달러에서 39억 달러로 1.8배 성장하고 공공용이 0.36억 달러에서 4.64억 달러로 12.9배 성장하며 상업용은 3.9억 달러에서 65억 달러로 16.7배 성장할 것으로 전망
- Teal Group은 글로벌 민수 드론 시장 분석 보고서(2016)에서 건설 분야의 경우 공사계획, 측량, 공정관리, 시설물 조사 등 광범위한 분야에서 활용도가 클 것으로 예상하고, 이에 대한 시장규모도 2025년에는 약 16억 달러 규모로 예상되는데 이는 2016년에 비해 10배 이상 증가한 수치임

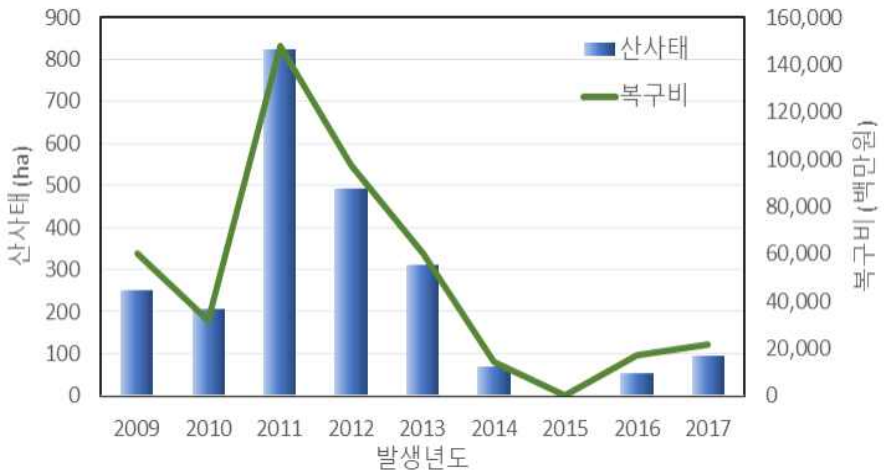


[글로벌 드론 시장 규모]

- 미 경제전문지 포천은 2016년 기준 전 세계 건설시장 규모가 8조 5,000억 달러(약 9,953조원)인데 이 중 사람이 하는 일을 드론으로 대체 가능한 금액만 1,273억 달러(약 149조원)에 달할 것으로 추정
- 글로벌 컨설팅 서비스 업체인 PwC사는 드론이 건설, 교통, 농업, 물류, 보험 등 1,250억 달러에 달하는 전통적인 산업의 혁신을 가져올 것으로 전망(로봇신문, 2016. 5. 9)
- 일본의 닛케이는 향후 10년 동안 드론 시장은 10배 증가하지만 드론 서비스 시장은 120배 증가할 것으로 전망
- 산업연구원은 드론 적용 가능한 SOC 분야 글로벌 시장규모는 452억 달러로 추정
- 2016년 기준 드론 활용 산업은 제작산업에 비해 3배 정도 크며 2025년에는 7.5배로 성장 속도도 빠를 전망(프랑스 EuroConsult, 2016. 5.)
- 드론 제작 산업과 활용 산업의 비율은 2016년 4.65 달러(제작) 대 13.43억 달러(활용)에서 2025년에는 30.7억 달러(제작) 대 229억 달러(활용)에 이를 것으로 전망

○ 비탈면 및 도로포장 유지관리 분야 현황

- 1980년대부터 집중 투자된 도로시설물의 내구 연한 도래에 따른 급속한 노후화가 진행되고 있고 그에 따른 SOC 유지관리비용이 해마다 증가하고 있음
- 광범위한 지역에 산재한 도로시설물의 효율적 점검을 위해서는 드론의 활용이 효과적임
- 국내 산사태 피해는 국지성 집중호우 등의 기후에 따라 연간 발생량의 차이가 있으며, 2011년 우면산 등에서의 대규모 발생 이후에 2017년 다시 증가 추세에 있음



[연간 산사태 피해현황]

- 비탈면 붕괴 위험성 조사를 위해서는 비탈면의 정밀변위 측정이 필요한데 이를 위해 현재 고가의 상시계측시스템을 비탈면 개소별로 설치하여 운영 중이지만 다음과 같은 문제점 발생

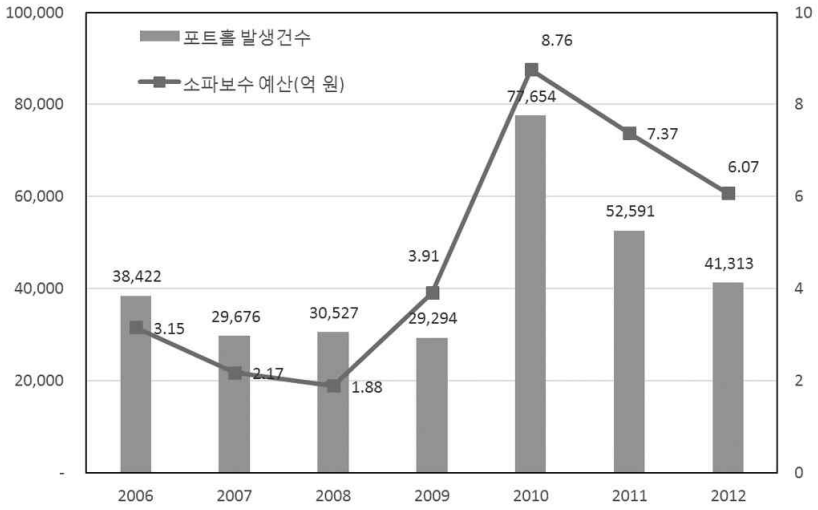
· 사람/동물의 이동, 바람 등에 따른 센서 파손 및 오경보 발생으로 인해 활용성에 제약이 있음
· 다수의 비탈면을 동시에 관리할 수 있는 드론과 달리 개별 비탈면 개소마다 모두 설치해야하기 때문에 설치비 과다 및 그에 따른 유지관리비 소요 문제가 있음

- 해마다 아스팔트 포장의 포트홀 발생 건수와 민원은 증가하고 있는 추세인데 포트홀에 대한 보수 현황을 보면 일반국도의 경우 2009년도 이후 1.5배(2012년 38,382개) 이상 증가하였고 이와 관련 민원 소송은 약 5배(2012년 50건) 이상 증가하여 행정력 및 예산이 낭비되고 있는 실정임

[일반국도 포트홀 보수 및 소송 현황]

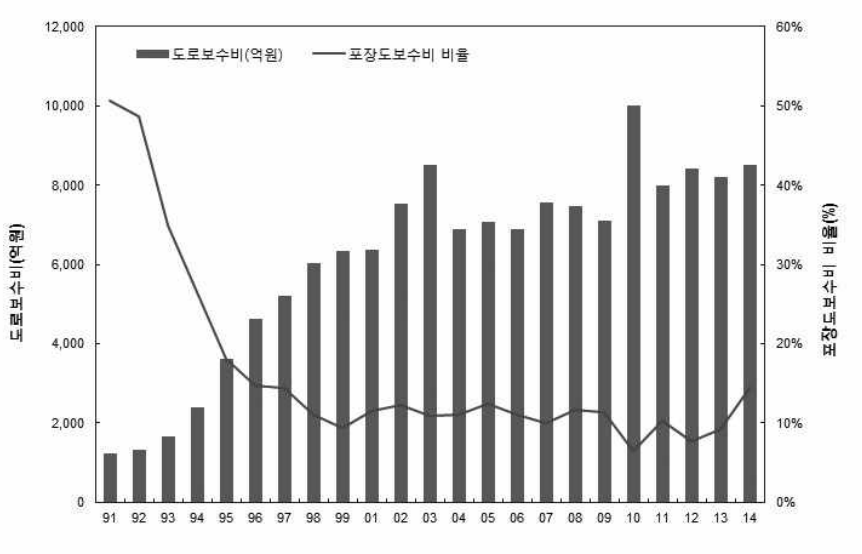
구분	2009년		2010년		2011년		2012년	
	포트홀 보수(개소)	소송(건)	포트홀 보수(개소)	소송(건)	포트홀 보수(개소)	소송(건)	포트홀 보수(개소)	소송(건)
국토청	25,231	10	44,378	28	42,424	57	38,382	50

- 포트홀을 복구함에 있어 경미한 포트홀보다는 차량의 손상이나 교통사고를 유발할 수 있는 대형 포트홀을 우선적으로 보수하는 등 합리적인 포트홀 보수전략이 필요함
- 그러나, 현재는 포트홀에 대한 정보수집 기술 부족으로 포트홀 보수의 우선순위의 불합리가 발생하고 이에 따른 잦은 민원이 제기되고 있는 설정임



[서울시 포트홀 발생건수 및 소파보수 예산]

- 일반국도 도로 노후화에 따라 도로보수비는 90년대에는 급격하게 증가하였다가 2000년대에 들어서는 완만한 증가세를 보이고 있지만, 실제 도로포장에 사용되는 포장도 보수 비율은 오히려 감소하고 있는 추세였다가 2010년 이후 차츰 증가하고 있는 추세임



[일반국도 도로보수비 및 포장도 보수 비율 현황]

- 국도 포장조사의 경우에는 전국 도로를 대상으로 조사차량을 이용하여 순차적으로 시행하며, 동일 지점에 대한 조사는 3년 주기로 조사가 수행되고, 모든 차로에 걸쳐 차량을 통한 포장면 조사 시 일개 차로별로 수집된 데이터를 인력 기반으로 분석하는 방식으로 수행되기 때문에 조사 업무의 비효율성을 초래

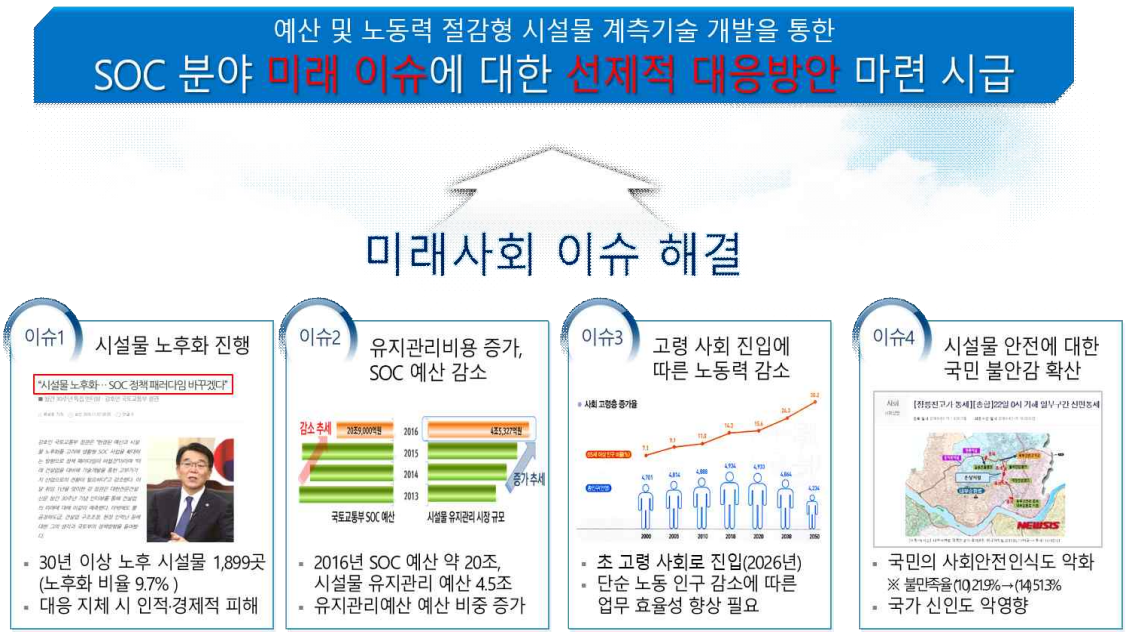
○ 드론을 활용한 시설물 유지관리 기술적 트렌드

- 미국의 스카이 캐치 등 유명 스타트업 기업들은 건설 산업시장에서 드론 기체보다 촬영 및 센싱 기술이 유망하다는 인식 하에 중국 DJI 등에서 제작한 상용 드론 기체를 활용하여 센서 및 데이터 처리기술 개발에 역량을 집중하고 있음
- 드론을 활용한 비탈면, 도로포장 등 SOC 시설물 계측을 위해서는 영상, 레이더, 초분광, 적외선 등 다양한 센서 데이터 처리기술이 필요하지만 현재 국내에서 상용화된 드론 센싱 기술은 영상 기반 3D 모델링 기술에 국한되어 있기 때문에 다양한 센서 데이터 처리 기술 개발 필요
- 현재 국내에서는 드론 영상의 단순 모니터링을 통한 비탈면 점검을 수행할 뿐, 비탈면 변위 계측의 계측이나 포장조사를 위한 고도의 센싱 기술 개발에 대한 사례는 존재하지 않음
- 국외의 경우에는 미국, 유럽을 중심으로 SAR, 영상, 초분광 센서를 이용한 비탈면 변위 측정 및 포장조사 기술을 개발 중이고, 일부 기술은 이미 상용화 단계에 있음
- 따라서 국내에서도 관련 기술을 개발하여 글로벌 트렌드에 부합하는 기술력 확보 및 전후방 연관 산업 파급효과를 극대화 할 필요가 있음
- 본 연구는 매년 수천억 원의 예산이 소요되는 비탈면 및 도로포장 관리 효율화를 위한 기술을 개발하는 것으로써, 개발기술의 시장 적용성이 우수함

다. 연구개발 필요성

○ 노후 도로 시설물 증가에 따른 유지관리 중요성 대두
- 도로규모는 지난 30여 년간 2.1배 증가하여 유지관리 대상은 2016년 기준 107,527㎞에 이르며 이중 고속·일반국도는 18,141㎞(16.9%)에 달함
- 지속적인 도로 신설에 따라 비탈면, 포장 등 주요 시설물이 증가하였고 그에 따른 노후화도 해마다 가중되고 있는 실정임

○ 도로부문 예산감소에 따른 효율적 유지관리기술 도입 필요
- 도로부문 세출예산은 2015년 9.1조원에서 2018년 5.4조원으로 최근 3년간 약 40% 감소함
- 그러나 SOC 시설물 안전성 강화에 대한 사회적 요구는 지속적으로 증가하고 있어 한정된 예산의 효율적 활용을 위한 첨단 유지관리기술 도입이 시급함



[연구 필요성]

○ 드론을 활용한 효율적 SOC 시설물 점검 등 융합산업 활성화
- 드론 관련 글로벌 시장조사 전문기관인 미국 Teal Group(2016년) 따르면 건설 분야 드론 시장 규모도 2025년에는 약 1.6조 원으로 예상되는데 이는 '16년에 비해 10배 증가한 수치임
- 넓은 지역에 산재한 도로 시설물의 효율적 점검을 위해서는 드론의 활용이 효과적이기 때문에 최근 드론을 활용한 교량 노후화 상태 점검, 송전탑 점검, 하천 측량 등에 대한 기술개발 및 시범사업이 활발히 추진 중
- 미국, 유럽 등 국외 선진국에서는 드론을 활용한 SOC 유지관리 효율화를 위한 각종 정책 및 기술개발을 활발히 추진 중임
- 드론의 경우 국토교통 7대 신성장 동력으로 지정(2016년)되어 있고 혁신성장 전략회의(2017.11.28) 에서도 대통령이 드론 산업 등에 있어 세계적 경쟁력을 갖출 수 있도록 정책과 사업에 속도 주문

○ 현행 비탈면 및 도로포장 유지관리 현황 및 이슈
- 비탈면 관리를 위하여 상시계측시스템을 개소별로(국도 146개소) 설치하여 변위를 측정하고 있지만 고가(1.5억원/개소)의 설치비, 오작동 등으로 인해 확대설치가 어려움
- 도로포장의 경우에도 포트홀 발생건수(38,382개/2012년)와 그에 따른 민원 및 소송건수(100건/2012년)가 증가하여 행정력 및 예산이 낭비되고 있는 실정임
- 덧씌우기 우선순위 선정을 위한 도로포장조사의 경우에도 고가의 조사차량을 이용하여 전체 도로에 대한 차로별로 조사함에 따라 조사비용 과다 및 효율성 저하 초래



도로비탈면유지관리시스템(CSMS)

- 개요 : 전국 국도에 분포된 비탈면(약 30,000개) 현황 관리 및 비탈면 보수, 보강을 위한 의사결정지원시스템 운영
- 변위측정 : 장기적인 지반 변위양상 파악을 위해 상시계측 시스템을 개발하여 현재 전국 146개소에 설치, 운영 중

도로포장관리스템(PMS)

- 개요 : 일반국도 포장의 효율적인 유지관리를 위해 도로 포장관리시스템(PMS) 운영 중
- 포장조사 : 매년 약 4,000㎞ 국도에 대해 노면 조사차량을 이용하여 포장상태를 조사

[비탈면 및 포장조사 현황]

○ 드론을 활용한 비탈면 및 도로포장 관리 효율화 기술 개발 필요

- 드론을 활용하여 지진, 해빙기, 우기 등에 발생할 수 있는 비탈면의 붕괴를 사전에 예방할 수 있는 변위(變位) 측정 기술 개발 필요
- 드론을 활용하여 기후변화 등에 따라 해마다 증가하는 포트홀의 긴급복구 및 포장조사 효율화를 위한 포트홀 및 포장상태지수(PCI) 측정 기술 개발 필요

구분	문제점	개선방안
비탈면 변위 측정	동물, 바람에 의한 센서 오작동 지점별 설치에 따른 예산 부담 (1.5억원/개소)	산짐승, 바람, 초목 영향 없는 변위 측정 기술 개발 예산 저감형 계측 기술 개발
포트홀 현황 조사	조사 시스템 부재 연간 행정소송 건수 약 100건	폭우, 폭설 등 악천후 종료시 신속한 포트홀 현황 파악 기술 개발
포장 노후화 상태 조사	전체 차로별 조사에 따른 조사 기간, 비용, 인력 과다 소요	조사기간, 비용, 인력 저감형 포장 노후화 상태 측정 기술 개발

[현황 문제점 및 개선방안]

라. 연구개발 대상 국내외 현황

(1) 국내 현황

○ 기후변화 및 사회간접자본 노후화로 인한 유지관리 중요
- 기후변화에 따른 폭우, 폭설의 증가와 신도시 건설 및 도로건설 등에 따라 도심지 주변과 도로인근에 조성된 인공비탈면 관련 시설의 노후화로 비탈면 붕괴 등의 위험이 증가되고 있으며, 노후도로는 지속적으로 증가하지만 한정된 예산으로 충분한 유지관리가 어려우며, 포트홀 등의 파손으로 안전사고 위험 증가

○ 위험지역 비탈면 관리 시스템 운영
- 정부와 지자체에서는 산사태 등을 예방하기 위해 시설물정보관리종합시스템(FMS)을 통해 2,990개의 산사태 취약지역, 인명피해 우려지역 등의 비탈면을 관리하고, 위험지역에 사방댐, 계류보전 시설구축, 산지보전사업 등을 추진
- 국토교통부와 한국건설기술연구원에서는 도로 절토사면 유지관리시스템(Cutting Slope Management System, CSMS)을 운영하여 전국 약 13,820km의 고속국도 및 일반국도국도 주변에 분포된 비탈면(2014년 29,850개소) 분포 현황을 조사하여 절토사면의 유지관리 및 안전사고 예방
- 전국적으로 비탈면을 점검·조사하고 있지만 일부지점의 센서매립 방법 이외에는 주로 인력식 방법으로 수행하고 있으며, 조사자의 접근이 어려운 산악지형이 많아 조사의 효율성이 낮고 지속적인 관리에는 한계가 있음

○ 고가 도로 조사 차량을 이용한 3-5년 주기의 도로포장관리 시스템
- 국도와 고속도로는 효율적인 도로관리를 위하여 자동 도로포장상태 조사차량, FWD(Falling Weight Deflectometer) 등을 이용한 도로포장관리시스템(PMS)을 운영하며, 최근 서울시, 경기도 등에서 일부 도입하였으나, 장비 구축 및 유지관리 비용이 높아 모든 지자체에서 구축하기 어려움
- 일반국도 2014년 현황을 보면, 국도 13,527km 중 3,852km의 조사대상구간에 대하여 노면상태를 조사하고, 이 중 2,015km는 다양한 조사차량을 이용하여 정밀한 상태조사 및 역학조사를 수행

○ 포트홀 발생 실시간 신고 및 보수 시스템 운영
- 포트홀은 강우, 강설 등의 기후변화 요인으로 최근 크게 발생되고 있으며, 정부와 지자체에서는 이에 따른 교통사고 저감을 위해 아스팔트 포장 재료와 시공기준 개선, 포트홀 발생 후 실시간 신고 및 복구 시스템 구축 등을 추진하고 있음.
- 악천후 발생 후 포트홀 관리는 주로 응급복구 차량이 관내를 순찰하면서 포트홀이 육안으로 발견되면 복구하지만 효율성이 낮아 포트홀 발생에 따른 교통사고, 민원 등이 지속적으로 발생
- 특히 도심지는 서울시 포트홀 실시간 신고시스템 등의 노선버스, 택시 등을 활용한 포트홀 신고가 이루어지고 있으나 신고자의 활동에 의존하고, 사전탐지가 불가능하며 광범위한 지역에 분포되어 있는 지방부 도로의 경우 신속한 조사가 어려움

○ 드론 특별 승인제 등 국내 드론 관련 제도 개선
- 최근 정부에서는 드론 산업을 국토교통 7대 신성장동력으로 선정하고, 드론 기반 도로관리 중장기 기술개발 로드맵 마련

- 2016년 규제개혁 장관회의 후속 조치로 항공법 시행규칙 개정, 드론 사업범위 확대, 비행전용 구역 확대 등을 추진하였으며 2017년에는 드론 특별 승인제를 시행하여 야간, 가시권 밖 비행허용, 특례적용 등의 기준 마련

○ 국내의 건설공사 및 유지관리시 드론 이용

- 2016년부터 국토교통부는 30만㎡ 이상의 중대규모 택지 · 산업단지나 댐 등에 대해서 토지보상 현장조사의 효율성을 높이고 불법 보상투기를 방지하기 위해 사업 초기단계부터 드론을 활용하여 현장 사진을 촬영하고, 지번 · 지장물이 표기된 지적지형중첩도와 연계해 자료 구축 · 활용
- 원주지방국토관리청에서는 도로변에서 육안 및 드론을 이용하여 무작위로 안전관리실태를 감찰하여 공사기간중 건설현장 관계자의 안전관리 관심도를 높이고 안전사고의 선제적 예방

○ 국내 시장 전망

- 국내에서는 비탈면 붕괴 조기감지 및 예방, 효율적 도로관리를 위해 일부지역에서는 관련 시스템을 운영하고 있으나 구축 및 유지관리 비용 등의 문제로 인해 전국적인 관리시스템의 구축과 지속적인 관측이 어려움
- 따라서 드론을 이용한 비탈면 정밀계측 및 도로관리 기술의 개발시 기존 시스템보다 저비용으로 구축가능하고, 위험지역이거나 도로 교통량 등에 상관없이 조사가 용이하므로 기존 기술을 보완할 수 있어 현장 적용성이 높음

- 국내 건설 분야의 드론 활용은 원주지방국토관리청의 광학영상을 활용한 육안 모니터링, KAIST의 대형구조물 원격진단, 부산교통공사의 교각하부의 자동 영상취득 등이 있음
- ㈜공간정보에서는 한국국토정보공사와 드론을 이용하여 촬영한 이미지 분석을 통해 공사현장의 토공량 · 체적량을 산출하여 공사비 산정 및 토공계획의 수립에 사용
- 2015년 울진군 봉평해수욕장을 태풍피해 전후에 드론으로 촬영하여 방파제, 제방 등의 손상피해 비교 분석 수행
- 한국건설기술연구원 CSMS 운영팀은 정밀점검, 붕괴비탈면 및 계곡부 정밀조사 등 연간 약 400개소 이상의 현장조사를 위해 인력의 접근이 어렵고 육안조사의 한계로 파악하기 힘든 위험요소들에 대한 보완대책이 필요하여 2015년 초부터 드론 영상을 이용한 육안 모니터링을 수행하고 있음



[국내 도로분야 드론활용 사례]

- 드론 촬영 영상을 활용해 조사자의 접근이 어려운 산악지 지형에 대하여 드론 촬영 영상을 활용하여 상부 자연 비탈면 내 인공시설물의 파악, 인장균열 발생여부, 지반침하 현상 인지

등을 수행하여 비탈면 위해 요인 파악 가능
- 특히 해빙기 합동 비탈면 점검, 2종 시설물 정밀점검 현장조사시 드론을 활용하여 효율적이고 안전한 현장점검 수행 가능
- 비탈면의 잦은 보수, 보강 등으로 현행화가 어려운 부분이 있는데, 이 경우 드론 촬영 영상의 3D 모델링을 통해 경사도, 높이, 길이, 면적 등 현황을 파악할 수 있음
- 한국건설기술연구원과 한국국토정보공사에서는 2016년 국토교통부 “드론을 활용한 도로관리 효율화 방안” 연구에서 드론이 촬영한 항공영상을 편위수정(rectification) 등의 다양한 보정 과정을 거쳐 정사 이미지(orthogonal image)로 변환하고, 이를 지적도와 맵핑하여 도로 점용현황 조사 시범사업을 추진함
- 국내의 드론 영상을 활용한 비탈면 3D 모델링, 토공량 측정, 측량 등 사진영상 처리기술은 세계적인 수준으로 즉시 사업화가 가능한 기술이지만, 초목이 있는 비탈면의 정밀변위 측정 및 도로포장 조사기술은 전무한 수준임

[국내 도로분야 드론 기술 및 활용사례 요약]

구분	분야	내 용	특 징
국내	도로건설	- 건설계획 및 보상을 위한 지장물 현황 조사 - 토공량, 체적량 산출 - 현장접근이 어려운 건설현장 안전점검	**- 영상 센서 데이터 기반 육안 모니터링, 측량, 3D 모델링에 국한** **⇒ 비탈면 : 초목(식생)이 있는 비탈면 변위측정 기술 없음** **⇒ 포장 : 포트홀 및 PCI 측정 사례 없음**
	유지관리	- 비탈면 현황 관리를 위한 비전센서 기반 육안 모니티링 및 3D 모델링 - 불법 도로점용 현황파악을 위한 도로점용조사 - 접근이 어려운 교량 점검 - 대형구조물 진단 및 원격 관리 시스템 개발	
	재난관리	- 도로 절토사면 유지관리를 위한 DB 구축 - 태풍, 집중호우에 의한 도로유실 현황점검	

(2) 국외 현황

○ 국외 드론관련 제도
- 호주는 1990년대부터 무인기 인증기준 제정을 세계적으로 선도하는 등 무인기 분야 기술발전에 적극적인 정책을 추진 중
- 미국은 드론 운영 관련하여 테러위협 및 안전 등의 문제를 이유로 인명피해 위험이 있는 상업적 목적의 무인기 이용을 규제하였으나, "2012 현대화 개정 법률" 이후 현재 상업적 목적 허용
- 취미용 25kg 미만 모형 무인기(Model Aircraft) 운영기준, 상업목적 25kg 미만 소형무인기의 가시거리 비행, 일몰 후 비행금지, 최고속도 100마일, 최고고도 500 피트 제한 등의 기준, 25~250kg의 무인기의 등록 기준, 비행공역(높이)에 따른 구분 및 제한기준 등 제정
- 캐나다는 모든 드론에 대한 비행면허를 요구하였으나 2015년 캐나다 교통국이 25kg미만 소형 드론의 이용 시 용도나 무게에 따라 구분
- 영국은 중량 20kg 이하의 드론을 소형 무인항공기라고 하여 보통 유인항공기에 적용되는 대부분의 규제를 면제하고 상대적으로 쉬운 운영허가를 받게 됨. 중량 20kg 초과 드론이나 항공업허가를 요하는 항공사진을 촬영하는 항공기는 유인항공기 수준의 강한 규제를 받게 됨
- 유럽항공안전청(EASA)은 무게 또는 상업용·비상업용 구분이 아닌 위험(risk)에 근거하여 저·중·고 위험군의 3개로 분류하여 제한

○ 국외 건설공사 및 유지관리시 드론 이용
- 드론 산업에는 인터넷통신, 물류, 방송, 서비스 기업 등 다양한 업종의 기업들이 참여하고

있음
- 드론은 건설, 유지관리, 재난관리 분야에 활용되며 주로 도로포장의 상황을 점검하거나 교량 슬래브 하부 등 사람의 접근이 어려운 지역에 대한 점검용으로 추진
- 미국 교통부는 교통정체 해소, 안전성 개선, 예산 절감 등을 목적으로 교량, 비탈면, 도로포장 노후화 측정에 드론 활용 방안 연구 중이며, 2016년 3월 현재 17개 주에서 관련 연구를 완료하였고 16개 주에서는 관련 정책, 기술 개발 지원 중임
- 미국 유타교통부에서는 인터체인지(Diverging Diamond Interchange)의 도로개선공사 전후 교통량 평가에 드론을 이용
- 유럽은 독일, 이탈리아를 중심으로 도로포장 노후화 조사 시 드론을 활용하여 조사 효율성을 높이고 있으며, 특히 독일의 Arsight社는 공항 등의 포장 상태 측정을 위한 드론 기술을 상용화
- 일본 국토교통성은 드론 기반의 건설사업 효율화를 위하여 “i-Construction” 이라는 슬로건을 발표하였고, 이에 따라 2016년에는 “UAV를 활용한 공공 측량 매뉴얼(안)”, “무인항공기를 이용한 기성부분 관리의 감독·검사 요령(토공편)(안)” 등을 마련
- 일본은 건설 현장의 생산성을 크게 높일 수 있는 신서비스로 드론을 활용한 측량 산업으로 인식하고 있으며, 특히 전자 센서 업체 소니는 로봇 업체 ZMP와 공동으로 설립한 에어로 센스를 통해 2016년부터 건설 현장 드론 측량 사업을 시작함
- 중국에서도 드론을 활용한 비탈면 붕괴 시 인명구조, 현황파악, 응급물자 수송 등을 위한 드론 활용방안 시범사업 및 연구를 추진함

○ 비탈면 변위 측정
- 이탈리아에서는 2011년~2014년 도로터널 굴착시 터널 상부 사면에 발생하는 변위를 고정밀 레이더(InSAR; Interferometric Synthetic Aperture Radar) 센서로 공사 전인 2003년부터 공사후인 2016년까지 측정하였으며, 지상에 매설한 센서의 값과 비교한 결과 위성에서 측정한 값은 잘 일치하는 것으로 나타났으나 지상에서 측정한 InSAR 센서값은 작은 차이가 발생
- Nebraska-Lincoln대학에서는 비탈면의 정밀 변위를 측정을 위해 드론에 InSAR 센서를 장착하여 매우 느린 속도로 진행되는 비탈면 포행(creep)을 관측하여 비탈면 붕괴 위험성을 사전 예측

○ 도로포장 점검 및 유지관리
- 드론을 이용한 도로의 유지관리를 위한 모니터링 기술들은 원격탐사(remote sensing)의 응용과 광학(가시광선) 센서 외에도 열화상센서, 다중분광센서, 초분광센서, LiDAR 등 다양한 센서의 특징을 활용하여 인프라의 정량적 노후도를 판단하고자 하는 연구들이 진행됨
- 비포장도로 상태의 모니터링 및 보수 시스템 개발을 위해 Zhang(2008)은 GPS/IMU 및 지자기 센서를 장착한 무인 헬리콥터로 사전 정의한 비행경로를 자율 비행하여 비포장도로의 상태를 사진측량 기술로 점검하였으며, 2012년에는 3D DSM(Digital Surface Model)을 구현
- Airsight 사의 드론 기반 지상 점검 기술은 GPS와 고해상도 카메라를 탑재한 드론을 이용해 공항지역을 빠르게 스캔하여 영상을 후처리하여 공항 관리자에게 포장현황과 균열정보 제공
- 미국 버몬트대학에서는 수해지역 유실도로의 복구를 위해 드론을 활용하여 도로 유실지역의 DSM을 기반으로 복구를 위한 토공량 산출
- 초분광(hyperspectral)센서 및 분광반사 특성을 이용하여 파장폭이 약 10nm 또는 400~1,000nm인 수십~수백 개의 연속적인 밴드로 취득한 영상을 분석하며, 식생 분석이나 지표면 객체의 분류뿐만 아니라 초분광센서를 탑재한 AVIRIS(Airborne Visible/Infrared Imaging Spectrometer) 영상을 이용하면 도로의 상태에 대한 분석 가능(Herold, 2003, Herold et al., 2004)하며, 도로 포장의 노후화 측정에도 응용될 수 있음
- Mei et al. (2014)의 연구에서는 수치영상처리(Digital Image Processing, DIP) 및 스펙트럼 분석을 통해 아스팔트 노후에 의한 도로 표면 변화량을 분석



[국외 도로분야 드론활용 사례]

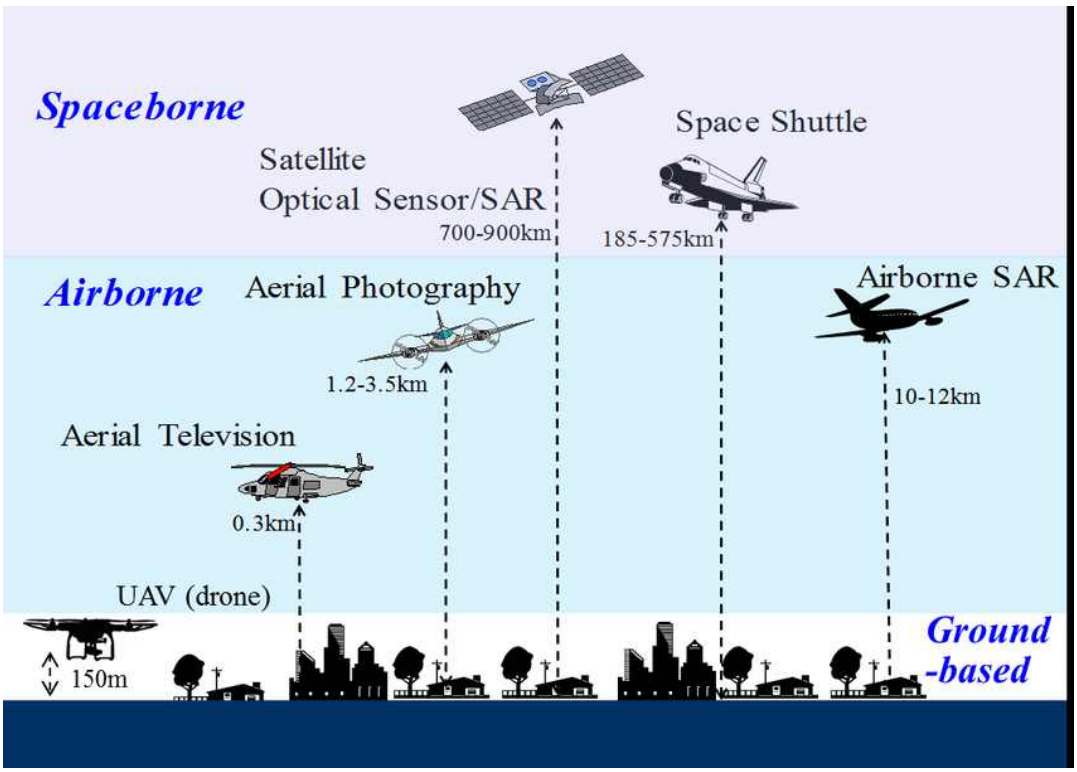
2. 연구개발과제의 수행 과정 및 수행 내용

[1차 년도]

가. 비탈면 변위측정 기술

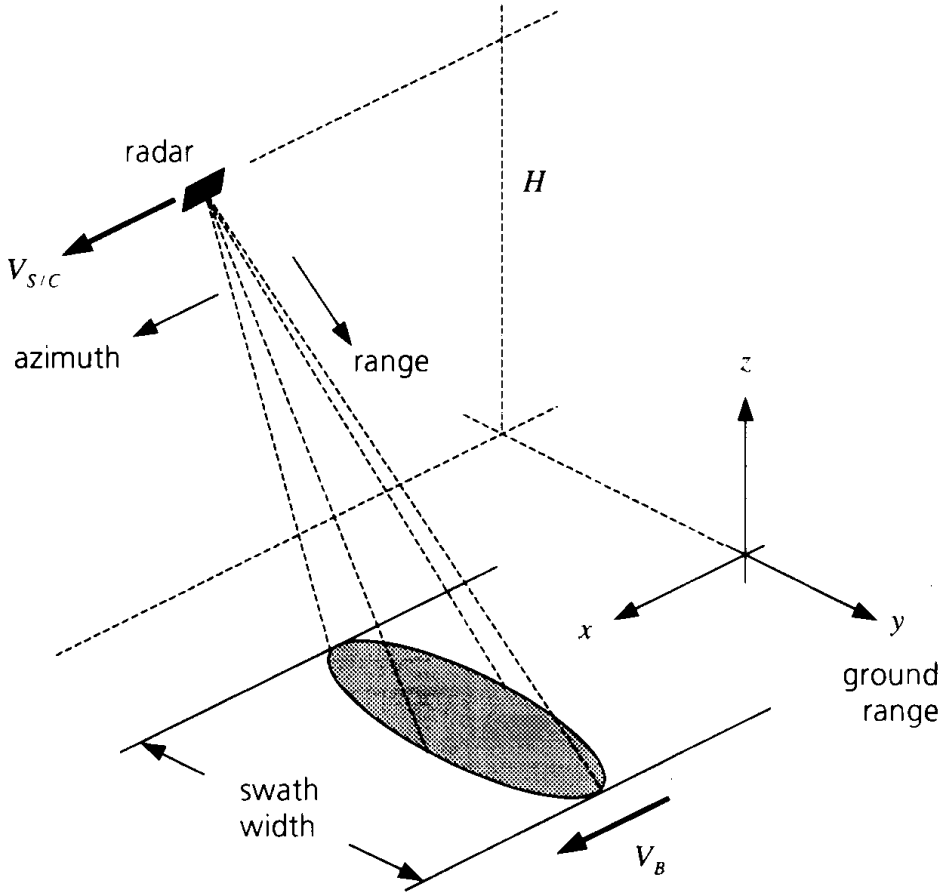
(1) 비탈면 SAR 영상 생성 기술 개발

- Synthetic Aperture Radar (SAR)는 비행체 혹은 이동 물체에 탑재된 안테나에서 마이크로 대역의 신호를 능동적으로 송신하고 지표의 산란체로부터 반사된 신호를 수신하여 획득한 영상레이더 자료임
- 마이크로파 대역의 전자기 신호는 수증기, 구름, 눈 등을 투과하는 성질을 가지는 능동형 센서로서 광학 전자기파 대역에 비해 전천후 환경에서 일조량, 시간, 기상 조건과 무관하게 영상 획득이 가능하다는 장점이 있음
- SAR 자료는 대상체로부터 반사된 신호의 진폭 (amplitude) 정보뿐만 아니라 안테나와 산란체 간의 거리를 나타내는 위상 (phase) 정보를 동시에 복소수 형태로 저장하고 있음
- 영상레이더의 주요 플랫폼에는 항공기, 우주왕복선, 위성, 드론, 레일, 차량 등이 있음



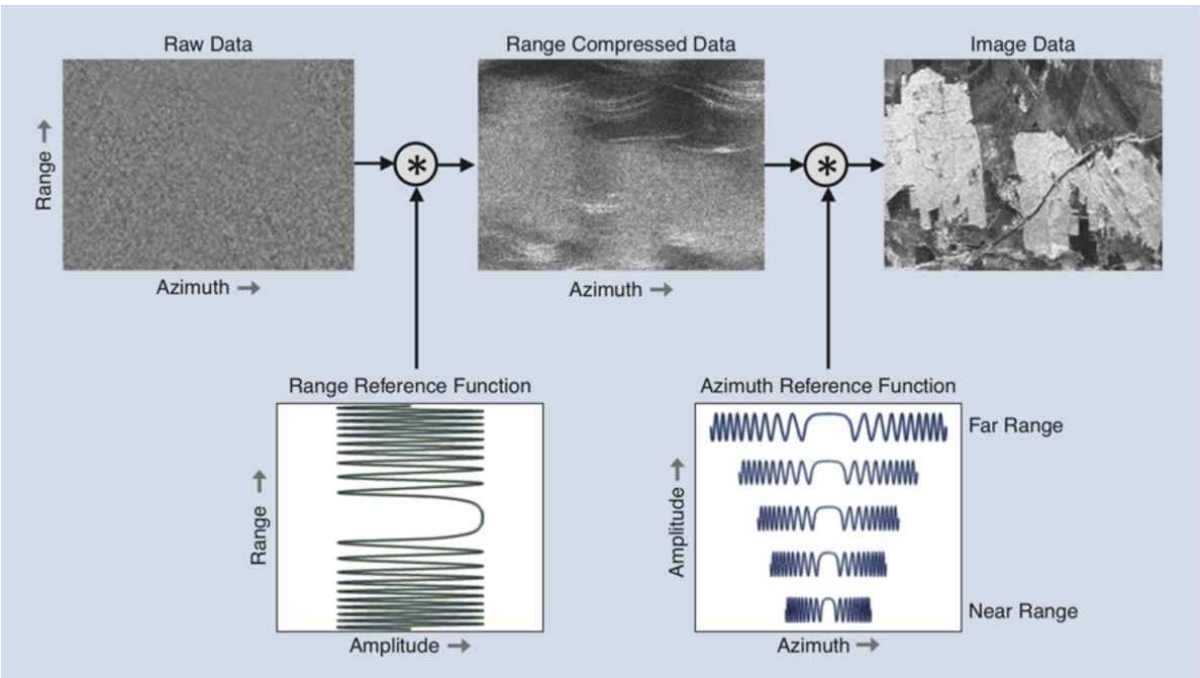
[다양한 원격탐사 플랫폼 (Yamazaki, Fumio, & Wen Liu., 2016)]

- 영상레이더는 비행체에 안테나를 탑재하여 일정한 주기 (pulse repetition frequency: PRF)로 획득된 전자기파를 합성하는 기술로, 지형의 고도 정보 추출에서부터 지진, 화산, 산사태, 지반침하와 같은 지질재난재해 감시 등과 같은 미세 지표변위 정보 추출 뿐만 아니라 토양 수분 조사, 빙하 이동 관측, 기름 유출 조사, 선박 감시 등 전 지구적 관측이 가능함
- 일반적인 SAR 영상은 위성 혹은 탑재체가 이동함에 따라 촬영되므로 range-azimuth 좌표계 상의 기하학을 가지며, azimuth는 비행체의 이동 방향, range는 비행 방향에 수직인 방향의 의미함



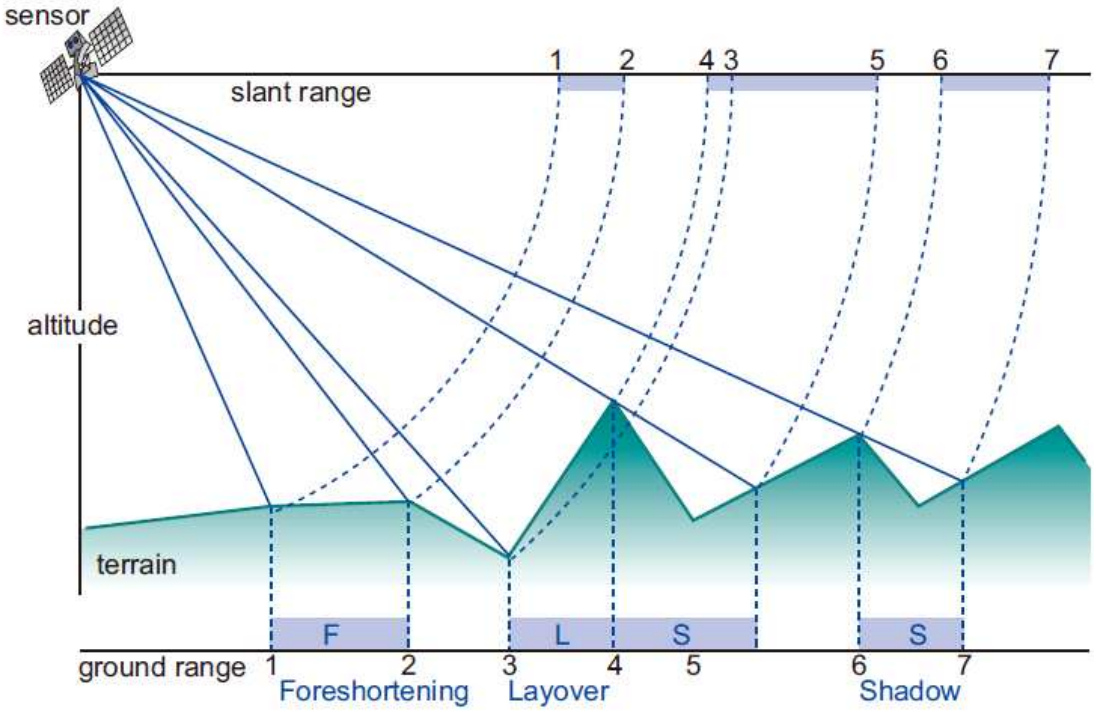
[SAR의 간단한 기하학 (Richard Bamler, 2000)]

- 이와 같이 영상레이더 탑재체는 지표를 관측할 때 직하부 방향이 아닌 측면 관측을 하기 때문에 경사 거리 방향에 대한 지형 왜곡이 발생하며, 측정된 slant range를 ground range로 변환하는 과정이 필요함
- 일반적인 영상레이더의 신호 복원 과정은 원시 데이터를 시간 도메인에서 range 레퍼런스 함수와 적분하여 range 압축 데이터를 획득한 후, azimuth 레퍼런스 함수와 적분하여 SAR 영상을 획득함 (Dell'Amore, Luca, Michelangelo Villano et al., 2019)



[SAR 영상 처리 과정 모식도 (Dell' Amore, Luca et al., 2019)]

- 지표의 기하 특성에 따라 foreshortening, layover, radar shadow 와 같은 기하학적 왜곡들이 발생하며 지형 표고 모형 등을 활용한 적절한 보정이 필요함
- Foreshortening은 위성의 측면 관측으로 인해 원래의 관측 지형보다 축소 느낌을 주는 현상임
- layover는 안테나에서 지형까지의 거리를 반영하는 SAR의 영상특성에 의한 왜곡 현상으로 높은 지점이 낮은 지점보다 안테나에 더 가까운 지점으로 영상화된 현상임
- shadow는 지표 고도에 의해 레이더 신호가 가려져서 전달되지 않는 영상 왜곡 현상임



[기하학적 왜곡 (Norman Kerle et al., 2004)]

- SAR 영상의 품질을 지시하는 해상도의 종류에는 공간해상도 (spatial resolution), 분광해상도 (spectral resolution), 방사해상도 (radiometric resolution), 시간해상도 (temporal resolution)가 있음
- 공간해상도는 영상 내 한 화소가 가지는 실제 공간적 크기를 뜻하며 일반적으로 정사각형 형태의 화소에 대한 길이로 표시함
- 분광해상도는 전자기파에서 사용하는 각기 다른 주파수 대역의 수를 의미하며, 영상레이더의 경우 단일 채널의 마이크로파 대역인 K, X, C, S, L, P-밴드가 주로 활용되고 있음
- 방사해상도는 센서로부터 들어오는 에너지의 세기를 구별하는 능력을 의미하며 대부분의 영상레이더의 방사해상도는 16 bit (2 byte) 영상으로 제공됨
- 시간해상도는 동일한 지역을 주기적으로 관측하는데 걸리는 시간 간격을 의미하며, 관측 주기가 짧을수록 시간해상도가 좋음. 특히 위상간섭기법의 성공적인 적용과 긴밀도 향상 측면에 있어 시간해상도가 가지는 역할이 매우 큼

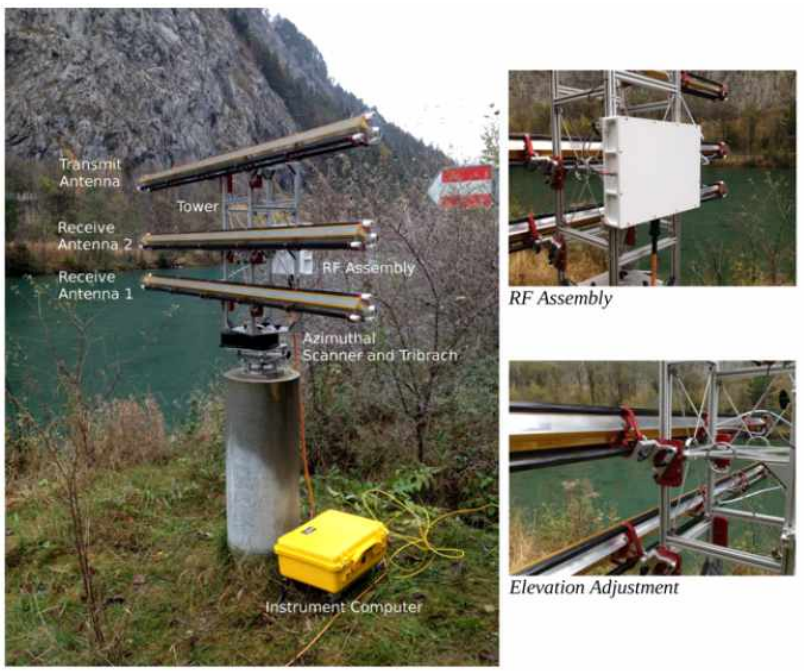
○ 지상레이더 SAR 데이터를 이용한 지반변위 측정기술 분석 사례 조사

- 높은 시·공간 해상도로 자료 획득이 가능한 지상레이더를 활용할 경우 인공위성 영상레이더 차분위상간섭기법 적용에 있어 지표 변위가 일정 파장 대역의 길이를 초과할 경우 관측 불가능하였던 부분을 개선할 수 있음
- 또한, 인공위성 영상레이더 자료 획득 당시 발생하는 대기 지연 및 이온층 지연 효과로 인해 발생하는 오차가 없으므로 매우 높은 정밀도를 지니는 지표 변위 결과를 획득할 수 있음
- 지상형 영상레이더는 위상간섭기법을 통한 변위 관측에 있어서 현장 측량의 이점을 고려하려 개발됨
- GPRI-II(Gamma Portable Radar Interferometer)는 우분투 리눅스 환경에서 구동되는 지상형 영상레이더 azimuthal 스캐너로, 차분위상간섭기법을 적용하여 변위를 관측함

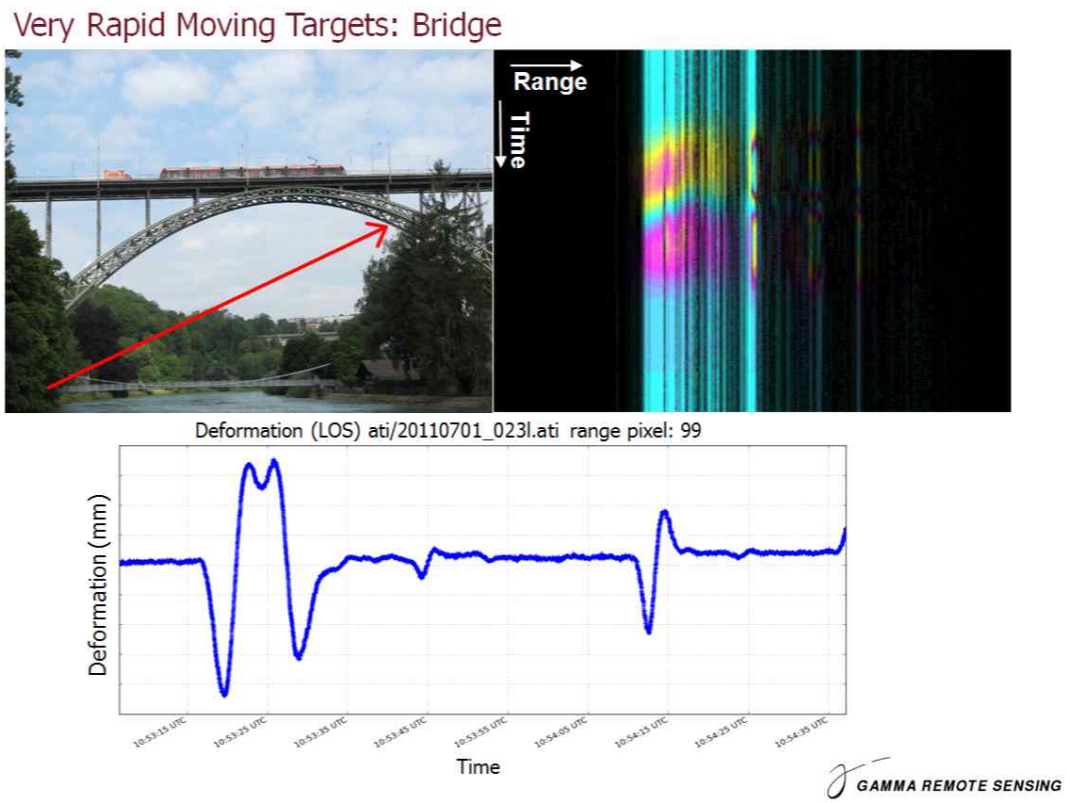
- 지표 변위 관측에 있어 0.1 mm의 정확도를 가지고 있으며, 공간기선거리의 변화가 없기 때문에 지형 위상의 영향을 거의 받지 않음
- 하나의 송신 안테나와 두 개의 수신 안테나로 구성되어 있으며, 약 2.5 m의 설치 면적이 필요함
- GAMMA 소프트웨어사는 GRPI에서 획득된 데이터를 인공위성 영상레이더 자료와 같이 처리할 수 있는 프로그램을 제공함
- 열차이동에 따른 교량의 수직적 변위 등과 같은 구조적 변위의 정량적 탐지를 수행할 수 있으며 매우 효과적으로 감시할 수 있음
- 또한, 이중 수신 채널을 이용하여 500 m 거리에서 0.25 m 의 공간해상도를 지니는 매우 높은 정밀도의 수치표고모형 제작이 가능하며, 지오코딩 또한 높은 정밀도를 보여줌
- 지상형 영상레이더 자료는 수치표고모형을 기반으로 한 영상처리를 수행하므로 영상레이더 자료와 제작된 제품들의 지형보정작업이 따로 필요하지 않다는 장점이 있음

[PRI-II 사양]

주파수 대역	17.1-17.3 GHz
레이더 유형	FM-CW
레이더 작동 범위	~10 km
Range 해상도	1.0 m (200 MHz chirp 및 Kaiser window weighting)
Azimuth 해상도	8m/1 km (slant range에 비례)
무게	Tower: ~8kg RF electronics: ~4 kg scanne:r 10 kg Controller/Power: ~14 kg Antennas: 각각 2.7 kg
크기	Tower: 80x28x28 cm RF assembly: ~ 30x26x6cm Controller/Power: ~52x42x30cm Antennas: 210x12.5x12.5 cm

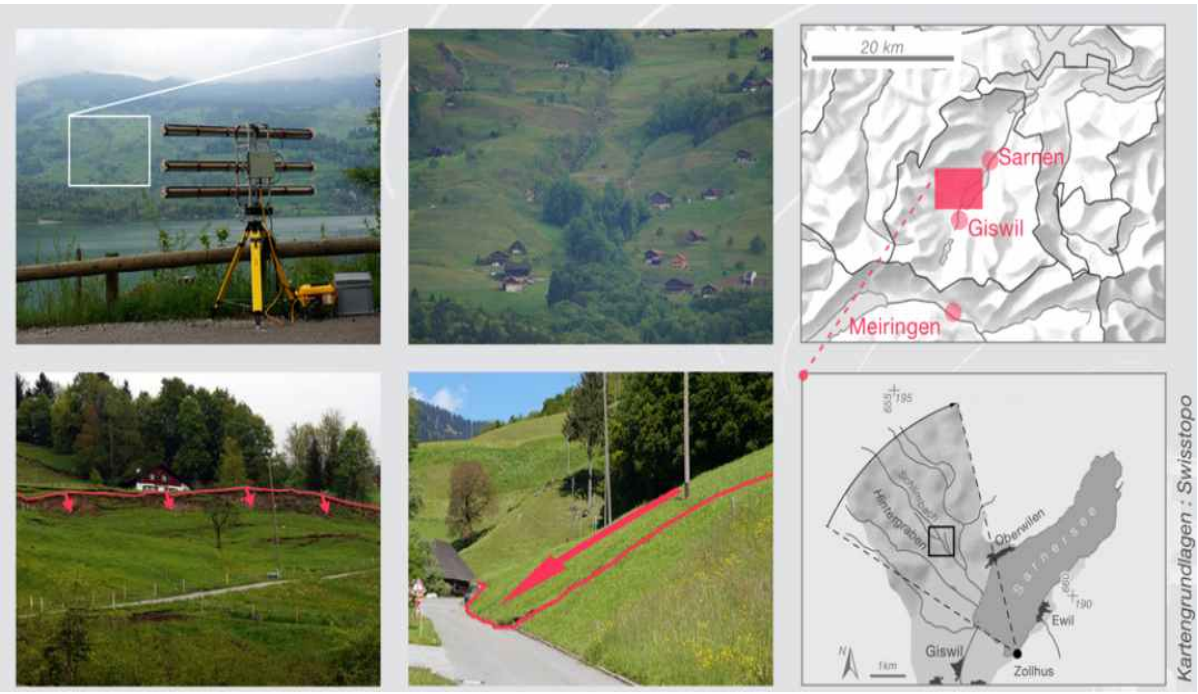


[고정형 지상레이더 관측장비 (Gamma Remote Sensing)

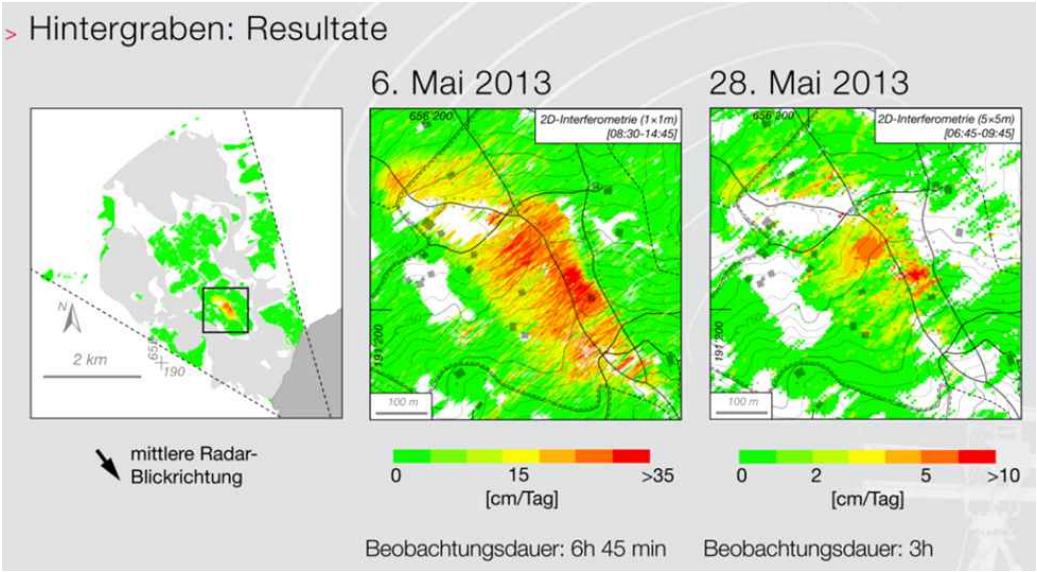


[교량수직변위 관측 결과 및 그래프 (Gamma Remote Sensing)

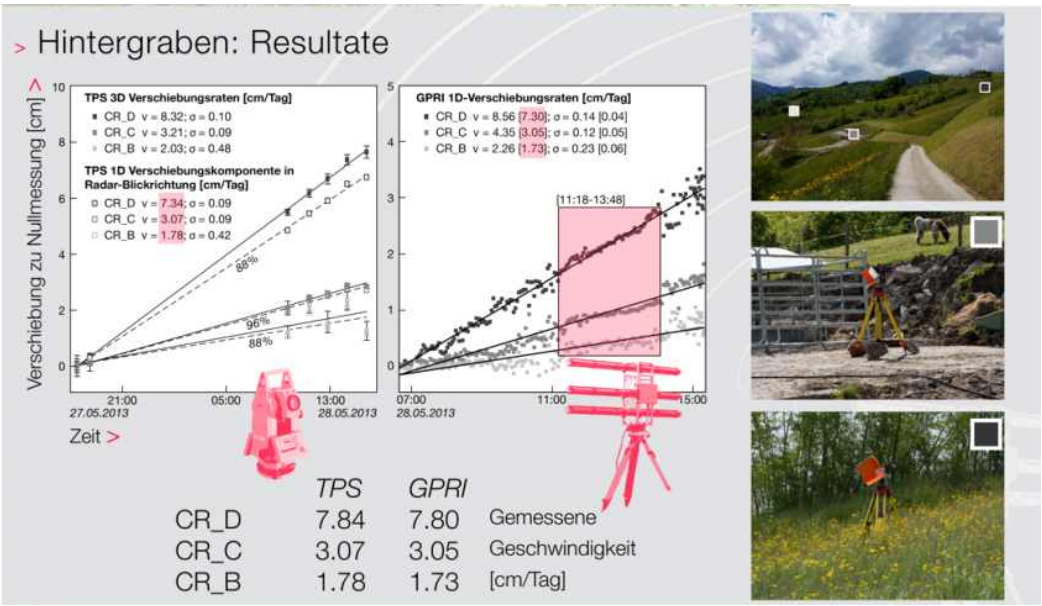
- GAMMA 사는 2013년 5월 스위스 Hintergraben에서 GPRI-II를 이용한 산사태 관측을 수행함
- 연구지역에서 5월 6일 6시간 45분 동안 관측한 결과와 5월 28일 3시간동안 사면을 관측하여 획득된 지표 변위 결과를 보여줌
- GPRI-II를 이용한 변위 측정 결과와 지상 total station을 이용한 변위 측정 결과를 비교함으로써 매우 유사한 관측 정보를 획득함으로써 산사태 관측에서의 영상레이더의 유용성을 확인하였음



[스위스 Hintergraben 지역 및 현장 관측 사진, 출처(: Gamma Remote Sensing)]



[GPRI-II를 이용하여 획득된 지표 변위 결과 (Gamma Remote Sensing)

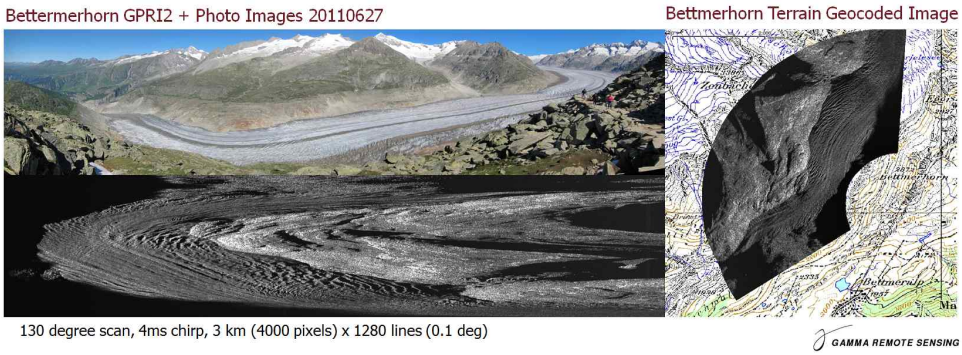


[Total station과 GPRI-II를 이용하여 관측한 지표 변위 측정 결과 비교 (Gamma Remote Sensing)

○ SAR 영상 생성 기술과 지도 제작을 위한 정사 보정 기술 개발

- 비탈면 변위 측정 기술 적용을 위한 single look complex (SLC) 영상 제작
- 수집된 SAR 영상의 좌표계 변환 기술 확보
- 레이더 영상에서는 경사 관측으로 영상을 획득하므로 산악지역에서 발생하는 기하학적 왜곡은 영상 활용에 있어 제한사항이 될 수 있음
- 산악 지형에서는 foreshortening, layover, shadow 등의 기하학적 왜곡이 발생하게 되고 이를 정사보정을 통해 동일한 평면 정확도를 가지는 정사 영상으로 바꿀 수 있음
- 수치표고모형 (Digital Elevation Model, DEM)은 수치 지형 또는 수심 측량 데이터에 관한 용어로 식생과 인공 건축물 등을 포함하지 않는 지형만의 높이를 나타내고 Digital Terrain Model (DTM)과 유사하게 사용되지만, DTM은 동일한 간격의 z값을 가지고 불규칙적인 간격을 갖는 불연속선이 존재한다는 차이점이 있음
- 수치표면모형 (Digital Surface Model, DSM)은 DEM에 자연 및 인공 지형지물이 포함된 것으로 실제의 모든 정보를 표현한 모형을 말함
- DEM은 이미 제작된 수치지도를 이용한 방법, 항공사진측량을 이용한 방법, 위성영상을 이용한 방법 등을 통해 제작이 가능하고 정확도는 데이터의 스케일과 공간해상도의 영향을 받음

- 정사영상 제작을 위해 지면의 고도 차이에 의해 발생하는 편위를 수정하는 수치미분 편위수정은 외부 표정 요소와 DEM을 통해 수치적으로 정사투영 영상 제작이 가능
- 수치미분 편위수정을 통해 보정된 영상을 재배열하여 정사보정 모자이크 영상을 제작
- 위성 탑재체 기반 SLC 제작기술과 지형보정 SLC 제작기술 비교 성능 검토



[지상용 레이더의 영상 생성 기술(좌), 비탈면 지도화 제작 기술(우)]

○ 드론 촬영 SAR 영상 생성 장치 입수를 위한 평가 테스트 진행
- 다중 시기 SAR 영상 생성을 위한 드론 플랫폼에서의 SAR 영상 생성 기술 장치 평가 테스트 진행 (네덜란드 Metasensing사 및 스위스 Gamma사)
- 진폭 영상에서는 일정 부분 좋은 품질을 보였으나 위상 정보에서는 간섭현상과 반복궤도마다 달라지는 위상 왜곡현상이 발견됨



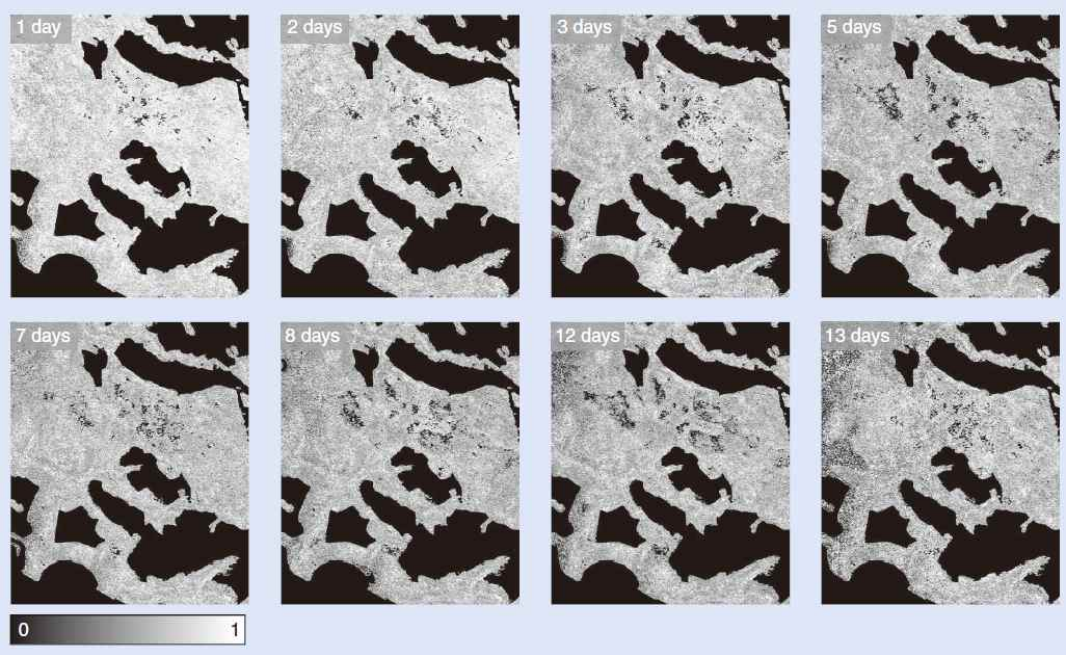
[드론용 SAR 관측 탑재 모습(좌)과 주요 시스템 제원(우)]

○ 식생에 따른 긴밀도(간섭도) 차이 문헌 사례 연구 조사
- 위상간섭도에서의 잡음 위상 φ_{noise}은 두 영상레이더 자료 간 상관관계를 나타내는 긴밀도(γ)와 관련이 있으며, 다음의 식 네 가지 세부 항목으로 표현 가능함 (Osmanoğlu et al., 2016)

$$\gamma_{total} = \gamma_{spatial} + \gamma_{Doppler} + \gamma_{temporal} + \gamma_{thermal}$$

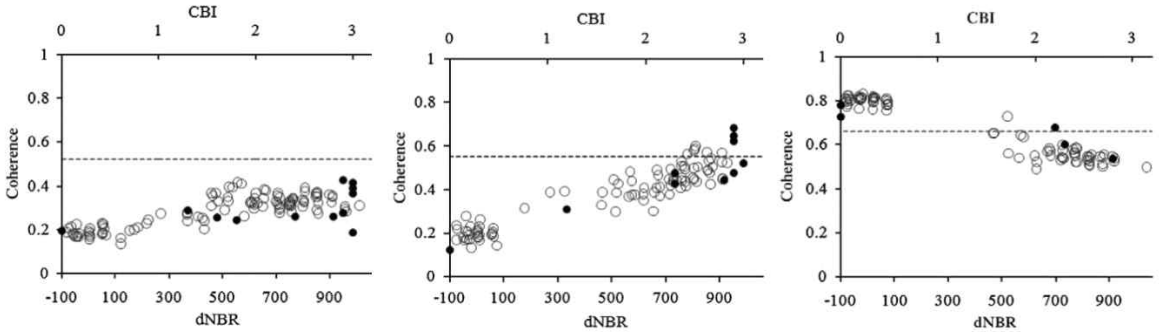
- $\gamma_{spatial}$은 기선거리 차이로 인한 상관관계 저하 효과를 나타냄
- $\gamma_{Doppler}$는 인공위성의 자세 차이로 인한 도플러 중심 상관관계 저하에 따른 효과를 나타냄
- $\gamma_{temporal}$은 시간 기선거리에 따른 상관관계 저하 효과를 나타냄
- $\gamma_{thermal}$은 센서의 신호 대 잡음 비율 (signal to noise ratio, SNR)에서 야기된 열 (thermal) 잡음을 나타내며, 일반적으로 무시됨
- 긴밀도는 0에서 1의 범위로 표현되며, 위상 오차가 클 경우 긴밀도 저하로 인해 고품질의 위상간섭도를 제작하는 것이 어려움. (Osmanoğlu et al., 2016)
- 하단 그림은 영상레이더 위상간섭기법 적용시 시간 기선거리별 긴밀도 저하 효과를 나타냄
- 시간 기선거리가 짧을수록 상대적으로 높은 긴밀도를 보여 영상에서 밝은 부분이 많이 관측되나, 시간이 지남에 따라 영상이 전체적으로 어두워짐을 확인 할 수 있음. 따라서 대상 지역을 자주 관측할수록 좋은 긴밀도를 가진 위상간섭영상을 제작할 수 있으며, 결과물의 품질에도

큰 영향을 주는 요인임



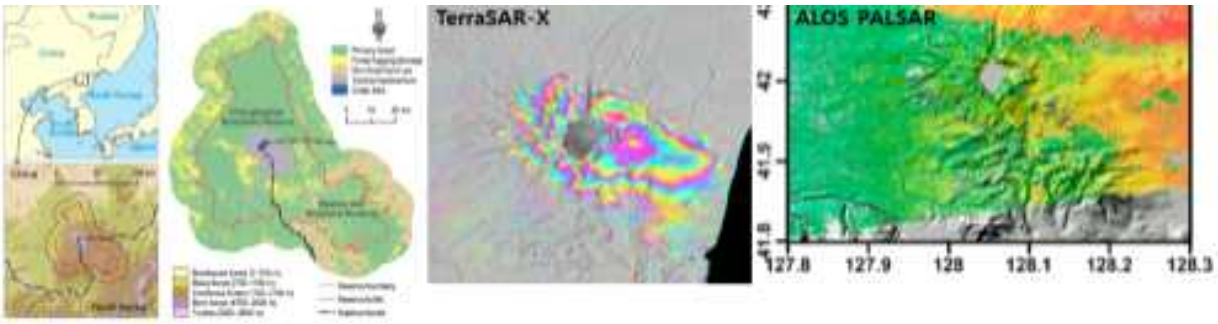
[$\gamma_{temporal}$ 에 따른 상관관계 저하 효과 비교 결과. 시간이 지남에 따라 영상 획득지역에서 전반적으로 긴밀도가 감소하는 경향을 볼 수 있음 (Moreira, Alberto, et al. 2013)]

- Tanase, Mihai A., et al. (2010)은 지중해 지역의 삼림 및 산불 피해 지역에 대해 X, C, L-밴드 영상레이더 위상간섭기법 적용을 통해 긴밀도 비교 및 분석을 수행함
- 연구 결과, 삼림 및 산불 피해 지역에서 L-밴드는 매우 높은 긴밀도를 보였으나, X 및 C-밴드는 낮은 긴밀도를 보이며 피해지역 조사 분석 수행 결과 상대적으로 낮은 정밀도를 보임
- 사면이 15°이상인 지역은 모든 파장대역에서 긴밀도가 감소하는 경향을 보였음
- 또한, 자료 획득 당시 기상 조건별로 긴밀도가 변화하였으며, 비올 경우 긴밀도가 저하되었음. 이는 지표면 수분에 따라 마이크로파의 반사 정도가 영향을 크게 받을 수 있음을 지시함



[삼림 지역의 긴밀도 비교, X-밴드(좌), C-밴드(중) 및 L-밴드(우) 간 긴밀도 비교 결과 (Tanase, Mihai A, et al. 2010)]

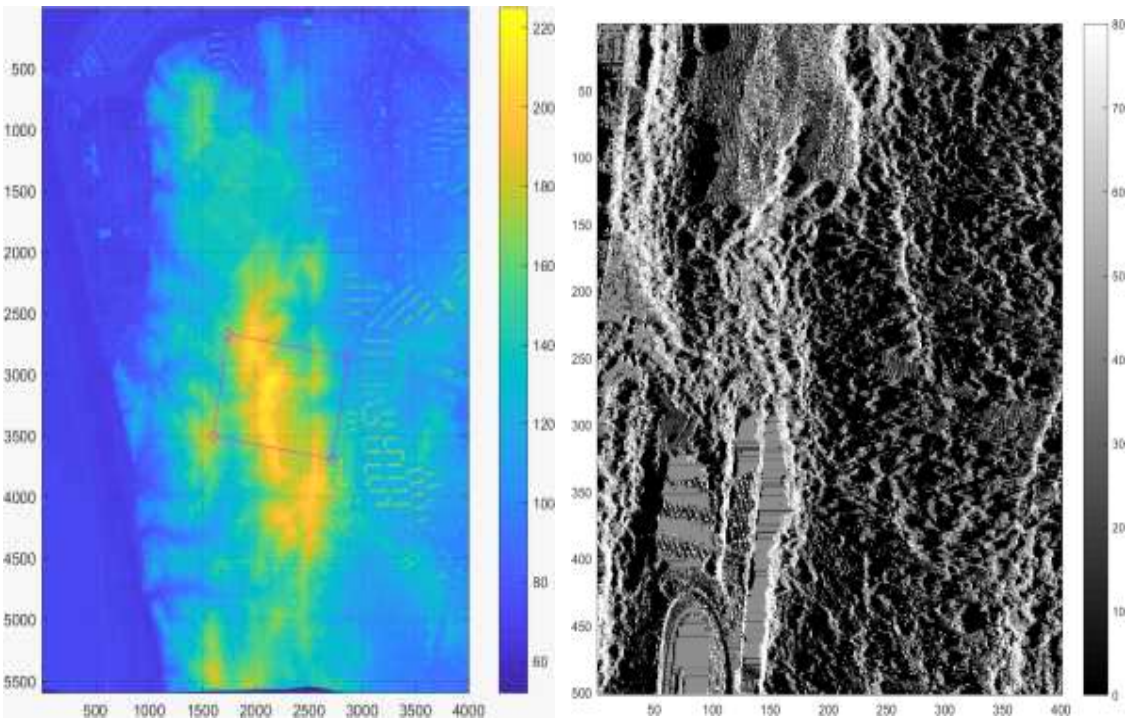
- 백두산과 같이 주변이 90%이상 산림으로 이루어져 있는 경우 영상레이더 위상간섭기법 적용 시 긴밀도 저하가 발생할 수 있으므로 파장대역 선정에 있어 주의가 필요함
- 단파장 영상레이더인 TerraSAR-X의 경우 백두산 주변부에서만 높은 긴밀도를 보여주며, 이는 백두산 천지 주변의 높은 해발고도로 인해 높이가 낮은 식생이 분포함이 주된 이유로 판단됨
- 백두산 주변지역은 산림으로 이루어져 긴밀도가 저하되는 모습을 보이나, 장파장 영상레이더인 ALOS PALSAR의 경우 영상 획득지역 전체에 걸쳐 높은 긴밀도를 유지하는 모습을 보여줌



[백두산 피복도(좌), X-밴드(중)와 L-밴드(우)로 생성된 백두산 위상간섭도]

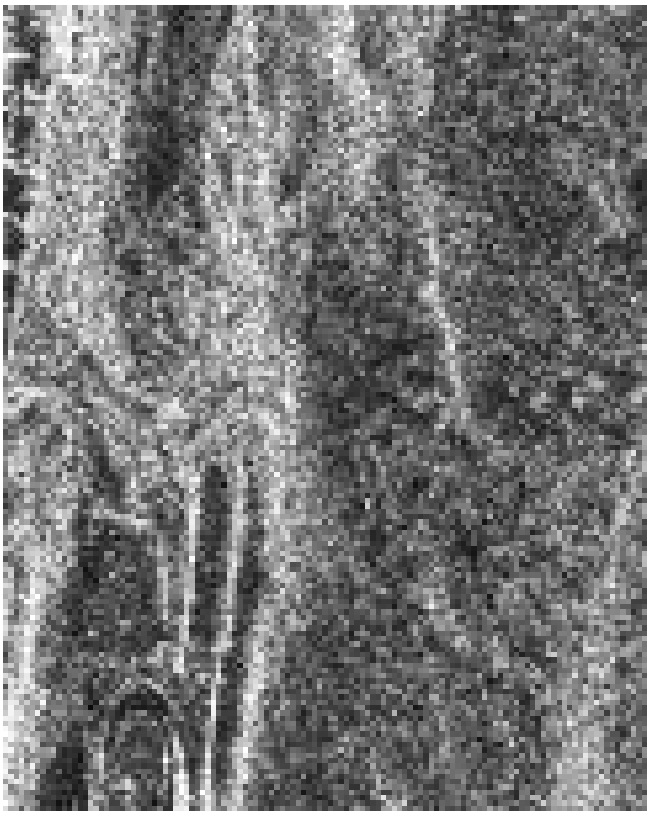
○ 드론에서 촬영된 SAR 영상의 좌표정확도 개선을 위한 알고리즘 개발

- 드론의 궤도정보만을 이용하여 지오코딩한 영상레이더 자료의 좌표는 드론 궤도정보 정확도에 따른 오차가 발생할 수 있음
- 변위 발생 위치를 포함하는 영상 전체의 지오코딩 정확도 개선을 위해 DSM과 드론 궤도를 이용한 SAR 영상 시뮬레이터 개발이 필요함
- 1 m 공간해상도의 DSM을 이용하여 SAR 영상 촬영경사각 약 40도, 공간해상도 1 m의 모사된 SAR 영상 생성할 수 있음



[사면을 포함한 DSM (좌), 개발시뮬레이터로부터 모사된 SAR 영상 (우)]

- 실제 드론으로 촬영된 SAR 영상과 모사된 SAR 영상간의 정합을 통해 좌표오차를 개선
- TerraSAR-X 영상과의 정합을 통해 azimuth 방향 14화소 (12 m), range 방향 16 화소 (15 m) 좌표오차 보정 수행
- 보정된 영상에서 두 개의 기준점을 통해 분석한 좌표오차는 최대 3 m 이내임



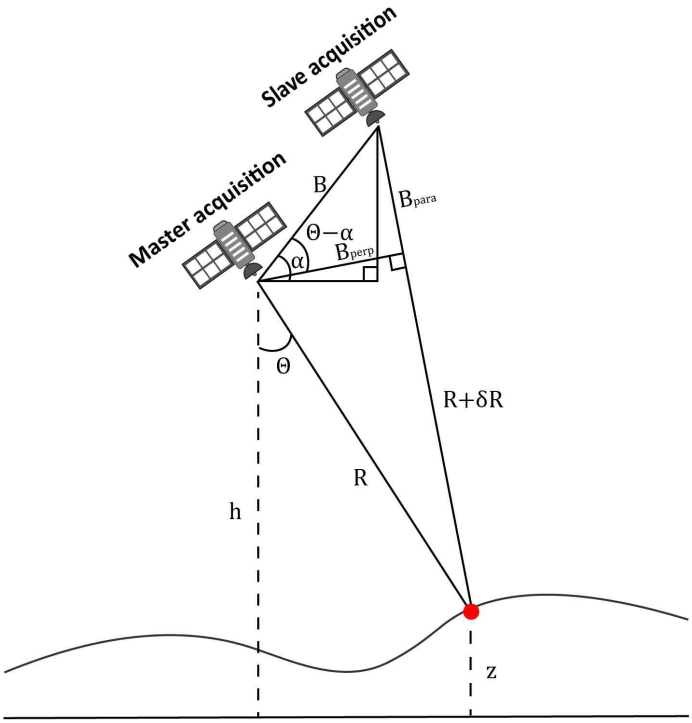
[TerraSAR-X SAR영상]

(2) 드론 촬영 비탈면 SAR 영상 기반 비탈면 변위 측정 기술 개발

- 능동형 원격탐사 센서 중 영상레이더는 마이크로파장대역의 전자기파를 사용하여 주·야간 및 기상 상황에 관계없이 지표 정보를 획득 가능함
- 영상레이더는 안테나로부터 신호를 송·수신하며, 획득된 신호는 진폭 (amplitude)과 위상 (phase)의 형태로 획득됨
- 서로 다른 위치에서 획득된 두 영상레이더 자료 간 위상차를 거리 차이로 변환하는 영상레이더 위상간섭기법 (SAR interferometry, InSAR)은 높은 경제적 및 시간적 효율성을 기반으로 지형 정보를 획득할 수 있음
- 위상간섭도 (Interferogram)에서 지형 정보에 대한 위상을 제거하여 mm ~ cm 스케일의 정밀한 지표 변위를 관측할 수 있는 차분위상간섭기법 (Differential SAR interferometry, DInSAR)은 지진, 화산활동, 지반침하, 습지 수위 변화, 빙하 이동 및 산사태 등 전 지구적인 지표 변위 변화 관측에 유용하게 사용됨
- 긴밀도 (coherence)는 위상간섭도의 품질을 평가할 수 있는 기준으로 영상레이더 신호의 획득 조건에 따라 다양한 형태로 영향을 받음
- 긴밀도는 두 영상레이더 자료에서 위상의 변화 유사성에 따라 결정되므로, 영상 획득 당시 시간 및 공간 기선거리에 영향을 받음
- 일반적으로 단파장 영상레이더는 높은 공간해상도를 지니나 장파장에 비해 투과성이 낮아 시간 기선거리에 따른 긴밀도 저하 효과가 더 크며, 이는 산림이 70% 이상으로 이루어진 우리나라의 경우 주요하게 고려되어야할 사항임
- 산사태가 영상레이더 파장 이상의 규모로 발생할 경우 긴밀도 저하로 인해 위상간섭기법을 통한 산사태 피해 규모 조사는 불가하므로 피해 규모나 이동 변위량에 따른 파장 선택도 중요함
- 위상간섭기법 적용이 불가능한 경우, 강도교차상관기법 (amplitude cross-corelation)을 적용하여 직접적인 변위를 계산할 수 있으며, 피해 지역에 대해 2차원 속도를 관측할 수 있는 오프셋 트래킹 (Offset tracking)기법을 적용하여 피해 규모 및 지역을 조사할 수 있음. 다만, 위상간섭기법에 비해 낮은 정밀도와 공간해상도를 지니므로 대규모 피해 조사에 활용 가능
- 최근 시간 기선거리와 공간해상도 개선을 통해 보다 높은 긴밀도를 지니는 위상간섭도를 획득하기 위해 항공, 드론 또는 지상 기반 (Ground-based) 영상레이더 원격탐사가 수행되고 있음

○ 영상레이더 위상간섭기법 (SAR Interferometry, InSAR)

- 영상레이더 위상간섭기법을 통해 서로 다른 영상레이더 자료 간 위상 차이는 산란체와 안테나 간 거리 차이로 표현 가능함 (Hanssen, 2001)
- 영상레이더 위상간섭기법의 자료 획득 방법은 안테나 구성 및 관측 방법에 따라 단일 경로 (single-pass) 와 반복 경로 (repeat-pass)로 분류됨
- 단일 경로는 하나의 플랫폼에 두개의 안테나가 탑재되어 하나의 안테나에서 송신된 신호가 두 개의 안테나로 수신되어 자료를 획득하는 관측 방법이며, Shuttle Radar Topography Mission (SRTM)과 같이 수치표고모형 제작 시 주로 사용됨 (Zebker et al., 1994)
- 반복 경로는 위상간섭기법을 통한 지표 변위 관측 시 주로 사용되는 방법으로서 각각의 플랫폼에 안테나가 탑재되어 서로 다른 비행체 또는 궤도에서 획득한 자료를 이용하는 방법임 (Moreira et al., 2013)
- 위상 차이는 안테나로부터 산란체 사이의 거리 뿐 아니라 영상레이더 안테나 간 기선거리, 관측각, 지구 타원체 및 지형 고도에 영향을 받음 (Hanssen, 2001)
- 아래 그림은 영상레이더 안테나의 관측 기하를 나타내며, B는 영상 간 기선거리를, B_{perp}는 수직 기선거리를, B_{para}는 수평 기선거리를, θ는 인공위성의 관측각도를, α는 안테나의 수직선과 기선이 이루는 회전각을, h는 지구타원체로부터 위성의 고도를, z는 지형고도를 나타내며 R과 $R+\delta R$은 안테나와 산란체 사이의 경사 거리를 나타냄 (Hanssen, 2001)



[영상레이더 위상간섭기법의 관측 기하 (Hanssen, 2001)]

- 영상레이더 안테나에서 획득된 신호는 진폭 $|v|$와 위상 φ의 정보를 지니는 복소수 형태의 자료로 획득됨 (Hanssen, 2001)

$$v_1 = |v_1| e^{j\varphi_1}$$
$$v_2 = |v_2| e^{j\varphi_2}$$

- 상기 식에서 서로 다른 시기에 획득된 두 영상레이더 자료를 동일한 위치로 정리하는 정합 (co-registration) 및 재배열 (resampling) 과정을 거친 후 켤레 복소수 곱을 통해 두 자료 간 위상의 차이를 나타내는 위상간섭도를 제작할 수 있음 (Hanssen, 2001)

$$v = v_1 v_2^* = |v_1||v_2| e^{j(\varphi_1 - \varphi_2)}$$

- 안테나와 산란체 간 거리를 고려하여 두 영상레이더 자료에서 획득되는 위상은 다음의 식으로 표현 가능함

$$\varphi_1 = -\frac{4\pi R}{\lambda} + \varphi_{scatterer,1}$$
$$\varphi_2 = -\frac{4\pi (R+\delta R)}{\lambda} + \varphi_{scatterer,2}$$

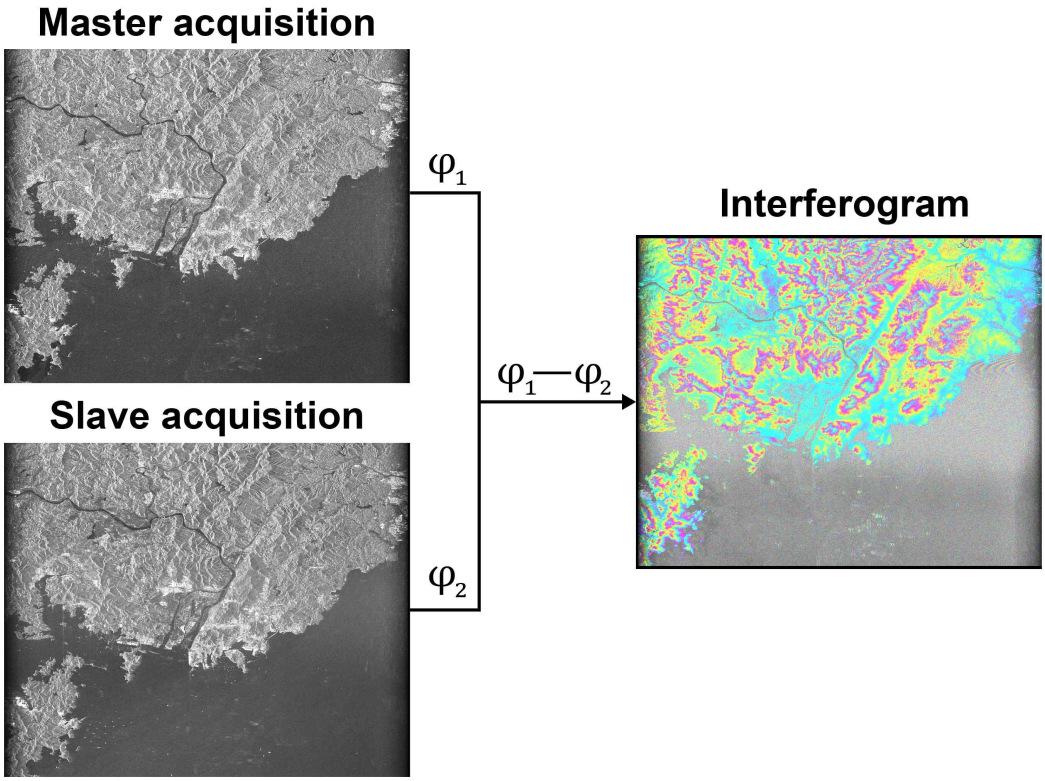
- 위상 차이는 지형 정보, 지표 변위, 지구 타원체, 위성간의 궤도, 대기 및 산란체의 전기적·유전학적 특성에 대한 성분으로 구성됨 (Colesanti et al., 2003; Ferretti et al., 2001).

$$\triangle\varphi = \varphi_1 - \varphi_2$$
$$\varphi_{flat} + \varphi_{topo} + \varphi_{\mathrm{orbit}} + \varphi_{disp} + \varphi_{tropo} + \varphi_{iono} + \varphi_{scatterer} + \varphi_{noise}$$

- φ_{flat}, φ_{topo}, φ_{orbit}, φ_{disp}, φ_{tropo}, φ_{iono}, $\varphi_{scatterer}$, φ_{noise}는 각각 지구 타원체, 지형 정보, 궤도 정보, 지표 변위, 대류권, 전리층, 산란체의 전기적 특성 및 잡음과 관련된 위상 성분을 나타

냄 (Osmanoğlu et al., 2016)

- 다음 그림은 영상레이더 위상간섭기법 적용의 모식도 이며, 제작된 초기 위상간섭도를 나타내며 지형에 대한 위상 정보를 포함하고 있음. 해당 위상간섭도는 지형에 대한 고도 정보를 포함하고 있으므로 절대 위상 보정 작업을 수행하는 경우 지형고도모형 제작에 활용할 수 있음



[영상레이더 위상간섭기법 적용 및 초기 위상간섭도 제작 사례]

- 영상레이더 위상간섭기법의 관측 기하에서 δR은 신호 획득 당시 안테나의 위치 차이로 인해 발생하며, 자료 획득 기하 구조에서 코사인 제 2 법칙과 근사를 통해 아래와 같이 표현됨 (Zebker & Goldstein, 1986)

$$\delta R = -B\sin(\Theta-\alpha)$$
$$\triangle\varphi = -\frac{4\pi}{\lambda}B\sin(\Theta-\alpha)$$

- 경사 거리에 따른 위상의 변화율은 다음의 식으로 표현되며, 위상 변화는 수직 기선거리와 경사 거리에 따른 관측각의 변화율에 영향을 받음

$$\frac{\partial\triangle\varphi}{\partial R} = -\frac{4\pi}{\lambda}B\cos(\Theta-\alpha)\frac{\partial\Theta}{\partial R} = -\frac{4\pi}{\lambda}B_{perp}\frac{\partial\Theta}{\partial R}$$

- Zebker et al.(1994)에 따라 지형 고도에 따른 위상 변화율을 아래와 같이 정리하여 지형 고도에 의한 위상의 민감도 (height sensitivity)를 계산 가능함

$$\frac{\partial\triangle\varphi}{\partial z} = -\frac{4\pi}{\lambda}\frac{\mathrm{B}\cos(\Theta-\alpha)}{R\sin\Theta}$$

- 영상레이더 자료 획득 기간 동안 지표변위 $\triangle R$이 발생한 경우, 위상간섭기법 적용을 통한 위상 차이와 지표 변위에 따른 위상의 변화율은 다음의 식으로 표현됨

$$\triangle\varphi = -\frac{4\pi}{\lambda}[B\sin(\Theta-\alpha)+\triangle R]$$

$$\frac{\partial\triangle\varphi}{\partial\triangle R} = -\frac{4\pi}{\lambda}$$

- 위 식들을 비교하였을 때, $\frac{B}{R} \ll 1$이므로 $\triangle\varphi$에서 표고에 의한 위상 변화는 지표 변위에 의한 위상 변화보다 민감도가 훨씬 작음 (Hanssen, 2001)
- 따라서 지형에 의한 위상을 m 단위 까지 측정할 수 있다면 변위에 의한 위상은 cm 에서 mm 단위까지 측정할 수 있으므로, 차분위상간섭기법 적용을 통해 정밀한 지표 변위를 획득 가능함
- 위상간섭도로부터 지표 변위를 획득하는 차분위상간섭기법은 초기 위상간섭도로부터 지구 곡률 위상과 지형 정보 위상 제거를 통해 적용 가능함
- 상기 식들의 비교 결과 지형 정보와 지표 변위가 위상에 끼치는 영향력이 다름을 알 수 있으며, 2-pass 차분위상간섭기법의 경우 cm 에서 mm 단위의 지표 변위를 관측하기 위해서는 m 급 공간 해상도를 지니는 수치표고모형을 이용하여 관측 가능함 (Hanssen, 2001).

○ 지구 곡률 위상
- 두 영상레이더 자료 간 위상 차이를 계산하여 제작된 초기 위상간섭도는 다양한 위상으로 구성되어 있으며 특히, 지구 곡률로 인해 영상 전체에 걸쳐 경사 거리 변화에 따른 지구 곡률 (flat earth) 위상이 생성됨
- 해당 위상을 제거하는 과정을 위상간섭도 플래트닝 (earth flattening removal)이 수행됨
- 지구 곡률 위상은 다음과 같이 나타낼 수 있으며, 계산 시 수직 기선거리를 사용하기 때문에 상태 궤도 벡터 (state orbit vector)를 이용한 정확한 궤도 보정이 선행됨 (Hanssen, 2001)
- 따라서 정밀 위상 보정을 위해서는 센서를 탑재한 이동체의 정밀 궤도 관측이 중요함

$$\frac{\partial\triangle\varphi_{flat}}{\partial R} = -\frac{4\pi}{\lambda R\tan\Theta}B_{perp}$$

○ 지형 정보 위상
- 위상간섭도 플래트닝을 통해 지구 타원체 위상이 제거된 위상간섭도의 모습은 등고선 지도처럼 반복되는 줄무늬를 보이며, 이를 지형 프린지 (topographic fringe)라 함
- 위상간섭도에서 지형 프린지는 2π 주기로 반복되며 한 프린지에서 표현 가능한 지표 높이를 고도의 모호성 (height ambiguity)으로 표현됨 (Massonnet et al., 1993)
- 고도의 모호성(h_a)는 높이 변화량에 대한 위상 변화량으로 다음의 식으로 표현됨

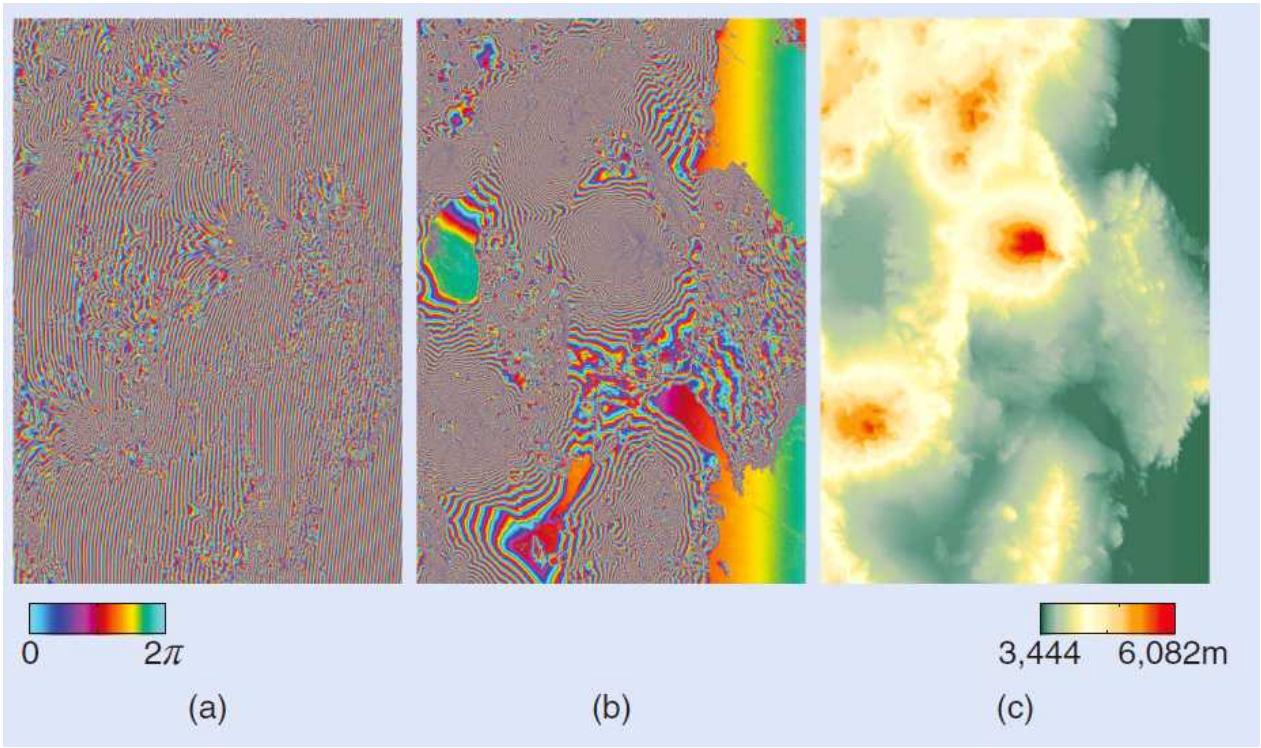
$$\frac{dz}{d\varphi_{topo}} = -\frac{\lambda}{4\pi}\frac{R\sin\Theta}{B_{perp}}$$

- $\partial\varphi = 2\pi$일 때, 즉 1 프린지에 해당하는 h_a는 다음과 같이 표현됨

$$h_{a,2\pi} = -\frac{\lambda}{2}\frac{R\sin\Theta}{B_{perp}}$$

- 안테나 간 기선거리가 증가할수록 h_a가 작아져 한 프린지 당 고도를 보다 세밀하게 표현가능
- 반대로 B_{perp} 값이 감소하면 h_a가 커져 한 프린지 당 표현 가능한 고도의 스케일이 줄어듦
- 임계 기선거리 (critical baseline)이하에서 수직 기선거리가 증가할수록 세밀한 고도 변화를 프린지로 표현이 가능해 보다 해상도 높은 수치표고모형 제작이 가능함
- 반대로, 수직 기선거리가 감소할수록 지형 위상의 영향이 줄어 차분위상간섭기법 적용에 용이함 (Massonnet et al., 1993)

- 따라서 수직 기선거리가 0인 경우 지형에 대한 위상정보를 제거하지 않아도 계산된 위상간섭 정보는 해당 지역의 변위 정보를 반영하게 됨



[TanDEM-X 영상레이더 자료의 위상간섭기법 적용 결과. (a) 초기 위상간섭도. 지구 곡률 위상으로 인해 관측 방향으로 주기적인 fringe가 생성된 것을 볼 수 있음 (b) 지구 곡률 위상을 제거한 위상간섭도 (c) 지형 고도 위상을 언래핑 하여 획득된 수치 표고 모형 (Moreira, Alberto, et al. 2013)]

○ 지표 변위 위상

- 지구 곡률 위상을 제거한 위상간섭도에서 대기 효과, 산란체의 전기적 특성 및 잡음이 없다는 가정 하에 위상 변화는 지형 정보 위상과 지표 변위 위상으로 이루어짐

$$\triangle\varphi = -\frac{4\pi}{\lambda}\frac{\mathrm{B_{perp}}z}{R\sin\Theta} + \frac{4\pi}{\lambda}\triangle\mathrm{R}$$

- 상기 식에서 지형 정보 위상은 수치표고모형을 이용하는 2-pass 차분위상간섭기법을 적용하여 제거 가능하며, 다음과 같이 지표 변위 위상을 계산할 수 있음

$$\triangle\varphi_{disp} = \frac{4\pi}{\lambda}\triangle R \;\rightarrow\; \triangle R = \frac{\lambda}{4\pi}\triangle\varphi_{disp}$$

- 위 식에서 $\triangle\varphi_{disp} = 2\pi$일 때 $\frac{\lambda}{2}$만큼의 지표 변위를 나타내며, 한 프린지 주기로 표현됨
- 차분위상간섭도에서 한 프린지 당 지표 변위는 L-밴드와 C-밴드에서 각각 약 12 및 3 cm로 환산되며 이는 파장의 1/2에 해당되는 양임

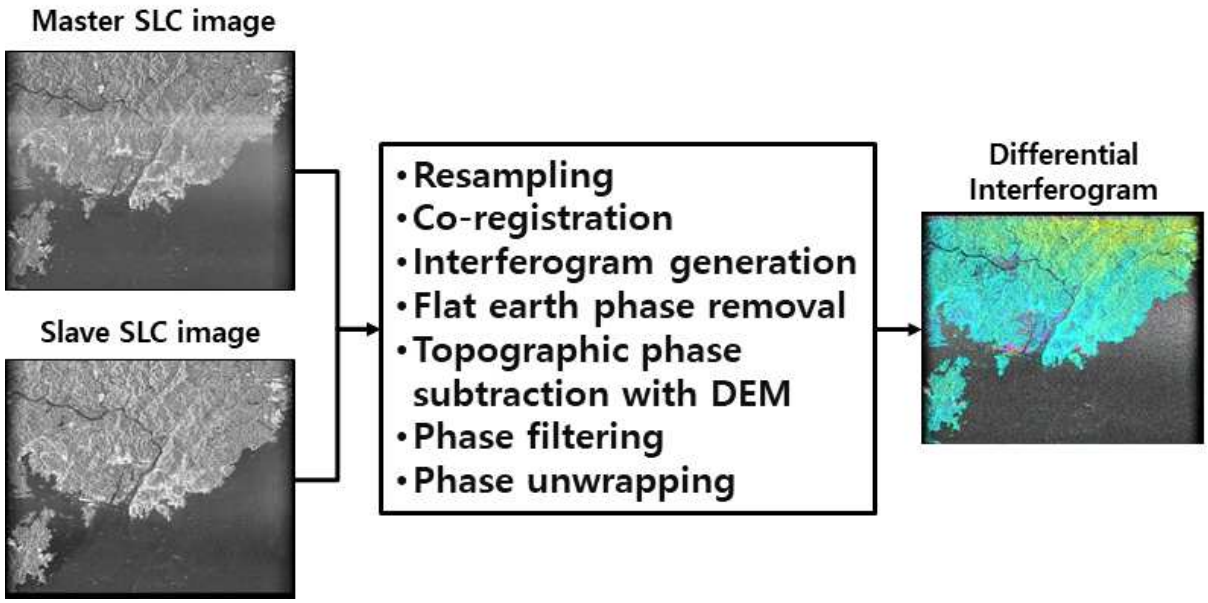
○ 영상레이더 차분위상간섭기법 및 사례 분석

- 위상간섭기법 적용 시 두 영상레이더 자료 간 지표 변위가 발생할 경우 지표 변위 위상 성분이 포함되며, 획득된 위상간섭도로부터 지구 곡률 위상과 지형 정보 위상을 제거하여 지표 변위 위상을 획득 가능함
- 차분위상간섭기법을 통해 지표 변위를 측정하는 방법은 2-pass, 3-pass와 4-pass 차분위상간섭기법이 있으며 (Gabriel et al., 1989; Massonnet and Feigl, 1998; Massonnet et al., 1993), 지속적인 지반침하 관측에는 2-pass 차분위상간섭기법이 가장 효과적임 (Massonnet and Feigl, 1998)

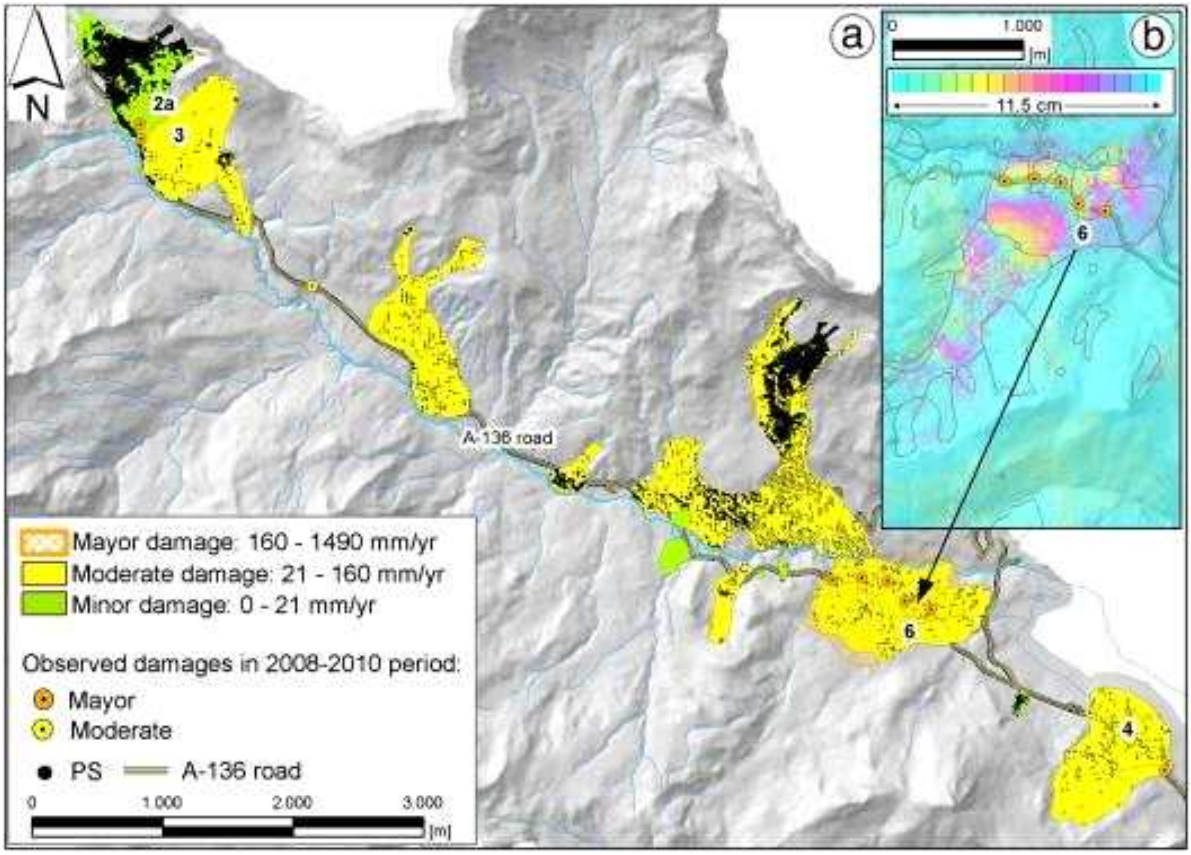
- 아래의 표는 차분위상간섭기법의 종류를 요약한 것이며, 영상레이더 차분위상간섭기법 적용에 대한 순서 및 결과의 모식도를 나타냄
- 2-pass 차분위상간섭기법은 위상간섭도의 지형 정보 위상을 제거하기 위해 제작되어있는 수치표고모형을 사용함
- 3,4-pass 차분위상간섭기법은 위상간섭도의 지형 정보 위상을 제거하기 위해 3장 또는 4장의 영상레이더 자료를 활용하여 더 긴 공간기선거리를 가지는 위상간섭도로부터 지형 정보를 추출하여 지표 변위를 계산함

[차분위상간섭기법의 종류 (Wegmuller et al., 1998)]

차분위상간섭기법의 종류	영상레이더 자료 수	수치표고모형	언래핑
2-pass DInSAR	InSAR pair (2 SAR images)	External DEM	No
3-pass DInSAR	2 InSAR pair (3 SAR images)	InSAR pair	Yes
4-pass DInSAR	2 InSAR pair (4 SAR images)	InSAR pair	Yes



[영상레이더 차분위상간섭기법 적용 과정 및 순서도]



[측정된 평균 산사태 속도 기반한 산사태 피해지도 (Herrera, G. et al., 2013)

- Herrera, G., et al.(2013) 은 스페인 중부 피레네 산맥의 Tena Valley에 대해 L-, C- 및 X-밴드를 사용한 다중 센서 및 다중 시간 SAR 데이터 조합으로 매우 느린 산사태의 모니터링 능력을 향상시킬 수 있는지에 대해 관찰함
- D-GPS 측정과 비교 및 검증을 수행하였으며 Tena Valley에서 발생한 산사태의 38%를 관측할 수 있었으며 산사태 속도는 관측된 도로 피해와 관련하여 분류하였음
- 이와 같은 결과는 위성 관측의 제한성을 지시하는 한 사례로 분석 가능함

○ 대기 위상

- 영상레이더 자료 획득 당시 서로 다른 대기 환경으로 인해 위상간섭기법 적용 시 대기 효과에 의한 위상이 존재하며 센서와 대상체 사이의 거리가 길수록 큰 영향을 끼칠 수 있음
- 대기 효과는 대기 중의 온도, 압력, 수증기압의 변화에 의해 위상 지연을 야기함
- 수증기에 의한 위상 지연이 가장 큰 영향을 미치며, 약 20 %의 상대 습도 변화는 10 cm 정도의 변위 오차를 야기할 수 있음 (Hanssen, 2001; Zebker et al., 1997)
- 대기 중의 수증기 분포로 인한 위상 지연 효과는 대류권에서 수직적인 분포로 인해 지표 높이에 따른 위상변화를 동반할 수 있음
- 대기 효과에 의한 위상 지연은 평지보다 지표 고도 차이가 큰 지역에서 더욱 커져 히말라야와 같은 고산지대에서 지진으로 인한 지표변위 관측하는 경우 보다 정밀한 결과를 얻기 위해서는 대기 위상이 필수적으로 고려되어야 함 (Jolivet et al., 2011)
- 위성 항법 시스템을 이용한 위상 지연 측정과 지역 대기 관측자료 등을 통해 효과적으로 대기 위상을 제거 할 수 있으나, 영상레이더 관측 지역에 비해 해당 지역에 관측 자료의 수가 부족하고 해상도의 차이가 있어 적용이 어려울 수 있음 (Jolivet et al., 2011)
- 일반적으로 위상과 고도의 관계를 이용한 경험식을 통해 대기 지연 위상을 계산할 수 있으나, 지형 고도가 높은 지역과 같이 지표 변위 위상과 지형 위상이 서로 연관될 때 궤도와 지표 변위 정보를 이용한 대기 위상 추출은 어려움 (Jolivet et al., 2011)
- Jolivet et al. (2011)에 의해 제시된 대기 효과 보정 방법은 유럽 중기 예보 센터에서 제공된 대기 분석 자료를 사용하여 차분위상간섭도에서 대기 위상을 제거할 수 있으며, 관측 방향의

위상 지연 ($\delta L^{s}_{LOS(z)}$)은 다음과 같이 표현 가능함 (C3S, 2017; Jolivet et al., 2011)

- Jolivet et al. (2011)은 대기 위상 제거를 위해 대기 성분을 건조 공기와 수증기 성분으로 나누어 분석함 (Jolivet et al., 2011)

$$\delta L^{s}_{LOS}(z) = \frac{10^{-6}}{\cos(\Theta)} \left\{ \frac{k_1 R_d}{g_m} (P(z) - P(z_{ref})) + \int_{z}^{z_{ref}} \left(\left(k_2 - \frac{R_d}{R_v} k_1 \right) \frac{e}{T} + k_3 \frac{e}{T^2} \right) dz \right\}$$

Θ = Local incidence angle $\quad k_1 = 0.776\,K/Pa$

$R_d = 287.05\,J/kg \cdot K \quad k_2 = 0.716\,K/Pa$

$R_v = 461.495\,J/kg \cdot K \quad k_3 = 3.75 \cdot 10^3\,K^2/Pa$

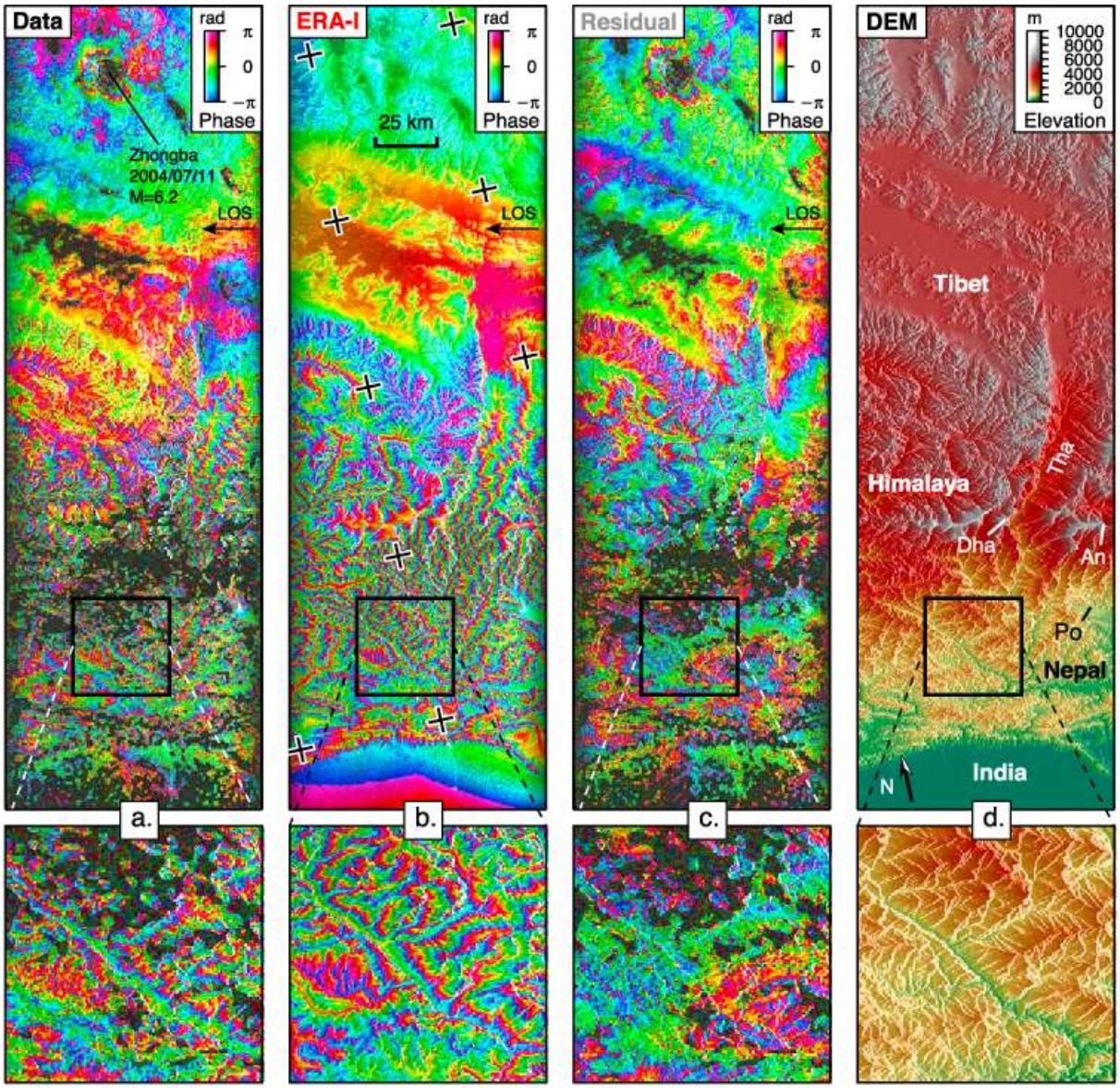
z = Surface elevation $\quad P$ = Partial pressure of the dry air

z_{ref} = Height above which the delay is assumed to be nearly unchanged with time (typically 10000 m).

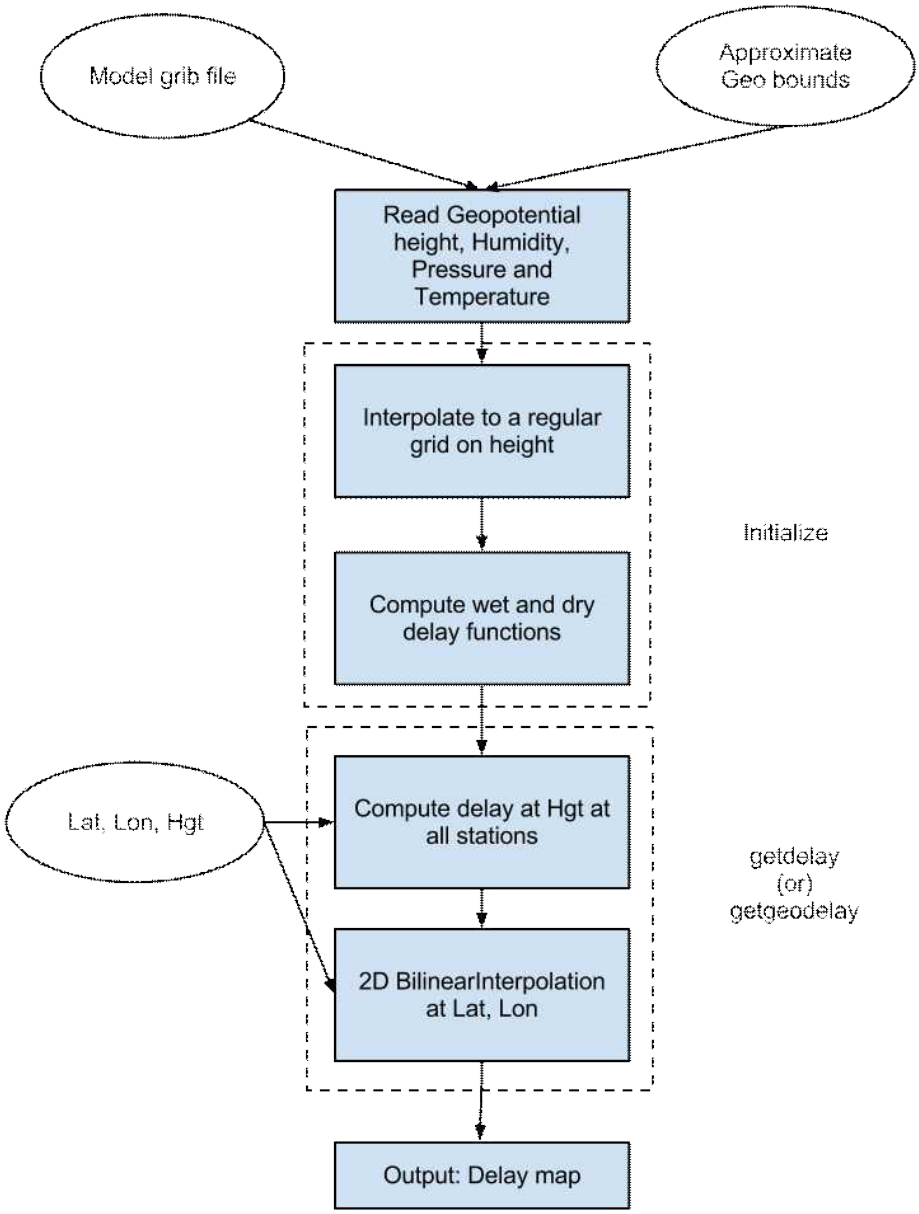
g_m = Weighted average of the gravity acceleration between z and z_{ref}

e = Partial pressure of the water vapor specific gas constants

- 위 식에서 첫 번째 항은 신호 지연에 따른 건조 성분을, 뒤의 항은 신호 지연에 따른 수증기 성분을 계산하여 관측 방향에서 지형 고도에 따른 대기 위상 지연 성분을 계산함
- 일반적으로 대기압 변화로 인한 건조 성분 계산 항의 변화는 매우 적으며, 수증기 성분을 계산할 수 있는 항을 통해 대기 위상 지연 성분이 결정됨 (Jolivet et al., 2011)
- 대기 보정을 위해 유럽 중기 예보 센터에서 제공된 1시간의 시간해상도와 31 km의 공간해상도를 가지는 대기 분석 자료 (ERA-I)를 상기 식을 기반으로 Jolivet et al. (2011)에 의해 개발된 Python based Atmospheric Phase Screen Estimation (PyAPS) 프로그램을 통해 대기 위상 지연을 계산가능함 (Jolivet et al., 2011)
- 아래의 그림은 히말라야 지역에서 PyAPS 프로그램을 사용하여 계산된 대기 지연 위상을 위상간섭도와 비교한 결과를 나타냄
- 위상간섭도와 대기 위상 지연 지도 간 차이를 계산하여 정밀한 지표 변위를 획득 가능함

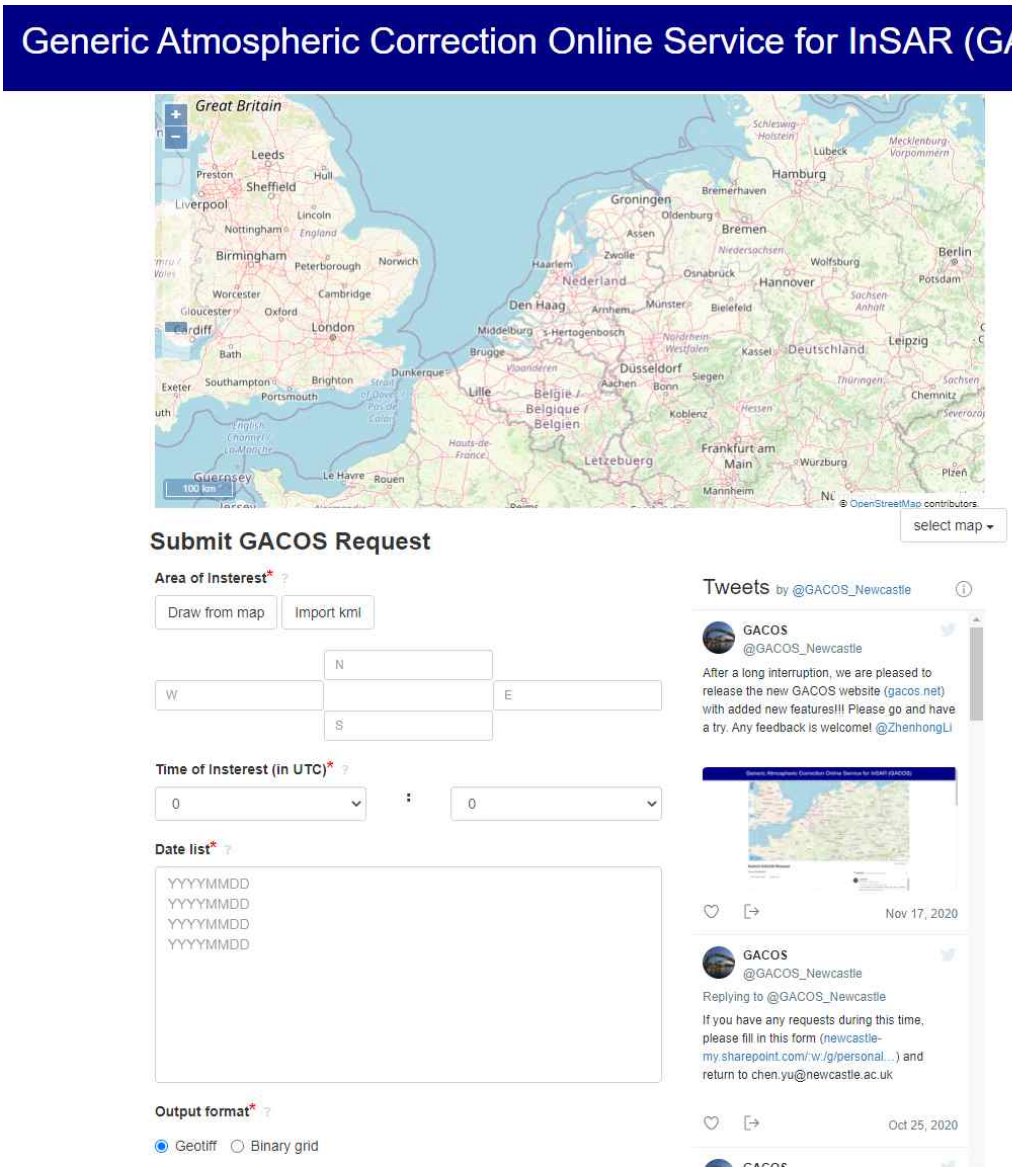


[히말라야의 간섭도 및 대기보정 예 (Jolivet et al., 2011): (a) SAR 영상으로 제작된 wrapped interferogram. (b) ERA-I 데이터로부터 예측된 계층화된 대기 위상 지연 지도. (c) 그림 a에서 b를 뺀 잔여 위상, (d) SRTM DEM]



[PyAPS 대기보정 방법 워크플로우 (Jolivet et al., 2011)]

- 위의 그림은 대표적인 대기위상 보정 알고리즘인 Python based Atmospheric Phase Screen Estimation (PyAPS)의 자료 처리 과정을 나타내며, 대기 모델로부터 2차원 대기 지연 위상 지도를 획득할 수 있음
- 대기 위상 지연 효과를 계산할 수 있는 또 다른 서비스인 General Atmospheric Correction Online Service (GACOS) 는 대기 효과에 의한 간섭도 보정에 사용하는 서비스로 반복적 대류권 분해 모델을 사용하여 ECMWF 수치 날씨 모델 및 높은 정확도의 GPS 포인트 단위 Zenith tropospheric delay (ZTD)를 통합하여 제공함
- Iterative Tropospheric Decomposition (ITD) 모델을 사용하여 대기 지연을 계층화된 구성 요소와 난류 성분으로 분리하여 대류권 보정 맵을 생성하여 대기 보정을 수행함
- GACOS는 범지구적 적용이 가능하고, 전천후로 사용 가능하며 실시간 보정과 보정 성능 평가 지표를 가져 보정 타당성 평가를 수행할 수 있다는 장점이 있음



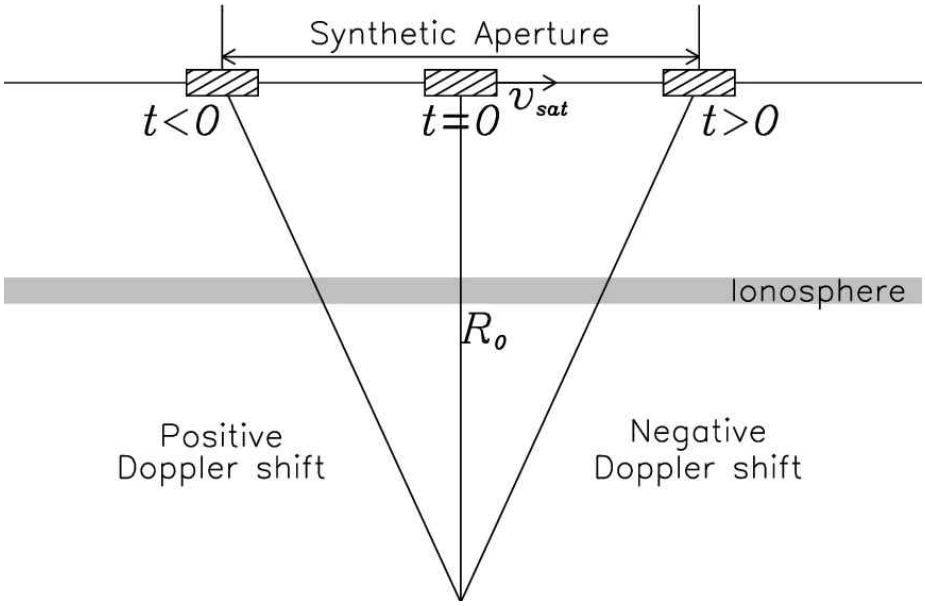
[GACOS 서비스 제공 사이트 (http://www.gacos.net/)]

○ 이온 지연 오차

- 이온층은 태양 에너지에 의해 기체 분자들과 몇몇 원자들이 이온화 (ionization)되어 자유 전자가 밀집된 층임
- 영상레이더 인공위성에서 안테나로부터 송수신되는 신호는 이온층을 통과하게 되는데 이 때 이온층 왜곡으로 인해 줄무늬 형태의 패턴으로 이온층 오차를 야기할 수 있음
- 이온층 내 자유전자 밀도는 태양활동, 계절, 시각, 위도 및 경도 등 여러 조건에 영향을 받음
- 극지방과 적도지방의 경우 지구 자기의 영향을 받아 전자 밀도의 차이가 있으며, 저위도 적도지방의 경우 전리층의 불규칙성이 높으며 고위도는 강하입자에 의한 전리작용으로 인해 이온층 왜곡의 해석이 어려움
- 이온층으로 인한 지연 효과는 영상레이더 파장과 비례하게 발생함
- 일반적으로 영상레이더 자료 획득 시 이온층 1 Total Electron Content Unit (TECU) 차이에 의해 X-밴드는 0.3 phase, C-밴드는 0.5 phase 및 L-밴드는 2 phase의 위상 지연 오차가 발생하는 것으로 알려져 있음 (Meyer et al., 2011)
- L-밴드와 같이 장파장 영상레이더의 경우 단파장에 비해 높은 긴밀도를 유지할 수 있어 산사태 관측에 매우 효율적으로 사용될 수 있으나, 이온 효과를 고려하지 않을 경우 정밀한 피해 규모 확인에 어려움이 있음
- 이온층을 통과하는 전파 지연은 다음과 같음 (Jakowski et al., 1992)

$$\triangle x_{\text{이온층}} = -40.28\int_0^x \frac{EC}{f^2}dx = -40.28\frac{TEC}{f^2}$$

- 상기 식에서 f는 주파수, EC는 단위 부피당 전자의 양, x는 전파 경로를 뜻함. total electron content (TEC)는 전파 경로에 따른 단위면적당 EC의 합이며, 해당 식을 통해 신호 획득 시 지연 효과를 고려할 수 있음



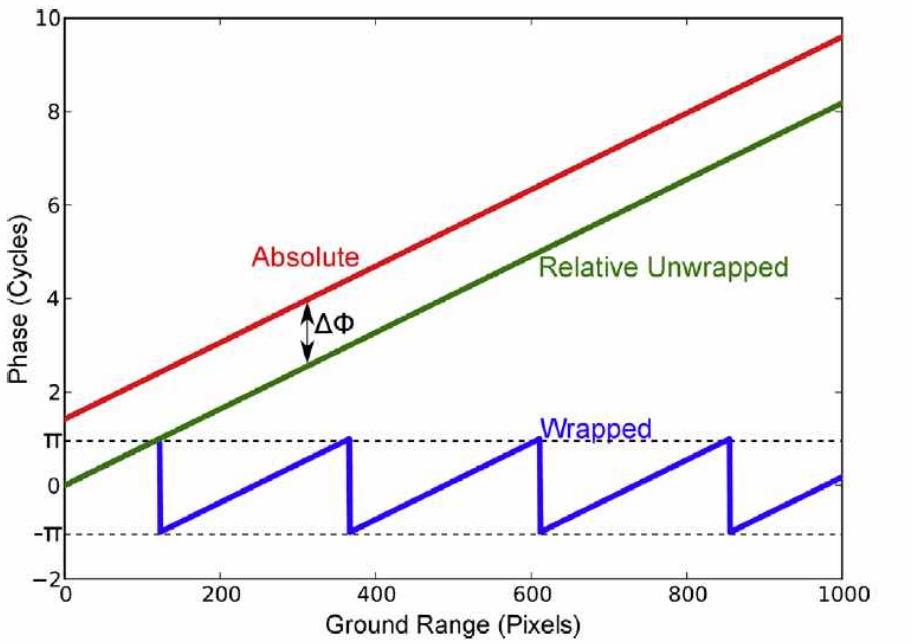
[이온층으로 인한 이온 지연 오차 발생 모식도]

○ 위상 언래핑

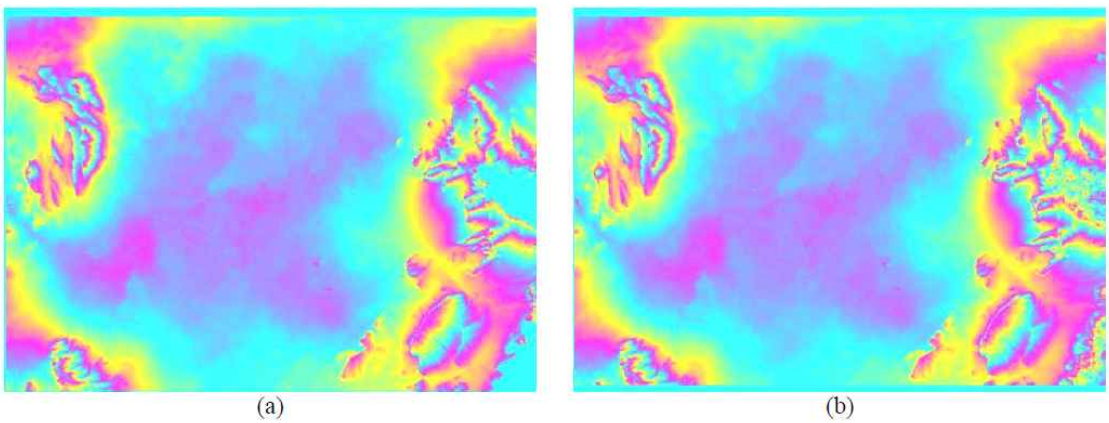
- 위상간섭기법을 통해 제작된 위상간섭도는 $(-\pi, \pi]$ 구간에서 반복적으로 표현되며, 이를 wrapped 되었다고 함
- 간섭도에서 반복되는 상대적인 위상 값을 절대적인 위상 값으로 unwrapped 하는 과정을 위상 언래핑이라 하며, 해당 과정을 통해 위상간섭도에서 전체적인 위상 변화를 계산할 수 있음 (Hanssen, 2001; Tribolet, 1977)
- 이상적인 위상 언래핑은 wrapped 된 위상 값은 인접한 픽셀에서 위상 차이가 π를 넘지 않아 단순 적분을 통해 절대 위상을 획득할 수 있음
- 그러나 위상간섭도에서 위상의 오차, 언더샘플링 (undersampling)과 불연속성 문제로 인해 정확한 위상 언래핑이 매우 어려움
- 해당 요소들을 고려한 다양한 위상 언래핑 알고리즘 방법이 제시되었으며, 대표적으로 경로 의존 (path following)과 경로 독립 (path independent) 방법이 존재하고 경로 의존 방법이 대중적으로 사용됨 (Costantini, 1998; Goldstein & Werner, 1998; Zebker & Lu, 1998)
- 경로 의존은 일반적인 위상 언래핑 방법으로 기준점으로부터 주변 픽셀과의 위상 차이를 계산하여 불연속적인 위상 값을 연속적으로 표현하는 방법으로서 브랜치-컷 (branch-cut) 알고리즘과 최소 비용 흐름 (Minimum Cost Flow, MCF) 알고리즘이 대표적임 (Costantini, 1998; Zebker & Lu, 1998)
- 브랜치-컷 알고리즘은 간섭도에서 위상 차이의 회전 합계를 사용하여 잔여 (residues)를 계산하며, 잔여는 위상 언래핑시 자료의 노이즈 또는 불연속성에 의해 발생하는 지역 오류로서 회전 합계의 계산에서 부호가 바뀌는 자리 또는 영상의 가장자리와 잔여 사이에 브랜치와 컷을 놓아 위상의 연속성을 조사함
- 위상 언래핑은 잔여를 통과하지 않는 브랜치를 따라 이루어지며, 유일 해를 제공함 (Zebker & Lu, 1998)
- 최소 비용 흐름 알고리즘은 개별 픽셀의 품질을 나타내는 지도를 이용하여 비용 계산을 통해 높은 품질을 가지는 픽셀을 따라 위상 언래핑을 수행하는 방법이며, 품질이 높을 경우에는 저비용, 낮을 경우에는 고비용으로 인식하여 가장 낮은 비용을 가지는 언래핑 경로를 선택함

(Costantini, 1998)

- 경로 독립 방법은 unwrapped 된 결과에서 전역 해 (global solution)를 최소화하는 방법을 이용하기 때문에 언래핑 경로를 사용하지 않음
- 대표적인 방법으로는 최소 제곱법 (least-squares)이 있으나 경로 의존 방법이 주로 사용되거나 두 방법을 혼용하여 사용하는 것이 일반적임 (Ghiglia & Romero, 1996)
- 아래의 그림은 위상 언래핑 과정을 나타낸 모식도이며, 반복적으로 표현된 위상을 언래핑한 후 절대적인 위상으로 보정하는 작업을 거쳐 영상에서 전체적인 위상 변화를 계산할 수 있음
- 영상레이더 위상간섭기법 적용 후 언래핑된 위상을 서로 다른 알고리즘을 적용한 결과들이 도시되었으며 영상 우측과 같이 위상간섭도에서 다소 긴밀도가 저하되거나, 상관관계가 낮은 지역에서 branch-cut 알고리즘보다 MCF 알고리즘이 더 정밀한 결과를 보여줌



[위상 언래핑 과정을 나타낸 모식도. 파란색 선은 wrapped 된 상태의 값을 나타내며, 초록색과 빨간색 선은 각각 unwrapped된 상대적인 위상과 절대 위상을 나타냄 (Osmanoğlu, Batuhan, et al., 2016)]

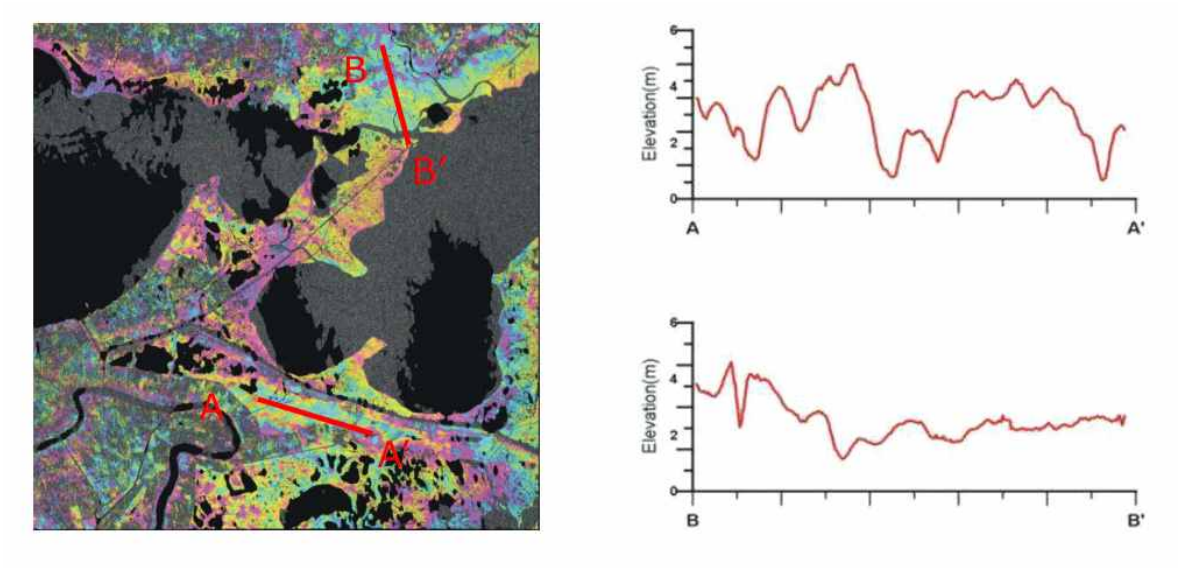


[위상 언래핑 결과. (a)는 branch-cut 알고리즘을, (b)는 MCF 알고리즘을 적용하여 위상 언래핑을 수행하였으며, 영상의 우측에서 낮은 상관관계를 보여 언래핑 결과에서 차이가 있음을 볼 수 있음]

○ 이종 영상레이더 자료 간 위상간섭기법 적용

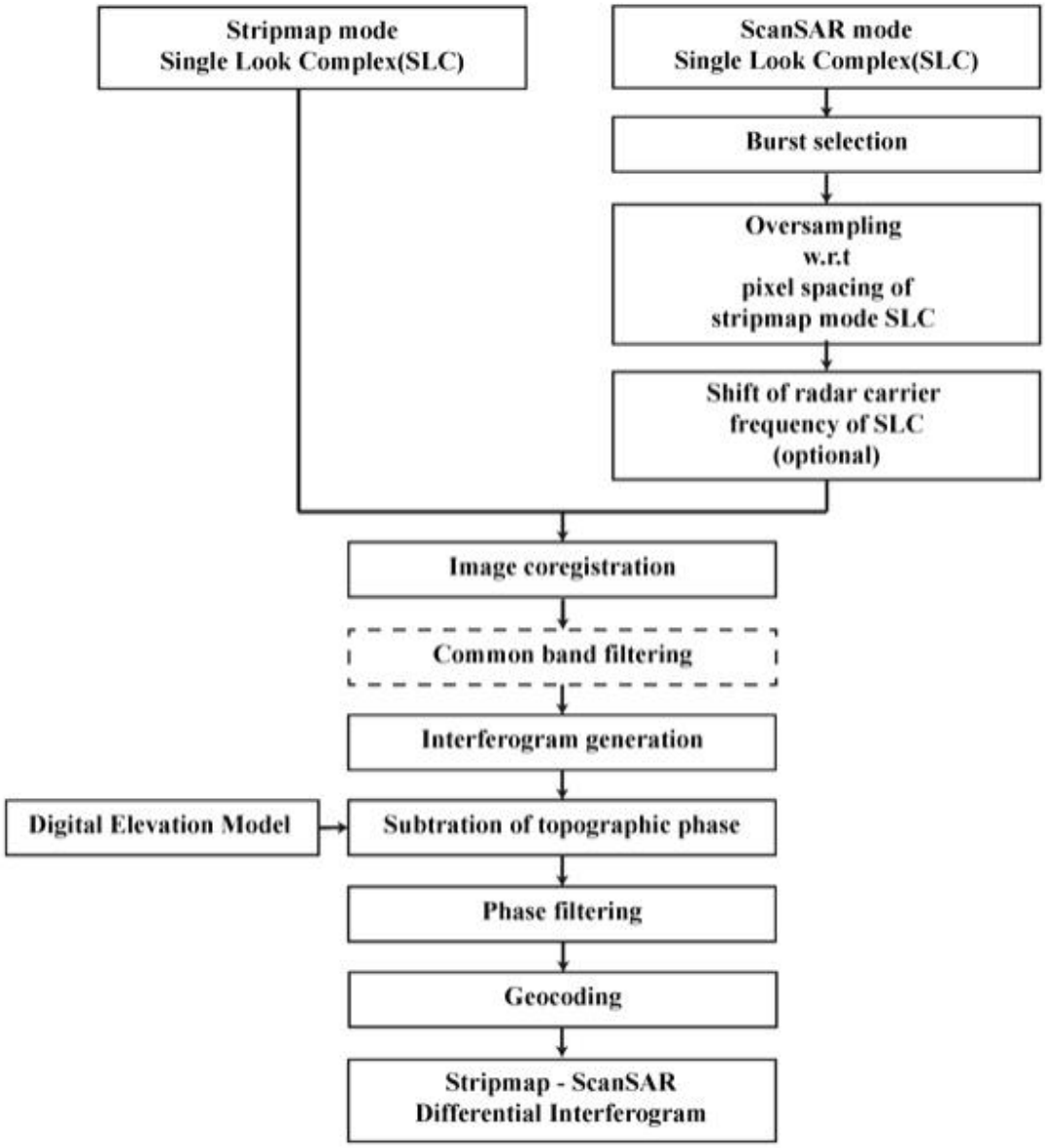
- 영상레이더를 이용한 원격탐사 시 파장 길이에 비례하여 투과성이 높아지므로 산림이 70 %이상 차지하는 대한민국에서 단파장 영상레이더에 비해 장파장 영상레이더는 높은 긴밀도를 유지할 수 있어 활용 가능성이 높게 평가됨 (Hong & Wdowinski, 2013)

- 장파장 영상레이더 인공위성은 Advanced Land Observing Satellite-2 Phased Array type L-band Synthetic Aperture Radar-2 (ALOS-2 PALSAR-2)가 현재 운용중이며 14일의 재방문 주기를 가지나 운영 계획에 따라 영상 획득 모드가 일정하지 않아 단일 모드를 이용한 위상간섭기법 적용 및 시계열 분석시 시간 기선거리 증가에 따른 긴밀도 저하가 야기될 수 있음
- 영상레이더 위상간섭기법 적용 시 시간 기선거리 저하로 인한 긴밀도 감소를 극복하고자 이종 영상레이더로부터 획득된 자료 간 위상간섭기법 적용 연구가 선행됨
- Hong & Won (2006)은 서로 다른 도플러중심 주파수를 가지는 이종 플랫폼인 ERS와 Envisat에서 획득된 영상을 비행 방향 (azimuth)과 관측 방향 (range)에서 common band filtering을 수행하여 스펙트럼의 잡음을 제거한 후 수치표고모형을 제작함
- 다음은 ERS-Envisat 영상레이더 위상간섭기법 적용 결과로, 30분의 시간 기선거리를 가지는 수치표고모형으로 연안지역에서 높은 긴밀도를 보여줌



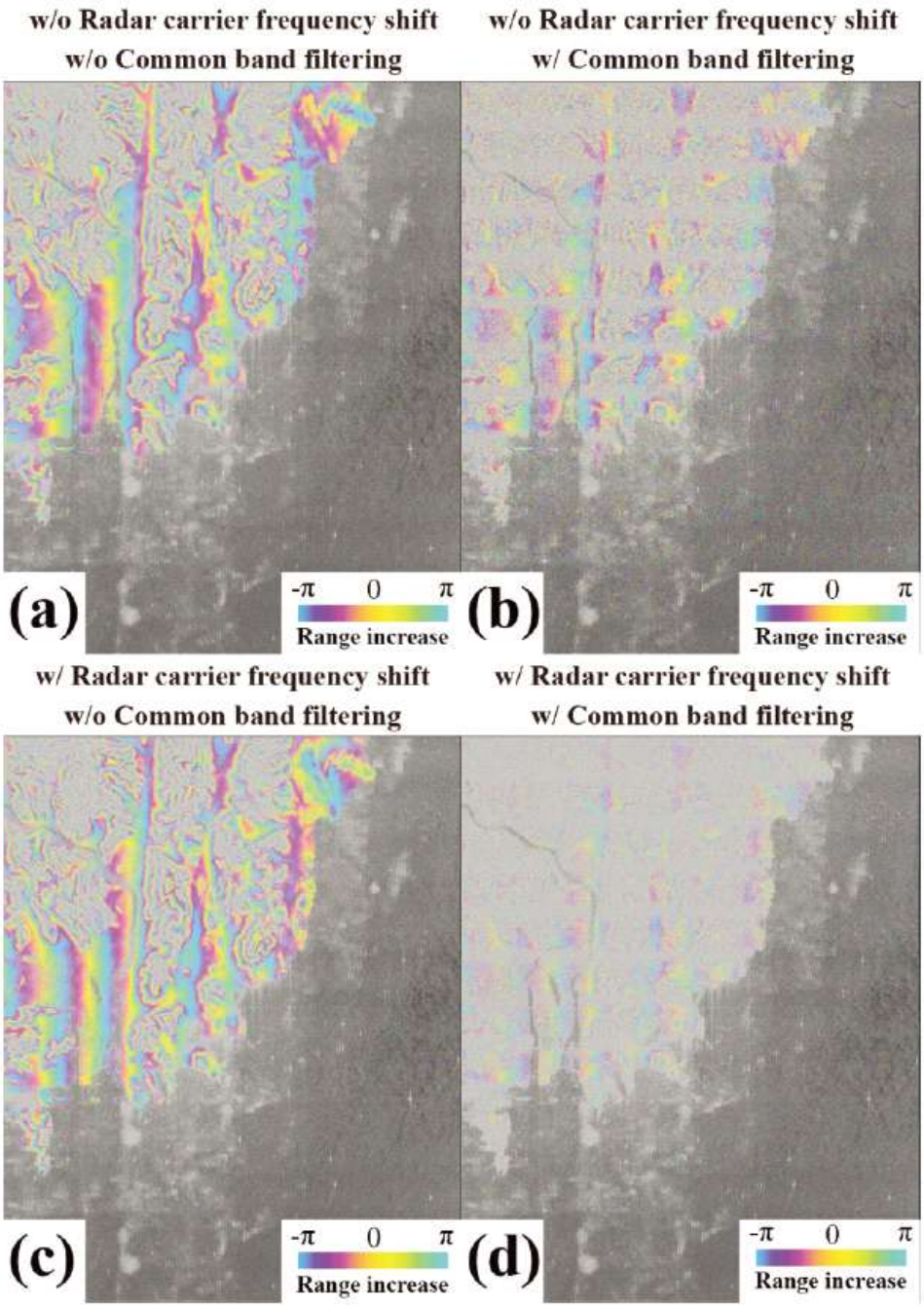
[ERS-Envisat 영상레이더 위상간섭기법 적용 결과와 수치표고모형 분석 결과]

- Park et al. (2020)은 일본 항공우주국에서 제공하는 ALOS-2 PALSAR-2 영상레이더로부터 서로 다른 영상 획득 모드로 얻어진 자료 간 위상간섭기법 적용 연구를 수행하였으며 비행 방향 및 거리 방향에서 common band filtering을 통해 ALOS-2 PALSAR-2 영상레이더 Stripmap - ScanSAR자료 기반의 위상간섭기법 적용 평가 및 각 방향의 긴밀도 분석을 통해 최적의 위상간섭기법 적용 연구를 수행함
- Stripmap과 ScanSAR영상 간 위상간섭기법 적용은 모드에 따른 신호 획득 방식이 달라 chirp 대역폭, 중심주파수와 PRF의 차이를 고려한 자료 처리가 요구됨
- 다음 순서도는 Stripmap-ScanSAR 영상레이더 위상간섭기법 절차의 모식도를 나타내며, 일반적인 위상간섭기법과 다르게 중심주파수 보정을 고려하여 자료 처리를 수행함
- Stripmap-ScanSAR 영상레이더 위상간섭기법 적용에서 중심주파수 보정 및 common band filtering의 적용 유무에 따른 결과가 도시되었음
- 이종 영상레이더 자료 간 위상간섭기법 적용은 시간 기선거리 향상을 통해 긴밀도 저하를 예방하여 영상레이더 원격탐사 기반의 산사태 감시에 효과적으로 사용될 수 있을 것으로 생각됨



[Stripmap-ScanSAR 위상간섭기법 처리 과정의 모식도. shift of radar carrier frequency 과정과 common band filtering 과정은 선택적으로 적용 가능함]

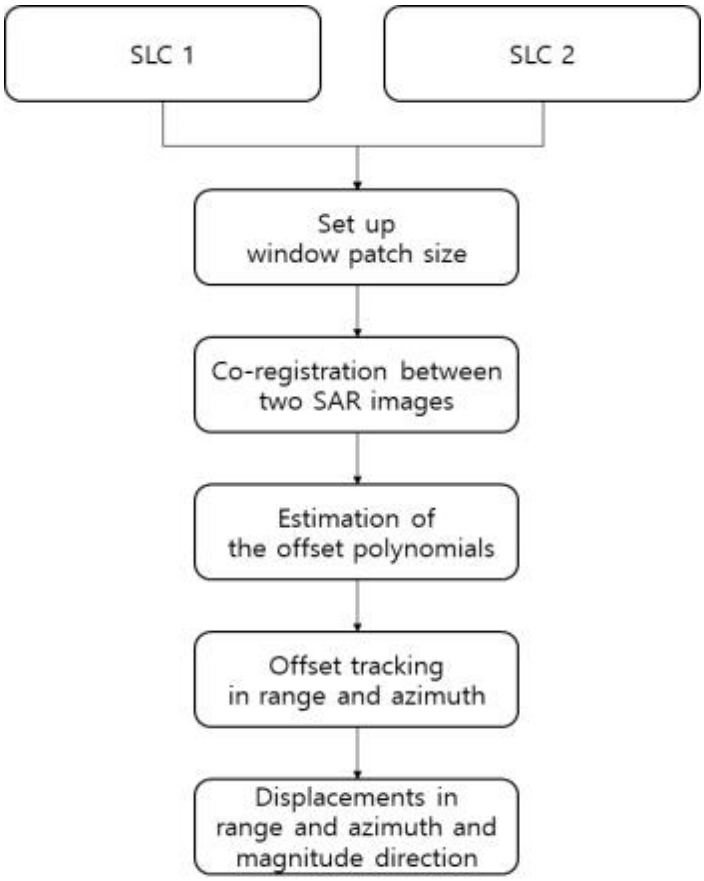
(Park et al., 2020)



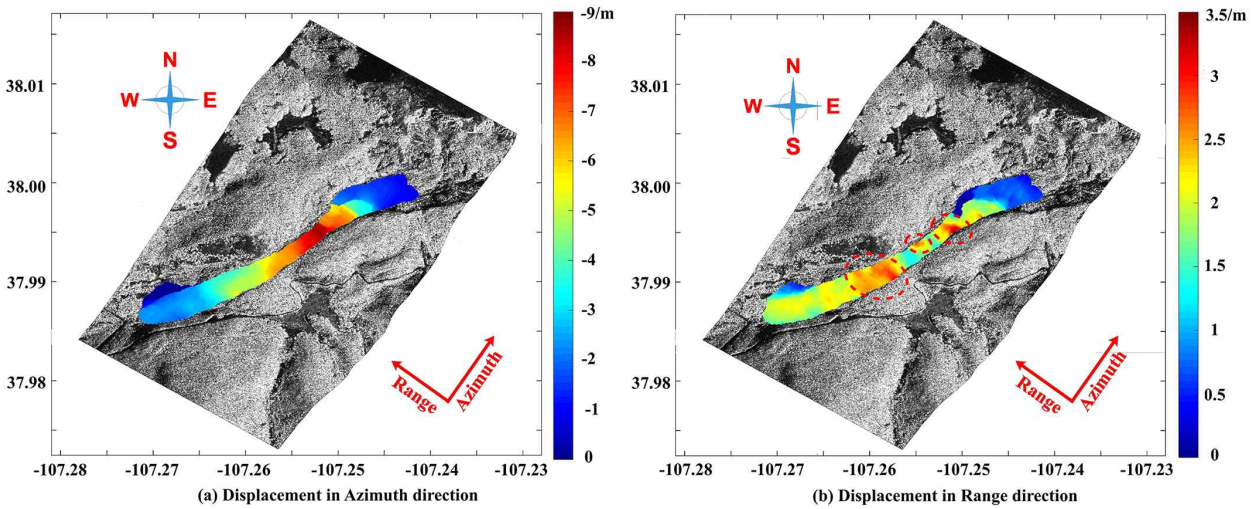
[ALOS-2 PALSAR-2 Stripmap-ScanSAR 위상간섭기법 적용 결과. 레이더 중심 주파수 보정 및 common band filtering 적용 유무에 따라 다양한 결과를 보여줌]

○ 오프셋 트래킹

- 오프셋 트래킹 기법은 강도교차상관기법(amplitude cross-correlation)에 의해 두 이미지 사이의 부합하는 패턴과 특징을 찾아 변위를 측정할 수 있음
- 지표변위가 클 경우 위상간섭기법의 적용이 어려우므로 변위가 큰 경우 빠른 속도 측정에 용이한 오프셋 트래킹 기법으로 처리하여 변위 지도를 얻을 수 있음
- 오프셋 트래킹 결과는 영상 패치 크기에 민감하게 반응하고 계산 속도를 결정짓기 때문에 적절한 패치 크기를 찾는 것이 중요함
- 작은 영상 패치를 적용하면 공간 해상도가 높은 변위 지도를 얻을 수 있지만, 빠르게 움직이는 곳에서는 적용할 수 없는 한계가 있음
- 경사거리 및 방위방향의 오프셋은 강도 교차상관관계를 이용하여 측정되며 측정된 오프셋은 변위로 변환됨



[오프셋 트래킹 순서도]



[2011년 8월 12일부터 2013 년 11월 1일까지의 슬럼 굴 리온 산사태 변위. (a) 방위각 방향 변위 및 (b) 범위 방향 변위 (Wang et al., 2018)]

- 콜로라도의 산후안 산맥에 있는 슬럼 굴 리온 산사태를 2011년 8월부터 2013년 11월까지 획득한 무인 항공기 영상레이더(UAVSAR) 자료를 사용하여 시계열 오프셋 트래킹 방법으로 측정하였음
- 일일 최대 1.2 ㎝의 속도로 이동하며 계절에 따른 속도 변화가 뚜렷함
- 산사태의 머리와 꼬리는 안정적인 경향이 있는 반면 산사태의 목은 가속되는 것을 확인할 수 있음

○ 드론 촬영 비탈면 SAR 영상 기반 비탈면 변위 측정 기술 개발

- 고정형 지상레이더 기반 비탈면/사면 변위 측정 가능성 평가
  · GAMMA 사는 네바다 주에 위치한 금 광산의 사면에서 발생한 붕괴를 관측하기 위해 GPRI를

이용하여 사면 안정성 연구를 수행함

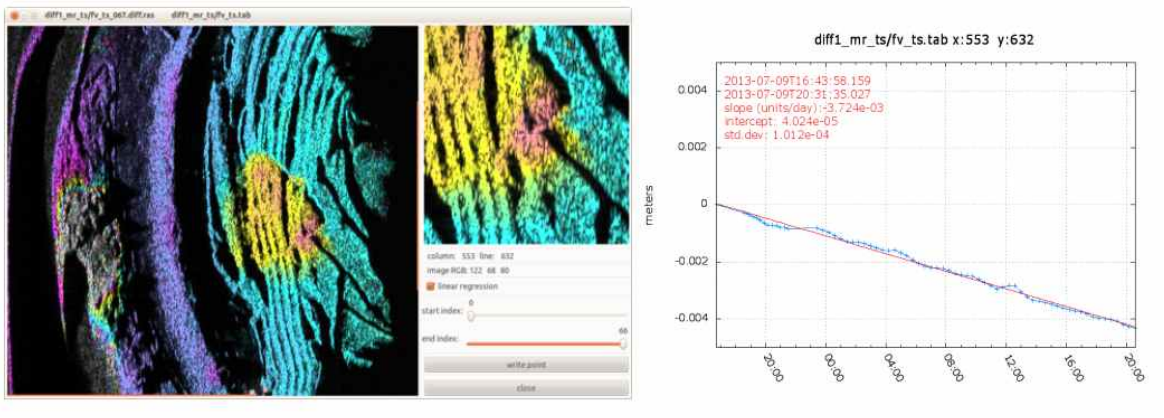
· 약 500 m 거리에서 자료를 획득하였으며, 2분 간격으로 총 450번의 관측을 통해 시계열 영상레이더 자료를 획득함

· 16분의 시간 간격을 가지는 차분위상간섭도를 쌓아 시계열 변위 분석을 수행하였으며, 대기 지연 위상에 대한 효과를 저감하기 위해 시간 필터링을 수행함

· 금광산에서 지상 레이더 관측 결과를 통해 수직 방향의 광산 붕괴를 매우 성공적으로 탐지함으로써 비탈면 지표 변위 관측에 있어 영상레이더 자료 기반의 적합함을 확인할 수 있음



[고정형 지상레이더 관측장비 및 금 광산]



[광산지역 지반수직변위 관측 결과]

○ 드론 SAR 관측자료의 비탈면 변위 측정 자료처리 기술 개발

- 정밀 비탈면 변위 측정을 위한 자료처리 기술 사전 개발 및 확보

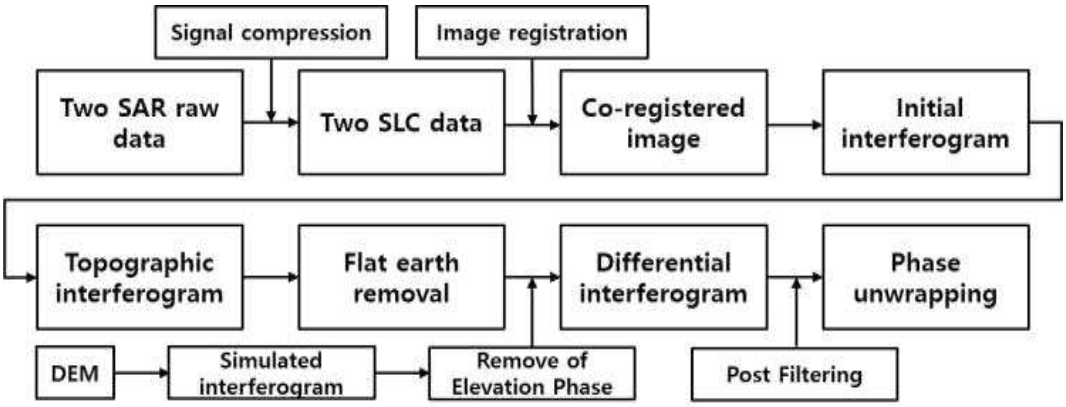
· 비탈면의 정밀 변위 측정을 위해 영상레이더 위상간섭기법 적용을 통해 변위 변화를 관측할 수 있으며, 변위 측정 기술 흐름도는 아래와 같음

· 송수신된 원시 자료는 관측 방향 및 비행 방향(또는 레일 이동 방향)으로 신호 압축을 거쳐 단일 복소 영상으로 변환됨

· 서로 다른 두 영상레이더 자료의 정합 과정과 재배열 과정을 거쳐 위상 차이를 계산한 후, 초기 위상간섭도를 제작함

· 제작된 초기 위상간섭도에서 지구곡률위상 및 지형 정보 위상을 제거한 후 지표 변위를 획득 할 수 있음

· 잡음 제거와 구속화(wrapped)된 위상의 불구속화(unwrapped)를 위해 필터링 및 위상 언래핑 과정을 통해 영상 획득기간동안 발생한 전체적인 변위를 관측 할 수 있음



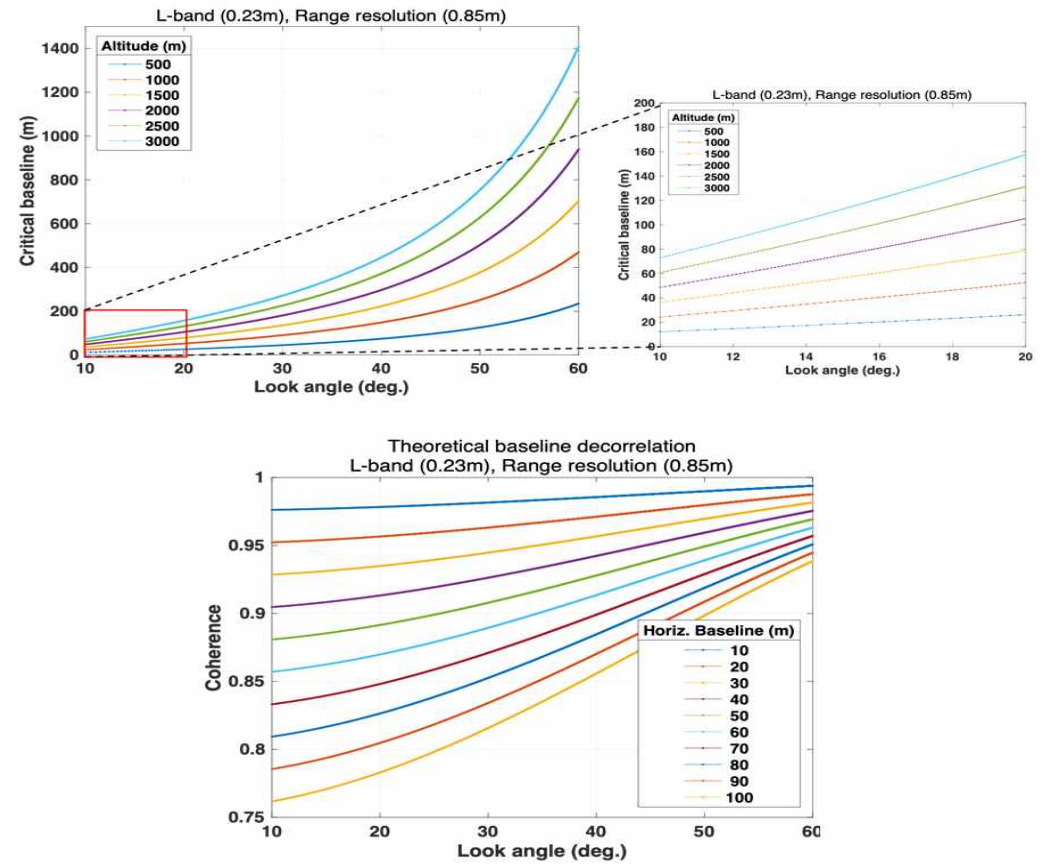
[비탈면 변위 측정 영상레이더 자료처리 기술 흐름도]

○ 드론 고도 및 궤도 간격 (기선거리)에 따른 최대 기선거리 및 긴밀도 모델링 수행

· 드론 기반의 영상레이더 위상간섭기법 적용 시 기선거리 및 관측각에 따른 긴밀도 변화 연구를 수행함

· 이론적으로 영상레이더 자료 획득 당시 입사각이 클수록 산란체의 투과성이 낮으나, 식생의 여부에 따라 결정됨

· 이론적으로 기선거리가 가까울수록 높은 긴밀도를 유지하며, 입사각이 증가하고 영상 획득 고도가 낮을수록 위상간섭기법 적용이 가능한 임계 기선거리가 증가함



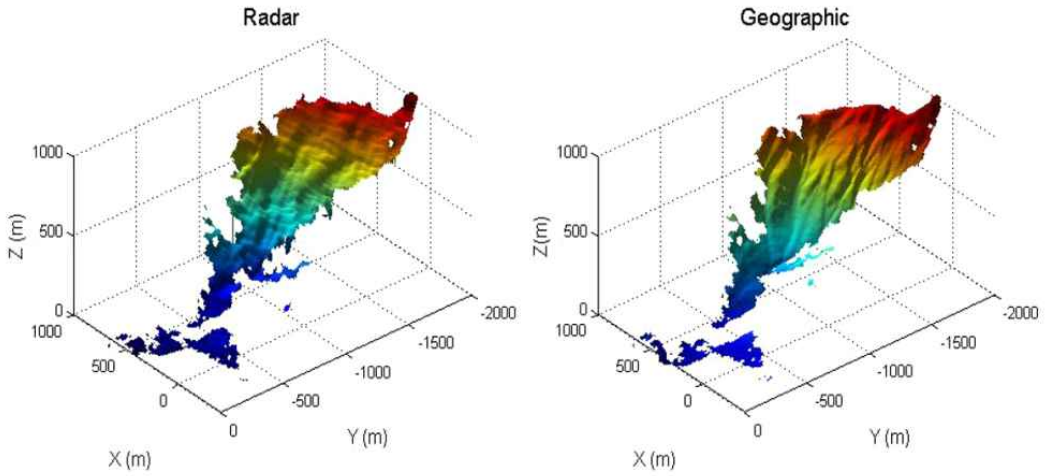
[드론 고도별 최대기선거리(위), 수평기선거리에 따른 이론적 긴밀도(아래)]

○ 드론 SAR 지형보정을 위한 모사된 간섭영상 생성

- 드론 SAR 시스템을 위한 모사된 간섭영상 생성 기술 개발

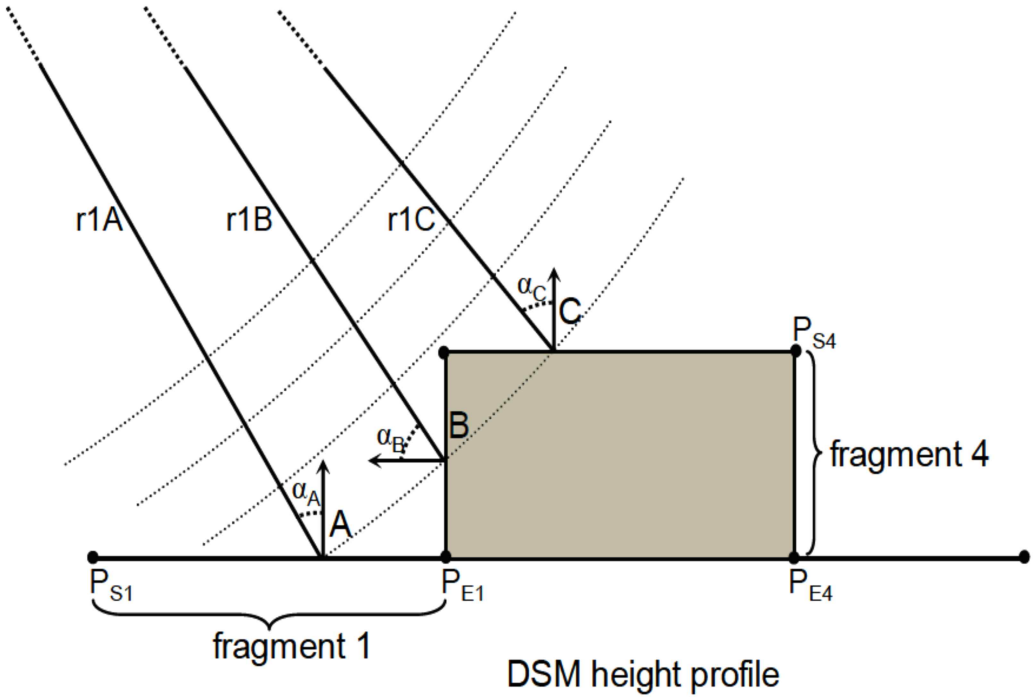
· 비탈면의 불연속적인 지형 특성을 반영하는 고정밀 시뮬레이터 개발이 요구됨

· Noferini, Linhsia, et al. (2006)은 지상기반 영상레이더를 활용하여 오스트리아 Lizumer Hutte 지역에서 지도와 비교하여 수치표고모형을 제작함



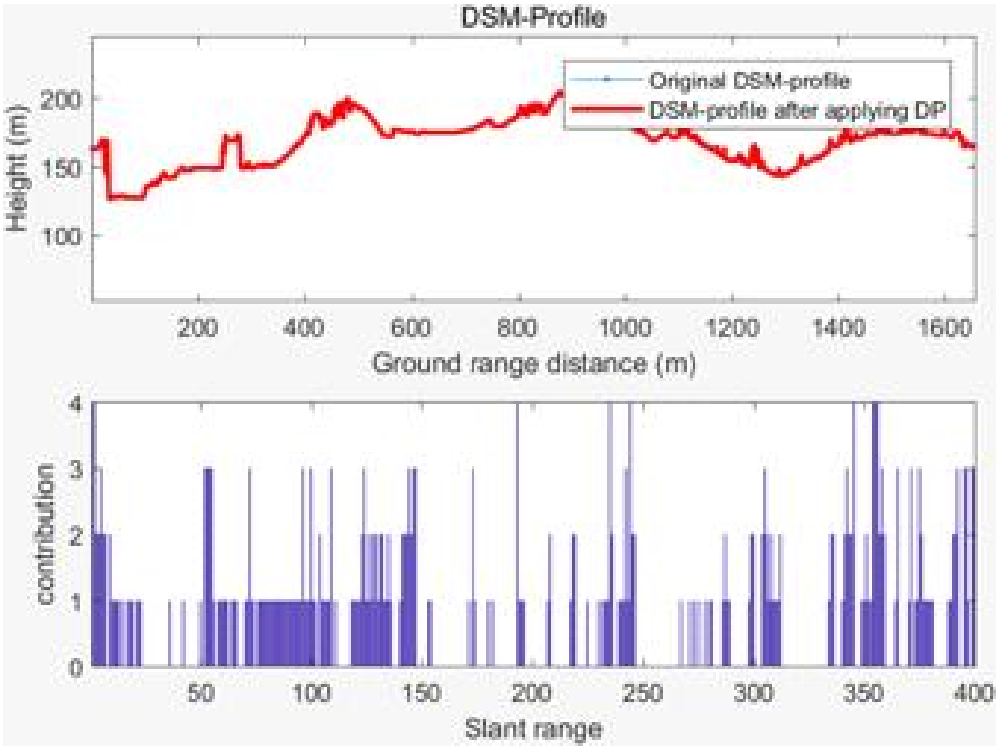
[지상기반 영상레이더를 통해 제작된 수치표고모형(좌) 및 지형도(우)]

· Thiele et al. (2007)연구를 기반으로 3D LIDAR DSM 자료를 이용하여 벡터분석 기반 확장 알고리즘 개발



[불연속 사면에서의 DSM 프로파일]

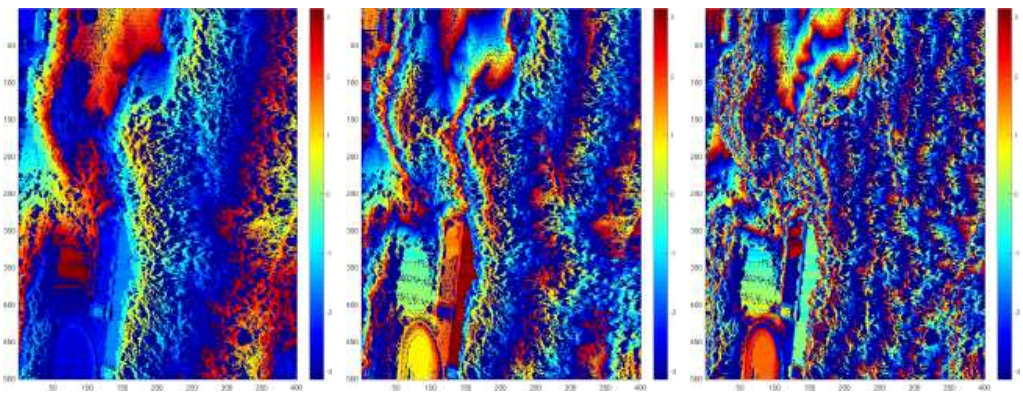
· LIDAR DSM에서 드론 궤도에 따라 각 라인별 고도 프로파일을 추출하여, 자료처리 효율성을 높이기 위해 Douglas-Peucker 알고리즘을 적용 후 각 선분별 반사값 및 간섭위상 계산
· 불연속 지형에 따른 다중산란을 고려하도록 각 resolution cell 안에 포함되는 복합 신호 분석 수행



[DSM 프로파일 및 각 resolution cell 안에 포함되는 다중반사 신호 수]

○ 드론 궤도간격 (기선거리)에 따른 모사된 간섭영상 생성

- 비탈면 변위는 반복 촬영 시 발생하는 드론의 궤도 차이에 의한 지형 위상 및 지구 곡률위상을 제거한 후 측정 가능함
- 위상간섭도에서 지형 위상 제거를 위해 DSM을 이용하여 지형 정보를 정밀 모사하는 알고리즘 개발
- 아래 그림은 각각 다른 기선거리 (100/200/400 m)에 따라 생성된 지형 정보 위상을 나타냄
- 위상간섭기법에서 고도의 모호성 (height ambiguity)으로 인해 기선거리가 길수록 보다 정밀한 지형 정보 표현이 가능하며, 400 m 수치표고모형에서 정밀한 프린지를 볼 수 있음
- 제작된 수치표고모형을 이용하여 위상간섭도로부터 지형 위상 제거 후 비탈면 변위에 의해 발생하는 위상은 한 프린지 주기에서 L-밴드의 경우 파장 길이의 반인 약 12 cm임
- 변위측정 정확도는 비탈면을 피복한 식생의 특성, 드론 궤도간격과 지형위상 제거에 사용된 DSM의 정확도에 영향을 받음
- 개발된 시뮬레이터를 이용한 지형 정보 위상 제거를 통해 잔여 위상 (변위측정 오차)은 최대 1 fringe (~12cm) 이내로 발생



[기선거리별 모사된 간섭영상 (좌) 100m, (중) 200m, (우) 400m]

(3) 비탈면 변위 측정을 위한 드론 제어기술 개발

○ Command 드론별 분리

- TAKEOFF

TAKEOFF				
PX4		Guided 모드로 변경	Arming	Takeoff
	Quad	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0xD9, 0x00000004)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 400, 0, 0x3F800000, 0, 0, 0, 0, 0, 0)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 22, 0, 0xBF800000, 0, 0, 0xFFC00000, 0xFFC00000, 0xFFC00000, 0x43F586C9)
		Vtol 모드로 변경	Arming	Takeoff
	VTOL	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x1d, 0x03040000)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 400, 0, 0x3F800000, 0, 0, 0, 0, 0, 0)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 22, 0, 0xBF800000, 0, 0, 0xFFC00000, 0xFFC00000, 0xFFC00000, 0x43f9472b)
APM		Arming	Guided 모드로 변경	Takeoff
	Quad	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 400, 0, 0x3F800000, 0, 0, 0, 0, 0, 0)	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0xD9, 0x00000004)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 22, 0, 0, 0, 0, 0, 0, 0, 0x40400000)
		Vtol 모드로 변경	Arming	Takeoff
	VTOL	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x1d, 0x0000000F)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 400, 0, 0x3F800000, 0, 0, 0, 0, 0, 0)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 22, 0, 0, 0, 0, 0, 0, 0, 0x41200000)

[드론 제어명령-TAKEOFF]

- LAND

LAND		
PX4		모드 변경
	Quad	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x9D, 0x06040000)
		모드 변경
	VTOL	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x9d, 0x06040000)
APM		모드 변경
	Quad	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0xD9, 0x00000009)
		모드 변경
	VTOL	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x01, 0x00000014)

[드론 제어명령-LAND]

- RTH

RTH		
PX4		모드변경
	Quad	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0xD9, 0x05040000)
		모드변경
	VTOL	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x9d, 0x05040000)
APM		모드변경
	Quad	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0xD9, 0x00000006)
		모드변경
	VTOL	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x01, 0x00000015)

[드론 제어명령-RTH]

- GOTO

GOTO		
PX4		Goto Command Long 명령어
	Quad	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 192, 0, 0xbf800000, 0x3f800000, 0, 0xffc00000, Latitude, Longitude, Altitude)
		Goto Command Long 명령어
	VTOL	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 192, 0, 0xbf800000, 0x3f800000, 0, 0xffc00000, Latitude, Longitude, Altitude)
APM		Goto Command Long 명령어
	Quad	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 192, 0, 0xbf800000, 0x3f800000, 0, 0xffc00000, Latitude, Longitude, Altitude)
		Goto Command Long 명령어
	VTOL	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 192, 0, 0xbf800000, 0x3f800000, 0, 0xffc00000, Latitude, Longitude, Altitude)

[드론 제어명령-GOTO]

- Mission UPLOAD

Mission UPLOAD						
PX4		MissionCountUpload	TakeoffMission Upload	ionUpload(모든미션아이템을 전송)		
	Quad	MISSION_COUNT(target_system, target_component, count) MISSION_COUNT(드론ID, 190, 미션갯수)	MISSION_ITEM_INT(target_system, target_component, seq, frame, command, current, autocontinue, param1, param2, param3, param4, x, y, z) MISSION_ITEM_INT(드론ID, 0, 0, 3, 22, 0, 1, 0, 0, 0, 0x7FC00000, Latitude, Longitude, Altitude)"	MISSION_ITEM_INT(target_system, target_component, seq, frame, command, current, autocontinue, param1, param2, param3, param4, x, y, z) MISSION_ITEM_INT(드론ID, 0, 몇 번째 미션, 3, 16, 0, 1, 0, 0, 0, Latitude, Longitude, Altitude)"		
		MissionCountUpload	TakeoffMission Upload	TransitionMission Upload	ad(마지막을 제외한 모든미션아	TransitionMission Upload
	VTOL	MISSION_COUNT(target_system, target_component, count) MISSION_COUNT(드론ID, 190, 미션갯수)	MISSION_ITEM_INT(target_system, target_component, seq, frame, command, current, autocontinue, param1, param2, param3, param4, x, y, z) MISSION_ITEM_INT(드론ID, 190, 0, 3, 84, 1, 1, 0, 0, 0, 0x7FC00000, Latitude, Longitude, Altitude)"	MISSION_ITEM_INT(target_system, target_component, seq, frame, command, current, autocontinue, param1, param2, param3, param4, x, y, z) MISSION_ITEM_INT(드론ID, 190, 1, 2, 3000, 0, 1, 0x40800000, 0, 0, 0x7FC00000, Latitude, Longitude, Altitude)"	MISSION_ITEM_INT(target_system, target_component, seq, frame, command, current, autocontinue, param1, param2, param3, param4, x, y, z) MISSION_ITEM_INT(드론ID, 190, 몇 번째 미션, 3, 16, 0, 1, 0, 0, 0, 0x7FC00000, Latitude, Longitude, Altitude)"	MISSION_ITEM_INT(target_system, target_component, seq, frame, command, current, autocontinue, param1, param2, param3, param4, x, y, z) MISSION_ITEM_INT(드론ID, 190, 미션의 마지막 count, 2, 3000, 0, 1, 0x40400000, 0, 0, 0x7FC00000, Latitude, Longitude, Altitude)"
APM		MissionCountUpload	TakeoffMission Upload	ionUpload(모든미션아이템을 전송)		
	Quad	MISSION_COUNT(target_system, target_component, count) MISSION_COUNT(드론ID, 0, 미션갯수)	MISSION_ITEM_INT(target_system, target_component, seq, frame, command, current, autocontinue, param1, param2, param3, param4, x, y, z) MISSION_ITEM_INT(드론ID, 0, 0, 3, 22, 0, 1, 0x41700000, 0, 0, 0, Latitude, Longitude, Altitude)"	MISSION_ITEM_INT(target_system, target_component, seq, frame, command, current, autocontinue, param1, param2, param3, param4, x, y, z) MISSION_ITEM_INT(드론ID, 0, 몇 번째 미션, 2, 16, 0, 1, 0, 0, 0, 0, Latitude, Longitude, Altitude)"		
	VTOL	확인작업 필요				

[드론 제어명령-Mission UPLOAD]

- Mission Start

Mission Start					
PX4		Set GuideMode Disarm	Arming		
	Quad	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x51, 0x04040000)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 400, 0, 0x3F800000, 0, 0, 0, 0, 0, 0)		
		Set GuideMode Disarm	Arming		
	VTOL	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x1d, 0x04040000)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 400, 0, 0x3F800000, 0, 0, 0, 0, 0, 0)		
APM		Set GuideMode Disarm	Arming	Takeoff	Mission Start
	Quad	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x59, 0x00000004)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 400, 0, 0x3F800000, 0, 0, 0, 0, 0, 0)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 22, 0, 0, 0, 0, 0, 0, 0, 0x40200000)	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0xD9, 0x00000003)
		Set GuideMode Disarm	Arming	Mission Start	
	VTOL	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0x59, 0x00000011)	COMMAND_LONG(target_system, target_component, command, confirmation, param1, param2, param3, param4, param5, param6, param7) COMMAND_LONG(드론ID, 0, 400, 0, 0x3F800000, 0, 0, 0, 0, 0, 0)	SET_MODE(target_system, base_mode, custom_mode) SET_MODE(드론ID, 0xD1, 0x0000000A)	

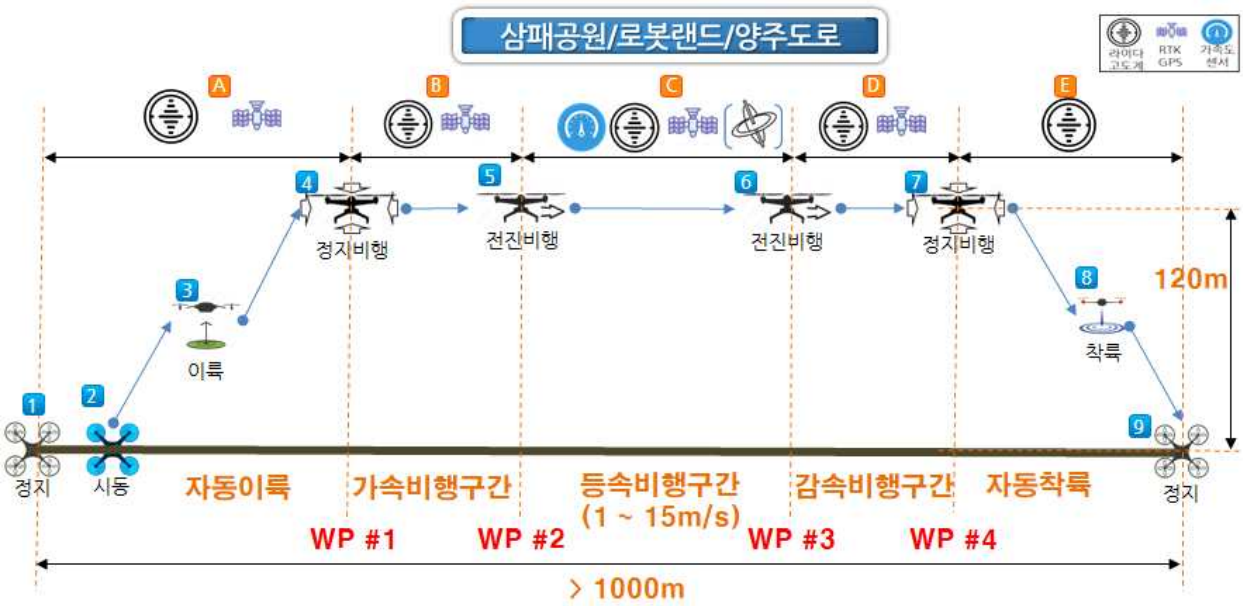
[드론 제어명령-Mission Start]

○ 연구수행 임무비행 시험장소의 특징 및 영향 분석

시험 장소	지리적 특징	환경적 특징	드론 영향
남양주 삼패공원		- 주변 고층건물 없음 - 강바람이 심함 - 주변 보호구역 있음 - 오전-> 오후 풍속증가	- GPS 위성 수신상태 불량(장시간) - RTK-BASE 0.5m급 GCP 구축 2~6시간
인천 로봇랜드		- 비행경로 오른쪽 빌딩 - 주변 장애물 없는 평지 - 해풍 3~5m/s - 오전-> 오후 풍속 증가 ※ 비행경로 중간위치에 (하수) 관로 매설 추정	- GPS 위성 수신상태 양호 - RTK-BASE 0.5m급 GCP 구축 2시간 - 비행방향이 건물방향으로 치우침 - 지자기 센서 에러로 비행 어려움 (자동비행 -> 수동비행 전환) - 드론의 비행자세, 고도, 위치 등이 비행계획 대비 현저히 변경
인천 로봇랜드		- 비행경로 왼쪽 고층빌딩 - 비행경로 오른쪽 레이싱 드론 연습비행 - 주변 장애물 없는 평지 - 해풍 3~5m/s - 오전-> 오후 풍속 증가	- GPS 위성 수신상태 양호 - RTK-BASE 0.5m급 GCP 구축 2시간 - 비행방향이 건물방향으로 치우침 - GPS 오차 증가로 비행궤도 벗어남 (건물구간 진입할 때 1m 수준) - 드론의 비행자세 : 매우 안정적 비행고도 : 하향 최대 1m 비행위치 : 물리적 위치 양호
양주시 도로변		- 주변 고층빌딩 없음 - 주변 장애물 없는 평지 - 평균 1~3 m/s - 오전-> 오후 풍속 증가	- GPS 위성 수신상태 양호 (최상) - RTK-BASE 0.5m급 GCP 구축 1시간 내외, HDOP 좋음 - (육안식별) 물리적 수평수직오차 0.5m 이내 - 드론 비행자세 : 매우 안정적 비행고도 : 하향 최대 0.5m 비행위치 : 물리적 위치양호 비행궤도 : 육안식별 안정적

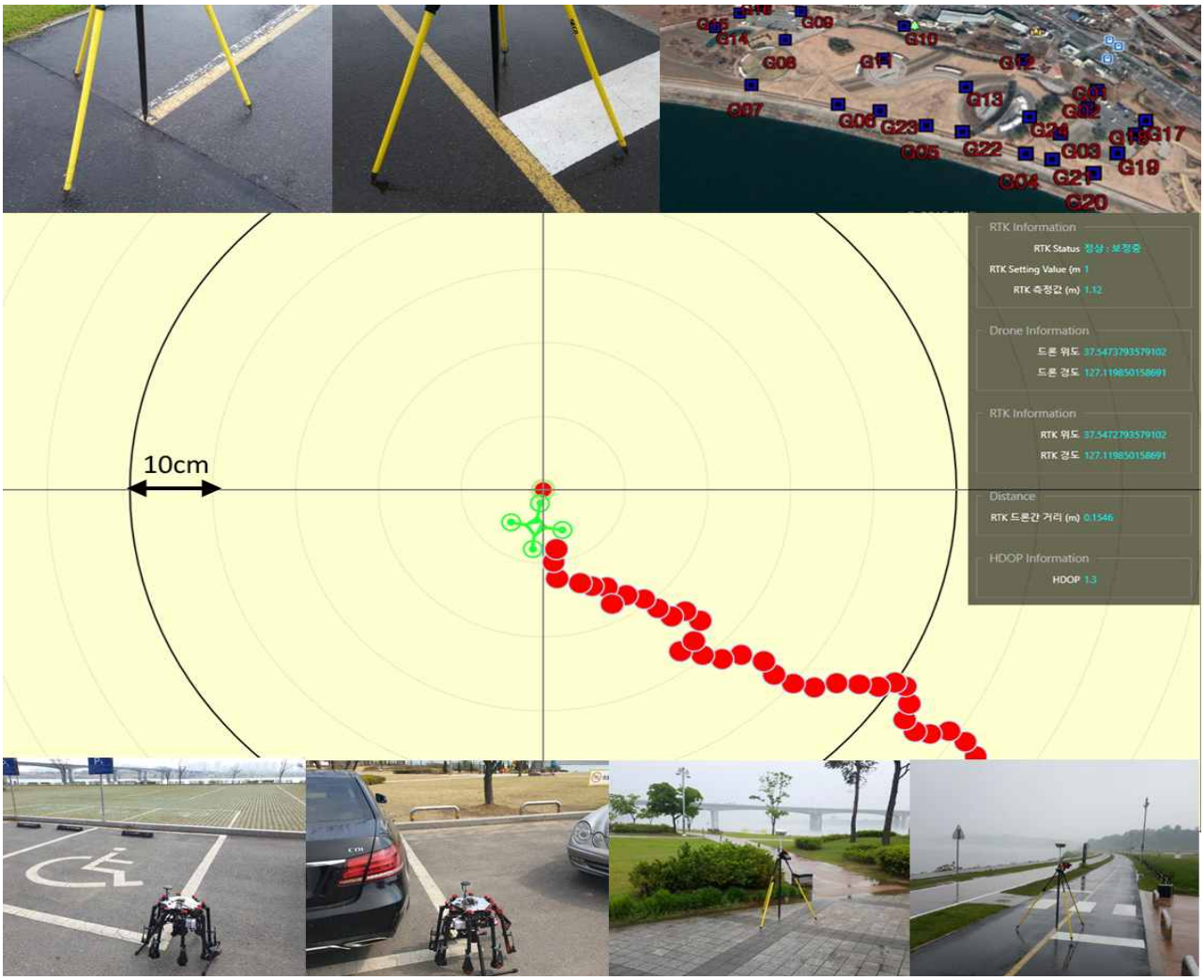
[드론 비행 공간적·환경적 특성]

○ 연구수행 임무비행 비행 패턴 및 상세 내역



[시험 임무비행 비행 패턴]

○ 임무비행이 가능한 RTK-GPS 정확도 체크



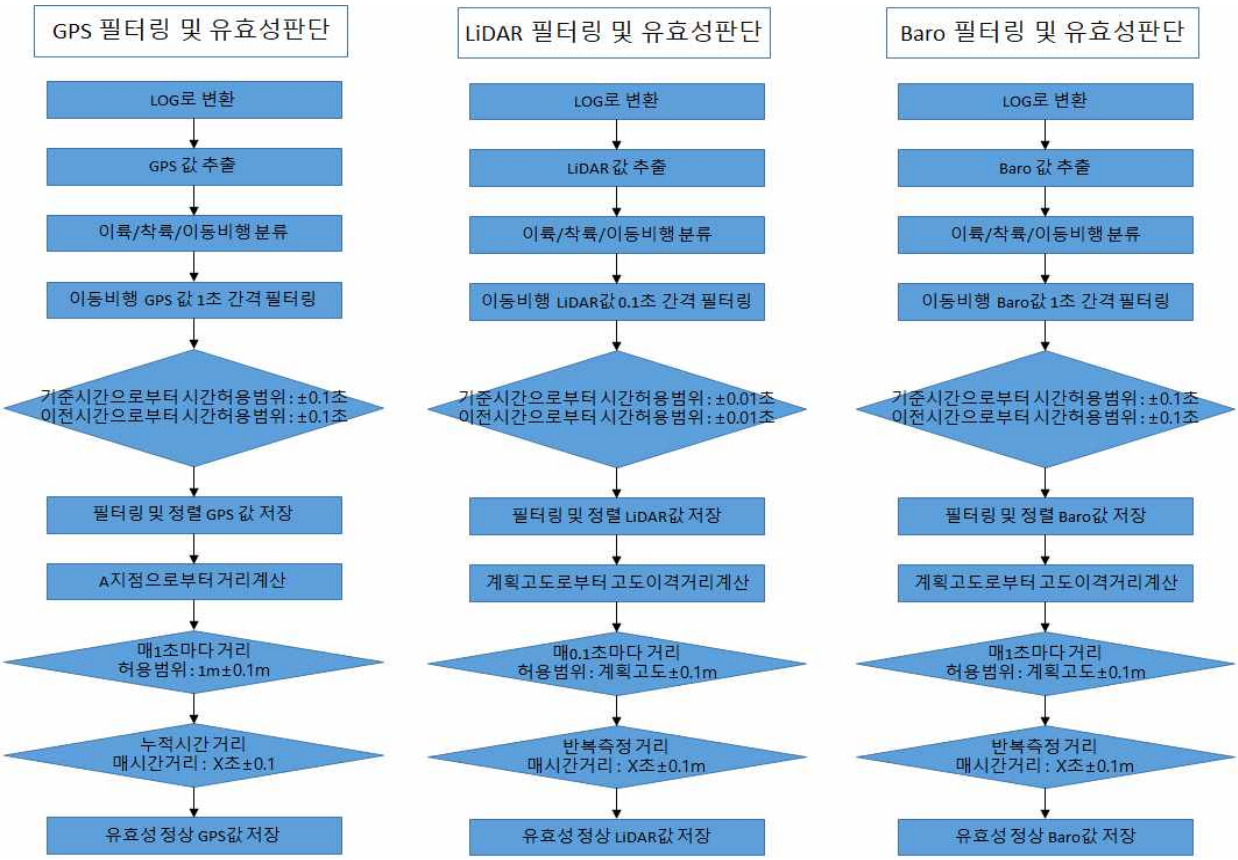
[RTK-GPS 정확도 측정]

○ 드론 유지관리정보를 활용한 안전진단 및 물리적 성능정보 취득 방안



[단계별 안전진단 및 물리적 성능정보 취득]

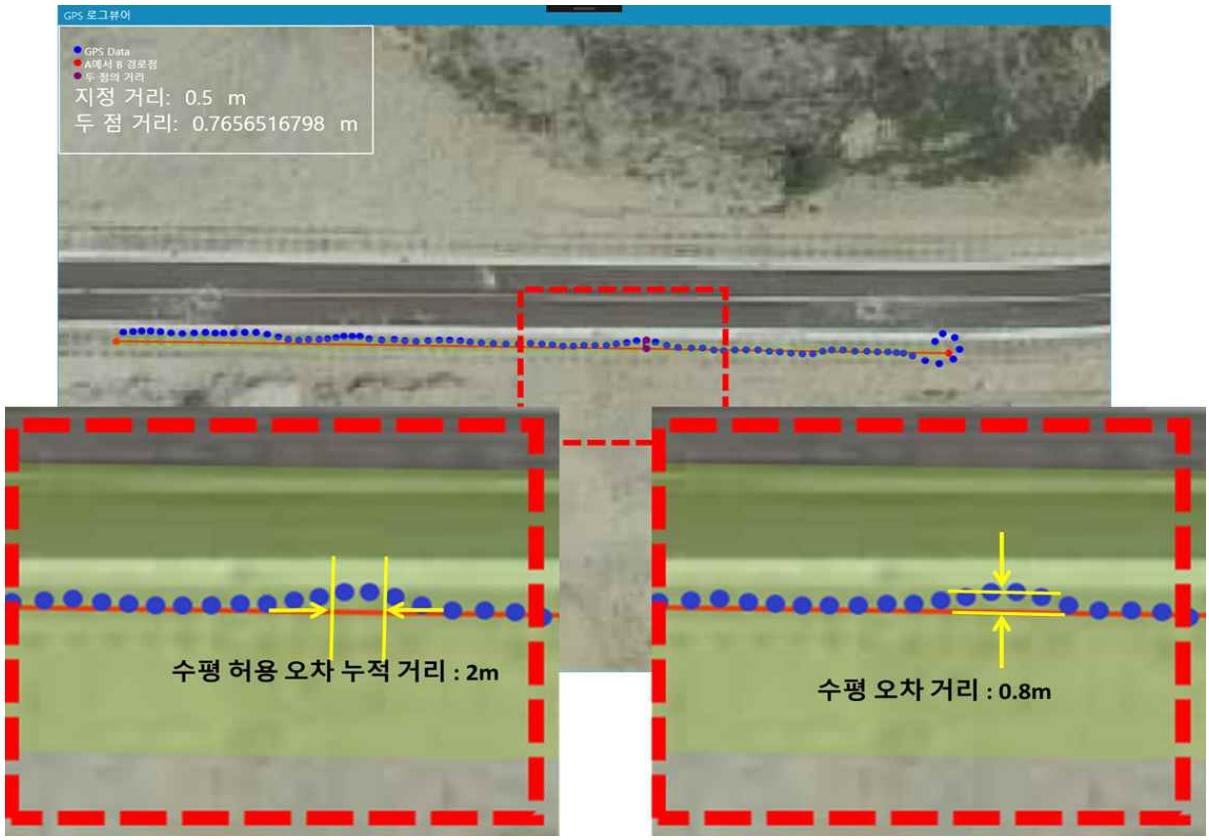
○ 비행로그 필터링 및 유효성판단



[필터링 및 유효성판단 순서도]

○ 비탈면 비행포인트 기초정보 분석·추출 알고리즘(비탈면·임무센서·비행정보)

- GPS 필터링 정보 가시화 및 수평오차 분석 화면



[GPS 필터링 정보 가시화 및 수평오차 분석 화면]

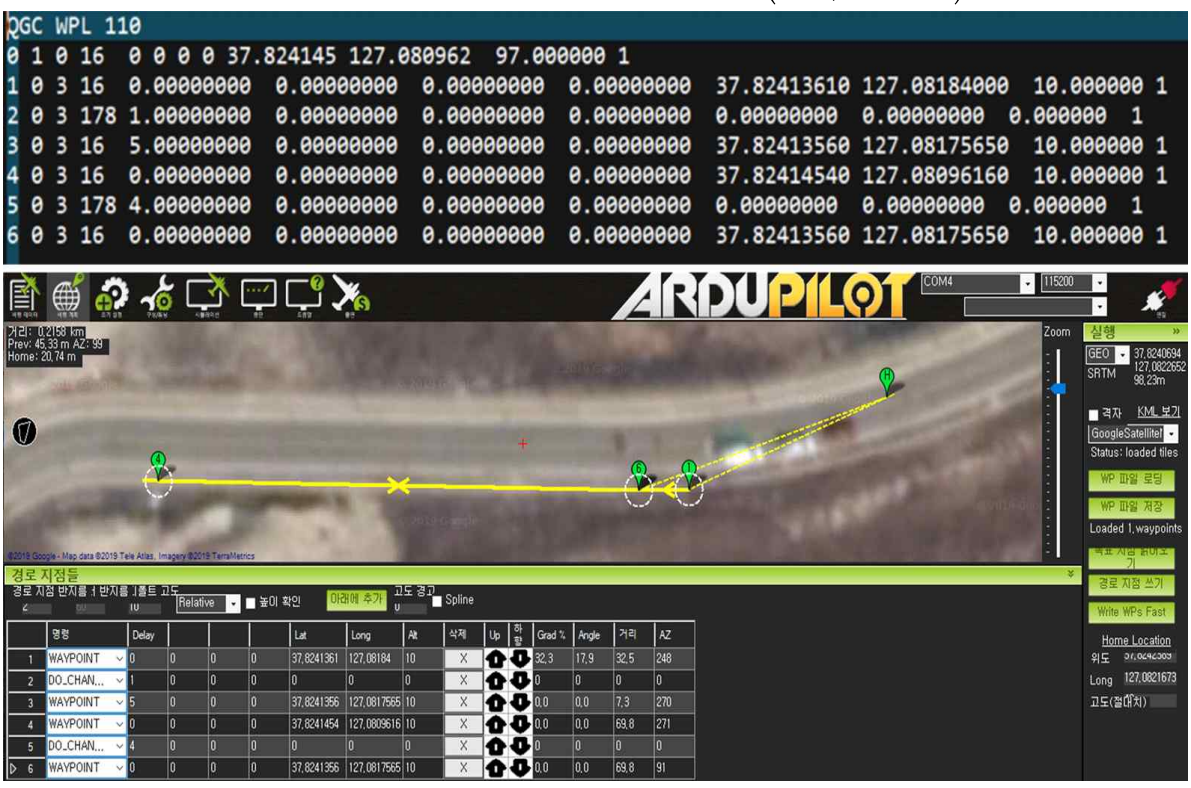
- 시험용 드론 기반 3D 임무 비행포인트 시험비행 결과



[카메라 화각을 이용한 3D 임무 비행포인트 시험비행 결과]

※ 테스트 임무비행계획 : 시작지점에서 1초(Time Interval Trigger) 마다 4K 정지영상 촬영
※ 고해상도 영상합성 시간 지연으로 해상도 Resize(640*480)
※ 수식 산출값(카메라, 짐벌, 위치, 고도 등)으로 생성한 임무비행포인트 현장 테스트 결과 이미지

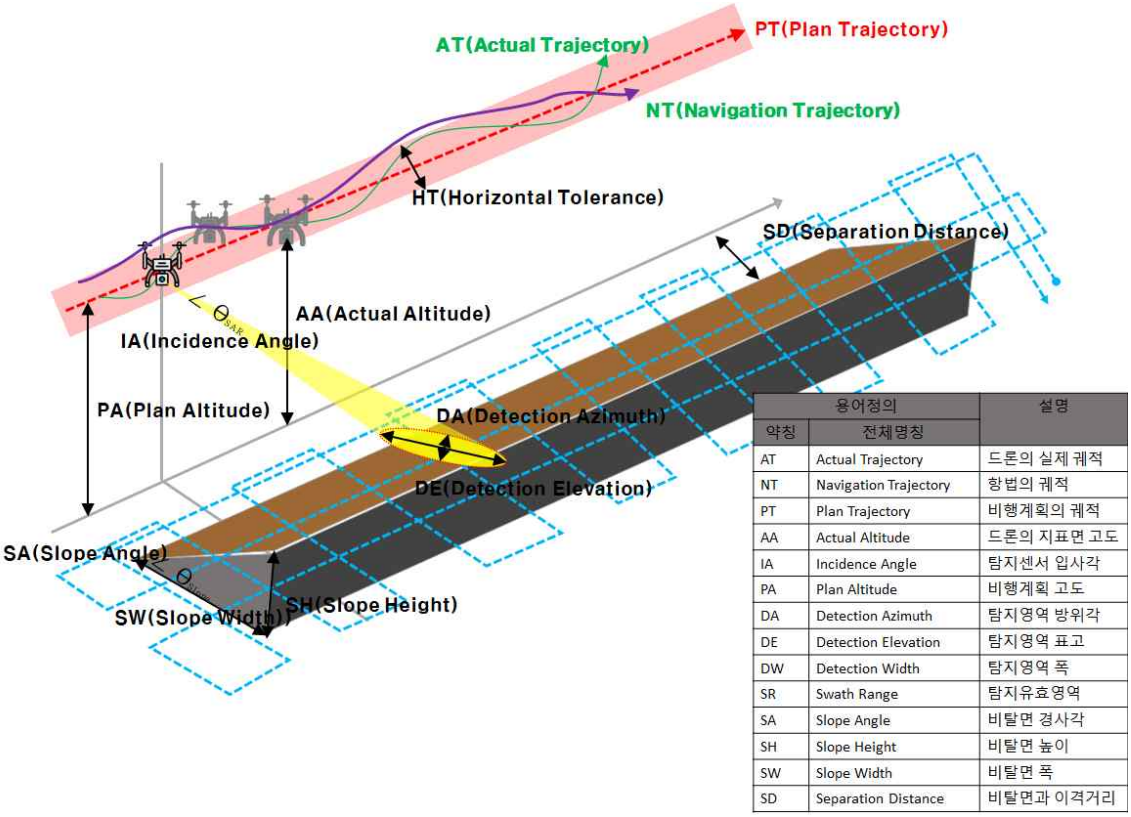
○ 시험용 드론 기반 3D 임무 비행포인트 시험비행 비행계획서(자체, 상용SW)



[임무비행포인트 비행계획]

○ 고정밀 항법센서 및 LiDAR 기반 비탈면 DEM 생성 기법 적용

- 드론을 활용한 고정밀 항법센서 및 고도센서 기반 비탈면 DEM 생성 기법



용어정의		설명
약칭	전체명칭	
AT	Actual Trajectory	드론의 실제 궤적
NT	Navigation Trajectory	항법의 궤적
PT	Plan Trajectory	비행계획의 궤적
AA	Actual Altitude	드론의 지표면 고도
IA	Incidence Angle	탐지센서 입사각
PA	Plan Altitude	비행계획 고도
DA	Detection Azimuth	탐지영역 방위각
DE	Detection Elevation	탐지영역 표고
DW	Detection Width	탐지영역 폭
SR	Swath Range	탐지유효영역
SA	Slope Angle	비탈면 경사각
SH	Slope Height	비탈면 높이
SW	Slope Width	비탈면 폭
SD	Separation Distance	비탈면과 이격거리

[드론을 활용한 센서 기반 비탈면 DEM 생성 기법]

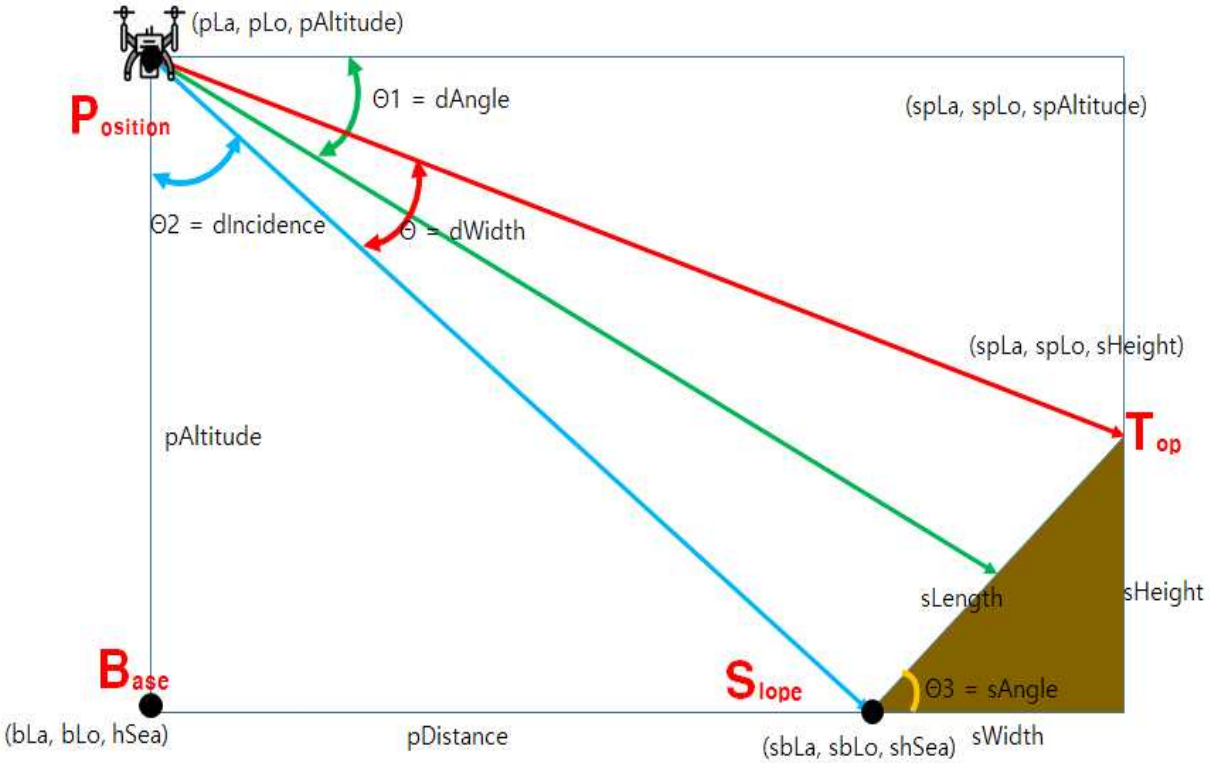
- 고정밀 항법센서 및 고도센서 기반 비탈면 DEM 생성 관련 설정

[센서 기반 비탈면 DEM 생성 설정값]

구분		적용	비고
LiDAR			
	LiDAR 측위방향	수직 하방	Install Margin : 30㎝
	임무비행고도	5m	AGL 기준
	임무비행계획	격자 패턴 비행	Pattern Margin : 0.5m, 1m
	임무비행속도	1m/s	등속도 비행
광학카메라			
	광학카메라 촬영방향	수직 하방	확인용
	영상촬영모드	Time Interval	셔터 1초, 중복도 80%
	광학 영상 해상도	4K	정지영상
RTK-GPS			
	GCP 구축	누적	1~4곳
	Survey In Acc	0.5m 급	RTK Base
	RTK 정확도	Float-Float단계	Fixed-Fixed 활성화 어려움
고도측정			
	고도유지모드	Hybrid Mode	LiDAR-BARO
	고도측정 시간	8Hz	

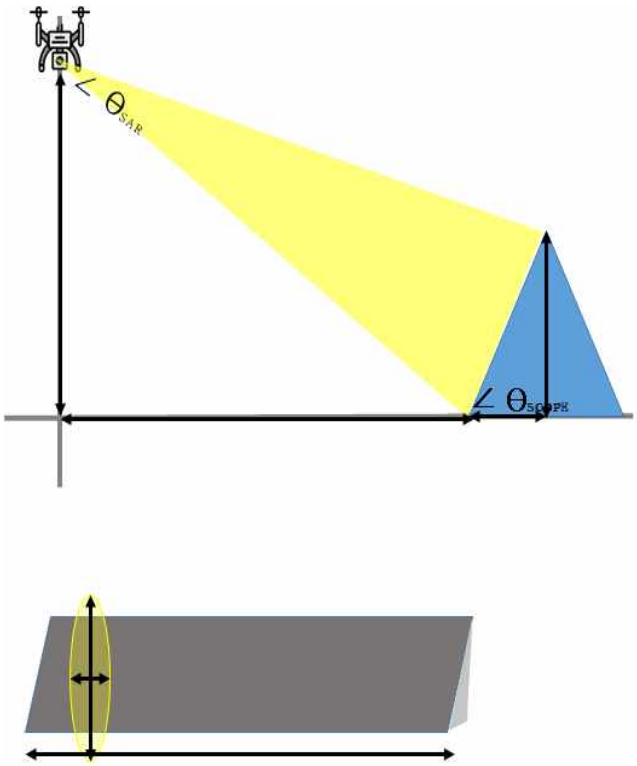
○ 고정밀 항법센서 및 LiDAR 기반 비탈면 3D 임무비행포인트 생성 기법 적용

- 3D 임무비행포인트 생성 개념



[3D 임무비행포인트 생성 설계 개념]

- 3D 임무비행포인트 생성 기반 정보리스트



설정 리스트	코드명	설정값
사면 높이	SET_sHeight	
사면 길이	SET_sLength	
사면 폭	SET_sWidth	
사면 경사각도	SET_sTilt	
착륙상태 해수면고도	SET_SeaLvlLanding	
이륙 설정고도	SET_SetAltTakeoff	
지표면 측정고도	SET_MeaAltGround	
보정 해수면고도	SET_SeaLvlCorrection	
이륙상태 지표면고도	SET_GrdAltTakeoff	

산출 리스트	코드명	산출값
사면과 이격거리	CALC_sDistance	
이륙고도	CALC_TakeoffAlt	
시작위치	CALC_LaStart	
	CALC_LoStart	
시작고도	CALC_AltStart	
끝위치	CALC_LaEnd	
	CALC_LoEnd	

표출 리스트	코드명	표출값
탐지 중심좌표	SHOW_CenterCoord	
탐지 장반경	SHOW_LongAxis	
탐지 단반경	SHOW_ShortAxis	

[3D 임무비행포인트 생성 정보 리스트]

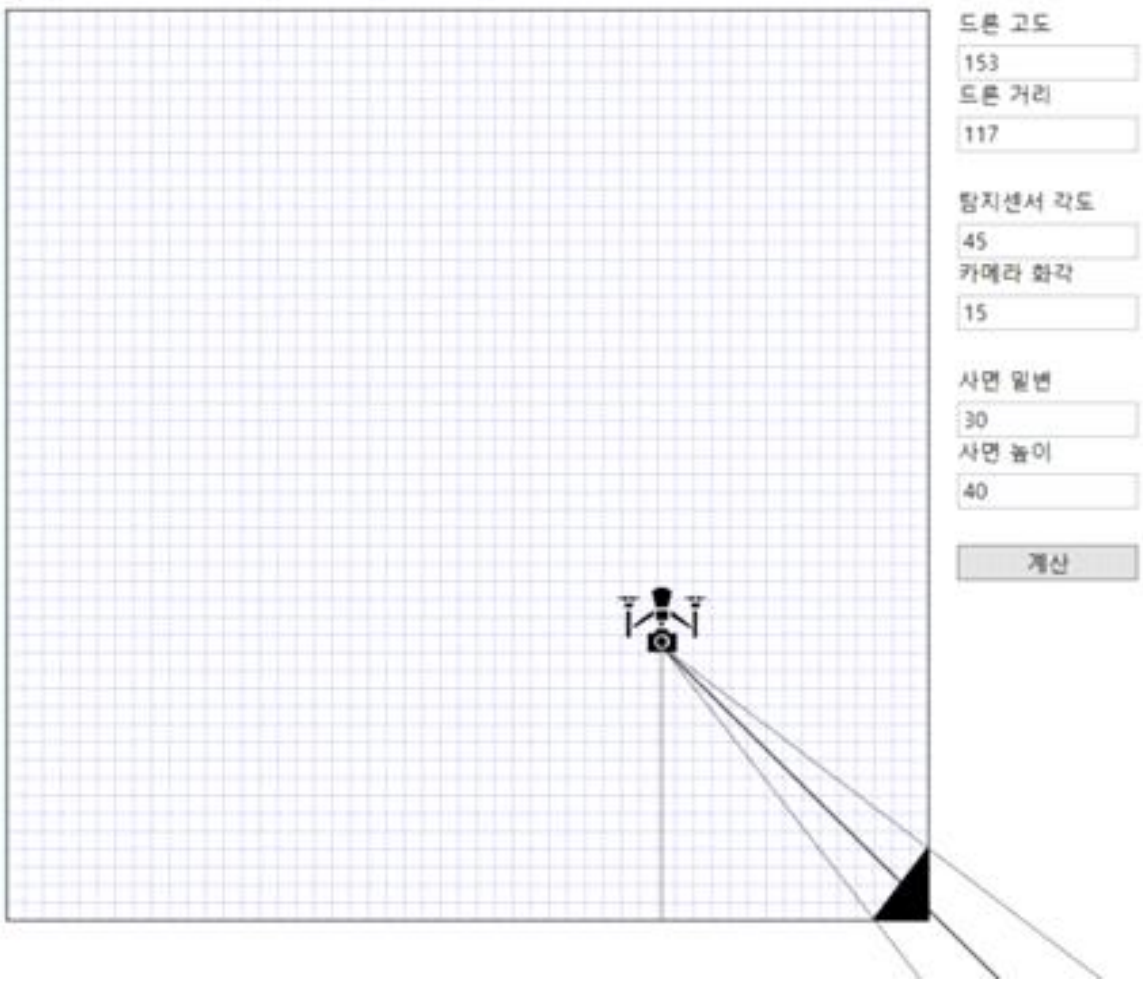
- 고정밀 항법센서 및 고도센서 기반 3D 임무비행포인트 생성 관련 설정

[센서 기반 임무비행포인트 생성 설정값]

구분		적용	비고
LiDAR			
	LiDAR 측위방향	수직 하방	Install Margin : 30㎝
	임무비행고도	산출기준	AGL 기준
	임무비행계획	직선왕복 비행	
	임무비행속도	1m/s	등속도 비행
광학카메라			
	영상촬영모드	Time Interval	셔터 1초, 중복도 80%
	광학 영상 해상도	4K	정지영상
	광학카메라 촬영방향	수직 하방 또는 임무방향	확인용
RTK-GPS			
	GCP	이·착륙	1~2곳
	Survey In Acc	0.5m 급	RTK Base
	RTK 정확도	Float-Float단계	Fixed-Fixed 활성화 어려움
고도측정			
	고도유지모드	Hybrid Mode	LiDAR-BARO
	고도측정 시간	8Hz	

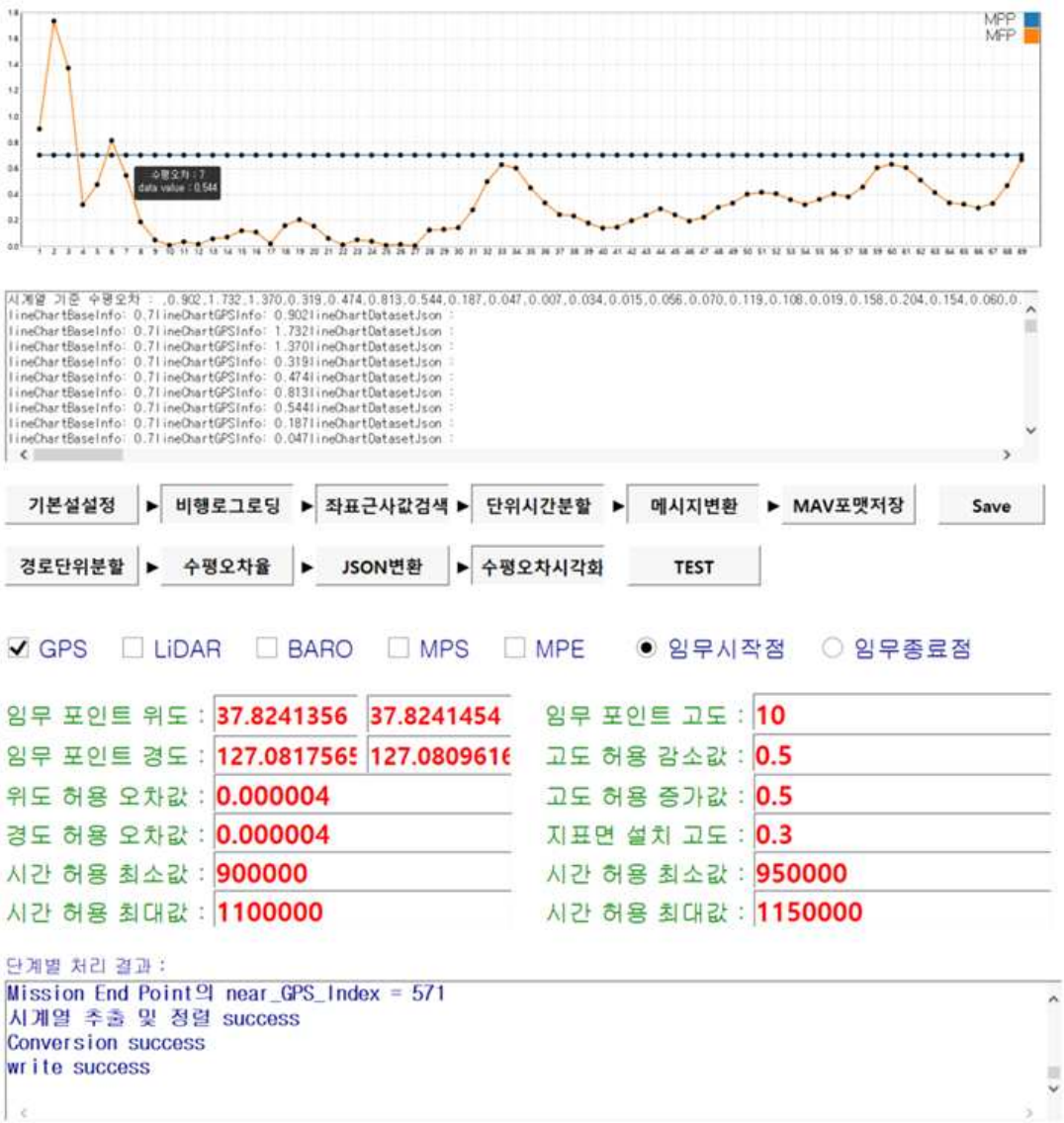
○ 고정밀 항법센서 및 LiDAR를 활용한 특정 임무영역 탐지를 위한 직선 3D 임무 비행 포인트 생성 기법 적용

- 3D 임무비행포인트 산출 검증 시뮬레이터



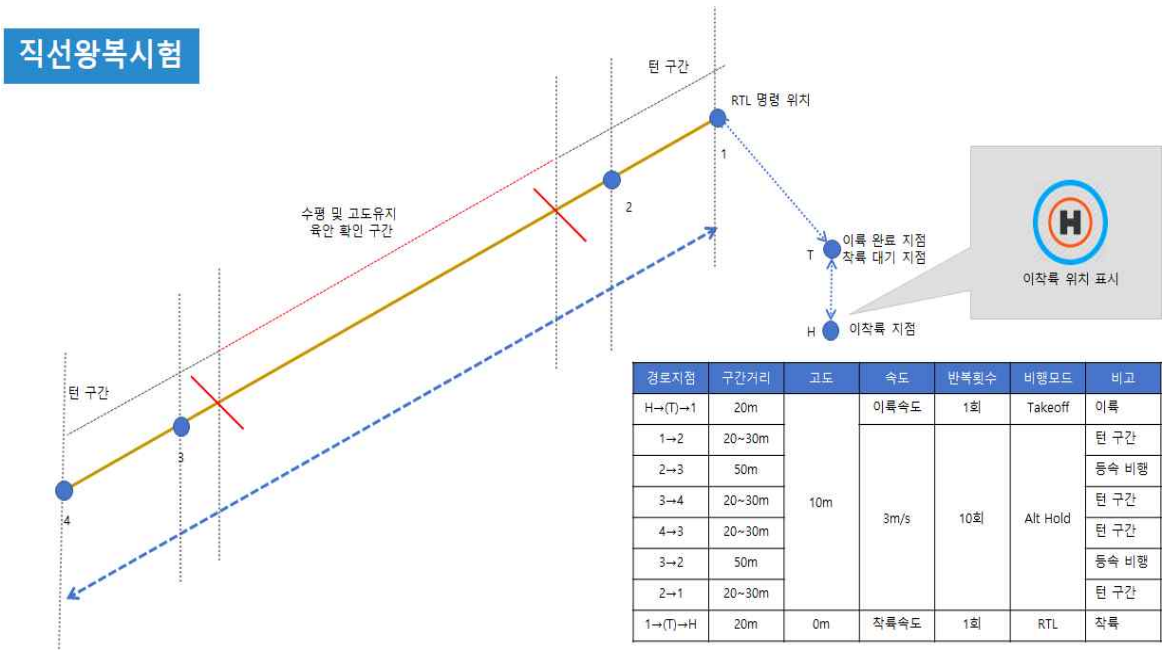
[3D 임무비행포인트 산출 검증 시뮬레이션 실행 화면]

- 3D 임무비행포인트 산출 및 임부비행데이터 분석 로직



[임무비행비행 포인트 산출 및 임무비행데이터 분석 로직 실행화면]

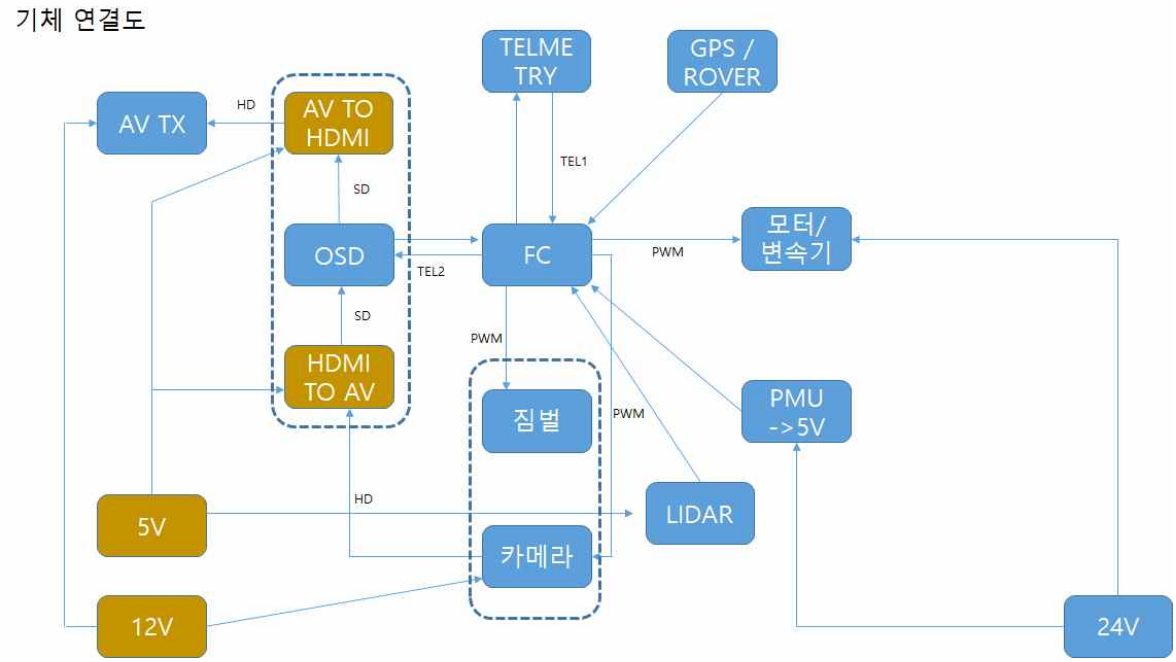
- 3D 임무비행포인트 산출 및 임부비행모니터링 분석용 비행데이터 생성



경로지점	구간거리	고도	속도	반복횟수	비행모드	비고
H→(T)→1	20m	10m	이륙속도	1회	Takeoff	이륙
1→2	20~30m		3m/s	10회	Alt Hold	턴 구간
2→3	50m					등속 비행
3→4	20~30m					턴 구간
4→3	20~30m					턴 구간
3→2	50m					등속 비행
2→1	20~30m					턴 구간
1→(T)→H	20m	0m	착륙속도	1회	RTL	착륙

[직선 왕복]

- 3D 임무비행포인트 산출 및 임부비행 모니터링 분석용 드론 구조



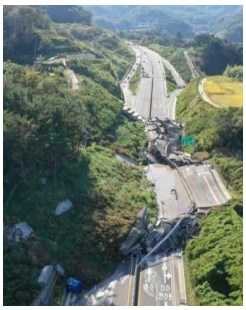
[시험용 제작 드론 구조도]

[시험용 제작 드론 스펙]

MB8F1100RT(가칭)			
비행성능	최대속도 : 7m/sec	순항속도 : 3m/sec	상승속도 : 3m/sec
기체규격	전폭 : 1045mm		전고 : 620 mm
기체중량	자체중량 : 8kg (배터리 포함)		최대 이륙중량 : 11kg
비행시간	22,000mAh 배터리 기준		최대 : 15 분
동력부	모터 : 400kv (8 개)		
	변속기 : 50A (8 개)		
	프로펠라 : 15인치8개		
배터리	리튬폴리머 : 6 셀 1p 사용 (15,000 ~ 22,000mAh 사용가능)		
조종기	8 채널 이상 조종기		
FC	Pixhawk 2.1		

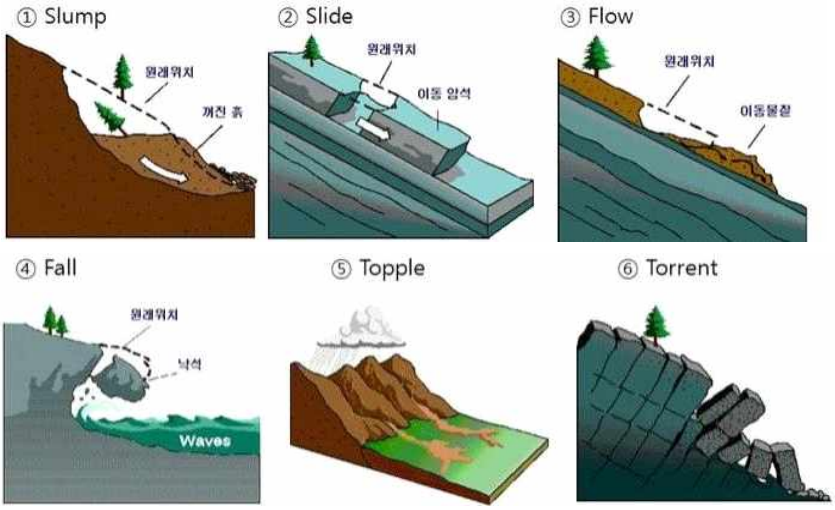
(4) 드론 촬영 비탈면 SAR 기술개발 요구사항 정의

○ 국내 비탈면 특성 조사
- 자연재해 : 집중호우와 같은 강우패턴 변화로 인해 비탈면 피해 증가
- 인공재해 : 건설부지 부족으로 부지 개발로 인한 급경사 비탈면 증가




[4번국도 산사태] [상도 유치원 비탈면 붕괴]

- 다양한 비탈면 파괴 유형을 대비해야 함
  • 비탈면 배면의 인장균열 파악시 대규모 비탈면 붕괴 대처 가능



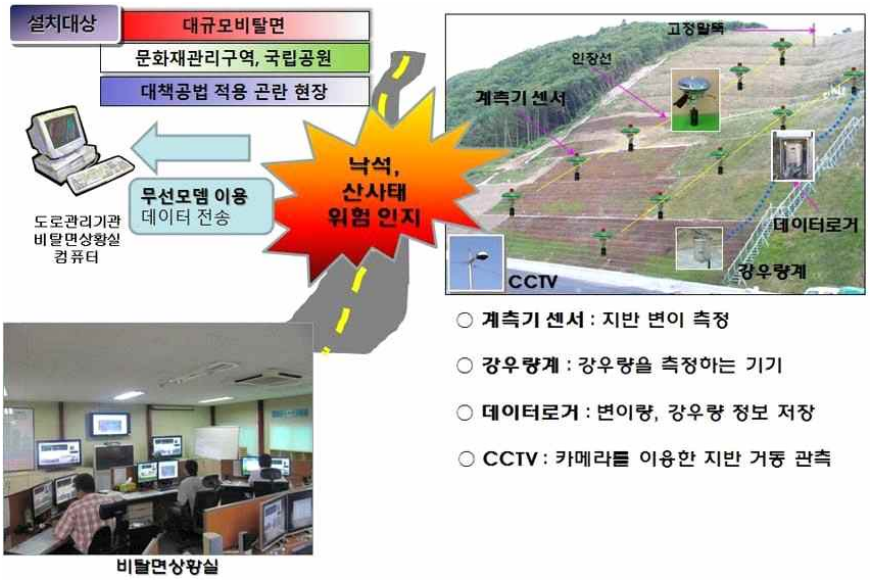
[비탈면 붕괴 유형 (교육부 우면산 산사태)]

- 국내 산지는 대부분 수목으로 이루어져 있으며, 절취된 표면도 수풀로 덮여있기 때문에 지표 변화 감지가 어려움



[비탈면 수목 및 절취면 수풀 식생 현황 (국도 비탈면)]

○ 기존 비탈면 변위 측정 방법 조사
- 비탈면에 무선망 관리 계측기 설치
· 설치 지점만 측정 확인 가능



[도로 비탈면 관리 시스템 (건설기술 2018.09.27.일자)]

- 비탈면 외부에서 거동 확인
  · 정기적 육안 관찰 조사 : 접근이 어려운 지점 발생, 소요 시간이 김
  · 지상 광파기 조사 : 타겟 설치 지점만 확인가능
  · 지상 3D 레이저 스캐너 조사 : 지상 관측시 비탈면 조사 음영 발생, 수풀 식생이 반영된 표면 측정
  · 실화상 드론 조사 : 수목, 수풀 식생이 반영된 영상 측정
  · 열화상, 다분광, 초분광 활용 드론 조사 : 수목, 수풀 식생에서 반사된 영상 측정
  · 라이다 활용 드론 조사 : 수풀 식생이 반영된 표면 측정



[육안 관찰 조사]



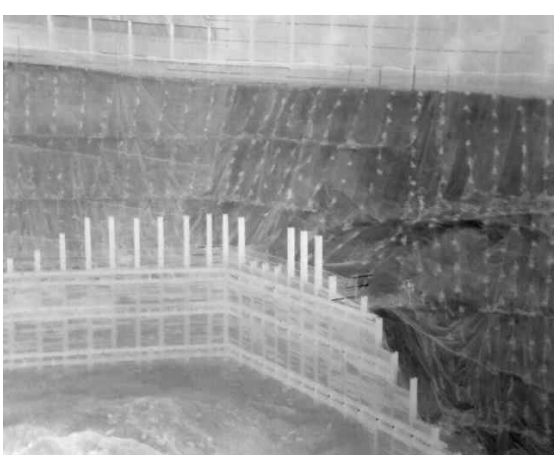
[드론 조사]



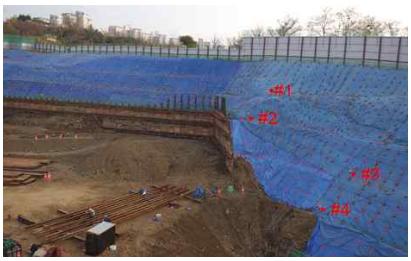
[레이저 스캐너 조사]

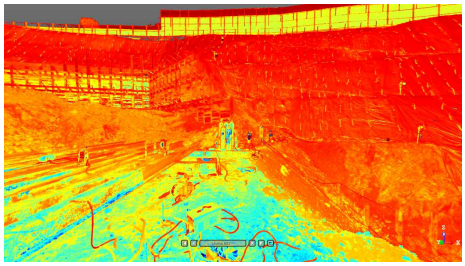
○ 비탈면 변위 측정 활용지침 요구사항 정의
- 우리나라 3만개의 국도 비탈면 및 건설부지 시공 굴착 비탈면 모두 관리할 수 있는 신속한 측정 기술이 요구됨
- 일부 지점이 아닌 비탈면 모든 지점의 거동 확인이 가능한 측정 기술이 필요
- 비탈면 수풀뿐만 아니라 비탈면 배면의 수목과 수풀 아래 지표 변위를 감지가 가능한 L밴드 SAR 활용한 기술 개발 추진

○ 변위 기준값 선정 지상 계측방법 개발
- 향후 L-band SAR 센서 드론 측정시 변위량 기준값 결정 계측 방법 개발
  · 열화상 드론 조사로 비탈면 리스크 구간을 예상하여 계측 지점 선정
  · 비탈면 내부 계측(기울기 센서)과 비탈면 외부 계측(광파기, 레이저 스캐너, 실화상 드론)을 통합하여 한 지점에서 다양한 계측이 가능한 통합 시스템 개발(스마트폴 특허 출원) 및 비탈면 시공 계측 수행

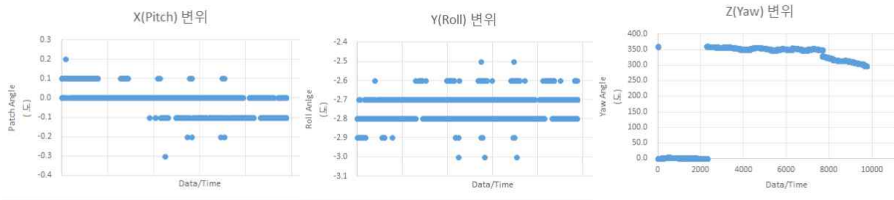



[드론 열화상 사진][통합 계측 시스템(스마트폴 특허 출원)]





[스마트폴 계측 비탈면 드론 실화상 사진] [레이저 스캐너 계측 자료]



[스마트폴 #1 계측 결과 - 1회/분 측정]

나. 드론을 활용한 포트홀 탐지 기술 개발

(1) 드론 촬영 영상 분석 기반 포트홀 객체 인지 · 추출 기술 개발

○ 포트홀 탐지 기술(Aparna, 2019)

구분	영상기반	3축 가속계 기반	레이저 기반	스테레오 카메라
사용 기술	2d 이미지처리	힘, 회전 및 방향	빛반사 이용 3d 이미지 생성	다중카메라 기반 3d 이미지 생성
응답 속도	높음	낮음	낮음	높음
처리	복잡한 이미지처리 알고리즘 또는 인공지능	영상기반에 비해 비교적 간단한 알고리즘	3D 포인트 클라우드 수집결과 이용	다중 카메라 인식 기반 조합 3d 이미지 구성
비용	렌즈에 따라 비용이 다름	낮음	높음	높음
포트홀 탐지특징	이미지 내 포트홀 특징 및 사이즈 기반 탐지	진동 기반 탐지	3d 이미지 기반	3d 이미지 기반
정확도	알고리즘 의존적 정확도	높음(맨홀, 이음매에서 오탐지 가능성)	높음	카메라 및 사용 알고리즘별 상이

○ 항공 영상 이미지 분석

No	촬영일/폴더명	파일명	비행시간	사진매수	지역	포트홀수
1	180629	Flight_1_R	4	59	전북 김제시 흥사동 229-7	2
2	180629	Flight_2_R	3	48	전북 김제시 부량면 신용리 225-1	1
3	180629	Flight_3_R	4	51	전북 김제시 백구면 삼정리 508-2	1
4	180629	Flight_4_R	7	123	전북 전주시 덕진구 용정동 282-37	3
5	180704	Flight_1_R	5	87	전남 함평군 월야면 외치리 335-7	1
6	180704	Flight_2_R	4	60	전남 영광군 법성면 대덕리 222-2	1
7	180704	Flight_3_R	3	39	전남 영광군 법성면 대덕리 788-8	1
8	180704	Flight_4_R	4	61	전남 영광군 법성면 덕흥리 876-3	1
9	180704	Flight_5_R	4	60	전남 함평군 학교면 사거리 1068	1
10	180704	Flight_6_R	4	57	전남 함평군 학교면 죽정리 산5-10	1
11	180704	Flight_7_R	4	59	광주광역시 남구 이장동 857-19	2
12	180704	Flight_8_R	5	81	광주광역시 남구 이장동 661-5	2
13	180704	Flight_9_R	5	86	광주광역시 남구 양과동 1276-7	3
14	180719	Flight_1_R	8	139	충남 논산시 채운면 화산리 539-1	3
15	180719	Flight_2_R	5	82	전북 익산시 용안면 교동리 86-15	1
16	180730	Flight_1_R	13	224	충남 공주시 계룡면 월암리 120-4	4
17	180730	Flight_2_R	7	114	충남 공주시 소학동 450-3	2
18	180730	Flight_3_R	10	186	충남 공주시 신기동 827	4
19	180809	Flight_1_R	3	54	전남 화순군 화순읍 감도리 913	1
20	180809	Flight_2_R	4	56	전남 화순군 화순읍 내평리 717	1
21	180809	Flight_3_R	4	62	전남 보성군 회천면 전일리 305-3	1
22	180809	Flight_4_R	4	84	전남 보성군 보성읍 봉산리 1073-4	1
23	180809	Flight_5_R	4	75	전남 장흥군 장동면 배산리 632-1	2
24	180809	Flight_6_R	6	115	전남 나주시 산포면 등수리 323	2
25	180809	Flight_7_R	7	138	전남 나주시 남평읍 교원리 721	4
26	180810	Flight_1_R	8	154	전북 정읍시 금붕동 1106-4	2

- 총 26개소, 총 26개의 정합 이미지

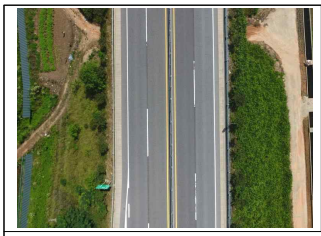


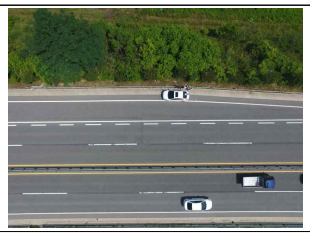
전라남도 함평군 월야면 외치리 335-7



전라북도 전주시 덕진군 용정동 282-37

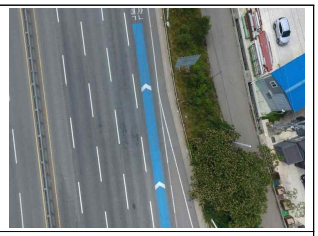
- 총 2,354개 원본 이미지





전라남도 함평군 월야면 외치리 335-7





전라북도 전주시 덕진군 용정동 282-37

- 총 91,716개 정합 이미지의 224p X 224p 타일



그림자



차선



비도로



도로



맨홀



포트홀

- 총 48개의 포트홀









포트홀 예시

○ 특징점 기반 포트홀 탐지를 위한 관련 연구

- OpenCV 3.2 for Windows

· 개념 : 입력된 영상을 통하여 사용자가 정의한 특징점을 탐지하기 위하여 제공되는 오픈소스 기반 영상처리 라이브러리

· 활용 : 윈도우 OS, C++ 기반하여 실시간으로 입력되는 영상으로 포트홀 탐지 사용

· 연구 및 실험 내용 : 도로주행 테스트 동영상 재생, 해당 라이브러리가 제공하고 있는 알고리즘을 사용하여 본 연구의 특징점 분류를 테스트하고, 본 연구의 적용 가능성을 확인

· 결과 : 본 연구의 기본 영상처리 라이브러리로 사용함

- RGB2Y

· 개념 : RGB(Red, Green, Blue) 기반의 3채널 컬러 영상을 Y(흑백) 단일 채널 영상으로 변환하는 알고리즘

· 활용 : 특징점 또는 딥러닝 기반의 알고리즘 연산 시 메모리 사용량을 줄이기 위하여 사용

· 연구 및 실험 내용 : 본 연구의 특징점 기반 알고리즘과 딥러닝 기반의 알고리즘 속도 향상에 따른 성능 저하 여부 테스트하고, 본 연구에 적용 가능성을 확인

· 결과 : 본 연구의 기본 영상처리 알고리즘으로 사용됨.

- 이진화 (Threshold)

· 개념 : Y(흑백) 단일 채널 영상의 값(0 ~ 255)에서 임의의 값 기반으로 작으면 최소 값(0), 크면 최대 값(255)으로 0과 255로만 표현하여 배경 및 물체 나타내는 알고리즘, 영상 프레임별 자동 임의의 값 설정하는 동적 및 전체 프레임에서 고정값인 정적 알고리즘 존재

· 활용 : 특징점 기반 알고리즘의 포트홀 영역 탐지하기 위하여 사용

· 연구 및 실험 내용 : 본 연구의 특징점 기반 탐지 알고리즘으로 사용하기 위해 중간 값(125) 이용 탐지 정확도 및 적용 가능성 유무 판단, 동적 알고리즘 적용 가능성 확인

· 결과 : 정적 이진화 알고리즘 사용 불가, 동적 이진화 알고리즘 사용됨.

- 오츠(OTSU) 알고리즘

· 개념 : 동적 이진화(Adaptive Threshold) 알고리즘 중 하나. 동적 이진화에 사용되는 임의의 값 설정하여 배경 및 물체 나타내는 알고리즘

· 활용 : 특징점 기반 알고리즘의 포트홀 탐지하기 위하여 사용

· 연구 및 실험 내용 : 본 연구의 특징점 기반 탐지 알고리즘으로 사용하기 위해 포트홀 영역 탐지 정확도 및 적용 가능성 유무 판단

· 결과 : 본 연구의 특징점 기반 알고리즘 중 하나로 사용됨

- SLIC(Simple Linear Iterative Clustering) 알고리즘

· 개념 : 유사 픽셀 클러스터링(군집화), 셀로 표현하는 알고리즘

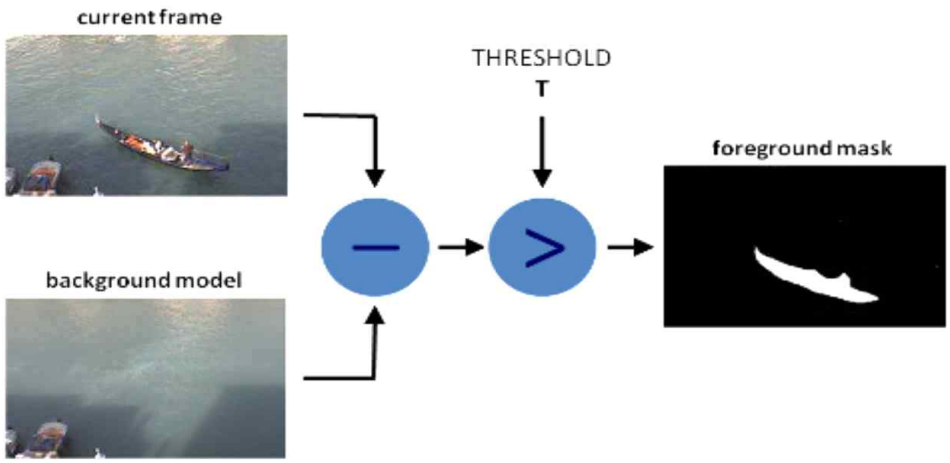
· 활용 : 특징점 기반 알고리즘의 포트홀 탐지하기 위하여 사용

· 연구 및 실험 내용 : 본 연구의 특징점 기반 알고리즘의 포트홀 영역탐지에 사용하기 위해 옵션(셀 크기, 블러, 연결성, 반복횟수) 별 테스트 하고, 본 연구에 적용 가능성을 확인

· 결과 : 본 연구의 특징점 기반 알고리즘 중 하나로 사용됨

- 배경 제거(Background Subtractor)

· 개념 : 영상 내부의 배경과 전경 분리 알고리즘. 기반하는 알고리즘에 따라 GMG, MOG, MOG2, KNN으로 구분됨



[배경 제거 알고리즘. 배경/현재 이미지 제어 통한 물체 획득(OpenCV 3.2.0, n.d.)]

· 활용 : 포트홀 영역탐지를 위해 사용

· 연구 및 실험 내용 : 도로 노면이 아닌 특징들을 드러내기 위해 사용됨. 화이트 포트홀, 패치, 크랙, 포트홀, 차선 등이 드러남. 다수결 기반 하이브리드 포트홀 탐지 알고리즘 중 하나로 사용됨

· 결과 : 본 연구의 특징점 기반 알고리즘 중 하나로 사용됨

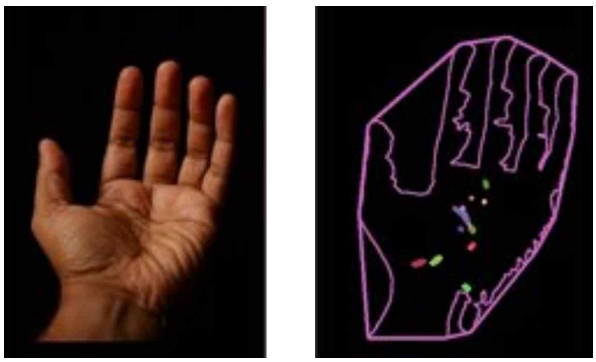
- 컨벡스 헐(Convex Hull)

· 개념 : 경계선의 꼭짓점을 이어 최소 크기의 다각형 그리는 알고리즘

· 활용 : 슈퍼픽셀 결과 셀 경계선의 다각형 검출

· 연구 및 실험 내용 : 본 연구의 특징점 기반 알고리즘의 슈퍼픽셀 셀 경계선 특징 이용 실

험. 도로 노면과 같이 유사 색상 및 넓은 이미지의 경우 사각형 경계선 돌출. 포트홀과 같이 원형 또는 타원형의 형태의 경우 사각형이 아닌 다각형 검출 확인
· 결과 : 본 연구의 탐지 정확도 향상을 위해 특징점 기반 알고리즘의 슈퍼픽셀 이후 사용됨



[컨벡스헐 사용 예(OpenCV 3.2.0, n.d.)]

- 시간축 프레임 차 영상(Frame Temporal Difference)
· 개념 : 시간(t) 축 기반 첫 번째 (t)와 두 번째(t+1) 프레임 차
· 활용 : 특징점 기반 알고리즘의 포트홀 탐지하기 위하여 사용
· 및 실험 내용 : 도로 노면이 아닌 포트홀과 같이 비 유사 영역 나타내, 본 연구의 특징점 기반 알고리즘의 포트홀 탐지를 위한 탐지 정확도 및 오탐율 실험
· 결과 : 본 연구의 특징점 기반 탐지 알고리즘 중 하나로 사용됨
- 모폴로지(Morphology)
· 개념 : 영상의 화질 개선 및 특징 돌출을 위해 사용, 영역을 넓히는 팽창, 영역을 좁히는 침식, 침식 후 팽창하여 경계선을 잇는 열기 그리고 팽창 후 침식으로 구멍을 제거할 수 있는 닫기, 4가지 연산 알고리즘
· 활용 : 본 연구의 특징점 기반 알고리즘에서 흰색의 잡음 제거 및 포트홀 영역 정확도 향상
· 연구 및 실험 내용 : 본 연구의 특징점 기반 알고리즘의 정확도 향상을 위한 침식, 팽창, 열기 그리고 닫기 연산 테스트, 본 연구에 적용 가능성을 확인
· 결과 : 본 연구의 탐지 알고리즘 정확도 향상을 위하여 사용됨



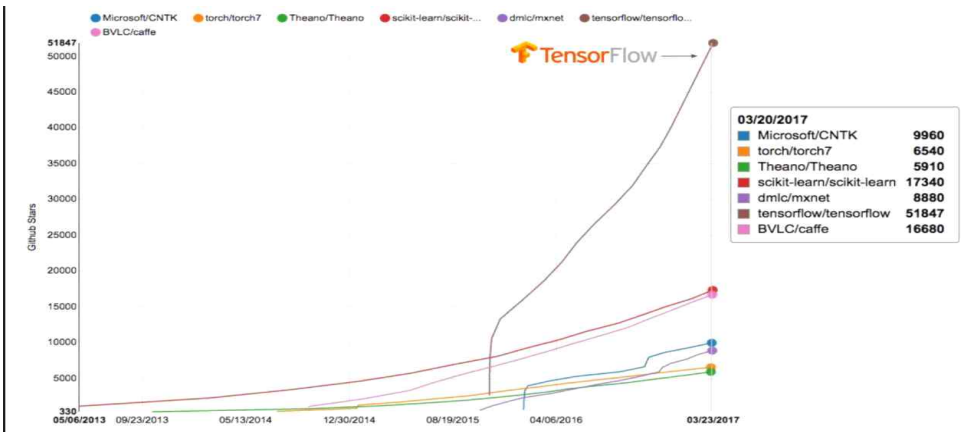
모폴로지 닫기 연산(OpenCV 3.2.0, n.d.)



[모폴로지 열기 연산(OpenCV 3.2.0, n.d.)]

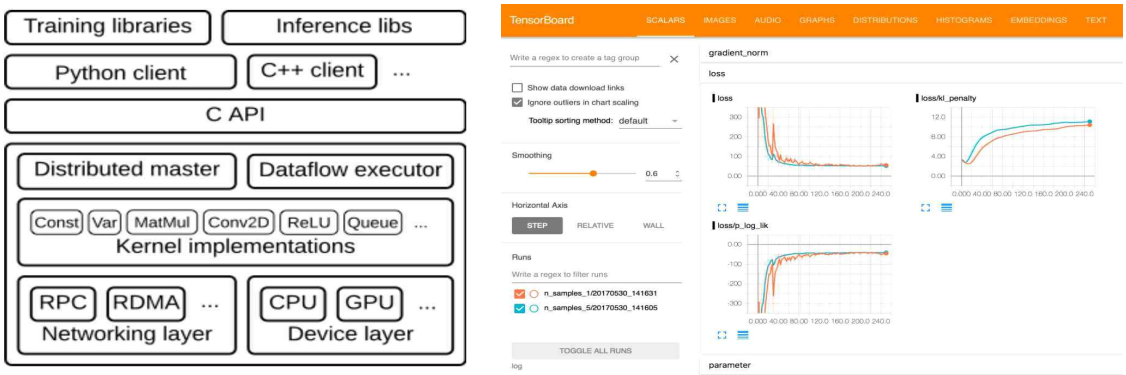
- 세일리언시 맵(Saliency map)
  · 개념 : 영상 내부의 특징되는 물체만 드러내는 알고리즘
  · 활용 : 포트홀 영역탐지를 위하여 사용
  · 연구 및 실험 내용 : 높은 오탐율 및 정탐율을 가짐
  · 결과 : 본 연구의 특징점 기반 탐지 알고리즘 사용됨
- 웨이블릿 에너지 필드(WEF: Wavelet Energy Field)
  · 개념 : 이미지 압축 방식인 웨이블릿 압축을 기반으로 이미지 질감을 나타내는 알고리즘
  · 활용 : 포트홀 후보 영역 질감을 판단하여 포트홀 탐지를 위하여 사용
  · 내용 : 본 연구의 탐지 알고리즘 초기 탐지 알고리즘. 타 논문의 높은 정확도 기반하여 적용하였으나, 낮은 정확도를 보였으나, 타 알고리즘 조합 시 포트홀 후보 검출 가능 확인
  · 결과 : 본 연구의 특징점 기반 탐지 알고리즘 중 하나로 사용됨

○ 딥러닝 기반 포트홀 탐지를 위한 관련 연구
- 딥러닝(기계학습) 프레임워크
  · 개념 : 딥러닝을 위한 하드웨어 및 소프트웨어 집합체. 개발에 있어 하드웨어 및 라이브러리를 제공하여 딥러닝 개발에 집중할 수 있도록 함. 대표적인 딥러닝 프레임워크로 구글 텐서플로우(Tensorflow), 테노(Theano), 카페(Caffe) 등이 있음.
  · 활용 : 포트홀 탐지를 위한 딥러닝 프레임워크 선정을 위한 분석
  · 연구 및 실험 내용 : 구글 텐서플로우가 가장 많은 사용자층을 보유
  · 결과 : 본 연구의 기계학습 프레임워크로서 구글 텐서플로우를 검토



[깃허브 딥러닝 프레임워크 인기 비교(Google Brain Team, n.d.)]

- 구글 텐서플로우(Google TensorFlow)
  · 개념 : 구글에서 개발한 딥러닝 프레임워크. C언어 기반으로 작성. 하드웨어 자원 제어 및 딥러닝 모델 작성을 위한 라이브러리를 지원
  · 활용 : 포트홀 탐지용 딥러닝 모델 생성을 위한 기반 프레임워크로서 사용



[텐서플로우 아키텍처(Google Tensorflow github, n.d.) 및 텐서보드 예제(Google Tensorboard, n.d.)]

· 연구 및 실험 내용 : 가장 많은 딥러닝 프레임워크 사용자층 보유. 많은 피드백 및 사용 예제가 많음. 텐서보드(Tensorboard) 도구를 통하여 모델의 학습 상태, 오류 상황, 모델 형태 분석 등을 시각적으로 제공. 본 연구에서 딥러닝에 집중할 수 있는 도구로서 가능성을 확인
· 결과 : 본 연구의 딥러닝 프레임워크 모델로서 사용

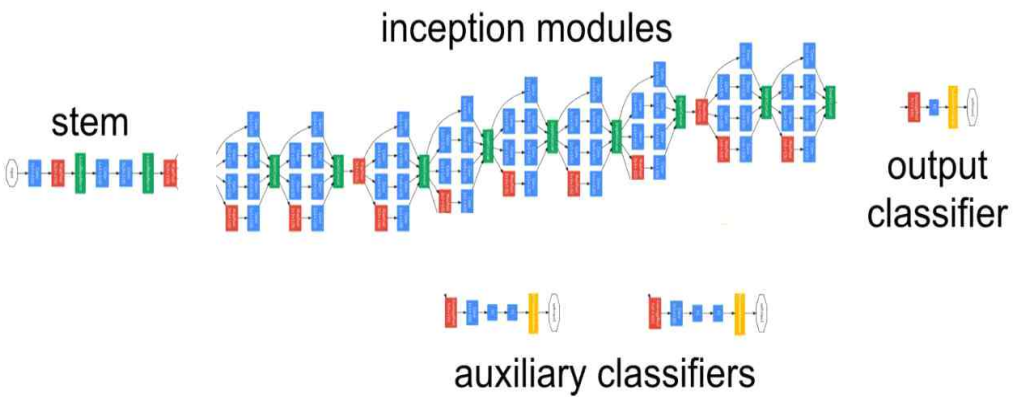
– 이미지 셋
· 개념 : 딥러닝 모델을 학습할 때 사용되는 이미지 집합으로 이미지들이 각 카테고리에 맞춰 저장하여 딥러닝 모델은 카테고리별로 이미지를 학습
· 활용 : 포트홀 및 비포트홀 이미지로 분류 및 딥러닝 모델에 적용
· 연구 및 실험 내용 : 도로주행 영상의 이미지화 후 포트홀 및 비포트홀 이미지로 분류
· 결과 : 본 연구의 딥러닝 학습 모델에 적용하기 위한 이미지

– 딥러닝 학습 모델
· 개념 : 딥러닝 모델이 이미지 셋을 통해 학습된 상태
· 활용 : 딥러닝 모델이 포트홀 이미지 셋을 학습하여 포트홀 탐지에 사용
· 연구 및 실험 내용 : 텐서플로우 슬림에서 제공되는 모델을 포트홀 이미지 셋을 사용하여 학습
· 결과 : 포트홀에 특화된 딥러닝 모델을 생성



[데이터셋과 학습기를 사용한 모델 생성 과정(Google TensorFlow Serving, n.d.)]

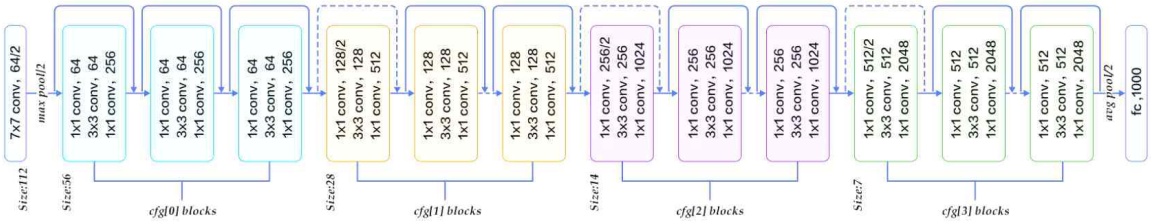
– 인셉션(Inception) 모델
· 개념 : 구글에서 개발하고 ILSVRC 2014에서 우승한 GoogLeNet 학습 모델의 개량형 버전. 최신 모델은 v4. 인셉션 모델을 사용하여 이미지 분류를 수행
· 활용 : 포트홀 학습 모델에 사용될 딥러닝 모델
· 연구 및 실험 내용 : 포트홀 학습 데이터셋을 사용하여 포트홀 이미지 학습 및 성능 평가
· 결과 : 본 연구의 딥러닝용 모델로서 사용



[구글 인셉션 모델 구조(hskang9, n.d.)]

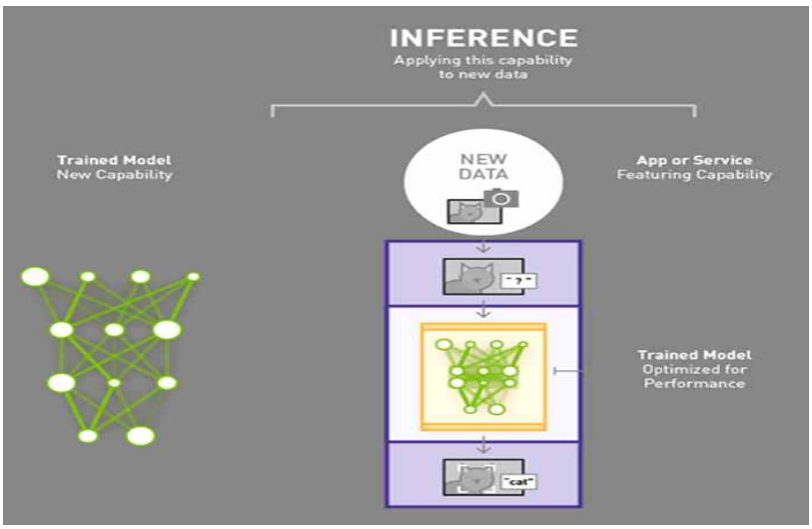
– 레즈넷(ResNet) 모델
· 개념 : 마이크로소프트에서 개발하고 ILSVRC 2015에서 우승한 학습 모델로 최신 모델 버전

은 v2로 학습 시간이 매우 오래 걸리는 반면 데이터 판단 시간이 빠른 것이 특징
· 활용 : 포트홀 학습 모델에 사용될 딥러닝 모델
· 연구 및 실험 내용 : 포트홀 학습 데이터셋을 사용하여 포트홀 이미지 학습 및 성능 평가
· 결과 : 본 연구의 딥러닝용 모델로서 사용



[마이크로소프트 레즈넷(ResNet) 모델 구조(Sai Kumar Basaveswara, n.d.)]

- 인셉션-레즈넷(Inception-ResNet) 학습 모델
· 개념 : 구글의 인셉션(Inception)모델을 레즈넷(ResNet) 모델의 특징을 적용하여 데이터 판단 속도를 향상한 모델
· 활용 : 포트홀 딥러닝 학습 모델에 사용될 딥러닝 모델
· 연구 및 실험 내용 : 포트홀 학습 데이터셋을 사용하여 포트홀 이미지 학습 및 성능 평가
· 결과 : 본 연구의 딥러닝 모델로서 사용
- 모바일넷(MobileNet) 모델
· 개념 : 인셉션모델의 저성능 모바일 장치 사용 목적 경량화 모델
· 활용 : 모바일용 포트홀 학습 모델에 사용될 딥러닝 모델
· 연구 및 실험 내용 : 포트홀 학습 데이터셋을 사용한 포트홀 이미지 학습. 대회 우승 모델과 성능 평가
· 결과 : 본 연구의 모바일용 딥러닝용 모델로서 사용
- 인퍼런스(Inference) 모델
· 개념 : 딥러닝 학습 모델이 데이터에 대한 판독을 수행할 수 있는 모델. 딥러닝 모델에서 판독에 사용되는 부분만 남겨진 상태. 한번 인퍼런스화 된 모델은 추가 학습이 불가능. 딥러닝 학습 모델을 라이브러리화하여 타 장치에서 호출하여 사용
· 활용 : 포트홀 딥러닝 학습 모델을 모바일에 사용할 수 있게 이식
· 연구 및 실험 내용 : 포트홀 학습 모델의 판독 부분만 추출 및 고정화 작업 진행. 학습 모델의 라이브러리화를 통해 모바일에서 사용할 수 있도록 인퍼런스화 진행
· 결과 : 모바일용 포트홀 인퍼런스 모델

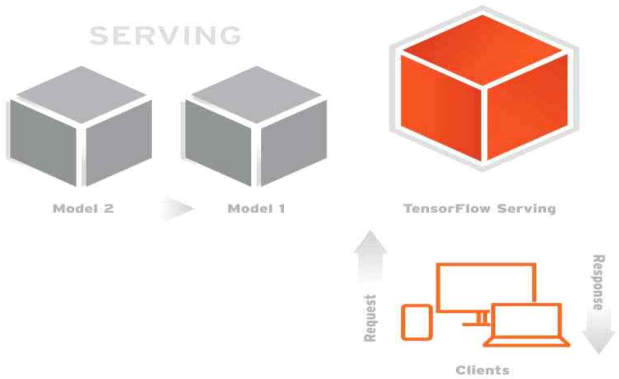


[인퍼런스(Inference) 모델 작동 방법(Nvidia, n.d.)]

- 텐서플로우 슬림(TensorFlow Slim)

· 개념 : 구글에서 제공하는 딥러닝 학습 모델 라이브러리 모음으로 텐서플로우에서 동작할 수 있도록 설계됨. ImageNet Large Scale Visual Recognition Competition (ILSVRC)에서 우승한 모델인 인셉션(Inception), 레즈넷(ResNet), 인셉션(Inception_ResNet), 모바일넷(MobileNet) 모델을 제공
· 활용 : 영상 내 포트홀 이미지 분석을 위한 높은 성능의 딥러닝 모델 적용
· 연구 및 실험 내용 : 본 연구의 딥러닝 기반 포트홀 탐지에서 각 모델을 적용, 모델별 성능을 평가하여 포트홀 탐지 가능성을 확인
· 결과 : 본 딥러닝 기반 포트홀 탐지에 텐서플로우 슬림 모델의 딥러닝 모델 라이브러리 사용

– 텐서플로우 서빙(TensorFlow Serving)
· 개념 : 서버 기반의 원격 딥러닝 학습 모델 서비스 라이브러리로 딥러닝 학습 모델들의 버전 관리 및 모델 버전별 서비스 기능을 제공하여 딥러닝 학습이 어려운 장치에서 딥러닝 학습 가능한 서버의 모델을 원격으로 호출하여 결과 획득 가능



[텐서플로우 서빙의 학습 모델 서비스 흐름도]

· 활용 : 최신 버전의 포트홀 딥러닝 학습 모델을 원격으로 모바일 기기에 판독 결과를 서비스
· 연구 및 실험 내용 : 텐서플로우 서빙 환경 구축 진행

○ 특징점 기반 포트홀 탐지 S/W를 위한 관련 연구

– 안드로이드 카메라 API
· 개념 : 안드로이드 카메라를 통한 영상처리를 효율적으로 진행할 수 있도록 구현된 라이브러리
· 활용 : 안드로이드 카메라를 통한 영상 작업 필요시, 사전에 구현되어있는 기능들을 사용
· 연구 및 실험 내용 : 안드로이드 카메라에 대한 제어와 입력 영상 데이터를 효율적으로 처리하기 위한 Native 라이브러리로, 카메라를 제어하기 위하여 본 연구에 적용 가능성을 확인
· 결과 : 본 연구의 안드로이드 기본 카메라 제어 라이브러리로 사용됨

– Native Development Kit(NDK)
· 개념 : C/C++로 작성된 라이브러리를 JAVA에서 사용할 수 있도록 하는 개발 도구
· 활용 : 기존에 제작된 특징점 기반 포트홀 탐지 알고리즘의 안드로이드 적용
· 연구 및 실험 내용 : 포트홀 탐지 알고리즘이 카메라를 통하여 입력된 영상을 통해 실시간 탐지가 되는지 테스트
· 결과 : 포트홀 탐지 알고리즘 안드로이드 정상 적용 확인

– EXchangable Image File format(Exif) for Android
· 개념 : JPEG, WAV 등의 이미지 확장 포맷에서 사용되는 사진정보를 저장하는 라이브러리
· 활용 : 포트홀 탐지 시 저장된 해당 이미지에 GPS 좌표를 기록을 위해 사용
· 연구 및 실험 내용 : 포트홀로 탐지된 이미지에 현재 좌표를 기록하여 해당 사진만을 가지고 사진이 저장된 위치를 찾아 표기
· 결과 : 정상적으로 표기함을 확인, 위치 정보 기록에 사용

– OpenCV for Android
· 개념 : 입력된 영상을 통하여 사용자가 정의한 특징점을 탐지하기 위하여 제공되는 오픈소스

기반 영상처리 라이브러리

· 활용 : 안드로이드 카메라를 통하여 실시간으로 입력되는 영상으로 포트홀 탐지에 사용

· 연구 및 실험 내용 : 안드로이드의 카메라로부터 입력된 영상기반, 해당 라이브러리가 제공하는 알고리즘을 사용하여 본 연구의 특징점 분류를 테스트하고, 본 연구 적용 가능성 확인

· 결과 : 본 연구의 기본 영상처리 라이브러리로 사용됨

- ffmpeg for Android

· 개념 : 영상 및 음성 데이터를 다양한 코덱으로 저장 또는 변환하는 오픈소스 라이브러리

· 활용 : 포트홀 탐지 어플리케이션 실행 시 해당 영상의 원본 영상을 저장하기 위하여 사용

· 연구 및 실험 내용 : 영상처리를 위하여 사용되는 OpenCV에서 제공되는 영상 저장 라이브러리 사용 시 실제 촬영 시간과 미 일치로 인하여 대체 방안으로 해당 라이브러리 사용

· 결과 : 저장 프레임율을 설정하여 영상을 저장할 수 있도록 설정

- 안드로이드 어플리케이션 UI/UX

· 개념 : 포트홀 탐지 어플리케이션의 사용자 편의성을 위하여 기능 추가 및 설정

· 활용 : 포트홀 탐지 어플리케이션 실행 시 해당 영상의 원본 영상을 저장하기 위하여 사용

· 연구 및 실험 내용 : 터치를 통해 탐지 영역 크기 조정, 설정 페이지를 통한 모드 설정 등을 손쉽게 설정할 수 있도록 적용 테스트 하였으며 정상 작동 확인

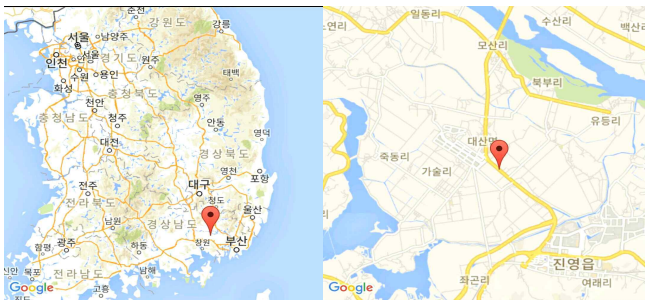
· 결과 : 설정 페이지 UI/UX에서 설정을 통해 160가지의 알고리즘 조합 및 어플리케이션 동작을 설정할 수 있음

- 구글 맵(Google Map) v2 API

· 개념 : 구글에서 제공하고 있는 지도 라이브러리

· 활용 : 입력/저장된 GPS 좌표를 받아 모바일 화면을 통하여 해당 위치 표기

· 연구 및 실험 내용 : 탐지 결과 관리 어플리케이션과 연동하여 포트홀 탐지 위치 지도상 표기



[구글 맵(Google Map v2 API)]

· 결과 : GPS 좌표 입력 시 위치 정상적으로 표기 확인

- ZGallery

· 개념 : 안드로이드 갤러리 어플리케이션 서드파티(3rd party) 라이브러리

· 활용 : 탐지 결과 관리 어플리케이션 기반 라이브러리로 활용

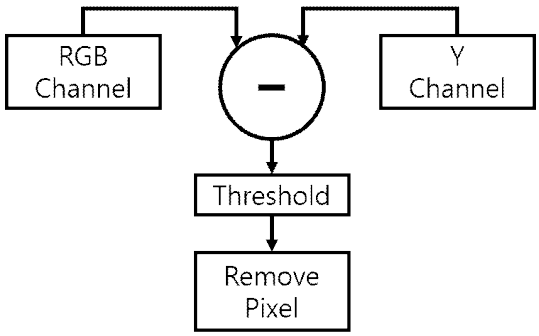
· 연구 및 실험 내용 : 오픈소스 라이브러리 기반 갤러리 어플리케이션 제작 진행, 탐지 이미지 파일 출력

· 결과 : 탐지된 이미지 파일 정상적으로 출력 확인

○ 특징점 기반 포트홀 탐지 관련 연구 분석

- RGB-Y 기반 컬러 오브젝트 제거 전처리 기법

· 도로 표면과 포트홀의 컬러값을 분석한 결과 흑백에보다 근접한 색을 나타낸다는 특징을 발견하여, 이를 기반으로 ROI 이미지 내의 노이즈를 제거하여 포트홀 탐지 성능을 향상

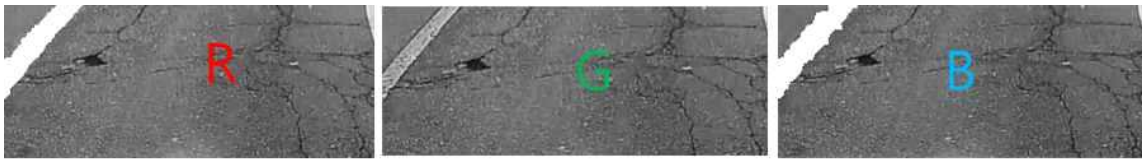


[RGB-Y 차이값 알고리즘]

· Y로 변환한 값을 RGB로 변환한 후 원본 RGB에 뺄셈 연산을 통해 컬러 이미지 부분을 추출하고, 임계 값 이상의 컬러 이미지를 흰색 이미지로 변환하여 포트홀 후보에서 제거

– RGB-Y

· RGB(Red, Green, Blue) 기반 3채널 컬러 영상의 각 채널에서 Y(흑백) 단일 채널 영상을 빼 RGB 중 비 포트홀 영역이 가장 많이 제거되는 영역 판단 연구



[RGB-Y 후 Y에서 돌출 영역 제거]

– 모폴로지 기반 전처리 기법

· 포트홀과 도로 표면의 질감 차이를 극대화하여 포트홀 탐지를 용의하게 하기 위한 전처리 기법 연구

· 3×3, 5×5, 7×7 필터의 모폴로지 침식을 활용한 도로 표면 노이즈 제거 연구

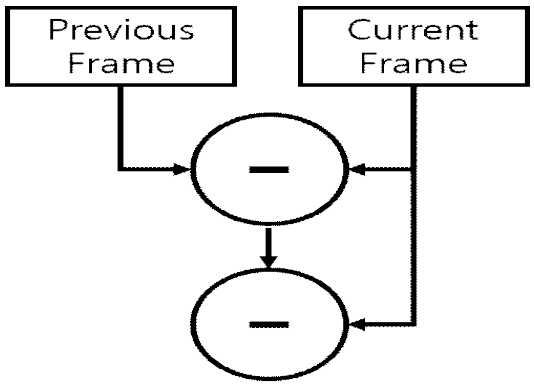
– 동적 이진화 OTSU를 활용한 도로 표면 마킹 탐지

· 도로 표면의 다양한 마킹을 탐지하여 포트홀 오탐지 요인 제거

· 동적 이진화를 통한 도로 표면 제거 방법 연구

– 시간 축 프레임 차 기반 포트홀 탐지 알고리즘

· 도로 표면과 차선에 비해 포트홀이 차량 진행 방향에서의 시간적인 연속성이 존재하지 않는 특성을 이용한 포트홀 탐지 기법을 연구

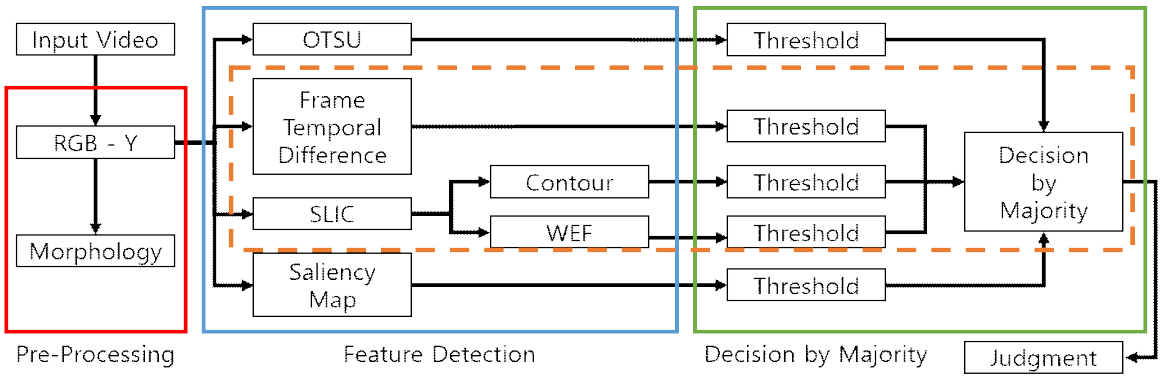


[시간 축 프레임 차 알고리즘]

· 이전 프레임에서 현재 프레임을 제외하는 연산과 이전 프레임에서 현재 프레임을 두 번 제외하는 연산을 통한 포트홀 탐지 기법 연구

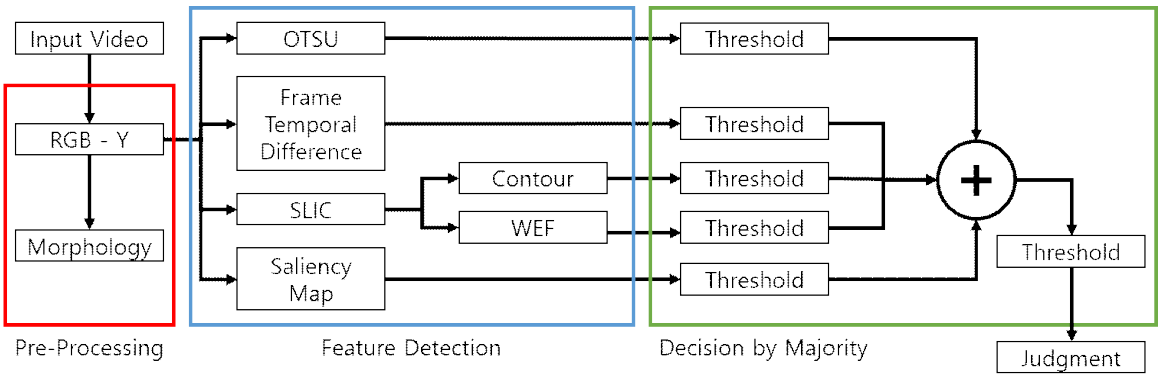
· 도로 차선 및 크랙 등의 노이즈를 효과적으로 제거하고 상대적으로 포트홀을 강조하여 포트홀 탐지 성능을 높이기 위한 방법을 연구

– 다각형 검출 기반 포트홀 탐지 알고리즘
· 슈퍼픽셀(SLIC)과 컨벡스 헐을 이용한 다각형 검출
· 슈퍼픽셀을 통해 유사 픽셀값을 일정 단위로 클러스터링 하고, 그 모양을 단순화시켜 컨벡스 헐을 통한 다각형을 검출하는 방법으로 포트홀을 탐지하는 알고리즘 연구
· 슈퍼픽셀의 특성상 도로 표면은 정사각형의 특징을 가지는 반면, 포트홀은 3각형 또는 5~6각형 이상의 다양한 형태를 나타냄
– 슈퍼픽셀과 WEF를 활용한 포트홀 탐지
· 슈퍼픽셀을 통해 클러스터링한 후 각 클러스터 내부의 질감을 WEF를 통해 수치화하여 포트홀을 탐지하는 알고리즘을 개발
· 후보 영역 내부 최댓값, 최솟값, 화이트 픽셀 분포도, 이미지 히스토그램을 통해 다양한 오브젝트와 구분되는 포트홀 영역의 WEF 특징값 연구
– 다수결 기반 하이브리드 포트홀 탐지 알고리즘
· 각각의 특징점 알고리즘은 포트홀을 탐지하기 위해 수정된 알고리즘을 개발
· 수정된 각각의 특징점 알고리즘을 통해 개별적으로 포트홀을 탐지



[다수결 기반 하이브리드 포트홀 탐지 알고리즘]

· 배경 제거 알고리즘 4가지(MOG, MOG2, GMG, KNN), WEF, OTSU, 디프런셜, 컨벡스 헐, 세일리언시 맵 이용 다수결 알고리즘 제공
– 가중치 기반 하이브리드 포트홀 탐지 알고리즘
· 각각의 특징점 기반의 알고리즘 결과를 수치화한 후, 각각의 값에 가중치를 부여하여 합계를 구한 후, 합계 값이 특정 임계 값을 넘는 경우 포트홀로 인식
· 각각의 특징점 알고리즘은 포트홀을 탐지하기 위해 수정된 알고리즘을 개발



[가중치 기반 하이브리드 포트홀 탐지 알고리즘]

○ 딥러닝 기반 포트홀 탐지 관련 연구 분석
– 도로주행 영상 딥러닝 타당성 연구
· ILSVRC 대회 우승 및 개선된 이미지 분류 모델들의 도로 주행 영상 내 포트홀 및 비 포트홀 탐지 성능 평가
· 이미지 분류 모델이 유사한 특징을 띄는 도로 영상에서 포트홀 탐지 가능성 유무 확인

· 라이브러리 : 텐서플로우 1.1
· 이미지 분류 모델 : Inception_v3, Inception_v4, Inception_ResNet_v2, ResNet_v1_152, ResNet_v2_152
· 지정된 ROI 영역만 Y(흑백) 이미지로 변환 (이미지 크기 800×200), 총 3,030장, 비 포트홀 2,879장, 포트홀 151장으로 분리, 입력 영상 이미지를 포트홀과 비 포트홀 클래스로 분류, 클래스별 이미지 장수의 5% 검증용으로 사용
· 학습 방법 : 도로주행 영상 이미지만을 사용한 학습. 1회 학습 (Epoch)에 5장씩 사용(GPU 메모리 한계로 인한 최대 허용량), 모델별 각 6,000회 학습

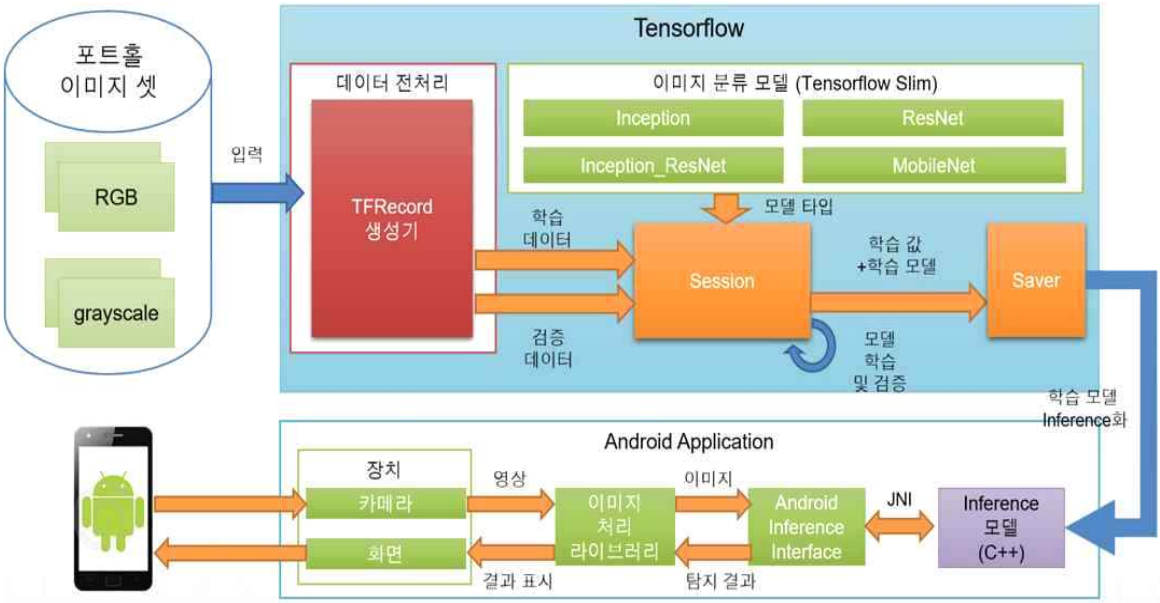
– 사전 학습 모델을 사용한 포트홀 탐지
· 학습 시 사전 학습 데이터를 사용한 성능 향상 가능성 확인을 위한 연구
· 텐서플로우 1.1
· 분류 모델 : Inception_v3, Inception_v4, Inception_ResNet_v2. ResNet_v1_152, ResNet_v2_152
· 사전 학습 모델 : ILSVRC 2012 데이터 학습 모델 (10만 장 이상 학습, 1,000개의 클래스로 분류)
· 학습 방법 : 모델의 가중치만 변경, 정규화 및 학습 속도는 사전 학습 모델 데이터 사용, 이미지 데이터 셋 y_v0.1 사용, 1회 학습 (Epoch)에 5장씩 사용(GPU 메모리 한계로 인한 최대 허용량), 모델별 각 20,000회 학습

– 포트홀 탐지 성능 향상을 위한 이미지 데이터 셋
· 이미지 재분류 및 데이터 셋 구성을 통한 모델의 95% 이상 성능 탐지율 획득 방안 연구
· 라이브러리 : 텐서플로우 1.2
· 이미지 분류 모델 : Inception_v4, Inception_ResNet_v2. ResNet_v1_152, MobileNet_v1
· 지정된 ROI 영역만 Y(흑백) 이미지로 변환 (이미지 크기 800×200), 총 3,186장, 비 포트홀 3,030장, 포트홀 159장으로 분리, 입력 영상 이미지를 포트홀과 비 포트홀 클래스로 분류, 클래스별 이미지의 5% 검증용으로 사용
· 학습 방법 : 1회 학습 (Epoch)에 5장씩 사용 (GPU 메모리 한계로 인한 최대 허용량), 모델별 각 5,000회 학습

– 이미지 데이터 색상 채널에 따른 모델 학습 및 성능
· 학습 모델이 이미지 내 포트홀 판단 시 채널 영향 확인
· 라이브러리 : 텐서플로우 1.2
· 이미지 분류 모델 : Inception_v4, Inception_ResNet_v2. ResNet_v1_152, MobileNet_v1
· 그레이 스케일 이미지 데이터 셋과 컬러 이미지 데이터 셋을 재구성
· 학습 방법 : 1회 학습 (Epoch)에 5장씩 사용 (GPU 메모리 한계로 인한 최대 허용량), 모델별 각 5,000회 학습 이미지 재분류 및 데이터 셋 구성을 통한 모델의 95% 이상 성능 탐지율 획득 방안 연구

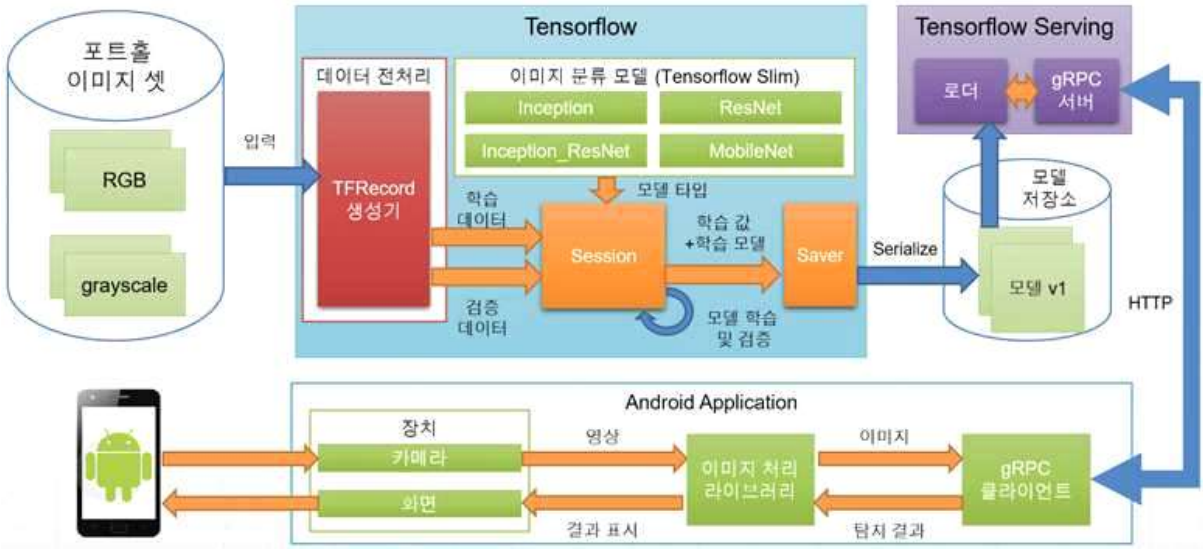
– 포트홀 오탐지 사례 분석
· 학습된 이미지 분류 모델이 검증용 이미지에서 오판 형태 분석
· 학습된 모델의 로그 분석을 통한 사례 분석
· 분석 대상 모델 : Inception_v4, Inception_ResNet_v2, ResNet_v2_152

– Stand-Alone 포트홀 서비스 모델
· 안드로이드 스마트폰 내에서 독립적으로 동작하는 딥러닝 기반 포트홀 탐지 알고리즘 서비스 모델 연구
· 학습 결과를 인퍼런스 모델로 추출한 후, 이를 안드로이드 스마트폰 환경에서 동작 가능한 형태로 이식 방안 연구



[Stand-Alone 서비스 모델 아키텍처]

- 서버/클라이언트 포트홀 서비스 모델
  · 딥러닝 학습 서버에 학습 결과를 두고, 텐서플로우 서빙을 통해 클라이언트의 포트홀 탐지 요청을 처리하는 서비스 모델 연구
  · 지속적인 기계학습이 가능하여 진화하는 알고리즘 가능성에 관한 연구

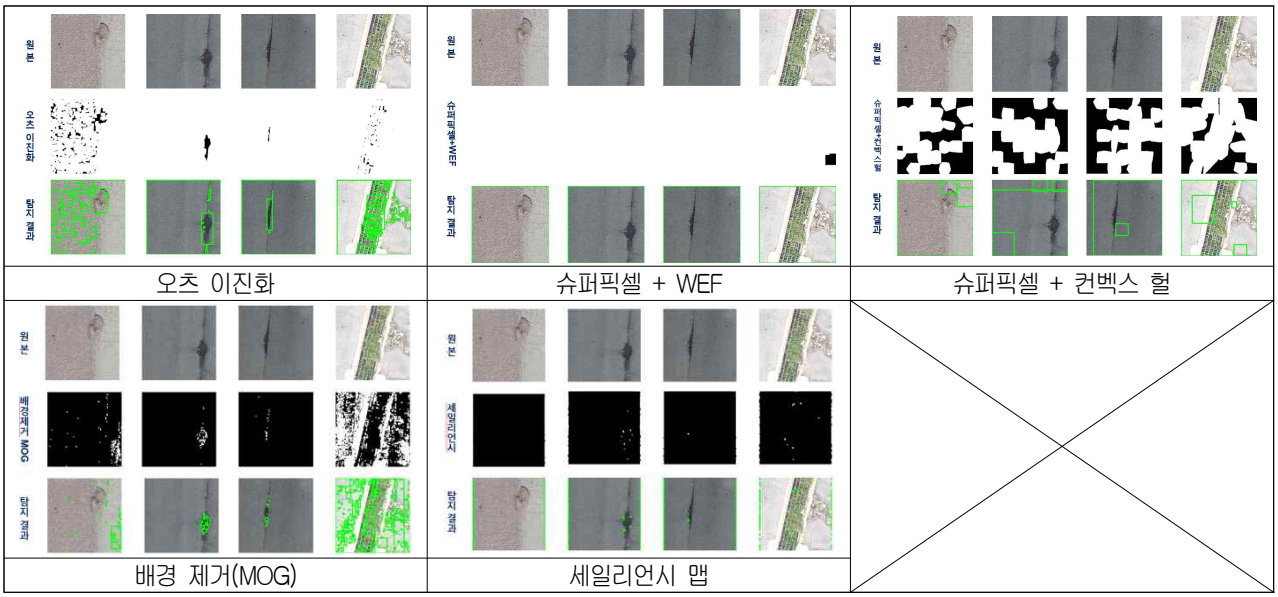


[서버/클라이언트 서비스 모델 아키텍처]

○ 포트홀 탐지 S/W 관련 연구 분석
- 안드로이드 카메라 API 최적화
  · 안드로이드 Native 카메라 API 분석
  · 오픈소스 기반의 카메라 API 분석
  · 다양한 카메라 API 분석을 통한 최적 성능의 통합 카메라 API 구축 연구
- 포트홀 탐지 어플리케이션 UI/UX
  · 안드로이드 스마트폰 어플리케이션의 사용자 친화적 UI/UX 연구
  · 카메라 동작 및 환경 설정, ROI 영역 선택 UI 개발
  · 탐지 알고리즘 사용자 직접 선택 인터페이스
  · 탐지 결과 이미지 또는 영상 저장 인터페이스
- 탐지 결과 관리 어플리케이션
  · 탐지된 포트홀 이미지를 시간별, 장소별, 종류별로 확인할 수 있는 어플리케이션 개발

· 각 포트홀 이미지와 시간, 위치를 한눈에 확인하여 도로유지보수에 활용 가능한 형태로 개발

○ 포트홀 특징 도출을 위한 컴퓨터 비전 알고리즘 실험
- 오츠 이진화, 슈퍼픽셀 + 컨벡스 헐, 슈퍼픽셀 + Wavelet Energy Field, 세일리언시 맵, 배경제거(MOG) 등을 사용한 특성 분석
- 포트홀, 패치, 크랙의 명확한 특징 추출 도출 불가
- 각 알고리즘별 결과

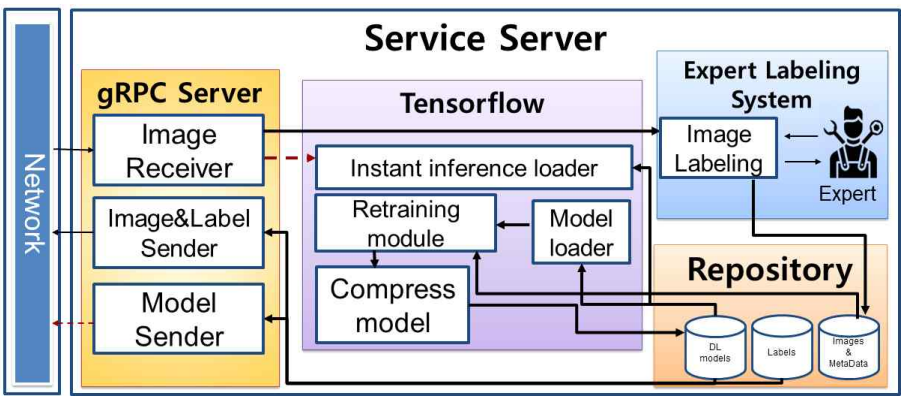


오츠 이진화 | 슈퍼픽셀 + WEF | 슈퍼픽셀 + 컨벡스 헐

배경 제거(MOG) | 세일리언시 맵

○ 시스템 구조 설계
- 컴퓨터 비전 기반 항공 영상 필터링 모듈
- 순환 학습을 위한 기계학습 모듈
- 재학습 추론 모델 버전 관리 모듈
- gRPC를 사용한 탐지 영상 및 결과, 모델 송수신 모듈

○ 인공신경망 기반 포트홀 학습 알고리즘 테스트
- 원본 및 정합 이미지의 타일 데이터 셋 생성 및 적용
- 원본 및 정합 이미지의 타일 기반 딥러닝 학습
- 데이터 셋 구성, 2개 분류: 포트홀(빨간색), 도로 영역(파랑색)
- 총 12,400개 타일 데이터 셋 (포트홀 : 328개, 도로 영역 : 12,072개)
- 항공 원본 영상을 활용한 포트홀 탐지 실험 진행 및 결과



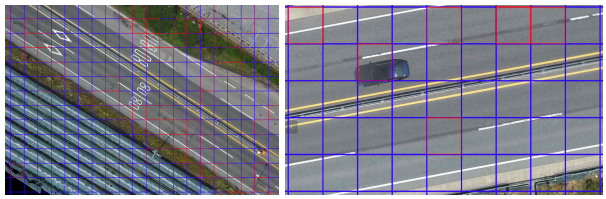
[시스템 구조 아키텍처]



오탐 예시



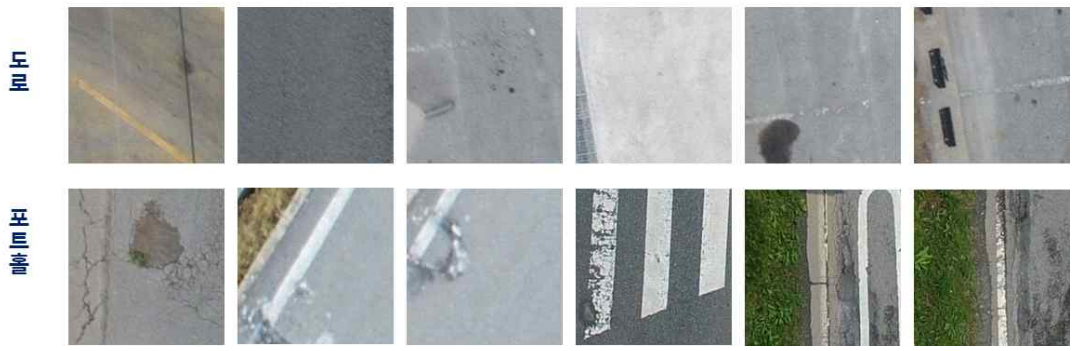
미탐 예시



Inception v3 추론 결과

○ 포트홀 평가/검증 계획 및 결과
- 검지율 : 탐지 포트홀 수 / 전체 포트홀 수
- 오검지율 : 비 포트홀 탐지 횟수 / 포트홀 탐지 횟수
- 탐지 에러율 : 비 포트홀 탐지 횟수 / 전체 탐지 실행 횟수
- MobileNet : 검지율 0% (0/25)
- InceptionNet : 검지율92%(23/25) 오검지율29.9(1001/345) 탐지에러율0.05%(2344/44,343)
- 타일 기반 MobileNet 딥러닝 모델의 부적함으로 인해 낮은 분류 정확도
- 타일 기반 Inception v3 딥러닝 모델의 낮은 도로 영역 분류 정확도

○ 정합 영상기반 포트홀 탐지
- 정합 이미지 내 합성된 영역 내 영상 변형 발생 → 오탐 발생 요인
- 원본 촬영 이미지 기반 도로 파손 영역(포트홀)탐지



[도로 이미지와 포트홀 이미지 예시]

(2) 포트홀 탐지를 위한 드론시스템 기술 개발

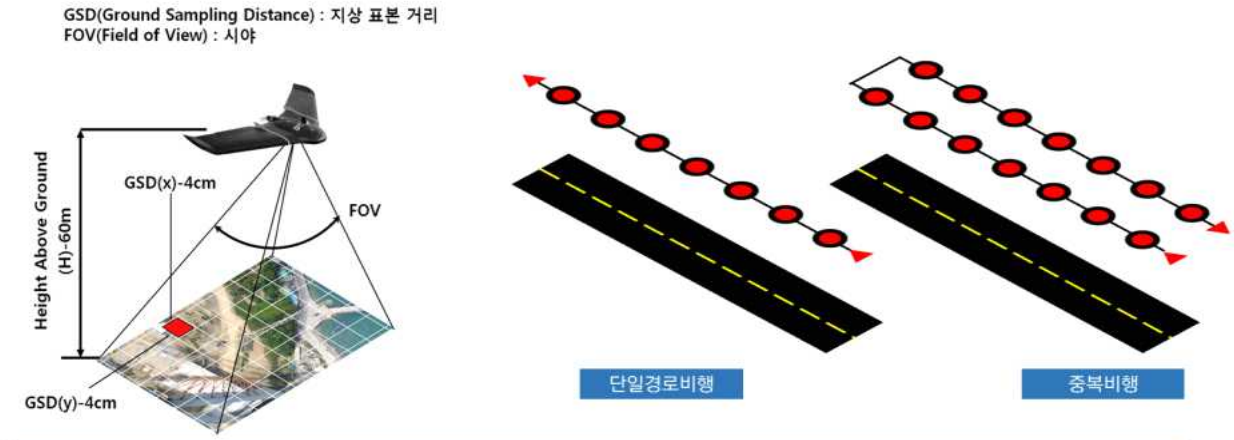
○ AI학습용 포트홀 상시 데이터 수집
- 폭우, 폭설 이후 포트홀 발생구간 드론촬영 수행
- 총 26회의 비행으로 48개의 포트홀 데이터 수집
- 평균 작업시간 80분 소요(외업 20분, 내업 60분)



○ 고정밀 GPS 탑재 드론 및 지상기준점을 활용한 포트홀 위치 정확도 향상
- 고정익드론 : RTK/PPK 기능 탑재 드론을 활용한 위치정확도 향상(RMSE 0.1m)
- 회전익드론 : 지상기준점 사용으로 위치정확도 향상(RMSE 0.6m)



○ 포트홀 탐지를 위한 최적의 드론 비행방법 도출
- 테스트베드 활용 해상도, 중복도, 경로 결정
- 드론촬영의 신속성 및 안정성, 영상처리의 신속성 및 정확성 검토결과 해상도는 2cm 미만, 중복도는 70%, 2라인 이상의 경로비행을 최적 드론 비행방법으로 결정

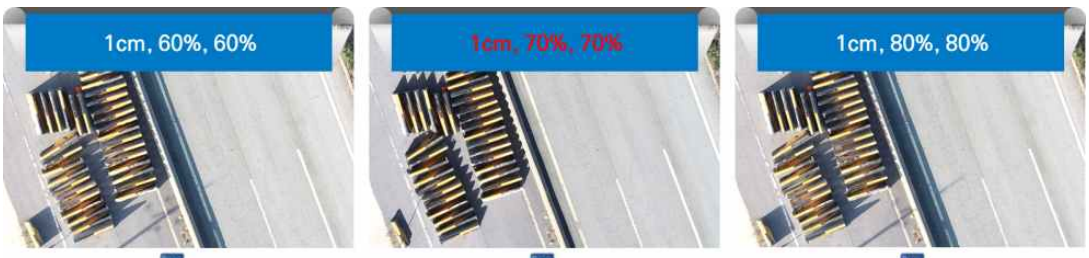


▪ 포트홀 탐지 최소 품질 보증을 위한 비행방법(경로, 고도, 중복도) 도출
▪ 정사영상 기준 포트홀 탐지가능한 최적의 해상도 도출

- 해상도 결정(2㎝ 이내)
· 고정익 및 회전익 드론에 대해 1cm, 2cm, 3cm, 4cm에 대한 해상도별 촬영수행
· 해상도 2cm 이상의 경우 깊이, 음영에 따라 포트홀의 구분이 어려움
· 포트홀 육안판별 최적해상도 2cm 미만으로 선정



- 중복도 결정(70%, 70%)
· 고정익 및 회전익 드론에 대해 종횡 모두 60%, 70%, 80% 촬영 수행
· 영상품질은 중복도가 높을수록 좋아지나 정확도는 RMSE 5cm 이내의 변화로 중복도에 따른 영향이 크게 없음
· 영상처리시간과 품질을 고려 70% 이상의 중복도가 적합



- 경로 결정(2라인 이상)

· 고정익 및 회전익 드론에 대해 해상도별 1경로, 2경로 비행 수행

· 고정익드론의 경우 낮은 해상도일 때 1경로 비행에서 정사영상 제작이 누락됨(고도가 낮고, 해상도가 높을수록 중복도 누락이 발생)

· 정사영상의 품질적인 부분은 1경로비행에서 노이즈, 블러링 현상이 발생한 반면, 2경로 비행에서는 노이즈, 블러링 현상이 없어짐

· 정확도 측면에서 고정익의 경우 1경로 RMSE 0.38m, 2경로 RMSE 0.11m, 회전익 1경로 RMSE 0.72m, 2경로 RMSE 0.60m로 2경로 비행이 월등히 좋음



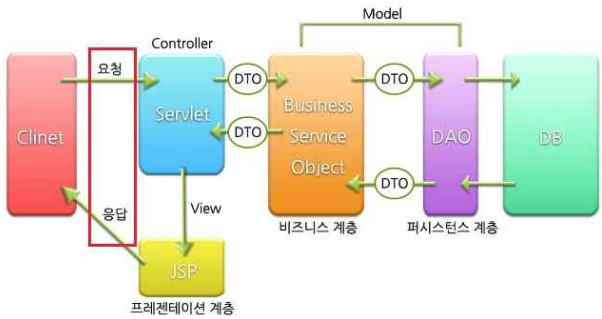
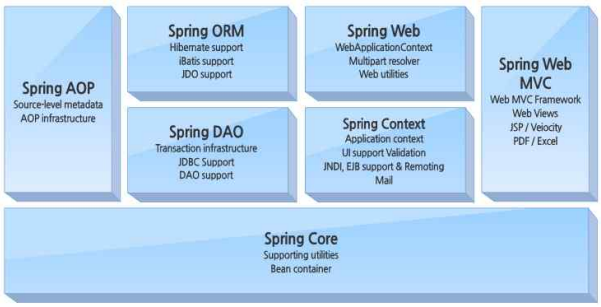
(3) GIS 맵기반 드론 측정 포트홀 데이터 가시화 기술 개발

○ 서비스 서버 구축

분류	프레임워크 & 라이브러리	비고
OS	CentOS	가시화 시스템 구축 OS
Framework	Spring Framework, Mybatis, Jquery	가시화 시스템 구성 Framework
WAS	Tomcat	Web application server
Database	Oracle xe 11g	

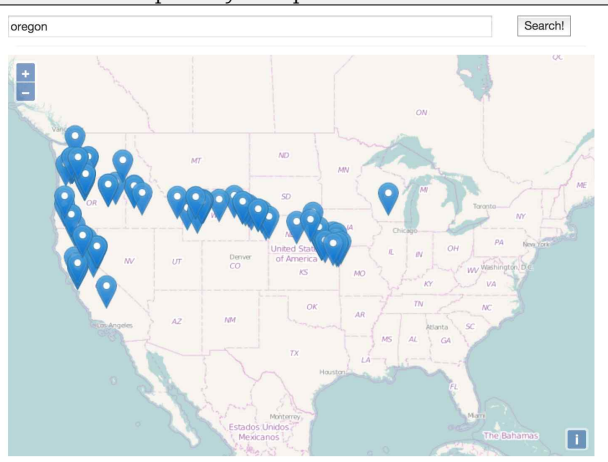
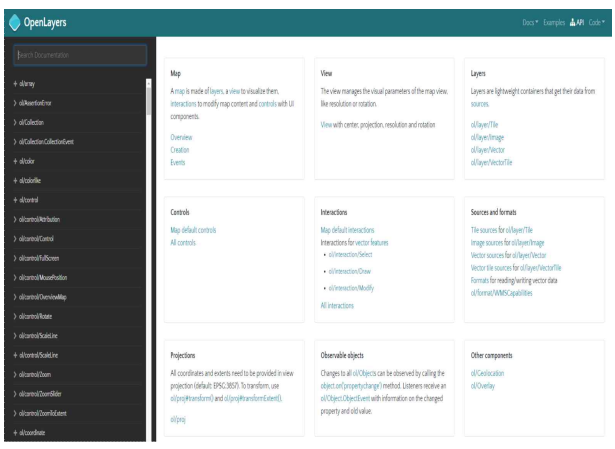
○ 웹 서비스 프레임워크

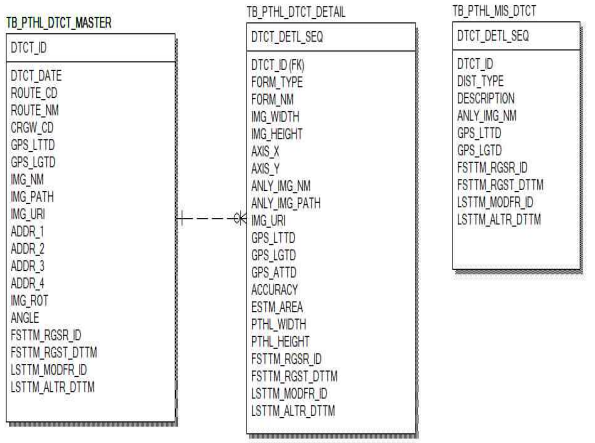
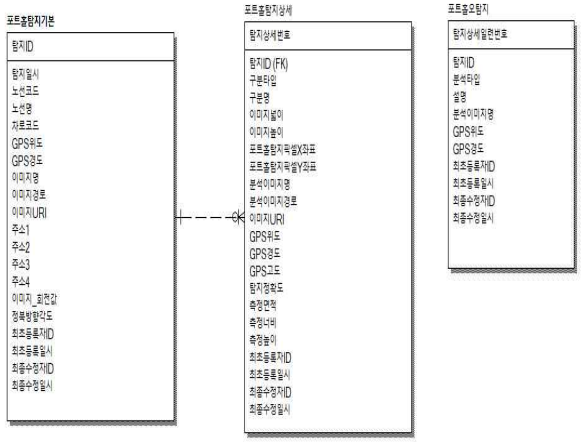
- 자바 플랫폼을 위한 오픈소스 어플리케이션 프레임워크인 Spring Framework를 메인 서비스 개발에 사용함. 프레임워크는 프로그램의 기본적인 흐름과 구조가 정해져있기 때문에 개발 팀원간 부분적인 코드만 추가하여 하나의 일관성있는 어플리케이션 구축이 가능하며 소프트웨어 개발의 생산성 측면에서도 막대한 이점을 가짐

Sring Framework MVC Model2	Spring Framework 구조
	

○ 지도 API활용 기술 연구

- 웹 브라우저에서의 지도레이어는 vworld 레이어와 google map 두 지도 레이어를 비교하였을 때 vworld레이어가 google map레이어보다 대한민국 기준으로 지도정보 정확도 측면에서 우수한 것을 확인하였으며 vworld를 선택하였음.
- 웹 브라우저의 지도레이어에 포트홀의 탑지정보를 표시하기 위하여 오픈소스 자바스크립트 라이브러리인 openlayer를 활용함. vworld api를 화용하여 openlayer에 연동하여 가시화 시스템의 지도API의 기본 모듈을 개발하였음

openlayers point marker	openlayers 공식 API문서
	

물리ERD	논리ERD
	

○ Database 설계

- Database는 조사한 결과 Oracle 데이터베이스가 타 오픈소스 데이터베이스와 비교하였을 경우 성능과 안정성 측면에서 이점을 가진 것으로 판단되어 사용을 결정함
- 포트홀 탐지 데이터에 대한 Database ER-Diagram을 Logical Data Model과 Physical DataModel을 구분하여 작성하였으며 각 Entity 항목별로 DB Table을 작성

○ Spring Security 웹보안 프레임워크

- Spring 기반의 애플리케이션의 보안(인증과 권한, 인가 등)을 구현하기 위하여 Spring Security 보안 프레임워크를 적용함. 사용자의 시스템 권한을 인증(Authentication)하기 위한 Filter와 인증된 사용자가 요청한 자원에 접근 가능한지를 결정하는 인가(Authorization) 모듈을 구현하여 포트홀 시스템의 주요 보안자료(포트홀영상, 분석데이터 등)의 접근을 제한하는 시스템 모듈을 개발함

○ 포트홀 가시화시스템 SW API설계

- 포트홀 가시화 시스템의 Backend API 설계를 진행하였음. API의 URL Pattern을 정의하여 일관성있는 서버 데이터 요청구조를 가지도록 함. 데이터를 처리하는 방식에 따라 유의미한 요청형식을 부여하여 서버 어플리케이션 자원의 효율적인 배분을 유도하도록 구현

API URL	TYPE	Return
/pthl/main	GET	포트홀 메인페이지
/pthl/options	GET	포트홀 검색옵션 데이터 조회
/pthl/pos-list	GET	포트홀 드론촬영 위치정보 데이터
/pthl/list	GET	포트홀 상세 리스트 조회
/pthl/reg/misdetect	POST	포트홀 오탐지 등록
/pthl/reg/misdetect-all	POST	오탐지 포트홀 배치등록
/pthl/sta	GET	포트홀 통계페이지
/pthl/sta/summary	GET	포트홀 일자별 탐지건수 현황
/pthl/sta/cnt-addr3	GET	포트홀 탐지건수현황 (읍면동, 위험도별),일별 위험도 분포 조회

다. 드론을 활용한 포장 노후화 상태(PCI) 측정 기술 개발

(1) 드론 수집 초분광 데이터 분석 기반 PCI 정보 산출 기술 개발

○ 하이퍼스펙트럴 센서 및 분석 기술조사

- 드론 탑재용 초분광 카메라 조사
  · 파장범위 400-1000, 밴드폭 10nm 이하
  · 파장범위 1000-2500nm는 일부 밴드만으로 구성하여 멀티스펙트럴 카메라 제작 적용 가능

[드론 탑재용 초분광 카메라]

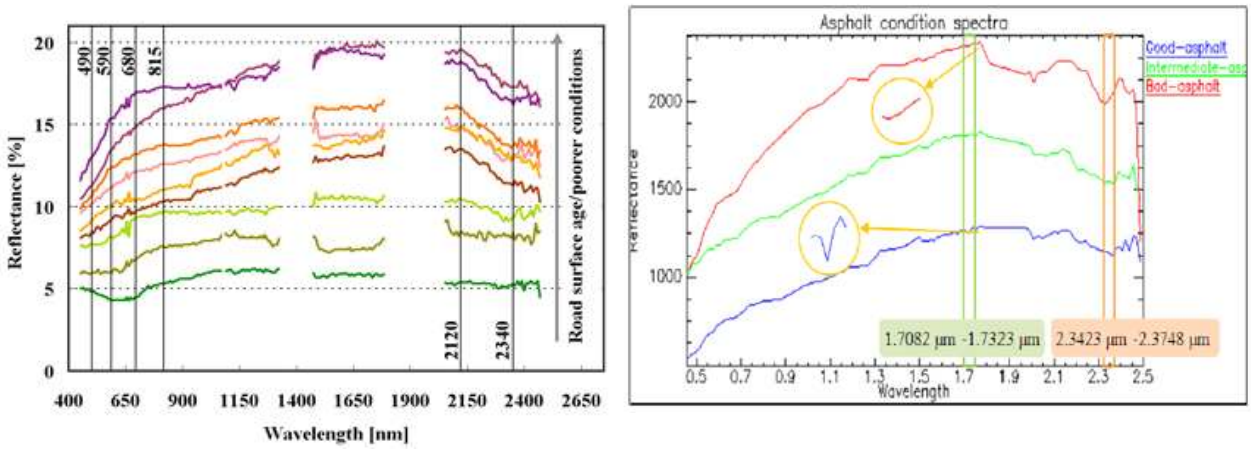
센서	Nano-Hyperspec	Micro-CASI1920	FX-10	SOC710-GX	Shark
제조사	Headwall (미국)	ITRES (캐나다)	Specim (핀란드)	Surface Optics (미국)	Corning (미국)
크기(mm)/ 무게(kg)	76 x 76 x 120 / 0.52kg	102 x 190 x 178 / 1.5	150 x 71 x 85 / 1.4	103 x 200 x 200 / 1.25	? / 0.45
밴드 수	270	288	220	120	180
밴드 폭(nm)	6	2.1		4.2	3.3
파장대(nm)	400-1000	400-1000	400-1000	400-1000	400-1000
센서사진					

- 초분광 파장 분석 모델 조사

[초분광 파장 분석 모델(1000 nm 이하)]

종류	모델식
ARI1(Anthocyanin Reflectance 1)	$\mathrm{ARI1}=\left(\frac{1}{\rho 550}\right)-\left(\frac{1}{\rho 705}\right)$
ARI2(Anthocyanin Reflectance 2)	$\mathrm{ARI2}=\rho 802\left\{\left(\frac{1}{\rho 550}\right)-\left(\frac{1}{\rho 705}\right)\right\}$
NDVI(Normalized Delta)	$\mathrm{NDVI}=\left(\frac{\rho 802-\rho 638}{\rho 802+\rho 638}\right)$
ISAVI(Optimized Soil Adjusted)	$\mathrm{OSAVI}=1.16\times\left(\frac{\rho 802-\rho 675}{\rho 802+\rho 675+0.16}\right)$
PRI(Photochemical Reflectance)	$\mathrm{PRI}=\left(\frac{\rho 531-\rho 570}{\rho 570+\rho 531}\right)$
PSRI(Plant Senescence Reflectance)	$\mathrm{PSRI}=\frac{(\rho 678-\rho 502)}{\rho 754}$
VIS1(Visible Imaging Spectrometer 1)	$\mathrm{VIS1}=\left(\frac{\rho 740-\rho 460}{\rho 740+\rho 460}\right)\mathrm{VISR1}=\frac{\rho 460}{\rho 740}$
VIS2(Visible Imaging Spectrometer 2)	$\mathrm{VIS2}=\left(\frac{\rho 830-\rho 490}{\rho 830+\rho 490}\right)\mathrm{VISR2}=\frac{\rho 490}{\rho 830}$
REI(Red Edge)	$\mathrm{REI}=\frac{\rho 705}{\rho 754}$
WBI(Water Band)	$\mathrm{WBI}=\frac{\rho 899}{\rho 967}$

- 도로 상태에 따른 초분광 파장 분석 방법 조사



○ 포장 노후화 유형별 초분광 실험

- 아스팔트 공시체, 아스팔트 등 실내시편 제작
  (공시체 형태, 표면 텍스쳐, 아스팔트 노화 상태 변화 등)



아스팔트
(직경 15cm)

아스팔트 공시체
(30×30cm)

아스팔트 공시체
(직경 15cm)

아스팔트 공시체
(컷팅, 직경 15cm)

[초분광 시험용 실내제작 시료]

- 현장 코어 채취



균열발생 현장 및 코어 채취



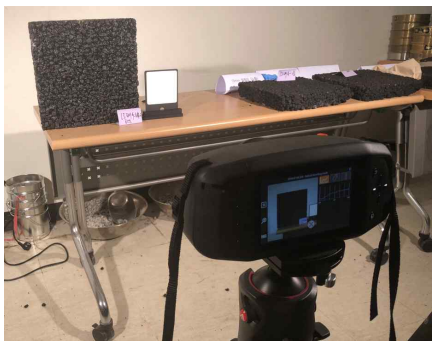
코어(균열 실링, 직경15cm)

코어(균열 2, 직경15cm)

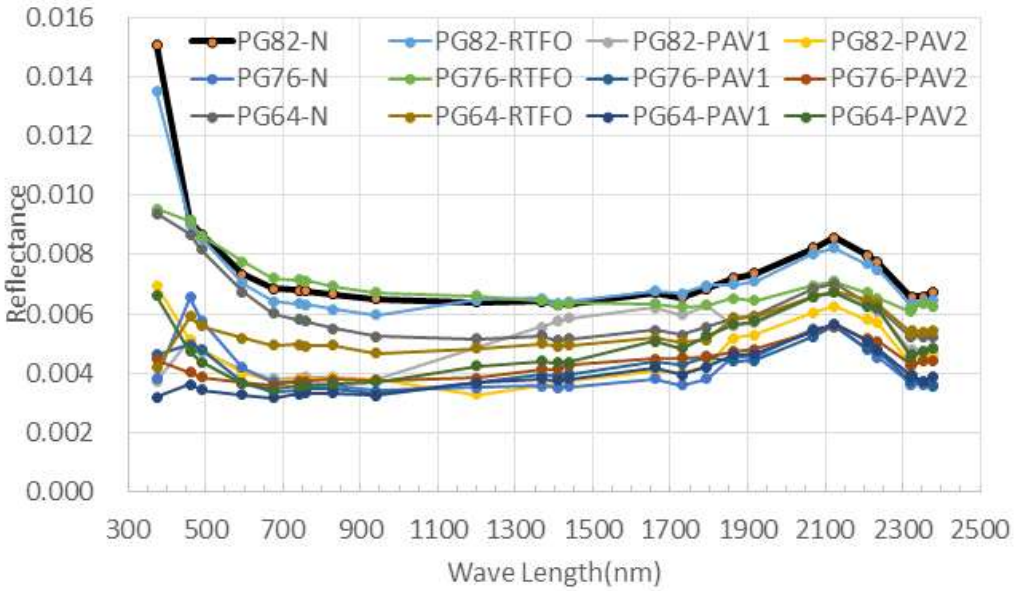
코어(균열 3, 직경15cm)

[초분광 시험용 현장 코어 시료]

- 하이퍼 스펙트럴 카메라, 분광계 등을 이용한 실험


- 아스팔트 노화에 따른 초분광 실험 결과(아스팔트 시료)

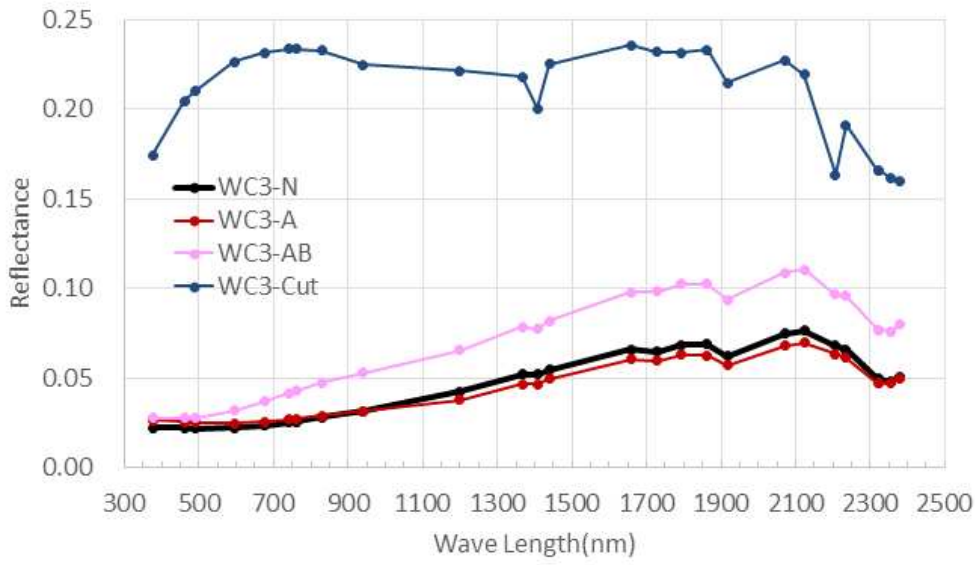


[아스팔트 시료의 노화에 따른 초분광 반사율]

(PG82 및 PG76: SBS 개질 아스팔트, PG64: 일반 아스팔트,
N: 신규, RTFO: 단기노화, PAV1: 장기노화, PAV2: 장기노화 2회)

- 포장 노화 및 표면 상태에 따른 초분광 실험 결과

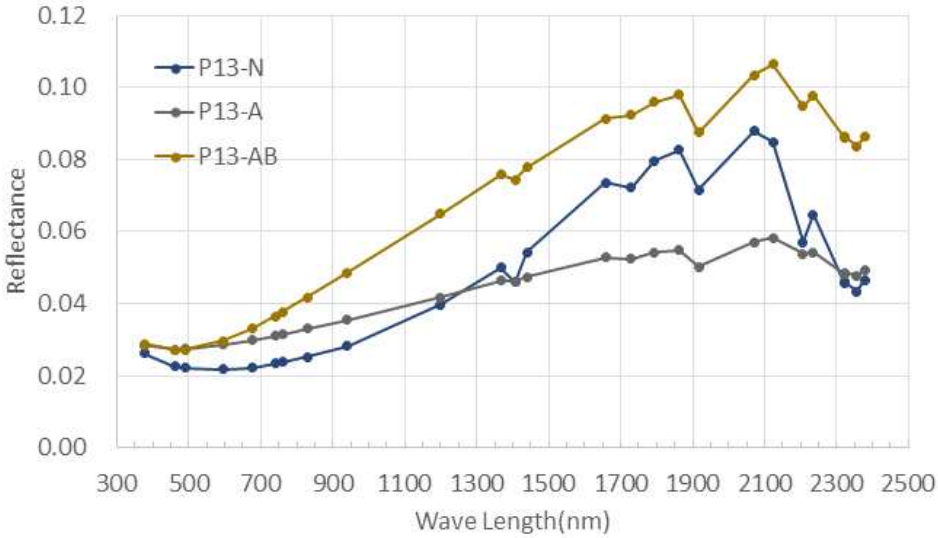
· 일반 밀립도 아스팔트 포장



[밀립도 아스팔트 콘크리트의 노화 및 표면에 따른 초분광 반사율]

(WC3: 밀립도포장, N: 신규, A:노화, AB:노화+표면탈색. Cut:표면커팅)

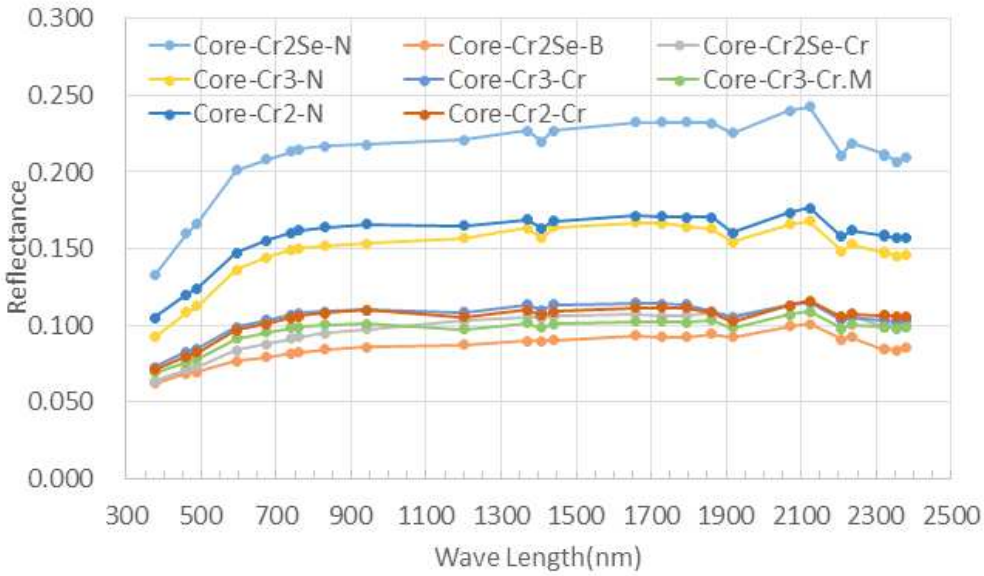
· 배수성 아스팔트 포장



[저소음/배수성 아스팔트 콘크리트의 노화 및 표면에 따른 초분광 반사율]

(P13: 배수성포장, N: 신규, A:노화, AB:노화+표면탈색)

· 포장 현장 코어 시료



[포장 현장 코어 초분광 반사율]

(Core: 현장채취코어, Cr2:2조각균열시편, Cr3: 2조각균열시편, Se:균열실링처리시편,
N:양호부분, B:실링시편 검은밴드부, Cr: 균열, Cr.M: 큰균열)

○ 포장 노후화 유형별 초분광 실험 데이터 분석 기술 개발

- 초분광 실험 결과의 특징 분석

· 노화시 1000nm이상 파장에서 반사율이 낮아지며, 표면이 탈색된 경우에는 반사율이 높아짐

· 실내 제작 공시체 시편과 현장 코어는 300~1000 nm에서 큰 차이 발생

· 실내 시편 표면을 커팅한 경우 현장 코어 시편과 유사

· 현장 코어의 양호한 구간과 균열부를 비교한 결과, 균열부의 반사율이 전반적으로 낮아지고, 1407 nm 파장대 반사율에서 유의미한 차이 발생

- Visual 파장대의 분광지수 분석 (아스팔트 노화 영향) 및 평가

· 초분광 파장 분석 모델에 따른 경향 분석결과 장기노화된 시료의 경우 VISR2의 값이 전반적으로 낮아지는 경향을 보임

· VISR2 이외에 기존의 모델 들은 경향성을 확인하기 어려웠음

· 다양한 파장대별 경향을 평가한 결과 688 nm, 836nm 파장을 이용하여 계산한 PVISI1(Pavement Visible Imaging Spectrometer Index 1, $\rho 688/\rho 836$)는 VISR2 보다 경향성이 뚜렷하여 이후 추가 분석 결정

[아스팔트 노화에 따른 초분광 지수 평가 결과]

(PG82 및 PG76: SBS 개질 아스팔트, PG64: 일반 아스팔트, N: 신규, RTFO: 단기노화, PAV1: 장기노화, PAV2: 장기노화 2회)

구분	VIS1	VIS2	VISR1	VISR2	PVISI1
PG82-N	0.148	0.1319	1.35	1.3	1.0252
PG82-RTFO	0.1731	0.1645	1.42	1.39	1.039
PG82-PAV1	0.1423	0.0989	1.33	1.22	0.9816
PG82-PAV2	0.1541	0.116	1.36	1.26	0.9788
PG76-N	0.2821	0.2379	1.79	1.62	1.0333
PG76-RTFO	0.1227	0.1093	1.28	1.25	1.0363
PG76-PAV1	0.1863	0.1633	1.46	1.39	0.9817
PG76-PAV2	0.0392	0.0105	1.08	1.02	0.9561
PG64-N	0.1994	0.1943	1.5	1.48	1.0891
PG64-RTFO	0.0864	0.0622	1.19	1.13	1.0005
PG64-PAV1	0.0489	0.0185	1.1	1.04	0.9588
PG64-PAV2	0.1385	0.092	1.32	1.2	0.95

- Visual 파장대의 분광지수 분석 (실내제작 공시체) 및 평가

· 초분광 파장 분석 모델에 따른 경향 분석결과 노화되거나 커팅된 시료는 대부분의 지수가 높아짐

· 다만 표면이 탈색될 경우 낮아지는 것으로 나타남

[공시체의 노화에 따른 초분광 지수 평가 결과]

(WC3: 밀립도포장, N: 신규, A:노화, AB:노화+표면탈색. Cut:표면커팅)

구분	VISI1	VISI2	VISR1	VISR2	PVISI1
WC3-N	0.0599	0.1213	0.89	0.78	0.8414
WC3-A	0.0222	0.0736	0.96	0.86	0.8813
WC3-AB	0.202	0.2634	0.66	0.58	0.7874
WC3-Cut	0.066	0.0507	0.88	0.9	0.9995
P13-N	0.0153	0.0638	0.97	0.88	0.8839
P13-A	0.0653	0.0932	0.88	0.83	0.907
P13-AB	0.1473	0.2121	0.74	0.65	0.7979

- Visual 파장대의 분광지수 분석 (현장 채취 코어) 및 평가

· 초분광 파장 분석 모델에 따른 경향 분석결과 양호한 부분과 비교하여 균열이 있으면 지수값이 낮아지는 경향임

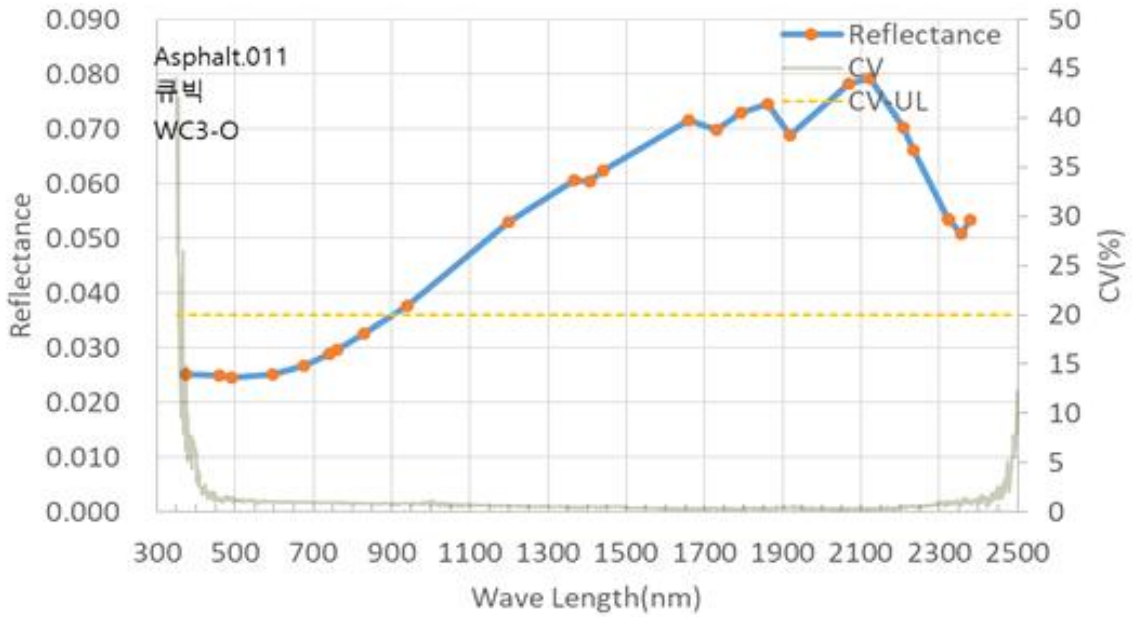
[포장 균열에 따른 초분광 실험 결과]

(Core: 현장채취코어, Cr2:2조각균열시편, Cr3: 2조각균열시편, Se:균열실링처리시편,
N:양호부분, B:실링시편 검은밴드부, Cr: 균열, Cr.M: 큰균열)

구분2	VIS1	VIS2	VISR1	VISR2	PVISI1
Core-Cr2Se-N	0.144	0.1322	0.75	0.77	0.9654
Core-Cr2Se-B	0.0873	0.0932	0.84	0.83	0.9472
Core-Cr2Se-Cr	0.1245	0.1283	0.78	0.77	0.9326
Core-Cr3-N	0.1589	0.1495	0.73	0.74	0.9581
Core-Cr3-Cr	0.1287	0.1259	0.77	0.78	0.9523
Core-Cr3-Cr.M	0.1288	0.1244	0.77	0.78	0.9549
Core-Cr2-N	0.1448	0.1394	0.75	0.76	0.9542
Core-Cr2-Cr	0.1361	0.1349	0.76	0.76	0.9396

○ 하이퍼 스펙트럼 데이터 품질관리 기술 개발

- 변동계수(Coefficient of Variation)를 이용하여 다중 측정된 스펙트럼 데이터의 품질을 추정한 결과 밴드별 문제점을 평가할 수 있었음
- 변동계수가 20% 이상일 경우 해당 밴드를 제외하고 분석

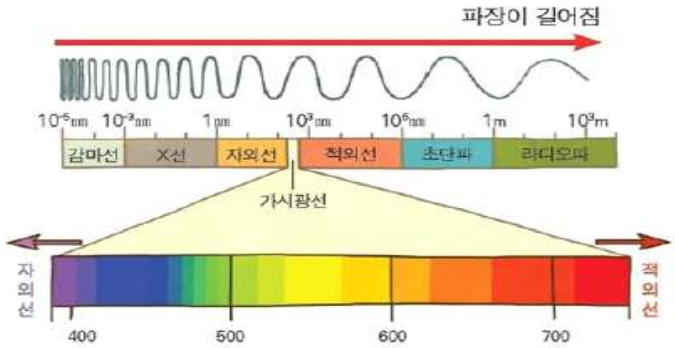


[변동계수를 이용한 하이퍼 스펙트럼 데이터 평가]

(2) 포장 노후화 상태(PCI) 측정을 위한 드론 드론시스템 기술 개발

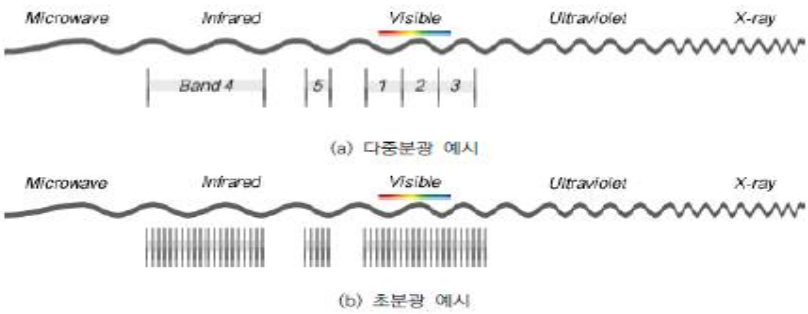
○ 초분광영상 정의

- 태양광선은 크게 가시광선, 자외선, 적외선 파장영역으로 구분할 수 있으며, 이를 구분하는 단위는 나노미터(nm) 혹은 마이크로미터(㎛)를 사용함
- 자외선 파장영역은 200 ~ 400nm이하이고, 가시광선은 400 ~ 700nm, 적외선은 700nm이상을 말하며, 적외선은 다시 근적외선(700 ~ 1,100nm)과 중적외선 파장영역(1,100 ~ 2,500nm)으로 구분함



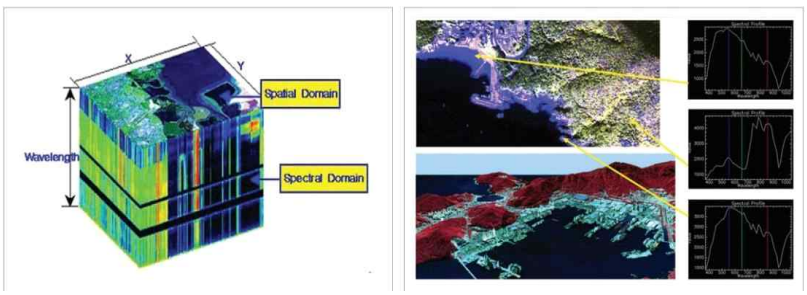
[빛의 파장영역(출처: 네이버 지식백과)]

- 이러한 파장영역을 몇 개로 구분하여 영상을 촬영하느냐에 따라서 다중분광영상(multi-spectral image), 초분광영상(hyperspectral image)으로 구분되는데 이러한 특성을 분광해상도라고 함
- 일반적으로 다중분광영상은 파장영역 10개 이하의 불연속적인 구간으로 나우어 촬영되는 영상을 의미하고, 초분광영상은 파장영역을 20nm이하의 좁은 폭으로 수 백개로 구분하여 연속적으로 촬영되는 영상을 의미함



[다준분광과 초분광의 차이(출처: https://gisgeography.com)]

- 초분광영상은 빛의 파장 대역을 다수의 밴드로 나누어 대상체가 가지는 고유의 광학적 성질 및 반사 특징을 분석할 수 있어 토지피복, 식생 등 대상물체 구별에 주로 활용됨



[초분광 이미지 구성 및 분광데이터]

- 지질, 식생, 암석, 물 등 복잡하게 구성되어 있는 지표물의 생물·화학적 특성에 따른 정확한 정보를 추출하기 위해서는 지표물의 완전한 분광정보가 요구되는데 이를 취득하기 위해 초분

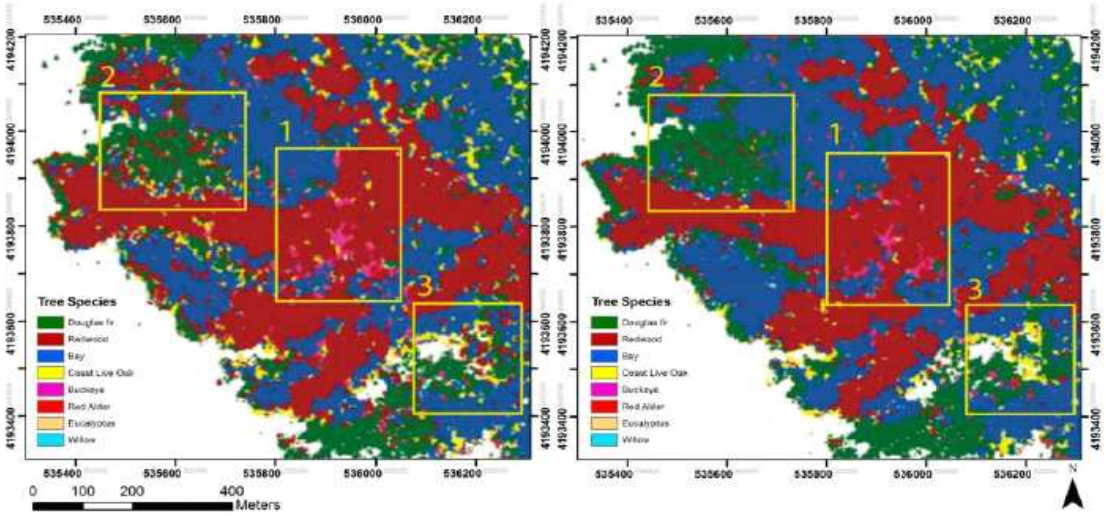
광영상 기술의 개발이 시작됨

- 초분광영상은 원래 'imaging spectroscopy'란 용어로 사용되다가 최근 'hyperspectral image,' hypwespectral remote sensing'이라는 용어가 통용됨
- 초분광센서는 높은 분광해상도의 영상을 취득할 수 있으며, 지상 대상물의 분광정보를 보다 세밀하게 표현함으로써, 기존의 다중분광센서에 비해 세밀한 분석이 가능함
- 이러한 지표면의 대상물체가 갖는 고유한 분광반사곡선을 이용하여 초분광영상에서 식생, 농경지, 도심지 등과 같은 공간적 영역을 결정하고 그 대상체의 형태 및 속성 정보를 획득할 수 있으며, 파장구간이 넓고 불연속적인 몇 개의 평면사진으로 구성된 다중분광영상에 비해 대상물체의 완전한 분광반사곡선을 획득할 수 있다는 장점을 가지고 있음
- 이와 같이 초분광영상의 분광 자료들은 광범위한 분야에 활용이 가능하며, 도로 노후화에 따른 포장상태 분석에 매우 유리한 특징을 가지고 있어 드론을 이용한 도로포장상태지수(PCI) 측정 기술에 활용하고자 함

○ 초분광영상 활용분야

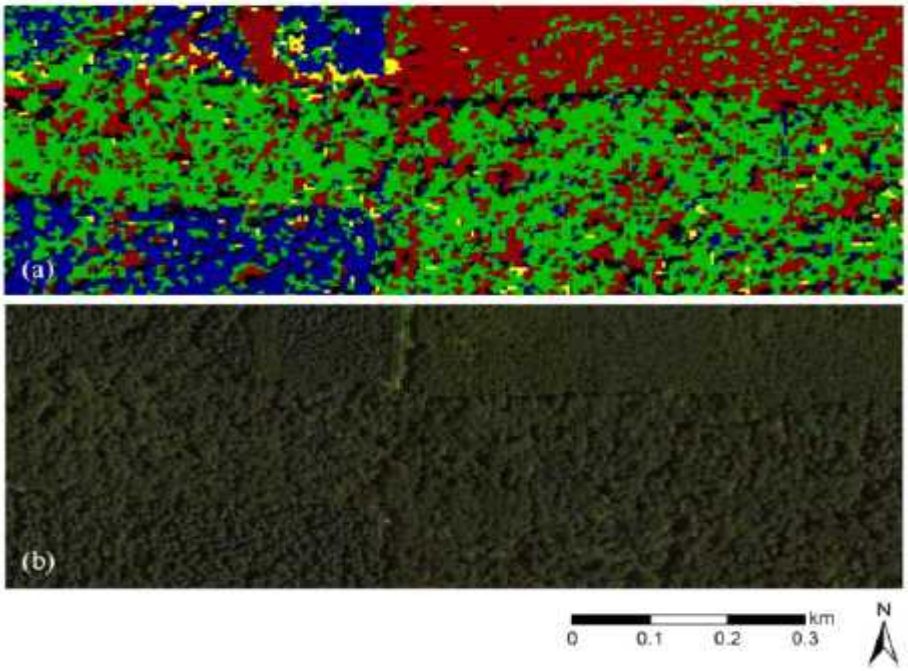
- **(산림분야)** 초분광영상로 취득된 산림의 분광정보는 수종분류, 산림 수종분포 파악, 특정 엽록소 및 수분 등 산림의 효율적인 계획 및 모니터링에 활용되고 있음

· Ballanti et al. (2016)은 ASIA(Airborne Imaging Spectrometer for Applications) Eagle 센서로부터 촬영된 초분광영상을 이용하여 미국 캘리포니아(California)주에 위치한 마린카운티(Marin County) 산림지역에 대해서 수종을 분류하였으며, 취득된 영상은 각각 SVM(Support Vector Machine)과 RF(Random Forest)기법을 적용하였으며, 적용 기법에 따라 다소 차이가 있었으나 전반적으로 약 90%의 분류 정확도를 나타냄



[대상지역 수종 분류결과 비교: RF 적용결과(좌측), SVM 적용결과(우측)]

· Tagliabue et al. (2016)은 APEX(Airborne Prism Experiment) 센서 자료 및 ML(Maximum Likelihood) 감독분류법을 이용하여 프랑스 알자스(Alsace) 산림지역에 대해서 수종을 분류하였으며, 분석결과 약 74.4%의 분류 정확도를 얻을 수 있었다고 하였으며, APEX 자료가 산림지역에 매우 유용하게 활용될 수 있으며, 향후 고해상도 초분광영상을 활용하여 정확도를 향상시킬 수 있다고 함

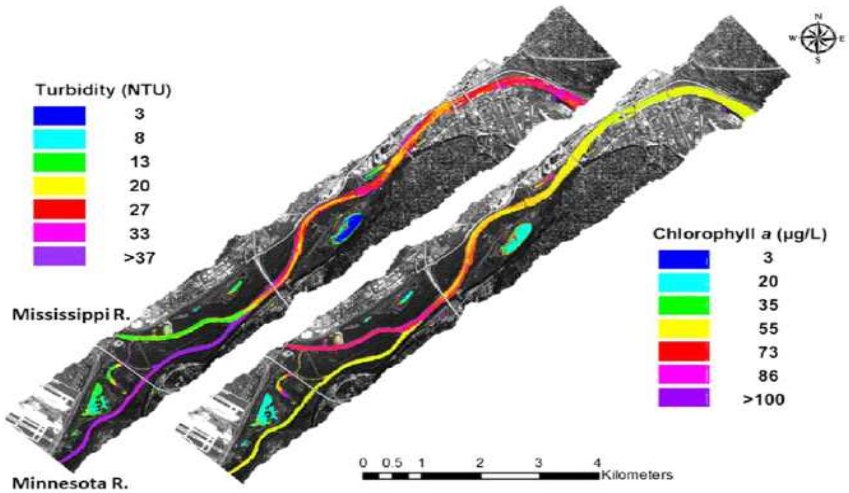


[수중분포 분석결과: 수중분포지도(위), 동일지역 정사사진(아래)]

· 국내에서는 송아람 등(2014)이 CASI(Compact Airborne Spectrographic Image) 초분광영상을 이용하여 독립기념관(충남 천안시) 주변 침엽수림과 활엽수림이 혼재되어 있는 2개 지역에 대하여 IEA(Iterative Error Analysis) 기반 분광혼합분석기법을 수정 및 적용하여 임상분류를 수행하였음.

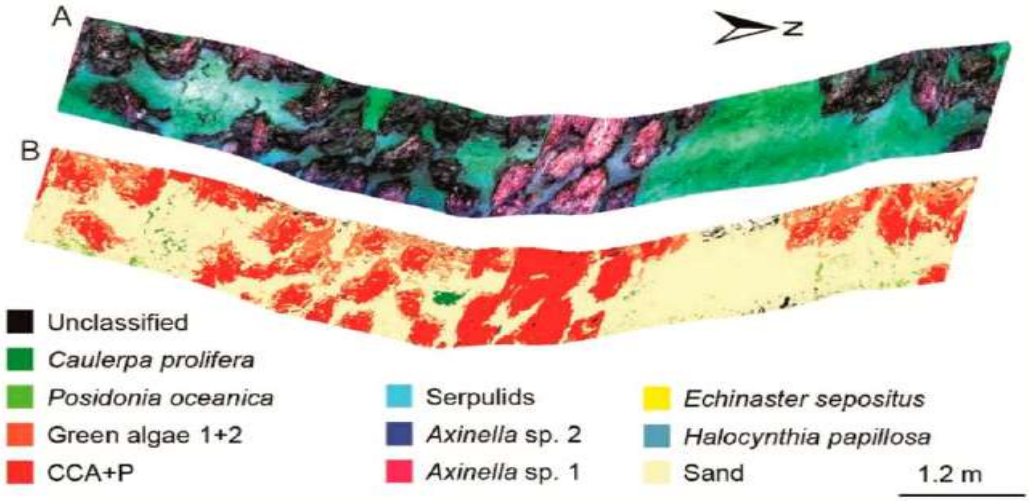
- **(하천·해양분야)** 과거 하천의 수질을 판단하기 위해서는 현장에서 직접 수질 샘플링을 한 후 실험이나 장비를 통해 분석이 이루어졌다. 이러한 기존 방식은 시간·비용·인력적인 측면에서 효율성이 매우 떨어지며 하천의 지점별 샘플데이터에 대한 조사결과를 토대로 하천 전체를 추정하기 때문에 신뢰성이 떨어지는 반면에 원격탐사 기반의 초분광영상은 기존 방식에 비해 효율적인 하천 조사 방법으로 각광받고 있음

· Olmanson et al.(2013)은 AISA 영상을 이용하여 미국 미시시피강에 대한 수질을 추정하고 현장조사결과를 기준으로 보정하였으며, 실험 결과 물의 고유한 광학특성(Inherent Optical Property, IOP)을 구별할 수 있는 분광특성을 찾아내고 이를 통해 수질지도를 표현하기 위한 모델을 개발하였음



[미네소타강-미시시피강 합류지점 탁도와 클로로필]

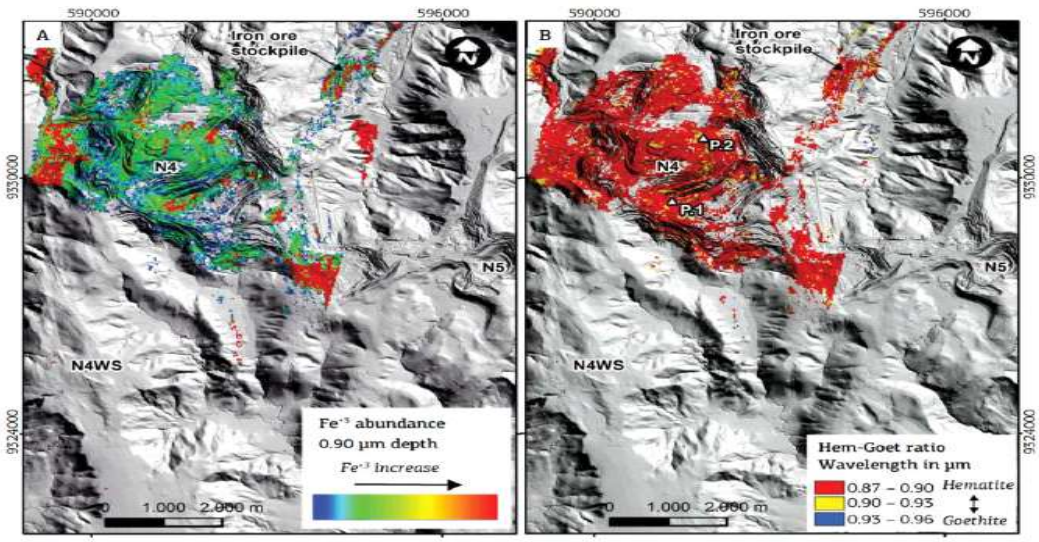
· Foglini et al.(2019)은 UHI(Underwater Hyperspectral Imager)를 이용하여 지중해 내심해 및 천해 수중 서식지를 조사하였다. 분석결과 각 조사지점에 대한 분광라이브러리를 토대로 감독분류를 적용한 결과 만족스러운 결과를 도출함



[해저 산호초 지역: 초분광영상(A), 감독분류결과(B)]

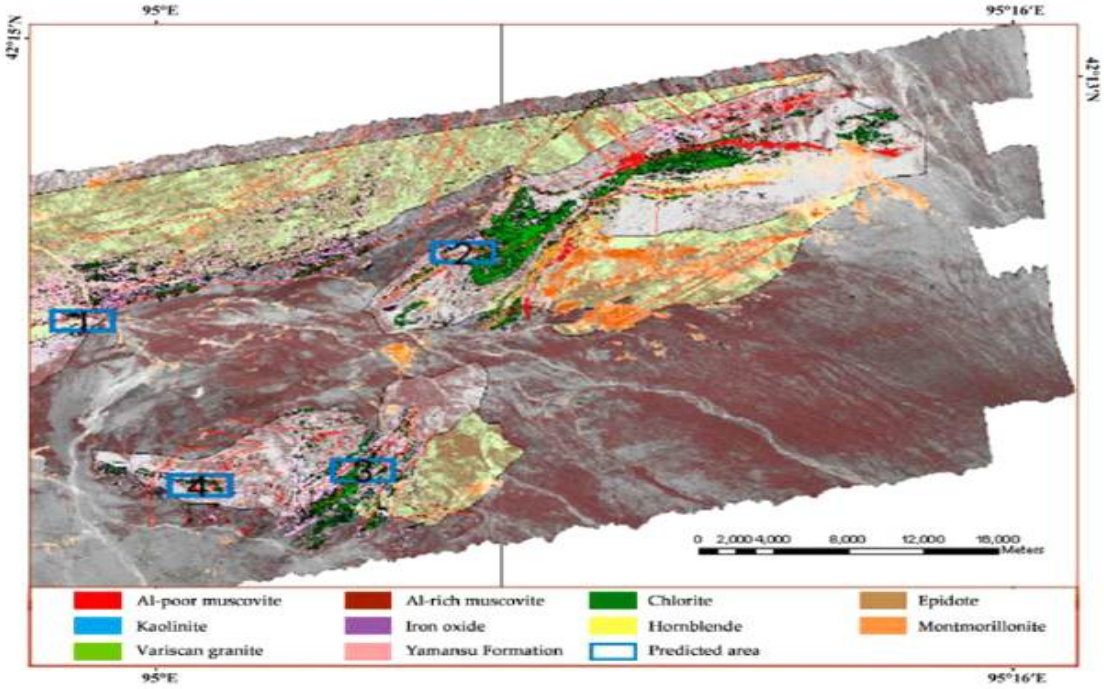
- **(지질자원분야)** 광물자원에 대한 정확한 위치 파악은 광물탐사, 폐기물 침전에 따른 오염도 추정과 위험관리에도 매우 중요한 정보로 활용됨

· Ducart et al. (2016)은 브라질 Carajás 광산지역에 대한 철(iron)의 침전물을 탐지하기 위하여 Landsat-8의 OLI(Operational Land Imager), EO(Earth Observing)-1의 Hyperion, 항공측량자료 등을 이용하여 OLI는 식생이 분포한 지역에 대해서도 산화철(ironoxide)을 탐지하는데 우수하였으며, Hyperion의 경우 폐광이나 처리시설뿐만 아니라 노천굴(openpit) 내 광물맵핑에 탁월하다는 결과를 도출함



[Hyperion 영상을 이용한 산화철 충만도(A), Hematite-Goethite 비율(B)]

· Pan et al.(2019)은 HyMap 초분광영상을 통해 대상지역에 대한 지질학적·광상성인론(metallogeny)적 특성, 광물변질대, 광물자원 등에 대한 포괄적인 정보를 얻을 수 있었으며, 현장조사결과를 토대로 검증을 수행함, 또한 MTMF(Mixed Tuned Matched Filtering)를 이용한 맵핑결과 열악한 자연환경 조건 하에서도 광물맵핑에 탁월한 결과를 도출하여 대상지역 내 기초적인 지형조사, 광물조사, 예측 등에 활용 가능성을 확인함



[HyMap 초분광영상을 이용한 변질광물 맵핑 결과]

- **(농업분야)** 초분광영상 기술은 작물의 분광특성을 토대로 작물의 스트레스 평가나 토양과 식생의 특성 분석을 통해 기존 방식보다 효율적으로 농작물 관리가 가능함
  · 초분광 영상은 농작물의 분광반사특성을 통해 식물의 종류와 상태에 대한 정보를 제공할 수 있으며, 특히 수분 또는 양분 부족에 의한 스트레스, 잎의 함수량 및 화학적 특성 등을 정규식생지수(NDVI, Normalized Difference Vegetation Index)로 산출하여 정량적 분석이 가능함
  · Colombo et al. (2008)은 잎의 EWT(Equivalent Water Thickness)가 변화하는 것은 잎의 가시 및 적외선 반사도가 변화하는 것에 기인한다고 하였으며, 초분광영상을 이용하여 잎 자체뿐만 아니라 경작지 전체의 수분함량을 추정할 수 있음을 인지함

○ 도로포장상태(PCI) 측정을 위한 드론시스템 기술 개발

- 본 연구 수행을 위해 투입된 드론 초분광시스템은 드론을 활용한 도로포장상태 평가 기술 적용을 목적으로 제작하였으며 시스템의 구성은 소형 초분광센서, IMU/GPS, 드론, SW로 구성됨



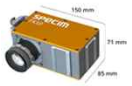
[드론초분광 시스템의 구성]

- 드론 탑재형 초분광 센서

· 도로포장상태 평가 적용을 위한 파장범위 400-1,000nm, 밴드폭 10nm 이하의 초분광센서를 조사하였으며, 도로조사 목적에 맞는 초분광센서는 뛰어난 공간 및 스펙트럼 해상도, 넓은 시야, 높은 SNR 등 고해상도로 영상을 취득이 가능한 Headwall社의 Nano-Hyperspec 센서를 선정함

· 이는 타 센서에 비해 크기가 작어 드론 탑재하여 안전적인 운영이 가능하며, 높은 신호대 잡음과 흔들림을 최소화하는 설계로 안정성 및 내구성이 뛰어난 강점이 있음

[드론 탑재용 초분광 카메라 조사]

센서	Nano-Hyperspec	Micro-CASI1920	FX-10	SOC710-GX	SHARK	OCI UAV
제조사	- headwall - (미국)	- ITRES - (캐나다)	- Specim - (핀란드)	- Surface Optics (미국)	- Corning - (미국)	- Bay Spec - (미국)
크기(mm)/무게(kg)	- 76×76×120/ - 0.52	- 102×190×178/ - 1.50	- 150×71×85/ - 1.40	- 103×200×200/ - 1.25	- 137×88×71/ - 0.680	- 80×80×30/ - 0.272
밴드수	- 270	- 288	- 220	- 120	- 180	- 100
밴드폭(nm)	- 2.2	- 2.1	- -	- 4.2	- 3.3	- 5
파장대(nm)	- 400~1,000	- 400~1,000	- 400~1,000	- 400~1,000	- 400~1,000	- 600~1,000
센서사진						

- 초분광 센서 탑재 드론

· 초분광센서의 탑재 드론은 DJI社의 Matrice 600 PRO기이며, 최대 이륙중량은 15.5kg로 소형 초분광센서를 탑재하고도 안정적인 운영과 성과도출이 가능함

[Matrice 600 PRO 드론 스펙]

장비	항목	세부 사양
 Matrice 600 PRO	- 크 기	- 1668×1518×727 mm - (프로펠러, 프레임 암, GPS 마운트)
	- 무 게	- 9.5 kg
	- 최대이륙중량	- 15.5 kg
	- 최대속도	- 65km/h
	- 최대 실용 상승 한도 - (해발 고도)	- 2500m

- GPS / IMU 시스템

· GPS(Global Positioning System)는 인공위성으로부터 수신된 전파를 해석하여 위치와 속도, 시간을 알 수 있는 범 세계위치결정 시스템이며, IMU(Inertial Measurement Unit)는 중력의 형상, 대상의 기울기 및 회전력을 감지하는 센서들로 구성되어있는 관성측정장치로서 GPS/IMU 시스템은 초분광센서에 장착되어 탑재 비행체의 위치, 시간, 속도, 방향 등의 종합적인 자세정보를 제공함



[GPS Antenna 및 IMU 센서]

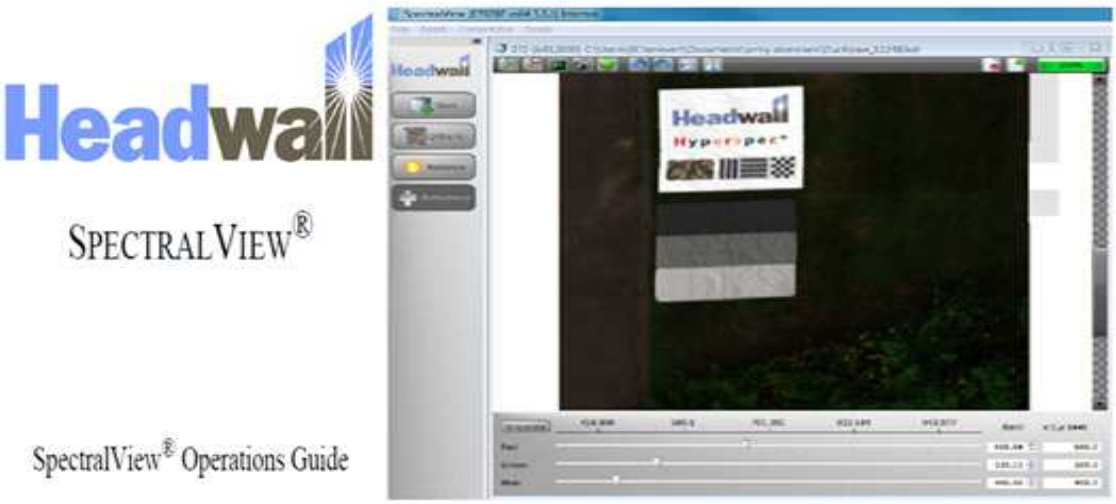
- 드론 초분광시스템 운용 및 처리를 위한 SW

· (드론 자동운용 SW) 도로포장상태 평가를 위한 객관적인 도로 분광라이브러리 구축을 위해서는 안정적인 성과도출이 필수적으로 요구됨에 따라 안정적인 성과도출에 많이 활용되어지고 있는 'Ugcs'드론 운용 SW를 활용하여 드론 영상촬영 활용함



[드론 운용 SW(Ugcs-Universal Ground Control Software)]

· (초분광영상 처리 SW) 기술개발된 드론 초분광시스템의 데이터는 제작사의 독자적인 파일형식인 큐브 형식으로 취득되어짐에 따라 최종성과물로 구축하기 위해서는 기하보정, 방사보정, 정사보정 과정이 요구되며, 이러한 전처리 과정은 제작사에서 제공하는 SpectralView SW를 통해 초분광영상을 구축함



[초분광영상 처리 SW(SpectralView)]

· (초분광영상 가공 SW) QGIS는 오픈소스 기반의 GIS 소프트웨어로 Widndow, Linux, Unix 등에서 실행되며 다양한 공간정보 포맷(vector, raster, database)을 지원한다. 또한 데이터 보기, 생성, 편지, 관리, 내보내기 등의 기능을 지원하여 영상데이터의 시각화, 데이터 분석에 활용할 수 있으며, SW를 통해 기존 PES장비로부터 산출되는 PCI(10m 단위)와 드론 초분광시스템 토대로 산출된 PCI 비교 분석을 위해 초분광영상 데이터 가공에 활용함



[초분과영상 가공을 위한 SW(QGIS)]

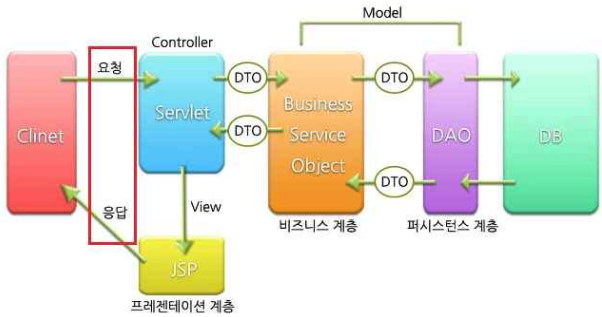
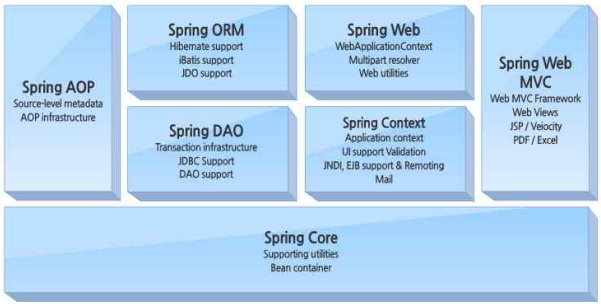
(3) PCI 측정 데이터 가시화 기술 개발

○ 서비스 서버 구축

분류	프레임워크 & 라이브러리	비고
OS	CentOS	가시화 시스템 구축 OS
Framework	Spring Framework, Mybatis, Jquery	가시화 시스템 구성 Framework
WAS	Tomcat	Web application server
Database	Oracle xe 11g	

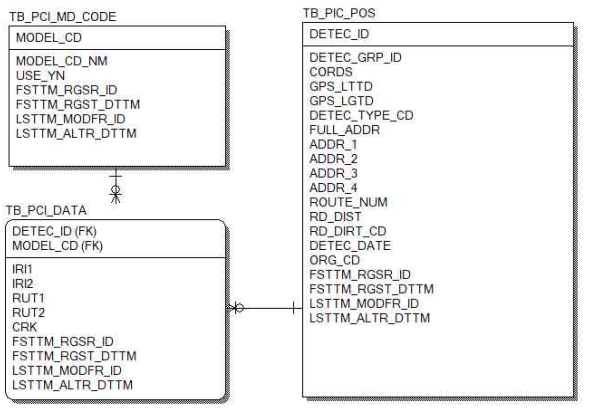
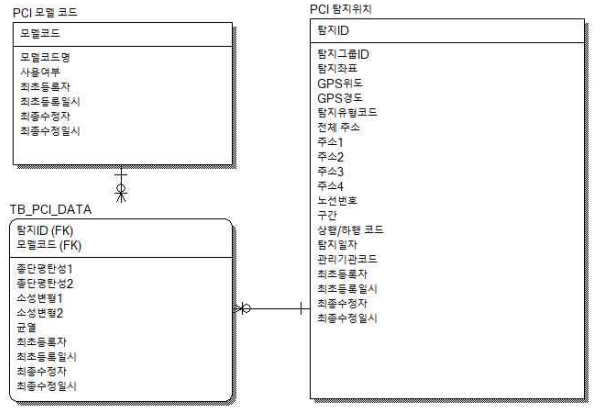
○ 웹 서비스 프레임워크

- 자바 플랫폼을 위한 오픈소스 어플리케이션 프레임워크인 Spring Framework를 메인 서비스개발에 사용함. 프레임워크는 프로그램의 기본적인 흐름과 구조가 정해져있기 때문에 개발 팀원간 부분적인 코드만 추가하여 하나의 일관성있는 어플리케이션 구축이 가능하며 소프트웨어 개발의 생산성 측면에서도 막대한 이점을 가짐

Sring Framework MVC Model2	Spring Framework 구조
	

○ Database 설계

- Database는 조사한 결과 Oracle 데이터베이스가 타 오픈소스 데이터베이스와 비교하였을 경우 성능과 안정성 측면에서 이점을 가진 것으로 판단되어 사용을 결정하였음.
- PCI분석 데이터에 대한 Database ER-Diagram을 Logical Data Model과 Physical DataModel을 구분하여 작성하였으며 각 Entity 항목별로 DB Table을 작성함.

물리ERD	논리ERD
	

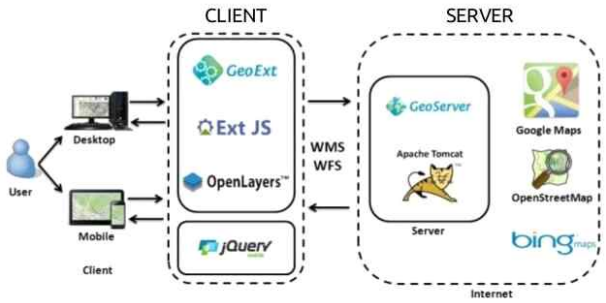
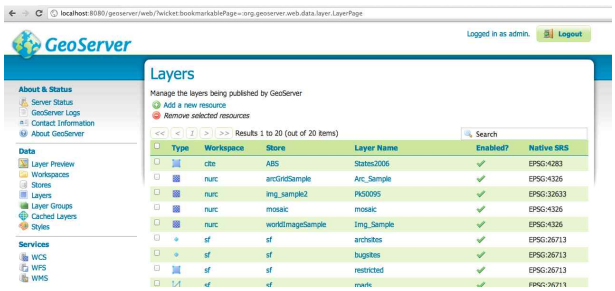
○ PCI 가시화시스템 SW API설계

- PCI 가시화 시스템의 Backend API 설계를 진행하였음. API의 URL Pattern을 정의하여 일관성있는 서버 데이터 요청구조를 가지도록 함. 데이터를 처리하는 방식에 따라 유의미한 요청형식을 부여하여 서버 어플리케이션 자원의 효율적인 배분을 유도하도록 구현

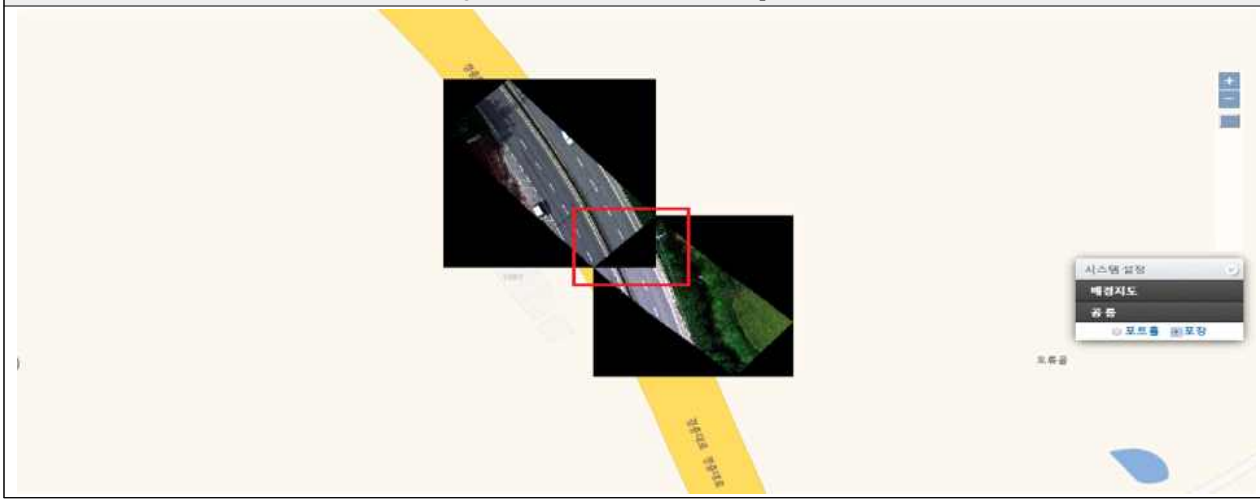
API URL	TYPE	Return
/pci/main	GET	PCI 가시화 시스템 메인페이지
/pci/options	GET	PCI 검색옵션 데이터 조회
/pci/list	GET	PCI 분석데이터 조회
/pci/collect	POST	PCI 분석 JSON데이터 시스템 연동
/pci/pos-update	POST	PCI 구간 좌표보정 API

○ 지도 데이터 연동을 위한 Geoserver 구축

- 정밀지도 구축을 위하여 지리공간 데이터를 공유하고 편집할 수 있는 오픈소스 GIS소프트웨어 서버를 설치하였음. 공간데이터를 WFS(Web Feature Service), WMS(Web Map Service), WCS(Web Coverage Service)표준에 따라 관리할 수 있는 서비스 인터페이스를 제공함.
- 초분광 데이터의 래스터 포맷인 GeoTiff형식을 지원함을 확인하였고 테스트를 통해 Vworld 레이어와의 연동을 확인하였음.

Geoserver Flow Diagram	Geoserver Web Interface
	

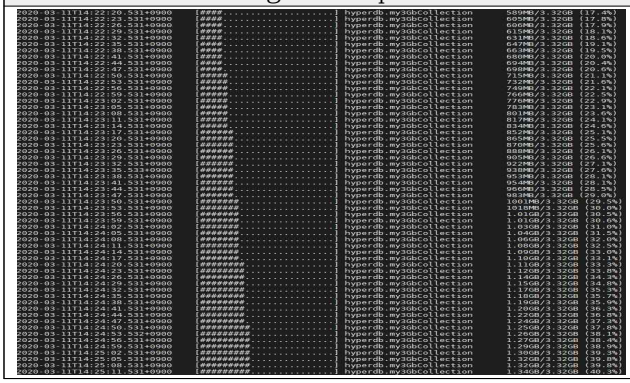
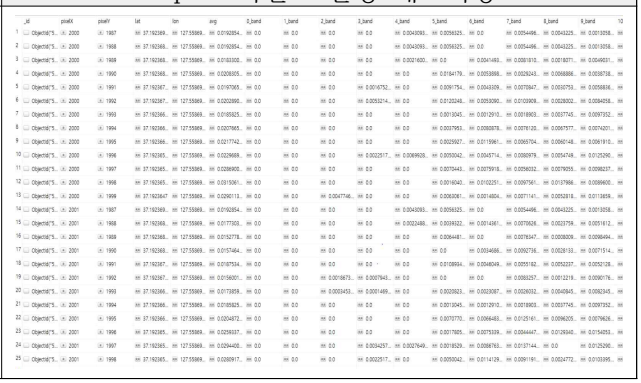
초분광 이미지 Geoserver import 결과



[초분광 이미지 Geoserver 연동]

○ 초분광데이터 분석을 위한 MongoDB 연동

- 대용량 초분광데이터의 데이터분석 효율성을 높이기 위하여 DB화 하려고 하였으나 초분광 데이터는 112개의 비정형적인 밴드 데이터의 특징과 하나의 이미지당 몇천만건의 데이터량을 고려하였을 때 RDB와는 구조적으로 맞지 않다는 것을 확인함. 대안으로 Nosql Database인 MongoDB를 활용하여 데이터저장 및 분석을 시도함.
- QGIS Tool을 활용하여 초분광 영상 데이터에서 래스터정보를 CSV포멧으로 Export시킨 후 MongoDB CLI Command로 초분광데이터의 Collection을 생성함.
- 테스트 결과 3Gb의 용량을 가지는 700만건의 초분광 영상 밴드정보가 10분의 처리시간으로 DB화처리되었음.

MongoDB import	Nosql DB화된 초분광 래스터정보
	

[초분광 래스터데이터 및 분석방안]

[1차년도 연구성과 요약]

□ 비탈면 변위측정 기술

ㅇ 비탈면 SAR 영상 생성 기술 개발

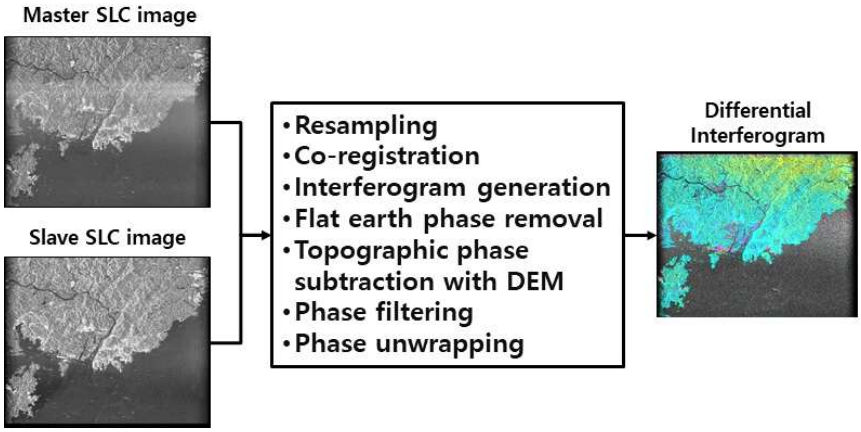
ㅇ 지상레이더 SAR 데이터를 이용한 지반변위 측정기술 분석 사례 조사

ㅇ SAR 영상 생성 기술과 지도 제작을 위한 정사 보정 기술 개발

ㅇ 드론 촬영 SAR 영상 생성 장치 입수를 위한 평가 테스트 진행

ㅇ 식생에 따른 긴밀도(간섭도) 차이 문헌 사례 연구 조사

ㅇ 드론에서 촬영된 SAR 영상의 좌표정확도 개선을 위한 알고리즘 개발



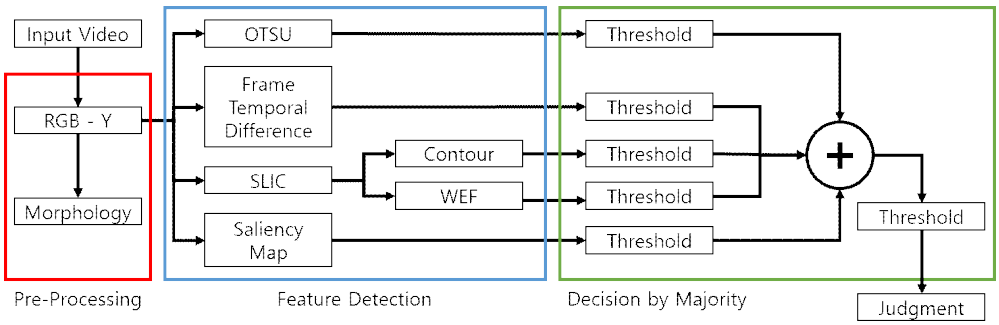
[영상레이더 차분위상간섭기법 적용 과정 및 순서도]

□ 포트홀 탐지 기술

ㅇ드론 촬영 영상 분석 기반 포트홀 객체 인지 · 추출 기술 개발

ㅇ 특징점 기반 포트홀 탐지를 위한 관련 연구 조사 분석

ㅇ 가중치 기반 하이브리드 포트홀 탐지 알고리즘 개발



[가중치 기반 하이브리드 포트홀 탐지 알고리즘]

□ PCI 측정 기술

ㅇ 드론 수집 초분광 데이터 분석 기반 PCI 정보 산출 기술 개발

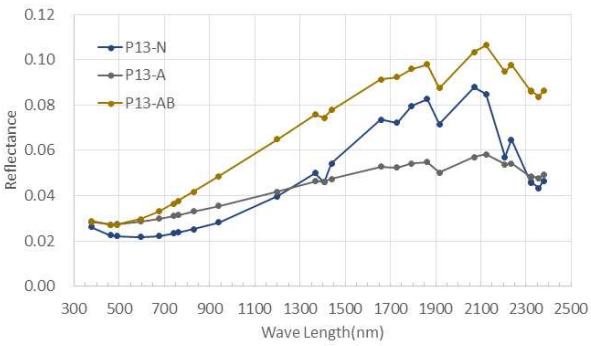
ㅇ 포장 노후화 유형별 초분광 실험



아스팔트 공시체
(직경 15cm)



아스팔트 공시체
(컷팅, 직경 15cm)



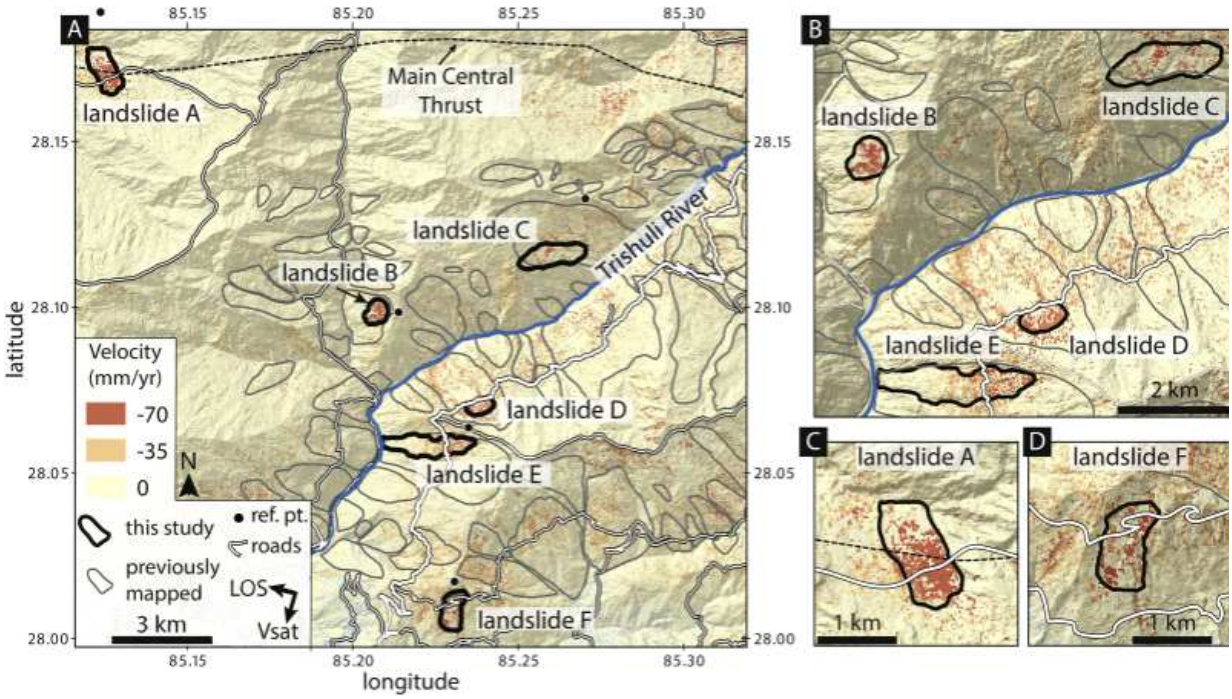
[2차 년도]

가. 비탈면 변위측정 기술

(1) 시계열 SAR자료 수집/처리 분석기술(알고리즘) 및 정밀 변위 해석기술 개발

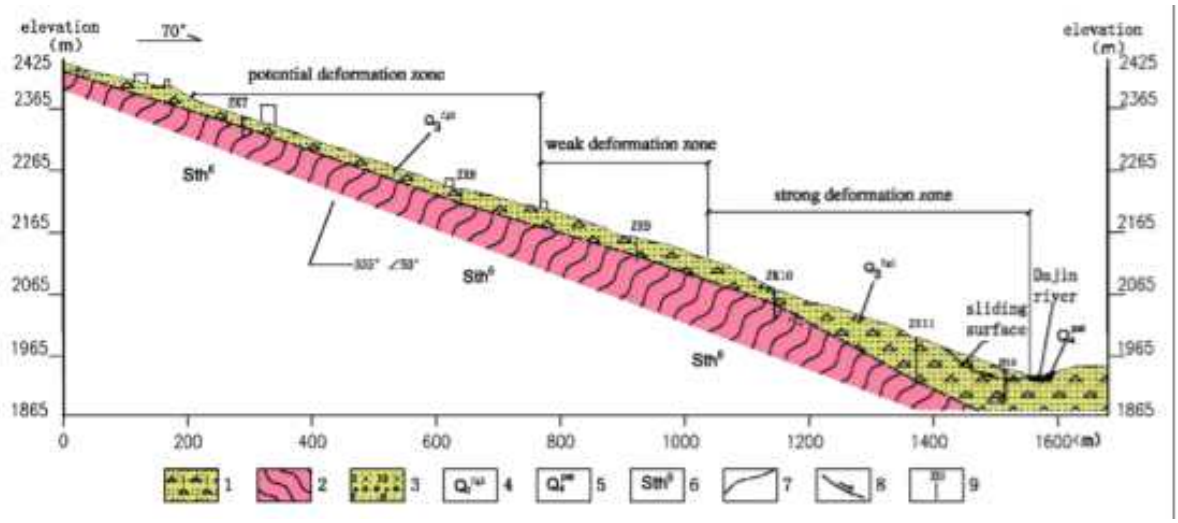
○ 위상간섭기법을 이용한 비탈면 정밀 변위 해석기술

- 영상레이더 위상간섭기법은 지상 기반 모니터링 기술이 불가능한 원격 지역에서 산사태를 식별하고 모니터링하는 데 사용할 수 있는 밀리미터 단위의 지표면 변형 측정을 제공
- D.P.Bekaert et al. (2020)은 새로운 위상간섭기법 지표 변위 감지 방식을 통해 지역 공간 필터 및 픽셀 클러스터링 방법과 함께 이중 차분 시계열 분석 방법을 사용하여 산사태 위치에 대한 사전 가정을 하지 않고 느리게 움직이는 산사태를 식별하고 모니터링 함
- 네팔의 Trishui 강 배수 유역을 중심으로 2014년에서 2017년 사이 수집된 Sentinel-1 위성 데이터를 사용하였음
- 암석학적 특징에 따라 발생하는 최소 6개의 느리게 움직이는 산사태를 발견하였고 산사태 침하량은 연간 2~9 cm이며 몬순 강우로 인해 발생할 가능성이 높음을 보고함

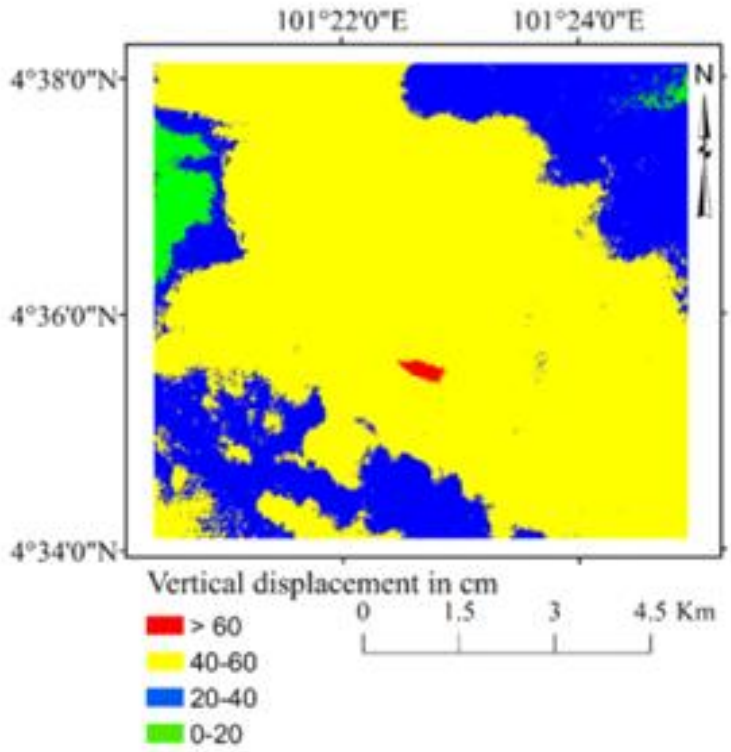


[고르카 2014년 10월에서 2015년 4월까지의 산사태 지도 및 InSAR 변위 속도]
(D.P.Bekaert et al., 2020)

- Jebur et al. (2014)은 산사태가 빈번히 발생하는 말레이시아의 Gunung 고개에 대해 적절한 계획 및 완호 전략을 펼치기 위해 산사태의 특징을 분석하고 산사태를 탐지, 지도화함
- 산사태의 수직 변위를 감지하기 위해 위상간섭기법 기술을 사용하였고 이는 차분위상간섭기법보다 요구되는 자료의 수가 적고 데이터 처리가 용이하여 연구비 절약에 적합하다 판단함
- L-band ALOS PALSAR 데이터를 획득하였고 SLC는 간섭도 및 긴밀도 데이터 생성을 위해 ASTER DEM과 결합하도록 생성한 후 긴밀도 데이터와 노이즈 감소를 위해 goldstein 필터를 사용하여 필터링을 진행하고 수직 변위 생성을 위해 도출된 영역의 위상을 unwrapping함
- 산사태에 대한 정확도는 0.19 m의 RMSE로 산출되었고 질량 이동량을 $100{,}000m^3$로 계산하여 부피로 나타내었음
- 본 연구의 성과를 유사한 열대 지역에 사용할 수 있다는 것과 L-band SAR 데이터가 식생이 많은 지역에 대해 허용 가능한 긴밀도를 제공한다는 것을 보여줌



[Jigaju 산사태의 지질학적 모식도]
(Jebur et al., 2014)

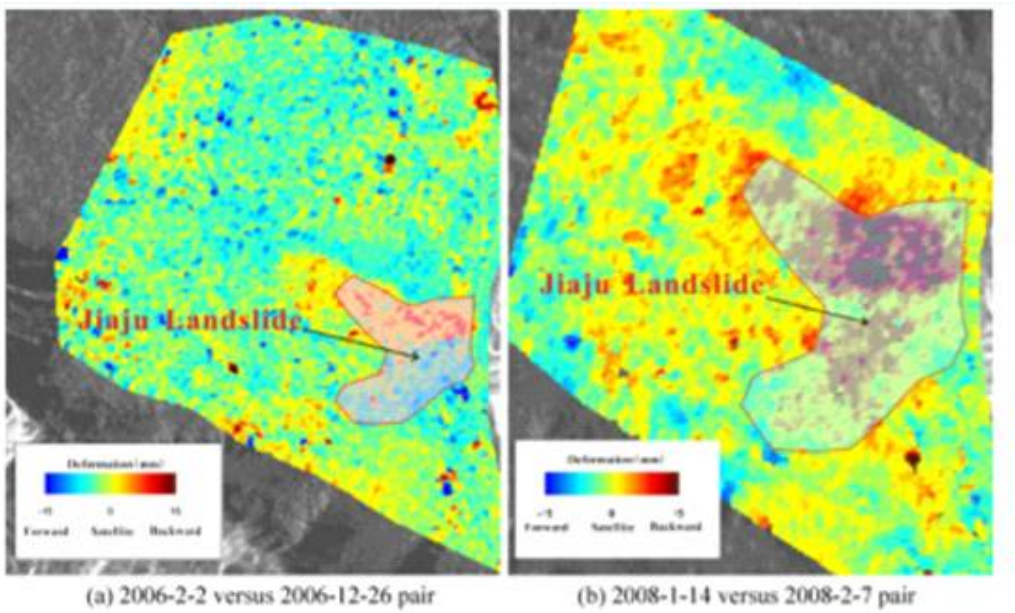


[Gunung 고개의 수직 변위 지도]
(Jebur et al., 2014)

- Yin, Yueping, et al. (2010)은 중국 쓰촨에 위치한 단바현의 Jiaju 산사태에 대해 GPS 및 InSAR 자료를 사용하여 관측함
- 해당 연구지역은 1200 m x 1000 m 면적으로 10-32도 정도의 경사를 가졌으며, 산사태 정도에 따라 전방의 강한 변형부분, 중앙의 약한 변형 부분, 후방의 변형 가능 부분으로 나뉘어짐
- 해당 연구는 수직 변위에 대한 GPS 측정의 정확도 개선을 위해 InSAR 자료를 사용하였으며, GPS 자료는 일일 수평 변위, InSAR자료는 월별 수직 변위에 대한 정보를 제공
- RADARSAT-1 자료를 사용하였으며, 2006년 6월 17일부터 2008년 5월 13일까지 총 25장의 영상을 사용하였음
- 연구 결과, 차분위상간섭도 상에서 남쪽과 북쪽 두 지역으로 구분되는 산사태 지표 변위 패턴이

나타났으며, 변위 속도가 2006년에 비해 2008년에 더 빠르게 나타난 것을 통해 시간이 지남에 따라 산사태가 점진적으로 가속되고 있음을 보여줌

- GPS 관측 변위는 지난 2년동안 북쪽에서는 수평 55-207 mm/a, 수직 -23 mm/a 로 발생하였으며, 남쪽에서는 수평 12-34 mm/a, 수직 -0.05~-12 mm/a의 속도로 변화가 일어났음
- 위상간섭기법 관측 변위는 2006년 12월부터 2008년 1월까지 북쪽에서는 수평 평균 41.6 mm/a, 수직 평균 43.9 mm/a으로 발생하였으며, 남쪽에서는 수평 평균 16.1 mm/a, 수직 평균 17.5 mm/a으로 나타나 GPS 관측 값과 유사한 결과를 보여줌



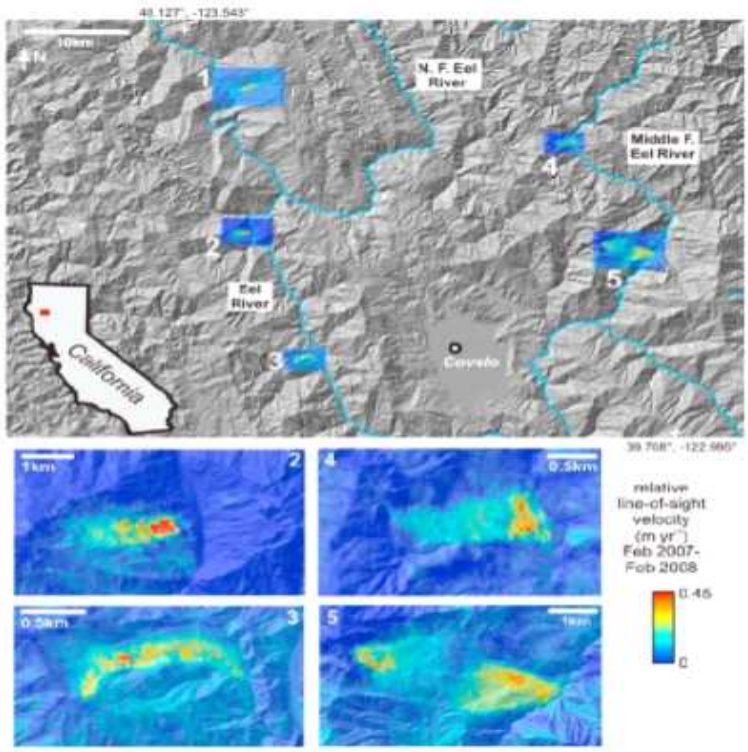
[지표 변위를 나타내는 차분위상간섭도]
(Yin, Yueping et al., 2010)

- Roering, Joshua J., et al. (2009) 는 북부 캘리포니아의 일 강에서의 대규모 저속도 산사태에 대해 17장의 ALOS 영상과 LiDAR 항공 영상 및 과거 항공 사진을 사용하여 비교 관측함
- 1964년에 촬영된 항공사진과 2006년 촬영된 LiDAR 항공사진을 정사 투영하여 산사태 발생 지역에서 관측되는 식생의 변화를 비교하여 산사태 변위 지도를 제작하였음
- 연구 결과, 위성간섭도 상에서 2007년 2월부터 2008년 2월 동안 5 곳에서 1 km 이상의 비교적 큰 규모의 산사태가 확인되었음
- 또한, 차분위상간섭도와 항공 사진 기반 변위 지도에서 확인되는 식생 변위 방향을 고려하여 비교하였을 때, 산사태 지형의 중앙 이동 구역에서 평균 하강 속도가 0.65 m/yr 로 발생함을 확인하였음

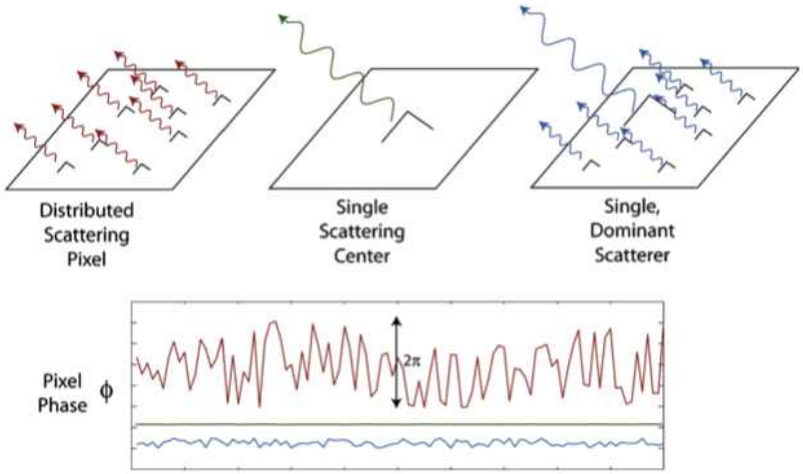
○ 고정 산란체 위상간섭기법 (Persistent Scatterer Interferometry: PSI)

- 고정 산란체 위상간섭기법 (PSI)은 SAR 영상에서 하나의 영상을 주영상으로 하여 간섭도를 생성하고, 간섭도 내에 건물이나 바위 같은 고정산란체 (Permanent Scatterer, PS)를 추출하며, 고정 산란체는 지표의 특성에 따라 신호가 일정하지 않은 분포 산란체와 달리 고정적이고 안정적인 신호를 획득할 수 있음
- 획득된 SAR 영상에서 고정 산란체 후보 (Permanent Scatterer Candidate; PSC)를 선택하고, 고정 산란체 후보에서 위상 성분을 구하여 차분위상간섭도로부터 모든 오차를 제거 후 시·공간 필터링으로 대기 오차를 보정하여 지표변위 값을 측정
- 지표변위 관측 정밀도는 SAR 자료들 간의 기선거리 (Baseline)에 영향을 받으며 시간적, 공간적

기선거리가 짧을수록 높은 정밀도를 가지며 넓은 지역 보다 좁은 지역에서 관측 정밀도가 높으며 지표변위 속도가 선형의 경향성을 가질 때 정밀도가 높음 (Ferretti et al., 2000 and 2001; Colesanti et al., 2003)



[북부 캘리포니아 일 강 저수지의 차분위상간섭도. 1 ㎞ 이상 범위에서 저속도의 산사태가 발생하는 지역을 지표 기복도 위에 나타냄]
(Roering, Joshua J., et al. (2009))

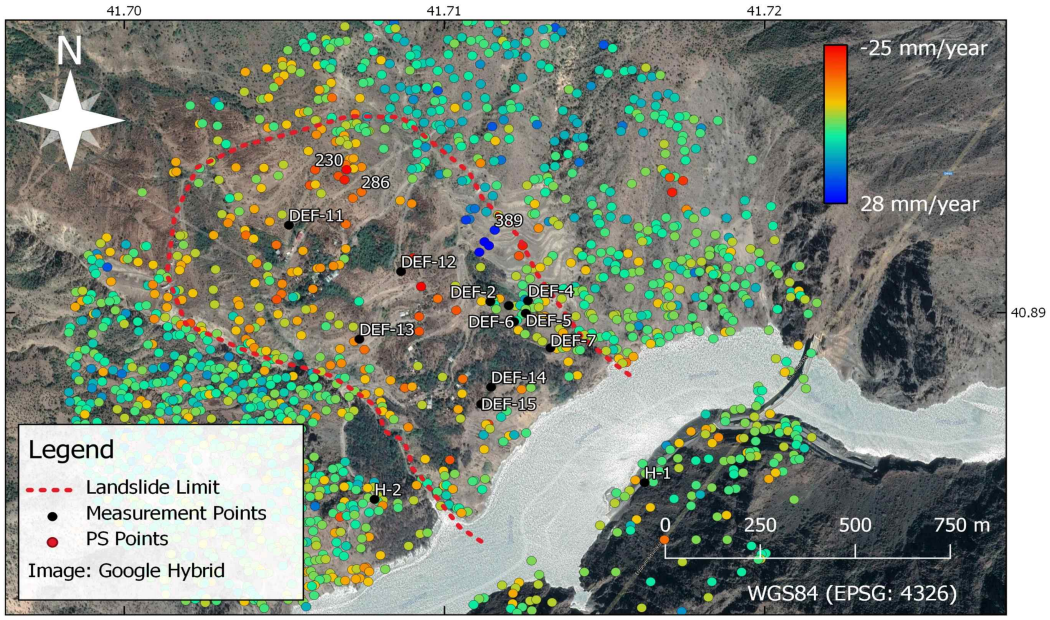


[분산 산란체 (왼쪽, 빨간색), 이상적인 단일 산란체 (가운데, 녹색) 및 지배적인 고정 산란체 (오른쪽, 파란색)의 모식도 및 간섭계 위상 시뮬레이션 (아래)]

- Yazici, B. V., & Tunc Gormus, E. (2020)은 Sentinel-1A SAR 영상을 활용하여 터키 Artvin 댐 지역에 대한 산사태 및 변형 속도를 PSI 방법으로 관측하였음
- 느리게 진행되는 산사태를 25 개의 Sentinel-1 이미지를 사용하여 관측하였고 두 가지 검증

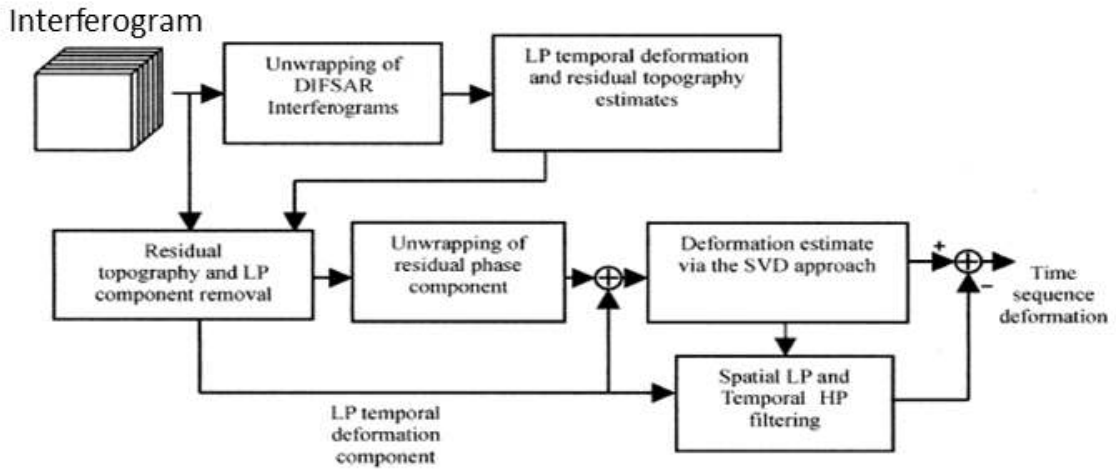
방법을 제안하여 PSI 기법의 효율성을 조사하였음

- 연구 결과 댐 주변 Havuzlu 마을 산사태 지역의 총 변위는 PSI 방법을 사용하여 연간 -25 ~ +28 mm 사이로 관측되었음



[산사태 변위 측정에 사용된 고정산란체 위치 (Yazici, B. V., & Tunc Gormus, E., 2020)]

○ SBAS (Small BAseline Subset)

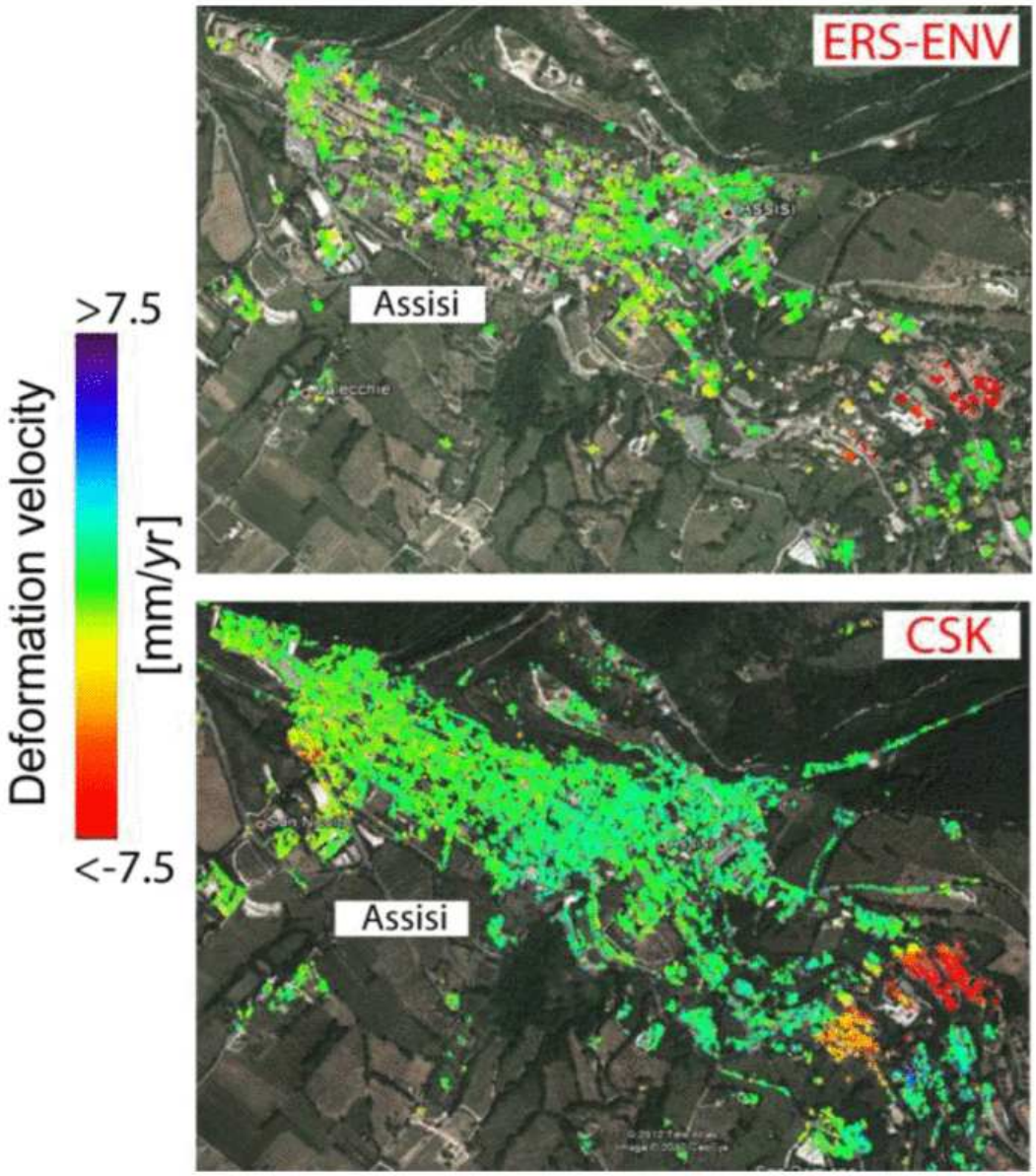


[SBAS 알고리즘 (Berardino et al., 2002)]

- SBAS 시계열 위상 간섭기법은 PSI 기법의 점 단위 분포 특성의 낮은 공간 긴밀도와 산악지역에서의 적용 한계를 극복하기 위한 긴밀도 기반 시계열 분석 알고리즘임
- 이는 간섭도로부터 계산된 위상에 포함된 오차 성분들을 효과적으로 제거하는 수단으로 정밀 지표 변위 관측을 위해 지표 변위에 해당하는 위상 성분을 제외한 위상을 제거하는 데 사용됨
- SBAS 기법은 PSI 기법과 달리 다수의 주영상 (reference image)을 사용하며, 기선거리가 짧고 공간적 긴밀도가 높은 차분위상간섭도를 작성함
- 공간적 긴밀도는 두 영상의 상관관계를 의미하며, 상관관계가 높을수록, 기선거리가 짧을수록, 공간적 긴밀도가 높게 나타남
- 절대 위상을 복원시킨 간섭도를 사용하기 때문에 발생하는 phase unwrapping 오차를 최소화하기 위해 잔류 차분 간섭도를 이용함
- 절대 위상이 복원된 간섭도가 가지고 있는 세 가지 정보인 변위 정보, 지형 정보, 잔류 오차 중

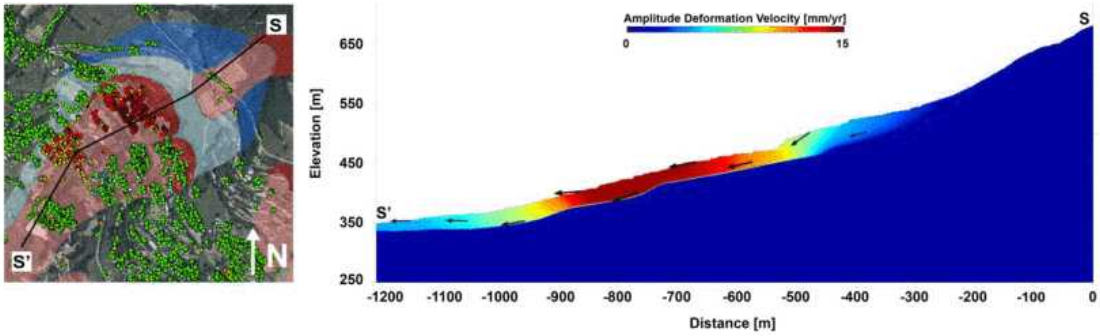
변위 정보만을 남긴 후 지형 정보를 제거하고 대기 보정을 위해 공간적으로 저주파 필터를, 시간적으로 고주파 필터를 적용함

- 오차를 제거한 변위정보는 각 위상 간섭도에서 불연속적으로 획득되었으므로 시간상으로 연결되지 않은 차분 간섭도로부터 지표 변위를 연속적으로 관측하기 위해 Singular Value Decomposition (SVD)을 사용함
- 선형으로 단순화된 지표변위 관측 결과의 왜곡을 보정하기 위해 시계열 분석을 반복적으로 수행하여 정밀도를 높임 (Berardino et al. 2002)



[이탈리아 중부 아시시 지역과 관련된 ERS-ENVISAT (위)와 COSMO-SkyMed (아래) 결과 비교]

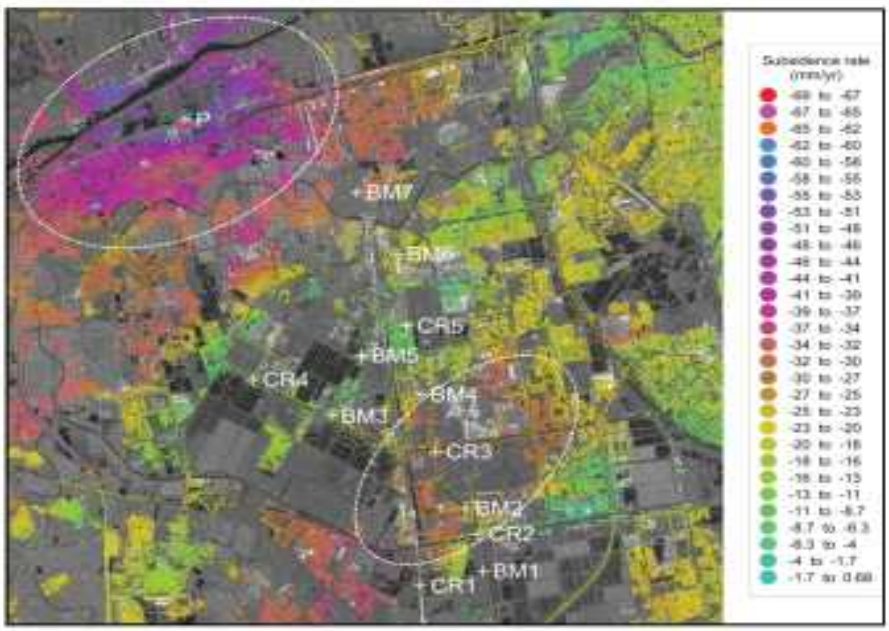
- Calò, Fabiana, et al.(2013)은 중부 이탈리아 Ivancich 지역에 대한 20년간의 C-밴드 ERS-1 / 2 및 ENVISAT SAR 센서와 X-밴드 COSMO-SkyMed 영상에 대해 DInSAR 및 SBAS 기술을 적용하여 산사태 및 침하를 관측하였음
- 수직 및 시간 기준선에 400 m 및 1500 일의 조건을 적용하여 347개의 위상간섭도를 생성하였음
- 최대 누적 변위 약 15 cm의 상당한 변형 추세를 보여주며 매우 긴 SAR 데이터 세트는 수십 년 동안 활동하는 느리게 움직이는 산사태를 조사하는 데 적합함을 보여줌



[Ivancich 산사태의 수치 모델링 결과. 왼쪽은 산사태 조사지도와 COSMO-Sky Med 전체 해상도 변형 속도지도 위에 중첩된 단면 SS '의 위치. 오른쪽은 S-S '단면을 따라 활성 산사태의 FEM 모델링으로 인한 2 차원 변형 필드]

○ TCPInSAR

- TCPInSAR는 SBAS와 동일하게 다중시기 주영상을 가지며 기선거리와 시간간격이 짧은 간섭도를 생성하고 PSI 기법에서의 PSC 추출과 동일하게 생성된 간섭도에서 긴밀도를 유지하는 점 TCP를 추출함
- 하지만 SBAS 기법과 달리 절대 위상이 복원되지 않은 간섭도를 사용하여 phase unwrapping 오차로 인한 분석 오류를 해결함



[23 쌍의 USB 간섭쌍을 이용한 모든 가능한 TCP에서 추정된 침하 속도]
(Guoxiang et al., 2014)

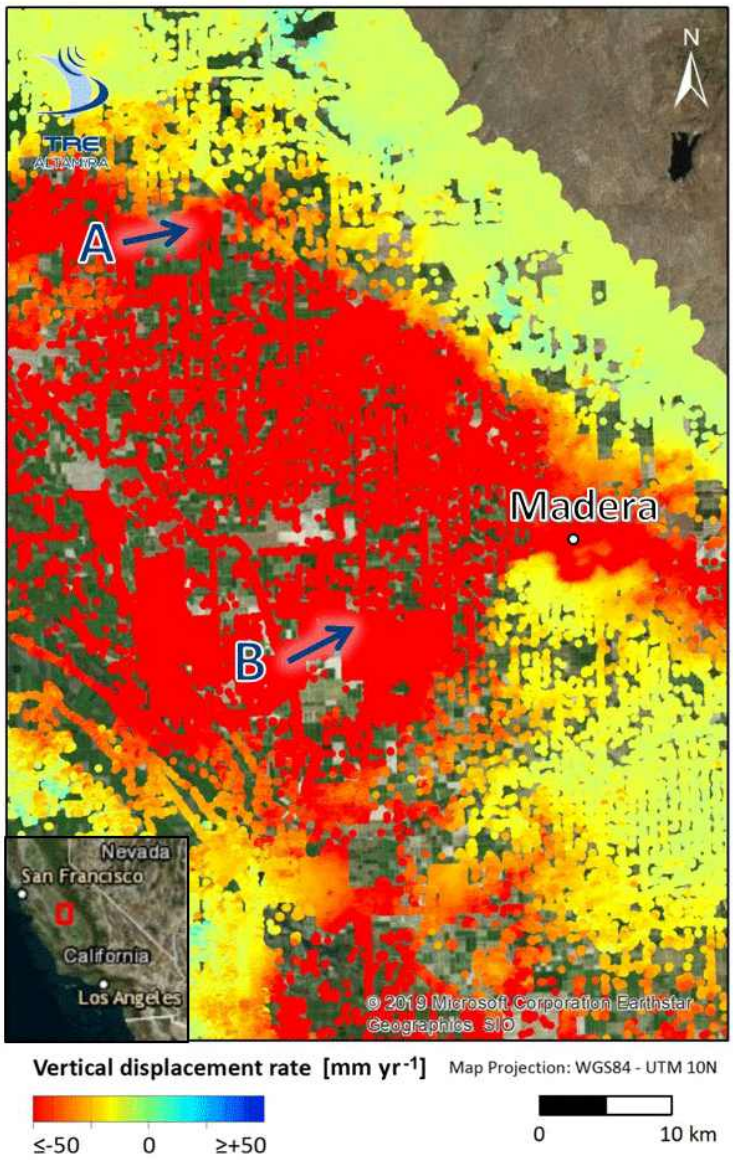
- 영상레이더 자료의 갯수가 적은 경우에도 침강을 탐지하는 데 사용할 수 있는 TCPInSAR기술의 기능을 확장하여 Ultrashort Space Baseline (USB) 간섭쌍 선택, TCP 식별, TCP 네트워킹 및 모델링, 그리고 최소 제곱 추정을 통한 오차 제거의 절차를 사용하여 중국 Tianjin 근처에서 획득한 고해상도 TerraSAR-X 이미지를 사용하여 지반 침하 속도를 측정함
- 해당 알고리즘을 사용하면 TCP의 밀도와 커버리지 범위를 획기적으로 높일 수 있으며 침하 측정의 품질을 높일 수 있음을 알 수 있음 (Guoxiang et al., 2014).

○ SqueeSAR

- SqueeSAR 기법은 permanent scatterer (PS)와 distributed scatterer (DS)를 같이 이용하는

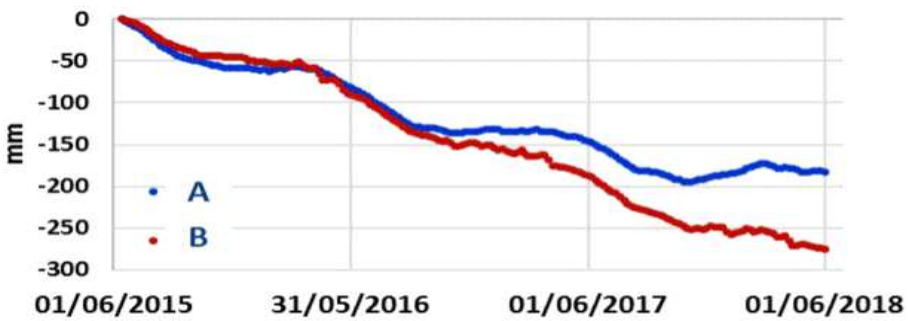
지표변위 관측 방법이며 PSI에서 고정 산란체를 선택하는 과정에서 발생하는 오차를 줄여 보다 정밀한 계산을 가능하게 함

- 고정 산란체가 불분명한 비도심 지역에서 효과적이고 PSI 기법에 비해 관측점의 밀도가 매우 높아 국지적 변위에 대한 분석이 가능함



[캘리포니아의 St Joaquin Valley에서 관찰 된 침하지도]

(Bischoff, Christine A., et al., 2020)



[위치 B와 위치 A에서 침하 행동의 차이를 보여주는 시계열]

(Bischoff, Christine A., et al., 2020)

- Bischoff, Christine A., et al.(2020)은 ESA의 Sentinel-1 센서를 이용하여 SqueeSAR 고급 처리 기술을 이용하여 국가 규모 단위의 재해 지도를 쉽게 제작할 수 있음을 보임
- 위성의 가시선 (Line-of-Sight: LOS)을 따라 각 레이더 표적에 영향을 미치는 변위 벡터의 투영을 측정함
- GNSS 데이터와 같은 다른 측정을 통한 교정은 전국 InSAR 데이터를 검증하고 기존 모니터링 프로그램에 연결하는 데 필수적임
- GNSS 시계열은 측정에서 노이즈를 줄이기 위해 이동 평균을 사용하여 필터링되며, 대규모 저주파 위상 패턴이 제거되어 다음으로 인한 바이어스를 방지함
- 위 사진들과 같은 변형은 LOS 및 수직 및 동서 방향 모두에서 ESA의 Sentinel-1 SAR 센서가 제공하는 빈번하고 안정적인 커버리지를 사용하여 국가 규모로 일상적으로 매핑할 수 있음

○ L-band SAR 드론 시스템 시험 운영
- UAV, 자동차 또는 레일에 장착 할 수있는 작고 가벼운 다목적 다중 채널 SAR 시스템
- 최대 4 개의 수신 채널을 동시에 사용하여 간섭계, 편광계 및 편광계 간섭계 SAR 데이터 수집
- 두 개의 반복 통과 SAR 이미지 (또는 더 긴 시계열의 반복 통과 측정)를 사용하여 레이더의 가시선 (LOS) 방향에서 표면 변위를 도출 가능
- Ku-band GPRI에 비해 LOS 표면 변위에 대한 시간적 긴밀도 측면에서 보완적임



[VTOL UAV를 이용한 L-밴드 SAR 드론시스템 시험 운영
(19.2.13, 19.3.28, 2회 수행)]

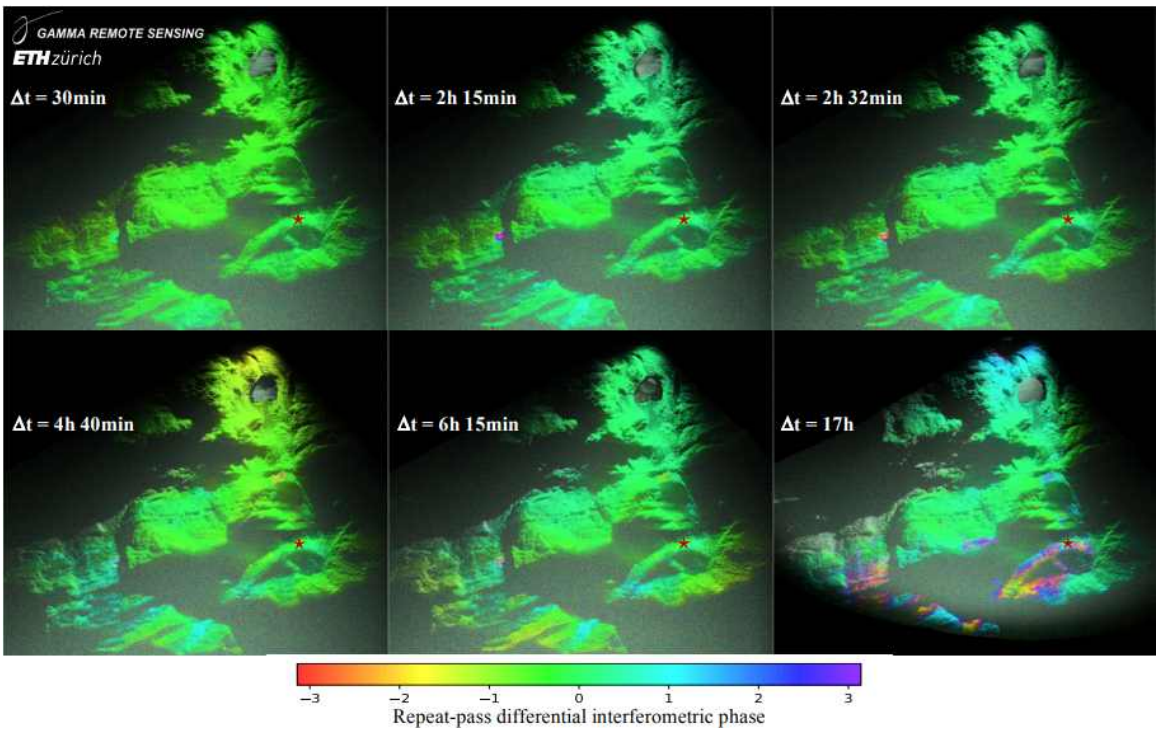
- 시험테스트에 사용된 드론 상세 정보
· 스위스 Aeroscout사 B1-100
· 길이: 3.3 m, 폭: 1.0 m, 높이: 1.3 m
· 주날개 길이: 3.2 m, 꼬리날개 길이: 0.65 m
· 공체 중량: 50 kg, 연료탱크: 5.0 L * 2
· 본체 재질: 알루미늄, 회전익 재질: 카본
· 상세정보 https://www.aeroscout.ch/index.php/scout-uav-helicopters/scout-b1-100-uav-helicopter

- 드론 임무 프로파일
· 시험비행시 레이더 관측 성능과 드론 운항 규정을 고려하여 적정 높이와 속도를 계산하여 운항되었음

· 비행고도: 약 70 m
· 비행속도: 6 m/s
· 비행거리: 트랙당 약 500 m

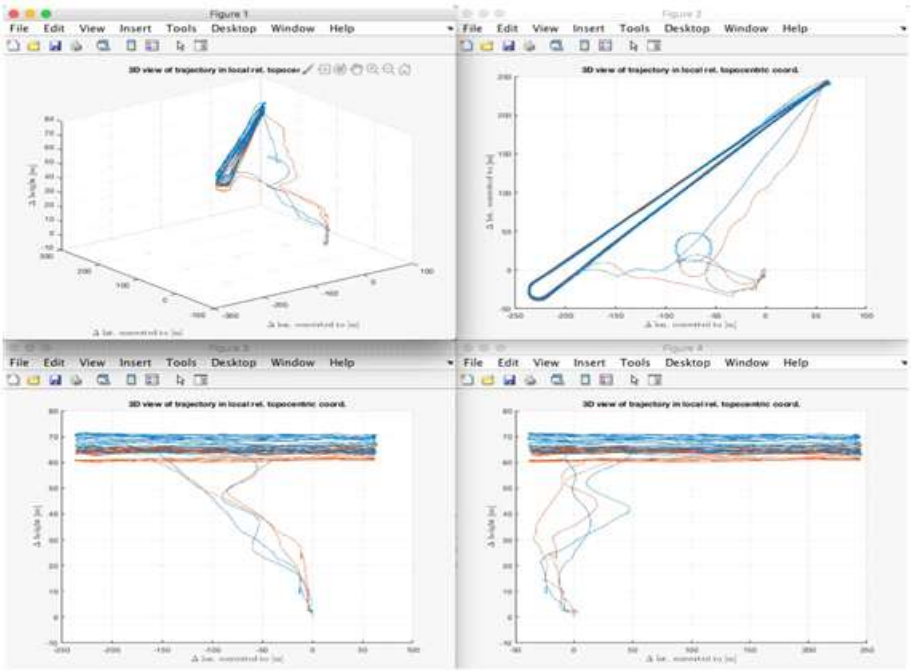
– 비행 시험 횟수 및 정량적 수치 데이터
· 위상간섭기법 적용을 위해 수 회의 동일궤도 관측이 요구되며 매우 정밀하게 반복 운항이 가능함을 확인하였음
· 시험비행 횟수: 각 테스트 당 13회 반복 비행
· 비행궤도 계산을 통해 궤도당 비행오차는 1 m이내임을 확인
· 정밀비행궤도 계산을 위해 INS/GNSS 추정기법을 사용하여 간섭기법 적용 가능한 충분한 궤도를 획득 가능함을 확인함

◯ Car-borne L-band repeat-pass InSAR



[Stein 빙하의 차분위상간섭도. 시간기준선 30분(왼쪽 상단), 135분(중앙 상단), 152분(오른쪽 상단), 280분(왼쪽 하단), 375분(중앙 하단), 1020 분 (= 17h, 오른쪽 하단)]

– 비탈면 시계열 변위측정을 위한 동일 비행 궤적 운항 시스템 구현
· 다중 시기 SAR 영상 생성을 위한 정밀 비행 궤적 운항 시스템 구현
· 관성항법장치와 GPS 위치정보시스템을 이용하여 정밀 비행 궤적을 계산함으로써 최적의 비탈면 시계열 변위 측정 솔루션 구현
· 구현 결과 매우 정밀한 궤도 정보를 이용하여 비행체가 비행할 수 있음을 확인하였으며 영상레이더 제작에 매우 필수적인 정보로 활용될 수 있음



[관성항법장치/GPS 시스템을 이용 정밀 측정한
드론시스템 비행 궤적]

- 자동차 탑재 모드로 작동한 GAMMA L-band SAR로 얻은 고산 빙하의 유속에 대한 간섭계 위상 측정
- 자동차 기반 SAR 데이터는 time-domain back-projection (TDBP) 접근 방식을 사용하여 디지털 고도 모델과 관련된 지도 좌표에 직접 초점을 맞춤
- 지오코딩된 SAR 이미지를 사용하면지도 좌표에서 차분위상간섭도를 직접 형성 가능

○ Rail-based repeat-pass InSAR
- 자동 repeat-pass InSAR 측정을 위해 레일에 장착 된 GAMMA L- 밴드 SAR 예시
- 레일은 각각 2 m 길이의 6개의 알루미늄 프로파일로 구성
- 운반이 용이하며 현장에서 2명이 설치가능



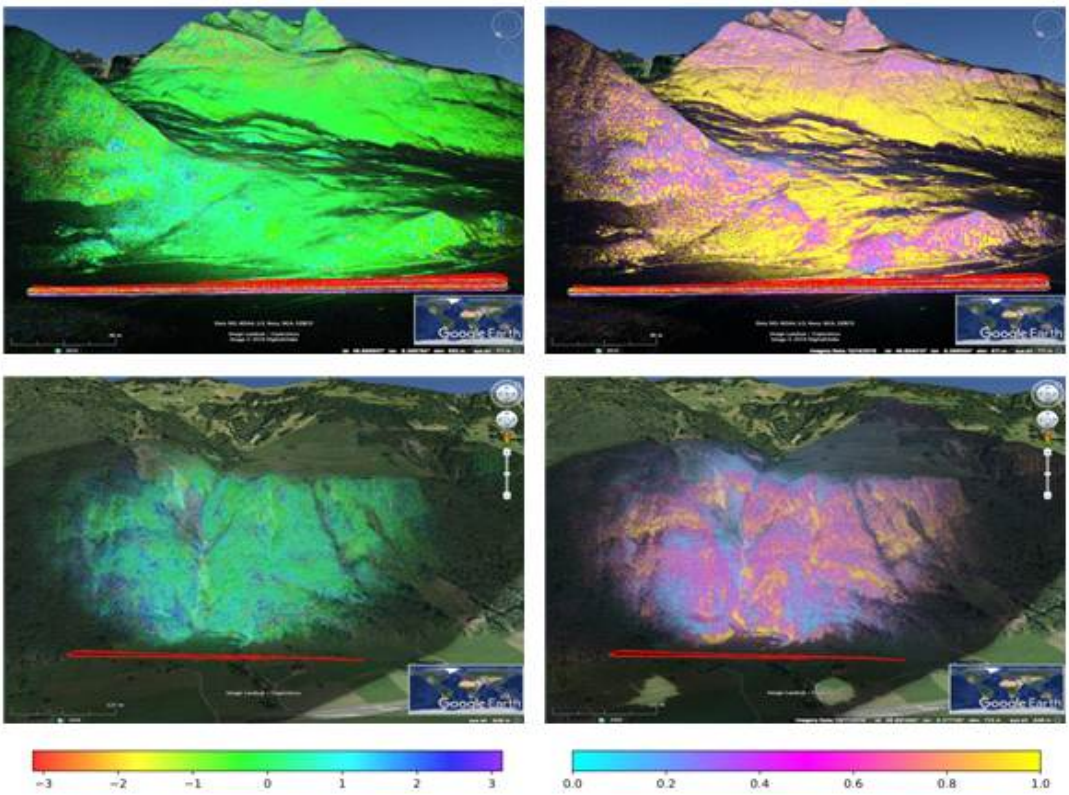
[12m 길이의 레일에 장착된 GAMMA L- 밴드 SAR]

○ 정밀 변위 해석기술 개발
- 정밀 변위 측정 소프트웨어 적용 기술 개발 구현

· 시험 비행 중 획득된 다중 시기 SAR 영상 생성을 이용한 비탈면 변위 지도 생성 기술 적용 구현

· 다중 시기 SAR 영상간 상관관계를 나타내는 긴밀도 평가 완료

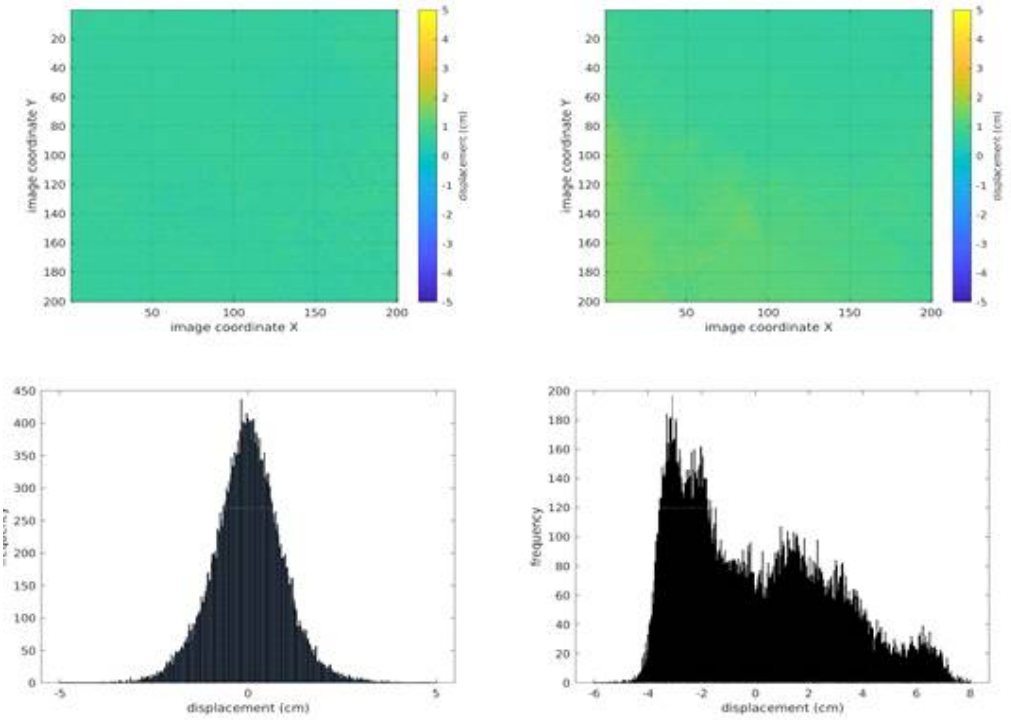
※ 긴밀도 값의 범위는 0(상관관계 없음) - 1(상관관계 완벽 일치)



[드론시스템으로 관측된 비탈면 변위 지도(좌),
비탈면 변위지도 제작을 위해 사용된 영상레이더 자료 사이의 긴밀도, 상관관계도(우)]

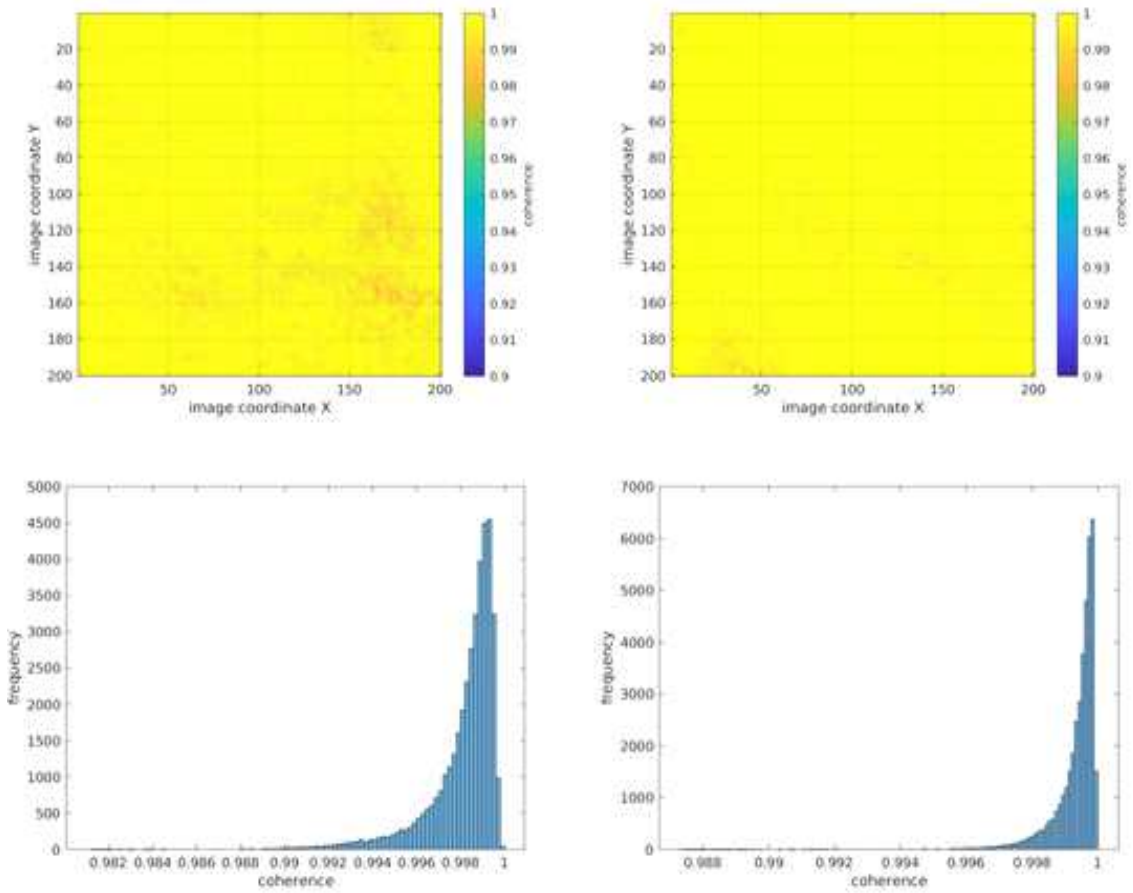
- 변위 측정 및 위치 정확도 평가 분석

· 변위 측정 정확도 분석 (평균 0.7-1.0 cm, 편차 0.2-0.5 cm)



[두 지역 변위 측정치 분포도(상), 변위분포 히스토그램 (하)]

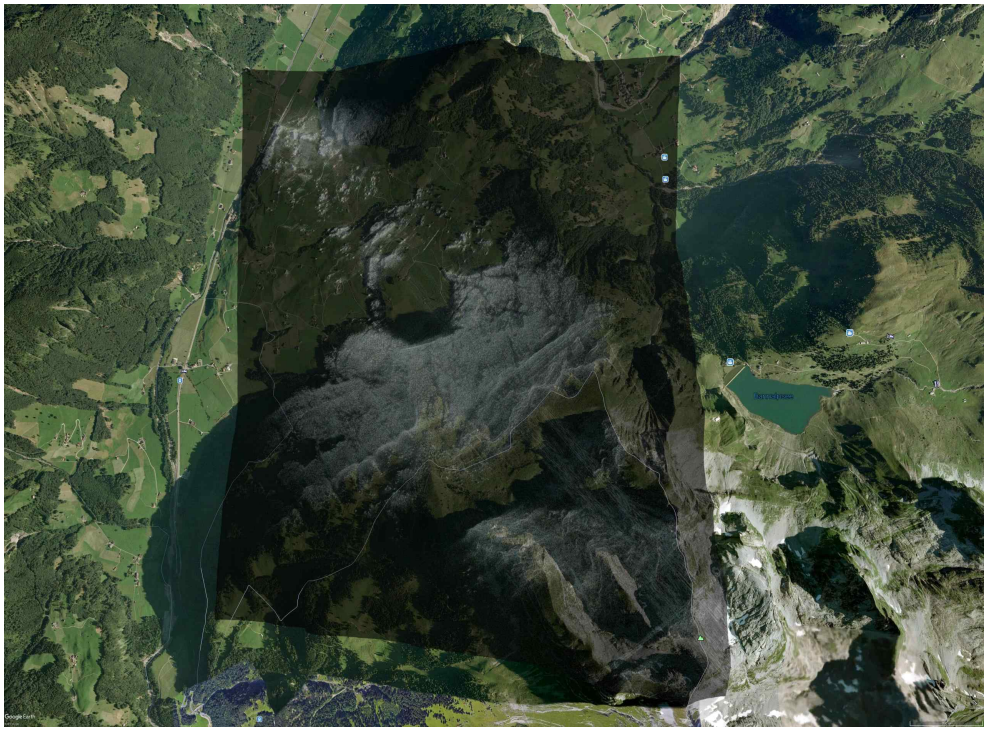
· L-밴드 파장인 23 cm을 적용하여 얻은 시험 변위의 평균과 표준편차 등 통계 계산 분석
· 최종 개발목표치인 5 cm 변위 측정 정확도 요구 성능을 만족하는 결과를 조기에 달성함
· 통계적으로 보다 안정적인 비탈면 변위 지도 작성을 위한 위상 잡음 제거 기술 개발 수행



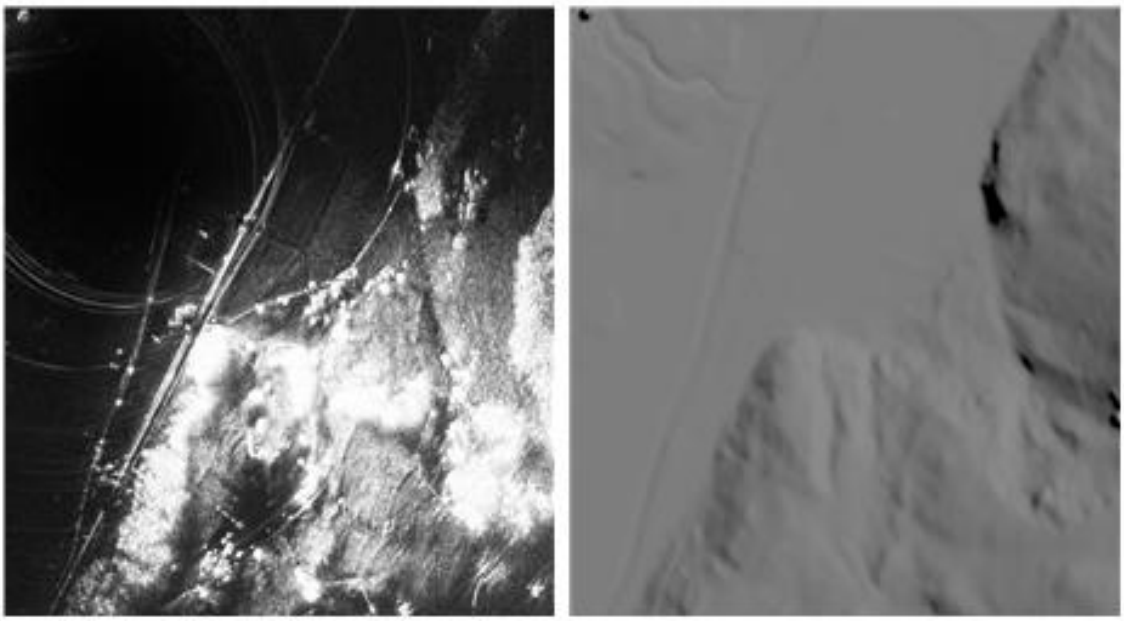
[두 지역에 대한 변위 분포 신뢰도를 나타내는 관측 영상간 긴밀도(상), 긴밀도의 히스토그램 분포도(하)]

· 긴밀도는 0.98이상을 상회하는 높은 값을 보여 매우 우수한 영상레이더 획득 시험임을 입증
· 비행 궤도 및 산림 분포, 밀도, 수종에 따른 긴밀도 저하에 대한 연구를 수행 예정

- 변위위치 정확도 분석 (평균: 0.975 (range: 1.1-1.4 m, azimuth: 0.4-1.0 m))
· 생성된 정사 영상은 약 1m 내외로 개발 목표치를 만족함
· 향상된 공간해상도를 가진 DEM을 이용하여 정사영상을 제작함으로써 변위위치 정확도 향상 기술 개발 진행
· 구글 어스와의 중첩을 통해 정확한 위치를 갖는 것을 추가 확인함
· 정밀 지상기준점 비교를 위한 corner reflecter(CR) 시험 제작 적용



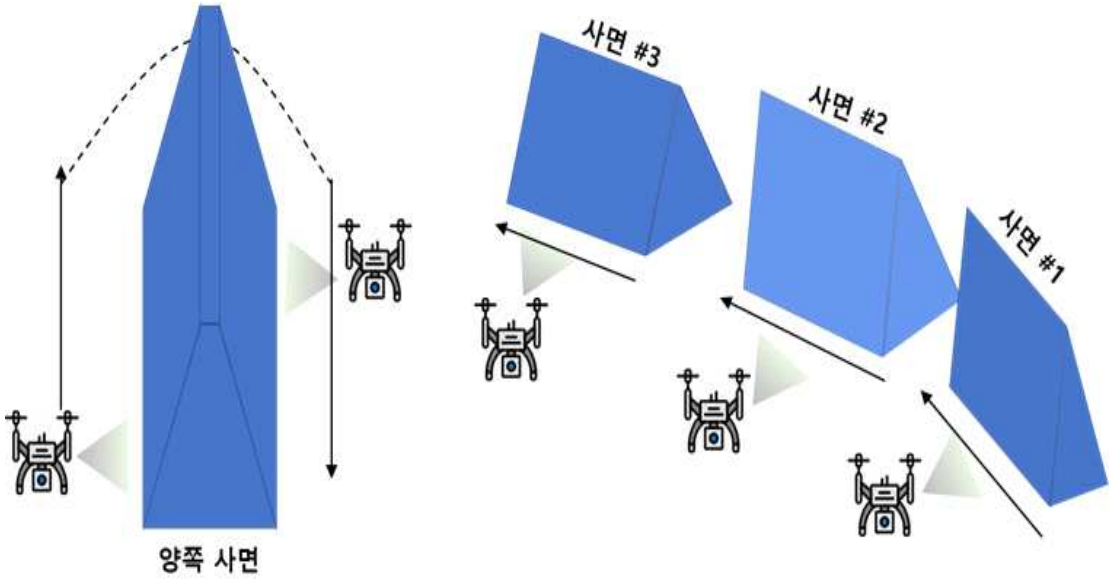
[관측된 영상레이더를 구글어스에 오버레이 도시]



[관측된 영상레이더와 지형표고모형과의 위치정확도 분석]

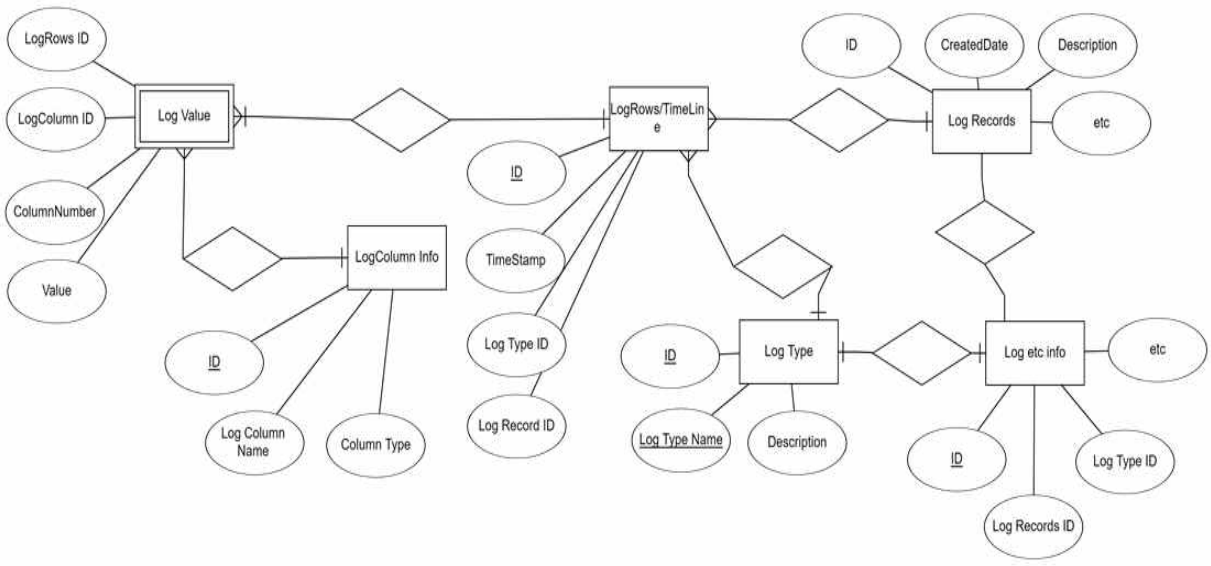
(2) MAVLink 기반 비행로그 정보 실시간 수집 SW 개선

○ 필요성
- 양쪽사면의 촬영시 카메라 방향을 조정하여 사면을 촬영할 수 있는 임무비행계획
- 크기와 기울기가 다른 복수의 사면을 촬영할 수 있는 임무비행계획
- 하나의 임무비행계획서에 위와 같은 기능을 포함할 수 있는 비행계획서 생성 기능 필요



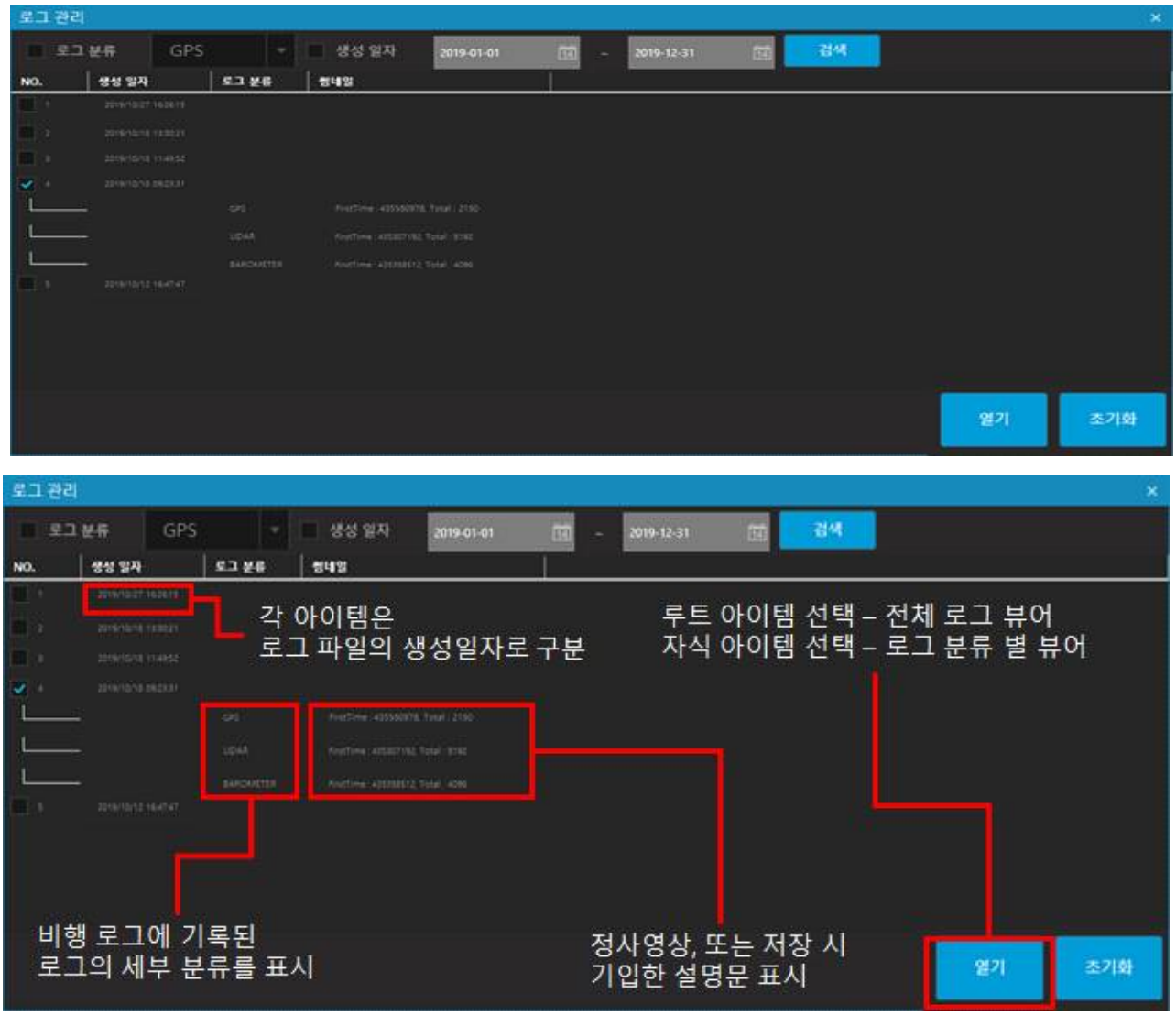
[사면의 양면 촬영과 연속적인 사면촬영]

○ MAVLink 기반 비행로그 정보 실시간 수집 SW 개선
- 비행로그 분석정보별 추출 및 시계열 적재
① ERD 설계 (비행로그 Raw Data 및 ETL Data)



[ERD 설계 아키텍처]

② 데이터베이스 정보 시각화 및 기능



[데이터베이스 정보 시각화]

○ 비행로그 기반 비행안전 및 임무상태 진단 기능

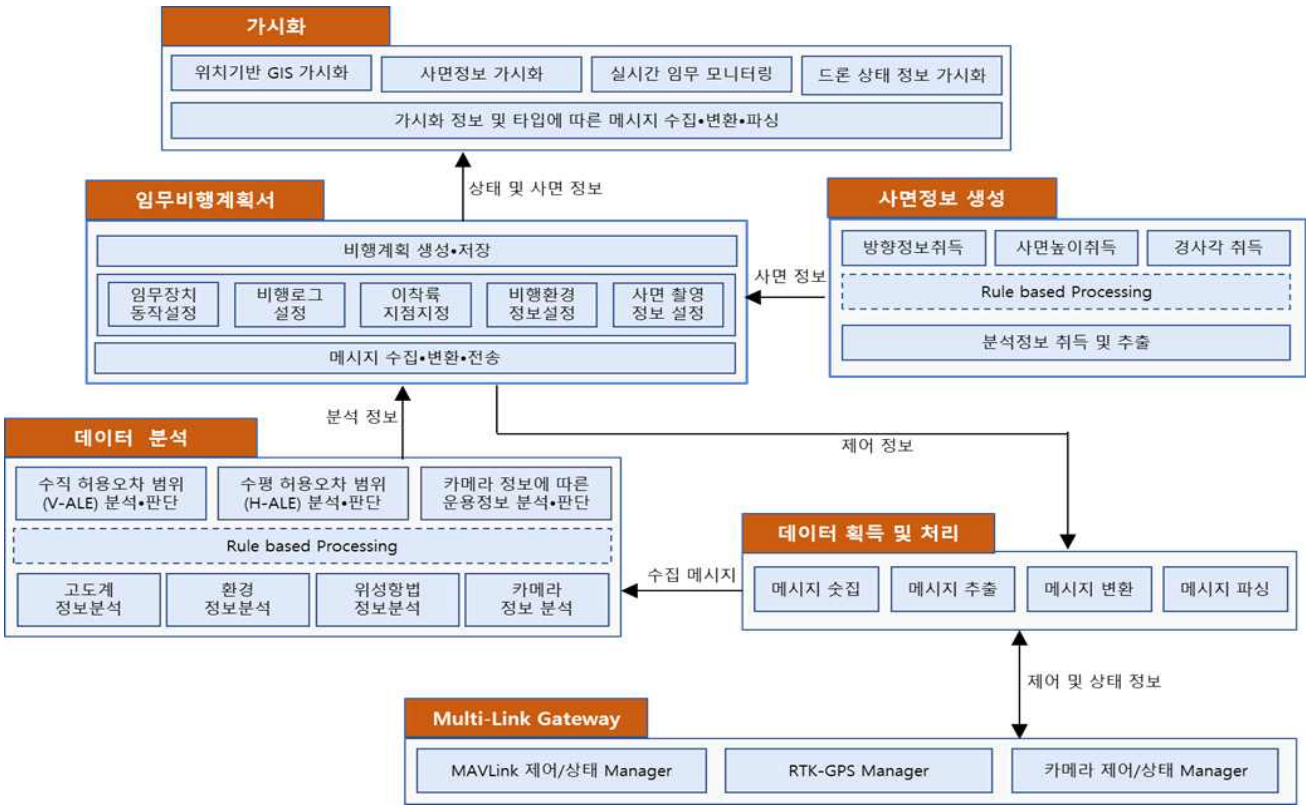
– 임무비행이 가능한 환경 및 드론의 상태 체크



[드론 상태 자동점검]

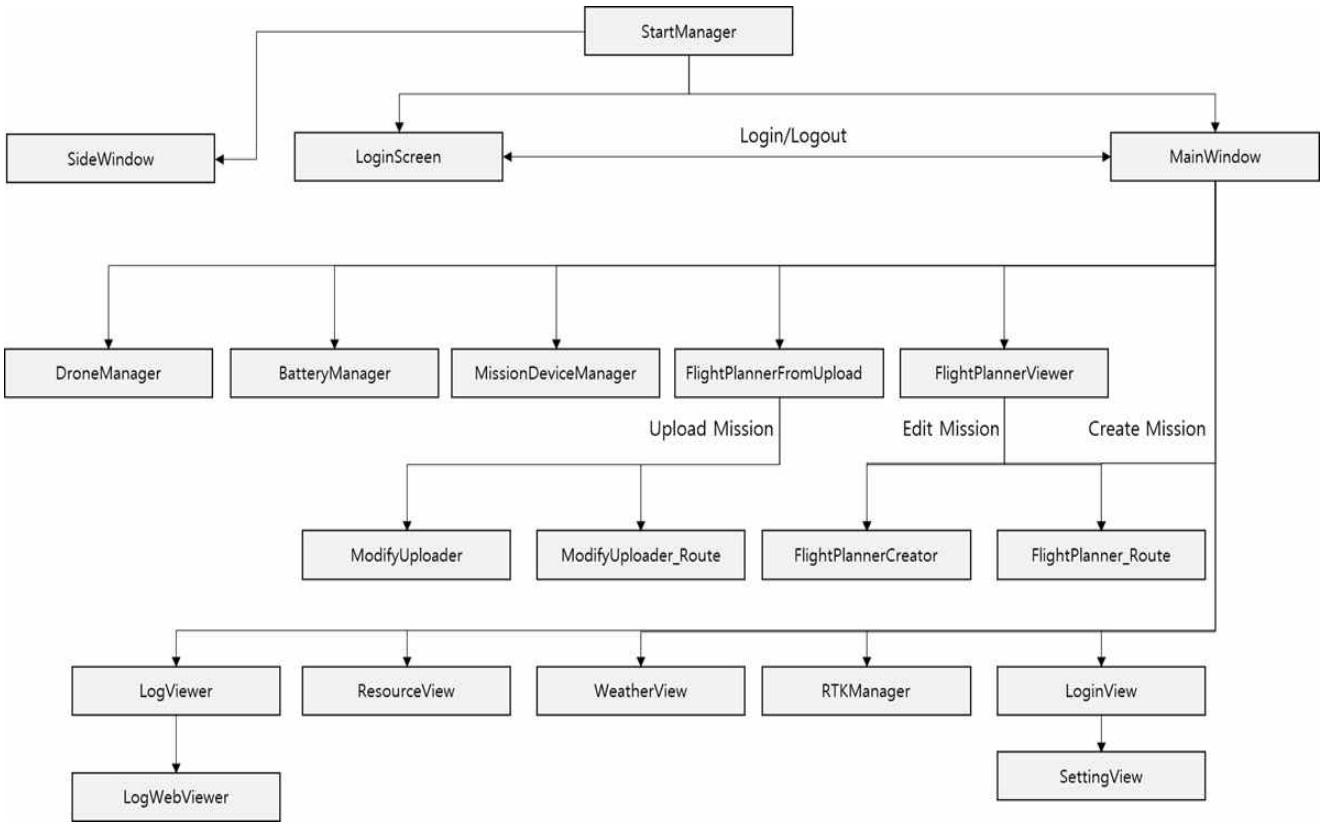
○ 요구사항 정의 및 설계

- 다중 사면 촬영 비행을 위한 기능별 구조 정의



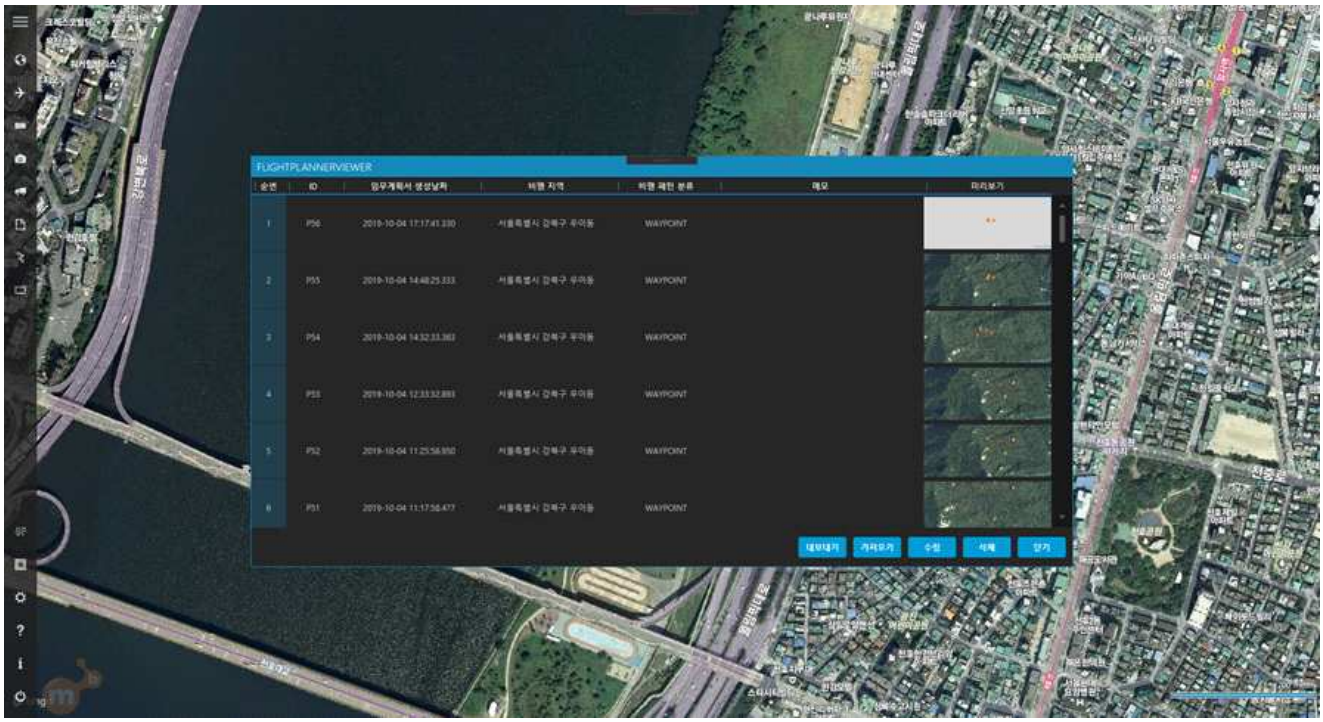
[임무비행계획서 기능구조도]

- 자동운항 기반 임무비행계획 UI 구조



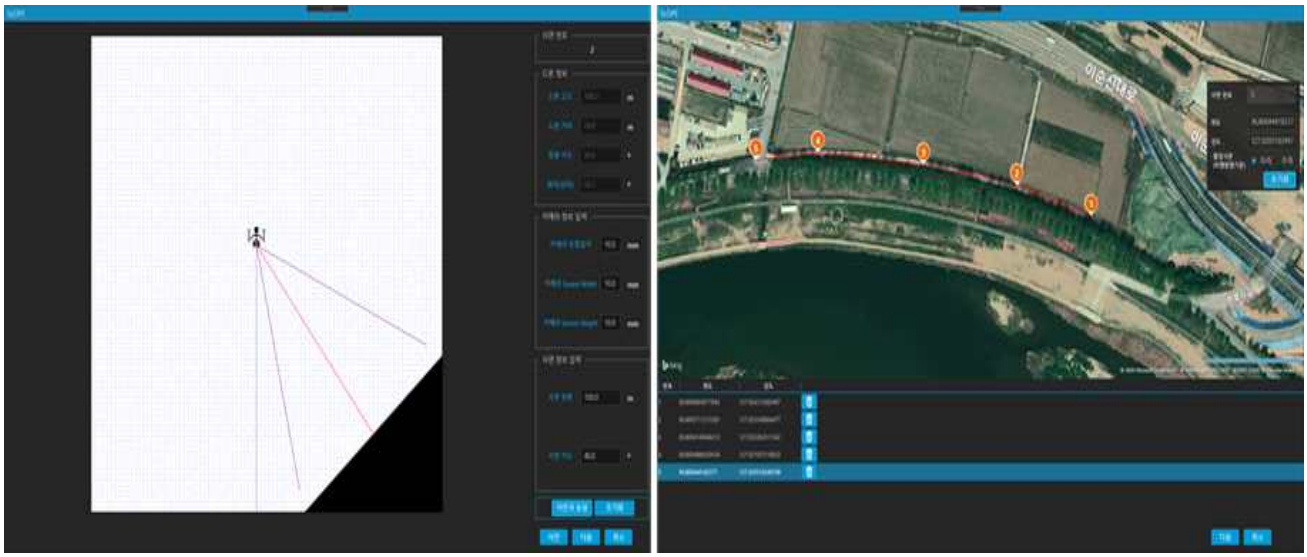
[자동운항 기반 임무비행계획 UI 구조도]

- 임무비행계획서 관리 기능



[작성된 사면 비행임무계획서의 조회 및 수정 기능]

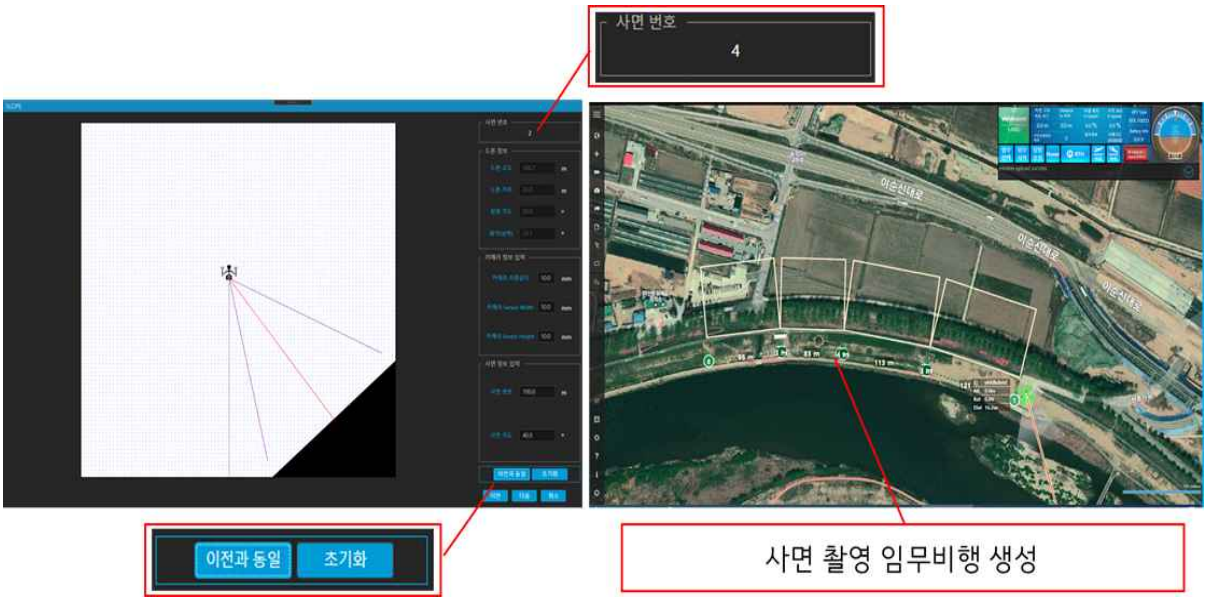
- 다중 사면 촬영 비행 및 비행경로의 좌측, 우측 선택하여 촬영을 위한 기능



[(좌)사면 촬영을 위한 드론의 운영 정보 입력, (우) 해당 촬영 지점에서의 카메라 방향 설정]

○ 다중 사면 촬영 기능

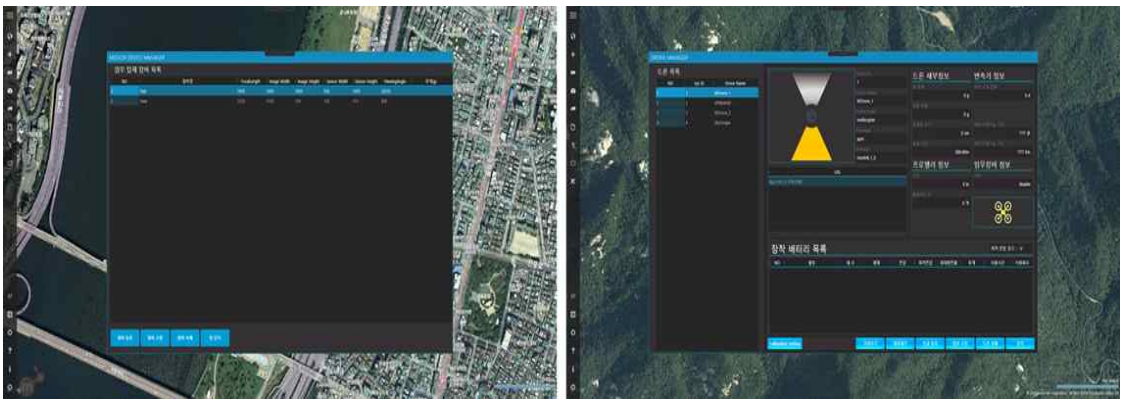
- 사면의 형태가 곡면의 형태인 경우 촬영을 위한 복수의 임무계획서
- 곡면의 사면을 다수의 직선임무비행계획서를 작성하여 사면 촬영



[드론의 임무고도 및 카메라 각도 입력 및 복수의 비행계획서 생성]

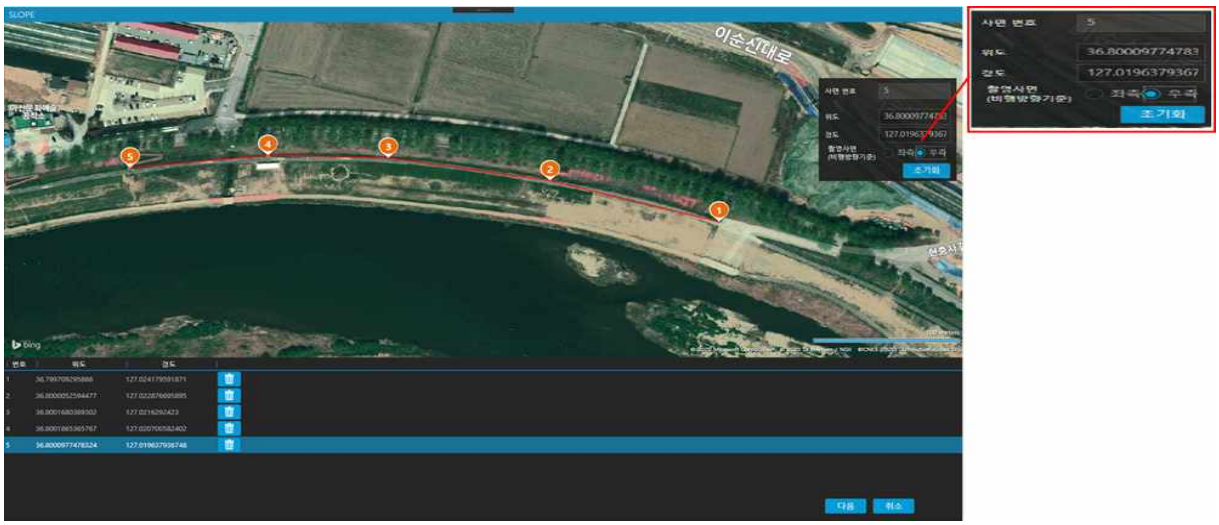
○ 사면의 위치에 따른 드론 카메라 촬영 방향 제어

- 드론 등록 및 케메라 정보 관리 기능



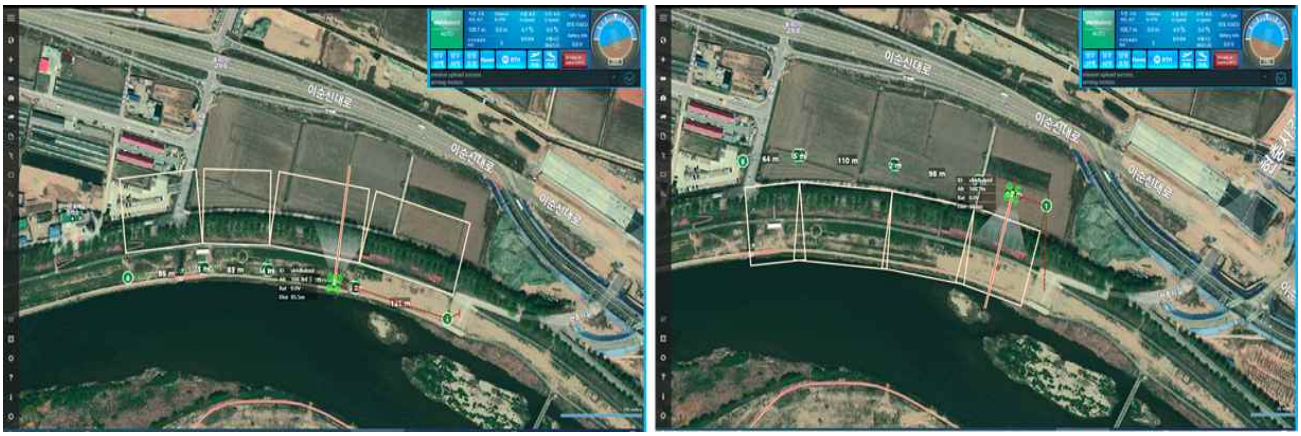
[(좌)카메라정보 등록, (우) 드론 등록]

- 촬영하고자 하는 사면의 위치에 따라 드론의 카메라 방향 설정 기능



[사면의 촬영방향에 따른 카메라 방향 설정 기능]

- 드론의 카메라 방향을 사면에 맞게 조정하여 드론 임무비행계획서 생성 기능



[드론 카메라 각도에 따른 촬영임무비행계획서 및 임무 수행]

○ 임무비행 환경에 따른 비행정보 특성 분석



(3) 드론탑재 SAR센서 기반 비탈면 변위 측정 및 가시화 결과 평가/검증 기술 개발

○ 드론탑재 SAR센서 기반 비탈면 변위 측정 및 가시화 결과 평가/검증 기술 개발
- 비탈면 변위 측정 절대 기준값 제시를 위해 실시간 변위 측정 센서 장치(스마트폴) 4곳 설치 및 무선 통신 수집 시스템 구축



[3축 가속도, 자이로 센서 기반 실시간 변위 측정 장치(스마트폴)
및 무선 수집 장치 (광명 철산 o단지 설치 사진)]

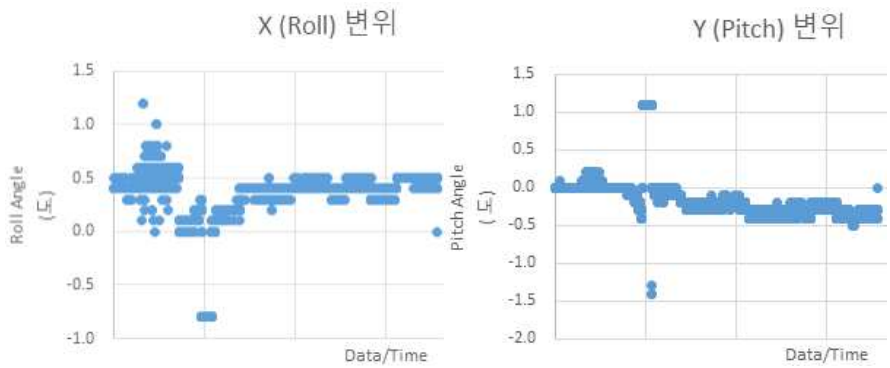
- 광명 철산 굴착 현장에 시험 시공 및 광파기/레이저 스캐너 지상 타겟 계측을 통해 변위 측정 장치 검증 수행
- 터파기 굴착에 따른 배면 비탈면 거동을 확인하는 비탈면 외부 계측(광파기, 레이저 스캐너) 수행



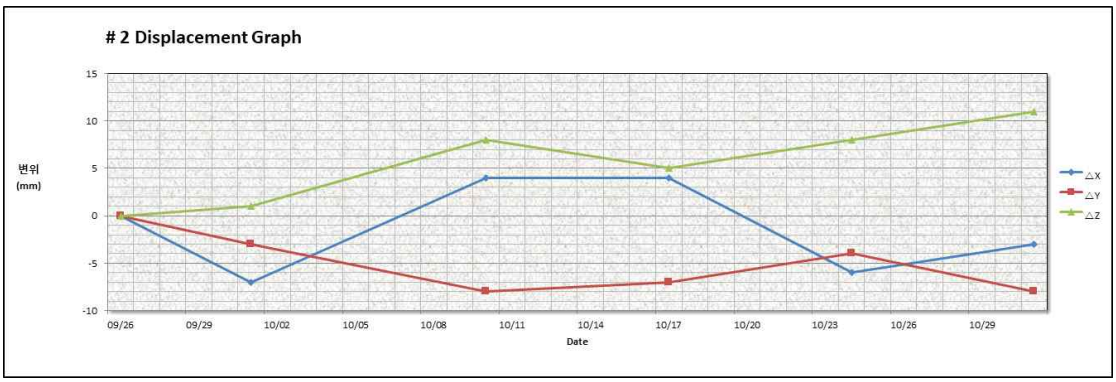
[스마트폴 변위 측정 및 검증 계획 모식도(광명 철산)]



[레이저 스캐너 계측 자료]



[스마트폴 #2 기울기센서 계측 결과 - 1회/분 측정]



[스마트폴 #2 광파기 타겟 외부 계측 결과]

- 스마트폴 자체 기울기 센서 계측 결과(폴 높이 1m일 때 각변위 0.5° 시 8.19㎜ 변위량)와 설치위치와 떨어진 위치에서 측정한 광파기 타겟 계측 결과는 8㎜내 변위로 유사한 경향을 보이고 있음

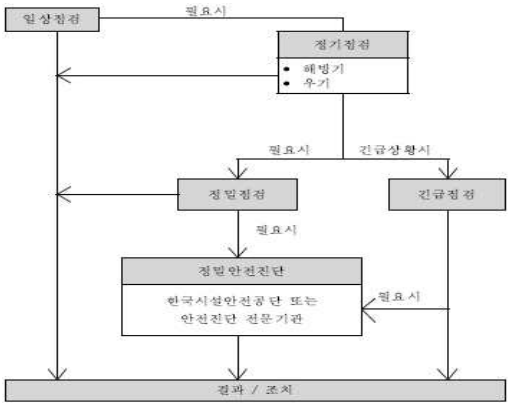
◯ 기존 비탈면 지침 조사 분석

- 절토사면 유지관리 매뉴얼 보완 · 개정 연구(2013.12), 한국시설안전공단
  · 지표면에 설치된 계측기기로 지표변위량 계측으로 설치 위치, 수량 등 계측 관리 한계 있음
  · 전문가 육안조사시 접근 불가 위치, 소요 시간, 전문인력 등 점검 한계 있음
  · 급경사 또는 대절토 비탈면의 경우 식생, 경사에 의해 비탈면 지표면, 비탈면 상단 측량이 불가함
- 건설공사 비탈면 설계기준(2011.12), 국토해양부
  · 지상 계측, 계측기기에 의한 계측 한계에 대한 언급이 없지만 계측목적에 부합하는 계측기 선정이 중요함을 언급함
- 철도설계지침 및 편람 : KR C-02050 계측 (2017.3), 한국철도시설공단
  · 지표면에 설치된 계측기기로 계측, 사진 촬영 측정 등 설치 위치, 수량 등 계측 관리 한계가 있으며,식생에 의한 육안, 지표 계측 관리 한계도 있음

◯ 비탈면 시공 또는 유지관리시 기존 방법으로는 육안조사, 지표면에 설치된 계측기기 지점만 점검, 계측 관리되는 한계점을 가지고 있음. 본 연구 기술로는 대절토 비탈면, 급경사지, 식생이 많은 곳, 계측기가 설치되지 않은 지점 등에서 발생되는 지표면 변위를 감지하여 비탈면 붕괴 조짐을 사전에 알 수 있음

4.4 절토사면의 계측관리

: 지표 변위량은 지표면 신축계, 지표면 경사계, 광파측량기 등



· 제5장 절토사면의 점검 및 진단 : 정기점검, 정밀점검, 긴급점검, 정밀안전진단으로 나뉨

: 정기점검 - 반기에 1회 이상 실시, 세심한 육안검사 수준의 점검으로서 간단한 점검장비(카메라, 필리도구, 줄자, 망치 등)

: 정밀점검, 긴급점검, 정밀안전진단 : 근접조사를 위한 접근장비 사용 가능, 측량기 사용, 외관조사

5.3 현장조사 및 시험법

: 절토사면 측량 : 외관 조사를 실시한 결과, 상태 평가 또는 안정성 평가를 위해 정밀한 사면형성이 필요한 경우 사면에 대한 측량을 실시

: 광파거리계에 의한 측량기를 사용하며 측정간격은 20~100m 범위에서 사면의 형상이나 손상정도에 따라 위치를 선정하여 측량

: 측점별 횡단 측량은 보통 20m마다 1개소씩 측량하며, 단면 변화구간 및 육안조사 단면의 이상부위는 보다 정확한 측량을 실시

5.8 깎기비탈면의 계측

: 계측계획 수립 : 비탈면의 변형이나 붕괴형태를 사전에 예상하고 계측목적에 부합하는 계측기 선정이나 배치, 계측방법, 관리기준치 설청 등을 검토하여 계측계획 작성

: 계측기준 : 계측기의 배치는 비탈면의 붕괴 및 활동 특성, 지형적 위치, 계측기 설치 편의성, 계측기의 관리 편의성, 비용 등을 고려하여야 하며, 비탈면의 변동 상황을 최소화의 계측기로 효과적으로 파악할 수 있도록 배치하여야 한다.

2. 깎기 비탈면 계측관리

: 비탈면의 안정성을 주의 깊게 관찰하기 위해 필요한 개소에 계측을 수행하도록 설계해야 한다.

: 암반 비탈면 변위에 대한 계측방법은 아래와 같은 방법들이 있다.

. 단순한 기계적 변위측정 : 지표의 암반변위를 직접계측

. 광학적 측정 : 광학사진 촬영

. 측지학적 측정 : 기준선, 거리측량, 삼각법 측량

. Extensomenter : 2점 사이의 길이 변화를 측정

. Deflectometer : 시추공의 휨 변형량 측정

. 기울기 측정 : 기울기 변화 측정

나. 드론을 활용한 포트홀 탐지 기술 개발

(1) 드론 영상 데이터 기반 포트홀 탐지 소프트웨어 개발

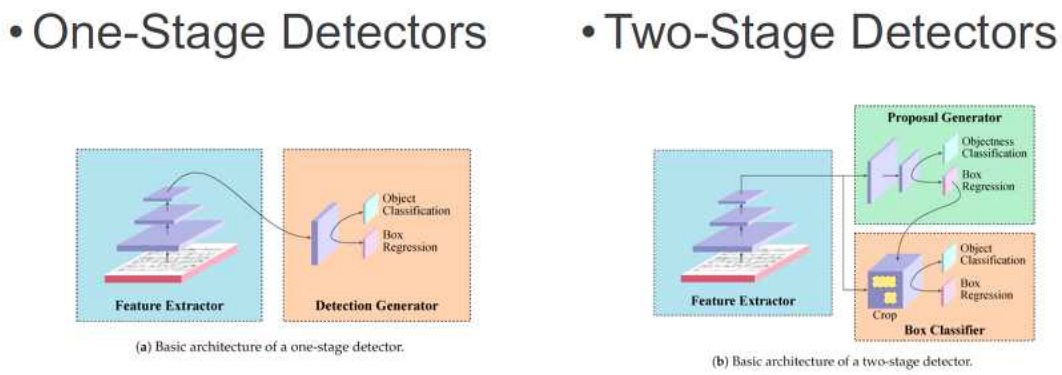
○ 딥러닝 기반 객체 탐지 관련 연구

- 인공신경망에서의 이미지 분류 및 객체 탐지 방식
  · 시각적 영상 분석에 주로 적용되는 합성곱 신경망을 기반으로 구성되며, 분류, 객체 탐지, 객체분할에 사용
- COCO 데이터셋 기반 딥러닝 모델 성능지표

[2017 Object Detection on COCO test-dev(paperwithcode, n.d)]

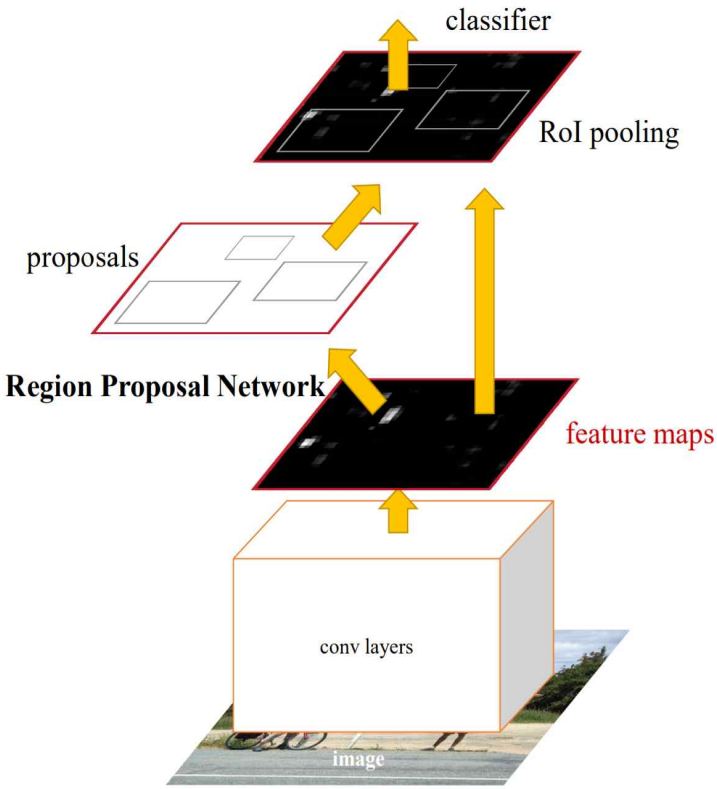
Model	BOX AP	AP 50	AP 75
Mask R-CNN(ResNet-101-FPN)	39.8	62.3	43.4
Faster R-CNN(ImageNet+300M)	37.4	58	40.1
RefineDet512(VGG-16)	33	54.5	35.5
SSD512	28.8	48.5	30.3
DeformConv-R-FCN(Aligned-Inception-ResNet)	37.5	58.0	-
YOLOv3 + Darknet-53	33.0	-	-
YOLO v2 + Darknet-19	21.6	-	-

- 1-stage Detector and 2-stage Detector 관련 설명
  · 객체 탐지모델은 물체의 위치를 찾는 Localization과 물체를 구별하는 분류가 필요
  · 1-stage Detector는 Localization과 분류를 동시에 수행하여 결과를 얻는 방법
  · 2-stage Detector는 Localization과 분류를 순차적으로 수행하여 결과를 얻는 방법
  · 대체로 1-stage Detector는 빠르지만, 정확도가 2-stage Detector에 비해 낮으며, 2-stage Detector는 1-stage Detector보다 느리지만, 정확도가 높다.



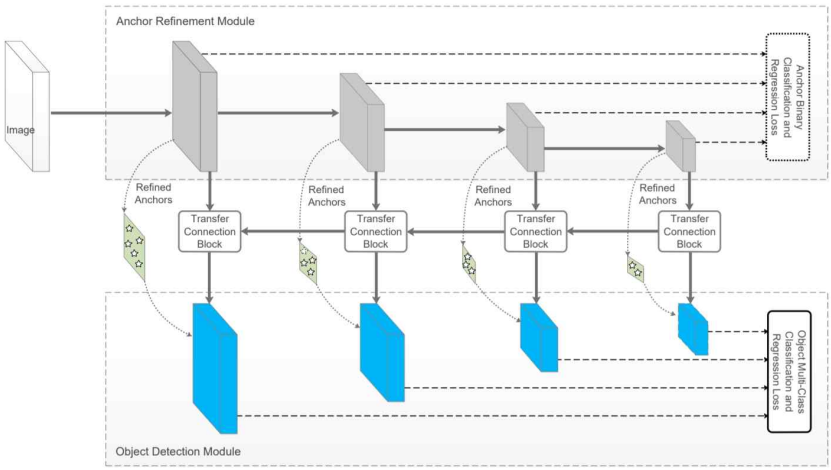
[1-stage Detector와 2-stage Detector 구조도(Pacha, 2018)]

- Faster R-CNN
  · 영상 내 객체를 추출하고 해당 객체를 분류하는 객체 탐지모델
  · 기존 Fast R-CNN과 R-CNN에서 객체 추출을 위해 사용하던 selective search 알고리즘을 제거하고 RPN(Region Proposal Network)을 통해서 RoI를 계산한다. 기존 Fast R-CNN과 R-CNN에 비해 더 고속의 처리속도를 보임
  · 2-stage detector로 region proposal을 하는 RPN과 classification 층이 구분되어 있음



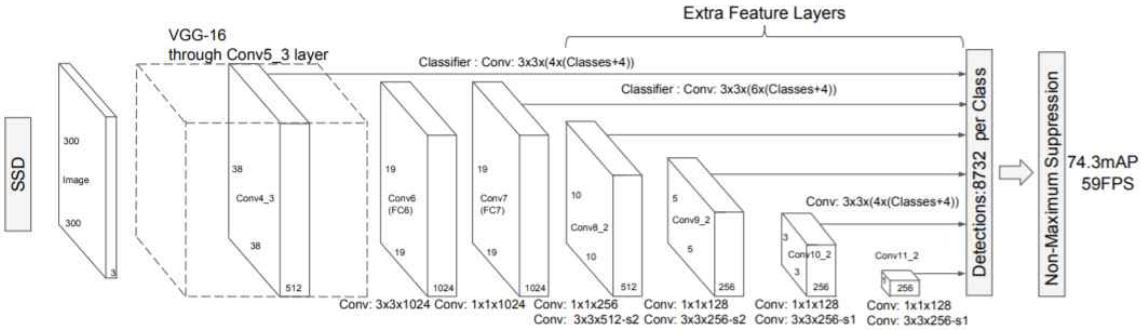
[Faster R-CNN 아키텍쳐(Ren, S., 2015)]

- RefineDet512(Refinement Neural Network for Object Detection)
· 두 개의 상호 연결된 모듈(ARM(Anchor Refinement Module), ODM(Object Detection Module))으로 구성된 1-stage 객체 탐지 프레임워크 구성
· 기존의 2-stage detector보다 정확하면서, 기존의 1-stage detector의 효율성 유지



[RefineDet 아키텍쳐(Zhang, S, 2018)]

- SSD(Single Shot Multibox Detection)
· 이미지 내 객체 탐지의 목적은 이미지 내의 객체의 위치 정보와 클래스 정보를 찾아내는 것
· Single-Shot detector는 말 그대로 사진의 변형 없이 한 장으로 학습과 검출을 하는 검출기를 의미
· convolution 과정 간에 생성되는 여러 크기의 특징맵을 이용하여 작은 물체와 큰 물체 모두 분류 가능



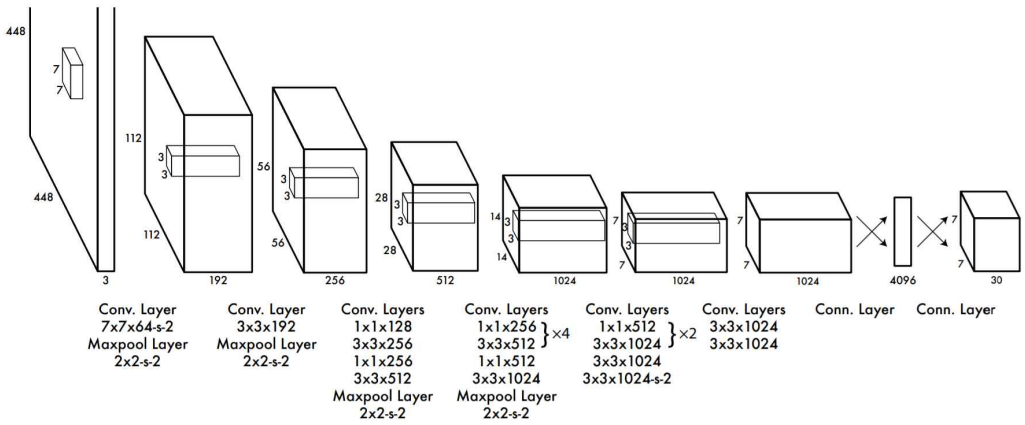
- YOLOv2(You Only Look Once)

· YOLO는 기존 모델들보다 더 높은 정확도를 추구하는 것이 아닌, 근접한 정확도를 가지면서 더 많은 양의 이미지를 처리할 수 있는 실시간 객체 탐지를 목적으로 함

· YOLOv2는 기존 YOLO모델에 정규화, Direct Location Prediction 등의 기법을 도입하여 FPS와 mAP를 높임

- YOLOv3(You Only Look Once)

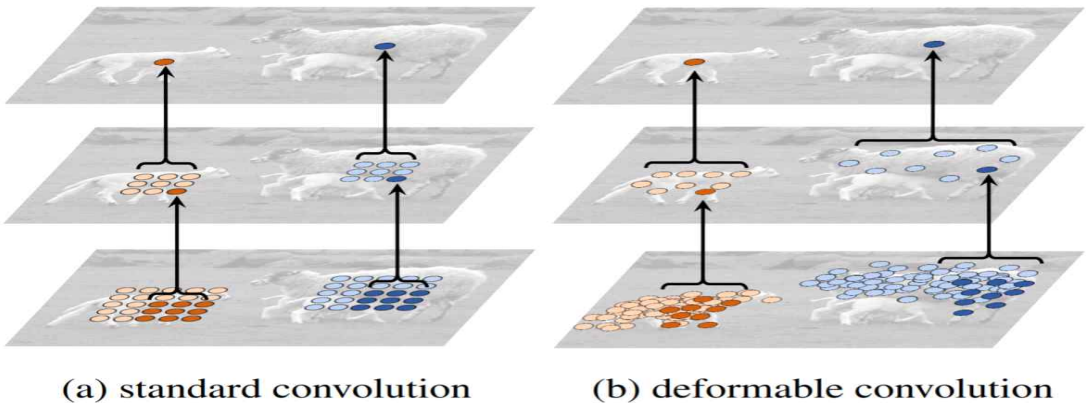
· YOLOv3는 YOLO v2에서 사용하던 백본망인 Darknet-19에 ResNet에서 제안된 skip connection 개념을 적용하여 Darknet-53으로 변경



[YOLO 아키텍쳐(Redmon, J., 2016)]

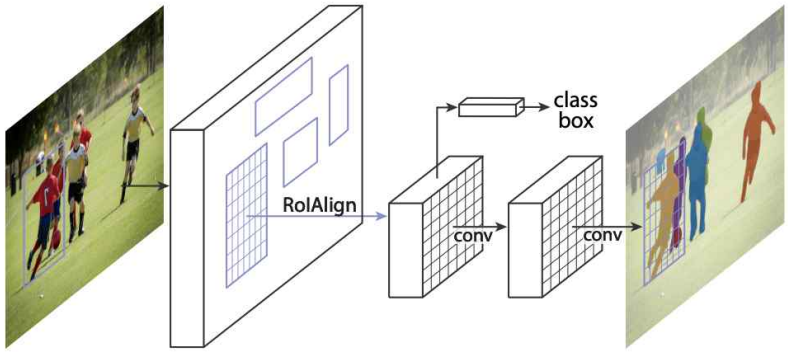
- DeformConv(Deformable Convolutional Network)

· 기존 CNN에서 사용하는 여러 연산(convolution/ pooling/ roi pooling 등)이 기하학적으로 일정한 패턴을 가정하고 있으므로 복잡한 변환에 유연하게 대처하기 어렵다는 한계가 존재한다. 위의 한계를 넘기 위해 만들어진 모델



[convolution 연산과 deformable convolution 연산 예시(Dai, J., 2017)]

- Mask R-CNN
· Faster R-CNN에서 객체 영역을 예측하는 객체분할이 추가된 모델
· Mask R-CNN은 2-Stage detector로 RPN(Region Proposal Network)을 통해 객체를 예측한 후 객체의 클래스와 영역을 예측
· 1-Stage detector인 YOLO(You Only Look Once), SSD(Single Shot Detector)보다 처리속도가 느린 것이 단점이지만 정확도는 비교적 높은 특징



[Mask R-CNN 아키텍쳐(He, K., 2017)]

- COCO(Microsoft Common Objects in Context) 데이터셋
· COCO 데이터셋은 객체탐지와 객체분할 등을 위한 데이터 셋으로 328k 이미지로 구성
· 객체 탐지 데이터셋의 경우 80개의 클래스로 이루어짐

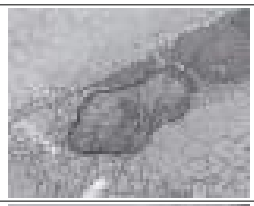


[COCO 데이터셋 예시(COCO dataset, n.d.)]

○ 데이터증강 관련 연구
- 데이터증강 기법은 학습하려는 데이터셋의 양이 부족하거나 클래스 불균형 때문에 데이터의 양을 늘려야 할 경우 사용되는 기법
- 데이터증강 기법

[데이터 증강 기법]

증강기법	설명	예시 이미지	증강기법	설명	예시 이미지
Original	원본영상		Cartoon	Cartoon Filter 이용 원본 영상 스타일 변환	

Additive Gaussian Noise	가우시안 분포로부터 영상내 요소에 잡음 추가		Canny	영상에 Canny edge detector 적용 후 영상과 edge간 무작위로 혼합된 영상 생성	
Cut out	영상 내 무작위 영역에 회색 픽셀 채우기		Enhance Contrast	영상 대비(contrast) 강화	
Flip ud	영상 상하 반전		Superpixels	영상 전체 또는 부분적으로 슈퍼픽셀화	
Fliplr	영상 좌우 반전		Clouds	영상 내 구름 추가	
Affine	영상 평행이동, 확대 및 축소, 등의 기하학적 변환		Fog	영상 내 안개 추가	
Rotate	영상 0~360‘ 회전		Rain	영상 내 비 추가	
AverageBlur	영상 내 각 픽셀 강도 주변 픽셀 값들의 평균 적용				

○ 포트홀 자동탐지 결과 관리시스템 관련 연구

– 웹프레임워크

· 개념 : 웹 어플리케이션, 웹서비스 개발 보조용으로 만들어진 어플리케이션 프레임워크이다. 웹 페이지를 개발하는 과정에서 개발자가 겪는 어려움을 줄이는 것이 주 목적으로 통상 데이터베이스 연동, 템플릿 형태의 표준, 세션 관리, 코드 재사용 등의 기능을 포함

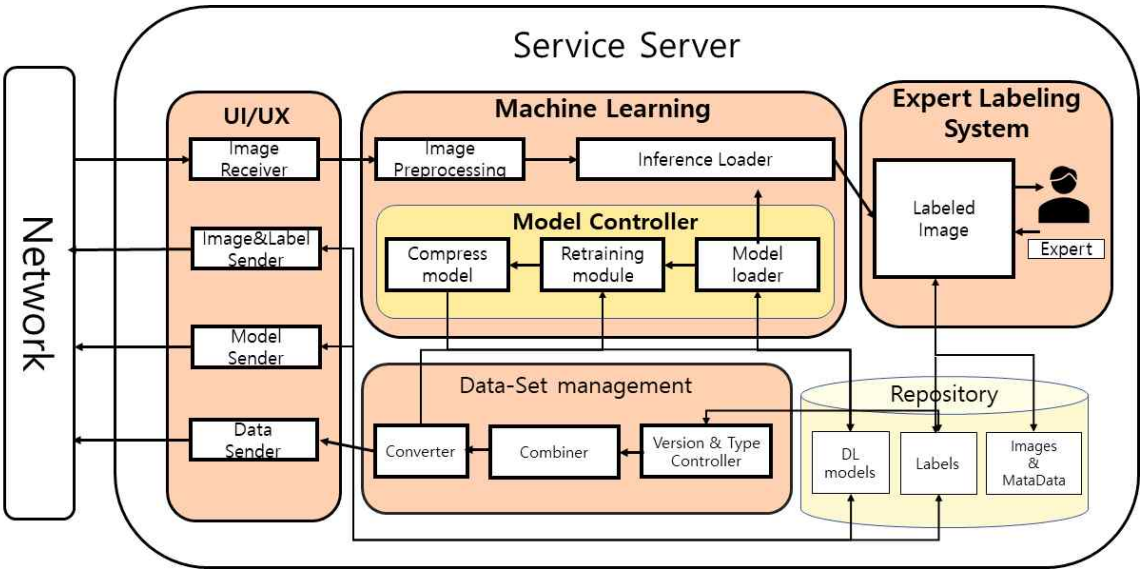
· 활용 : 포트홀 자동탐지 결과 관리시스템 구현을 위한 웹 프레임워크 선정을 위한 분석

· 연구 및 실험 내용 : Django와 Rails, CakePHP 비교 분석 진행, Django는 파이썬 기반의 웹 프레임워크이기 때문에 keras, tensorflow와 호환성이 좋으며 html을 통한 간단한 유저 인터페이스 구현이 가능하며, 데이터베이스를 직접 관리 가능

· 결과 : 본 연구의 웹 프레임워크로서 장고(Django)를 사용

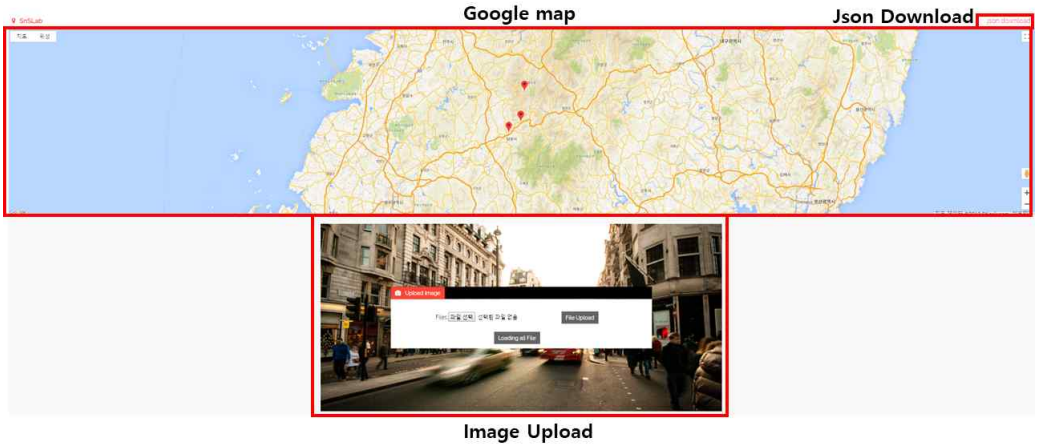
- 장고(Django)
· 개념 : 파이썬으로 작성된 오픈소스 웹 프레임워크로, 모델-뷰-컨트롤러(MVC)패턴을 따르고 있다. 고도의 데이터베이스 기반 웹사이트를 작성하는 데 있어서 수고를 더는 것이 장고의 주된 목표
· 활용 : 포트홀 자동탐지 결과 관리시스템 구현을 위한 웹 프레임워크로 사용
· 연구 및 실험 내용 : 포트홀 자동탐지 결과 관리시스템 구현
· 결과 : 본 연구의 웹 프레임워크로서 사용

○ 도로 항공 영상에서 포트홀 탐지 결과 관리시스템 구현



[관리시스템 아키텍처]

- MachineLearning : 딥러닝 기반 포트홀 탐지 알고리즘, 6개의 클래스별로 결과 출력
- Expert Labeling System : 탐지 결과를 전문가를 통해 검증
- Data-Set Management : 결과데이터 백업을 위해 버전별 관리
- Repository : 각 결과 및 데이터를 파일 형태별로 저장, 각 모델을 버전별로 관리하여 저장
- UI(User Interface)/UX(User eXperience) : 사용자의 입력을 받아 각 모듈로 전송, 결과데이터 사용자에게 전달
- 메인페이지 : 이미지 업로드, 이미지 GPS 기반 구글 지도 표시, 딥러닝 결과 다운로드 가능
- 타일 리스트 페이지 : 이미지 내 딥러닝 결과 표시, 각 클래스별 탐지 결과 타일 리스트 출력, 결과데이터 다운로드 가능, 타일 리스트 내 오탐 데이터 삭제 통해 데이터 셋 정제 가능

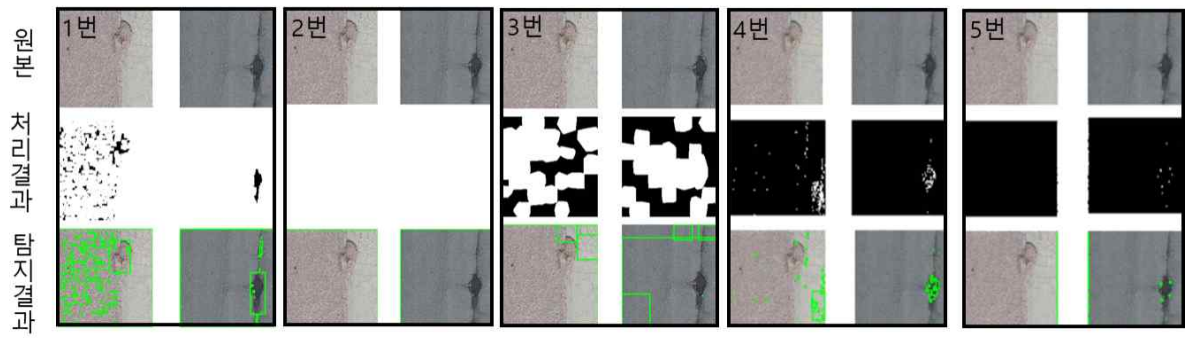


[메인 페이지]



[타일리스트 페이지]

○ 영상처리 기반 원본 이미지 내 포트홀 탐지 알고리즘 개발
- 영상처리 기반 포트홀 특징 도출 정확도 측정
  · 1번 : 오츠 이진화
  · 2번 : 슈퍼픽셀 + WEF
  · 3번 : 슈퍼픽셀 + 컨벡스 헐
  · 4번 : 배경제거(MOG)
  · 5번 : 세일리언시 맵



[상처리 결과 이미지]

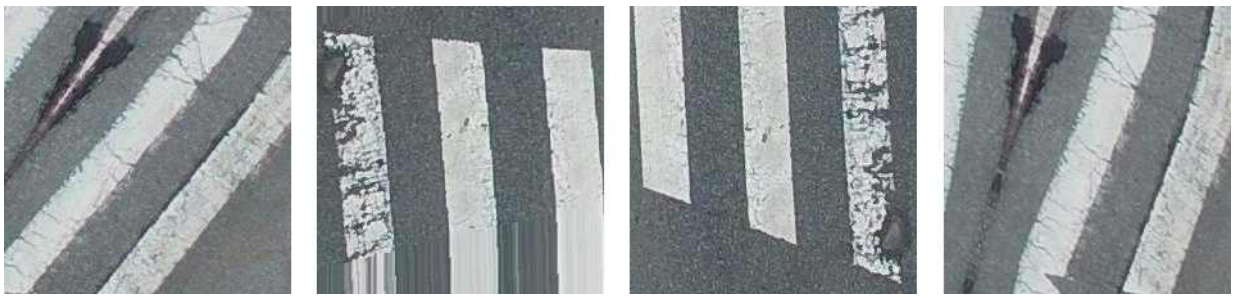
○ 딥러닝 기반 원본 이미지 내 포트홀 탐지 알고리즘 구현
- Inception v3 기반 포트홀 탐지 알고리즘 개발
  · 영상처리 기반 포트홀 탐지 알고리즘 보다 높은 검지율, 낮은 오탐율
  · 이미지를 처리 시 여러 크기의 필터 복합적으로 사용해 이미지의 특징을 더 잘 잡아냄
  · 컨볼루션 레이어(ConvolutionLayer)의 매개변수 연산 감소
  · 기존 ResNet 대비 높은 탐지율 확보 및 MobileNet 대비 낮은 오탐율 확보

○ 항공 영상기반 딥러닝용 데이터 셋 구축
- 1차년도 대비 데이터 증가 및 분류 세분화
  · 오탐 발생 가능 데이터 추가 확보 구축을 통한 낮은 오탐률 확보
  · 신규 포트홀 이미지 추가 및 데이터증강을 통한 포트홀 이미지 확보
  · 데이터 분류 단계 세분화 (총 6가지 분류 학습 – 포트홀, 도로, 비도로, 맨홀, 그림자, 도로 노면 표시)
  · 1차년도 포트홀 328개 대비 2,596개 포트홀 데이터 셋 확보 (약 1,000,000개 이미지 데이터 확보)

· 오탐 유발 특이 사례 분석을 통한 데이터 배제 (도로 노면 표시상 존재하는 포트홀 단 2건) → 3차년도 데이터 확보 계획 수립



[사용 데이터증강 기법]



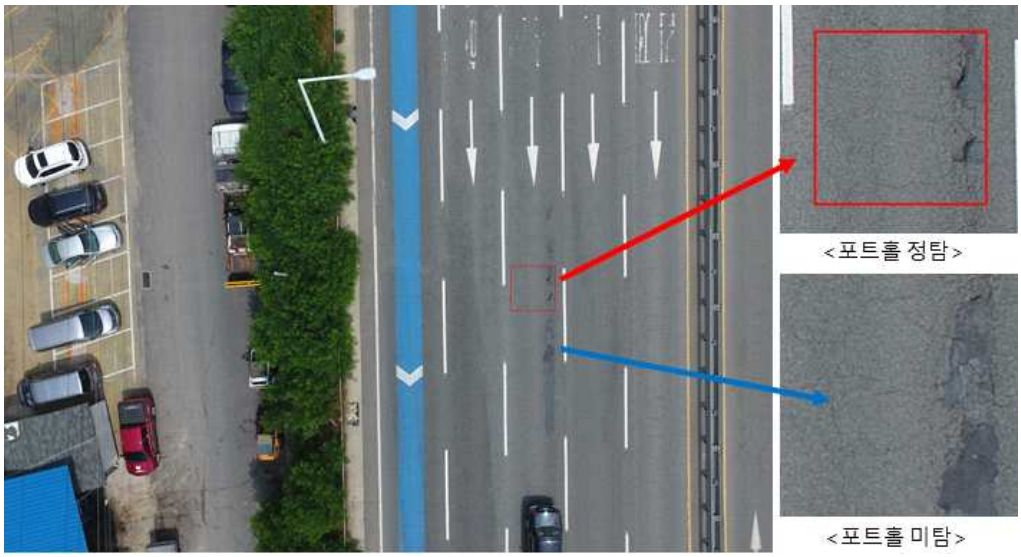
[횡단보도 내 포트홀 예시]

○ 항공 영상 내 딥러닝 탐지 결과 분석

- 포트홀 검지율 : 탐지 포트홀 수 / 포트홀 수 [TP/(TP+FN)]
- 포트홀 오검지율 : 오탐지 횟수 / 포트홀 탐지 횟수 [FP/(TP+FP)]
- 포트홀 검지율 : 78.78%
- 포트홀 오검지율 : 18.30%



[항공 이미지 내 포트홀 정탐 및 오탐 샘플]



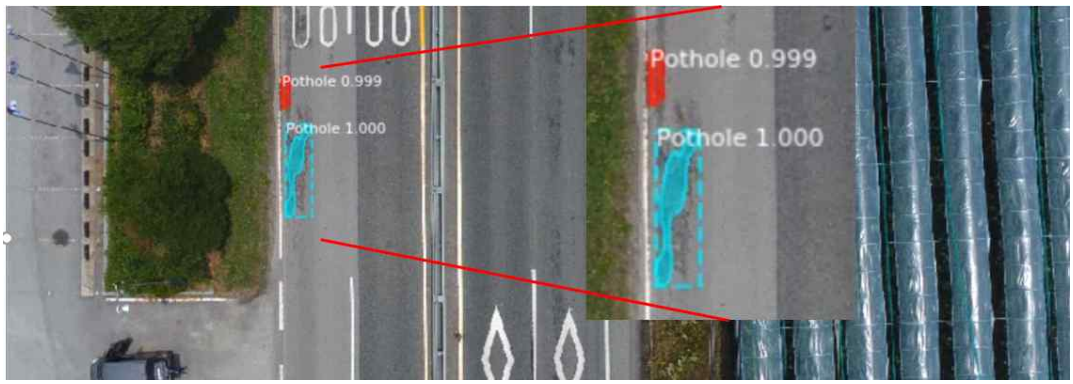
[ ]항공 이미지 내 포트홀 정탐 샘플

○ 1차년도 오탐 사례와 2차년도 정탐 사례 비교
- 1차년도 정탐율 92% 및 오탐율 29.9%
- 1차년도 대비 정탐율 20% 및 오탐율 11% 감소됨.
- 포트홀 특징 강조를 위한 데이터 셋 검증 및 수정을 통한 오탐율 감소 확인
- 1번, 2번 오탐 : (미세)크랙을 포트홀로 오탐
- 3번 오탐 : 노후화된 도로 노면 표시를 포트홀로 오탐
- 4번 오탐 : 도로 소성변형 및 표면 오염을 포트홀로 오탐
- 5번 오탐 : 도로 노면 표시상 물체를 포트홀로 오탐



[포트홀 오탐 사례]

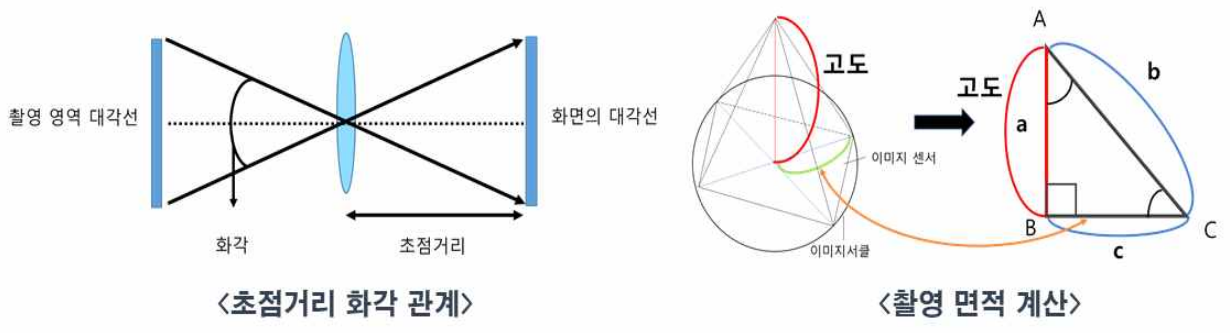
○ 항공 영상 내 포트홀 면적 계산 알고리즘 구현
- Instance Segmentation 기반 포트홀 오탐율 감소 테스트 및 면적 계산
  · Mask R-CNN(Region-Convolution Neural Network)을 통한 포트홀의 영역 추출 및 면적 계산
  · 총 134개의 테스트 데이터 셋 구축



[Semantic Segmentation 결과]

○ 포트홀 면적 계산 알고리즘 개발

– 영상 EXIF(EXchangable Image File format) 내 초점거리와 이미지 센서 및 촬영 고도를 이용해 픽셀 단위 면적을 계산 후 포트홀의 실면적 계산



[포트홀 면적 계산 관련 그림]

– 포트홀 실면적 측정을 위한 픽셀 단위 면적 계산

[계산 공식에 대한 기호 설명]

약호, 기호	설명	단위
d	센서대각선 길이	mm
f	초점거리	mm
a	고도	m
c	촬영 영역의 대각선	m
A	화각/2	m
C	A의 여각	°

· 객체의 실면적을 계산하기 위한 화각 계산 공식은 (1)과 같음, 삼각함수의 역탄젠트를 사용하여 초점거리(f)와 센서대각선길이(d)를 통해 화각(A)을 구함

$$A = \tan^{-1}\frac{d}{2\times f} \qquad (1)$$

· 촬영 영역의 대각선 길이(c)를 구하기 위한 계산 공식은 (2)와 같음, 삼각함수의 탄젠트를 사용하여 고도와 A의 여각인 C의 각도를 기반으로 계산

$$c = a\times\frac{1}{\tan C} \qquad (2)$$

· 이미지 센서의 대각선 길이(d)와 실제 촬영 영역의 대각선 길이(c)간의 비례식을 통해 촬영 영역의 가로와 세로길이 산출

· 영상 해상도 크기와 촬영 영역의 크기 간의 비례식을 통해 1pixel의 면적 산출 후, 포트홀 마스크의 픽셀 수에 곱하여 포트홀 실제 면적 계산

(2) 포트홀 탐지 영상제작 및 자료수집

○ 포트홀 상시 데이터 수집

- 드론 활용 AI학습용 DB구축 체계 정립



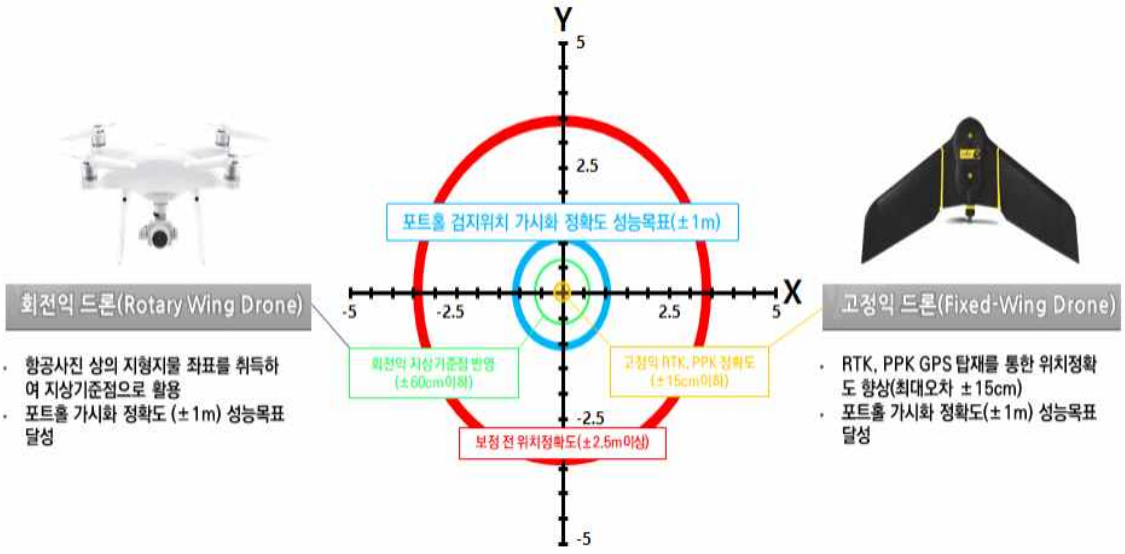
- 포트홀 AI학습용 DB구축, 총연장 105km 드론촬영 수행

- 포트홀 영상 61개 확보



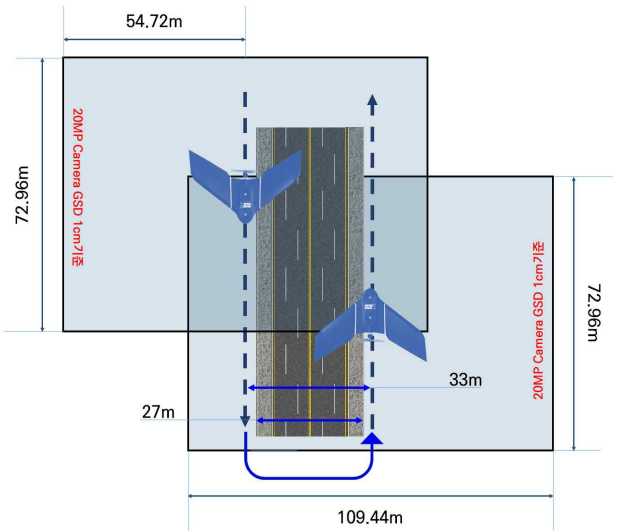
순번	촬영일/폴더명	비행시간	사진매수	노선번호	지역	포트홀수
1	190422	6	85	19	전라북도 남원시 식정동 55-1	2
2	190422	5	79	19	전라북도 남원시 산동면 대기리 1068	1
3	190422	5	79	19	전라북도 장수군 장수읍 개정리 659-3	1
4	190422	14	251	만선로	전라북도 무주군 설천면 심곡리 산47-1	10
5	190422	4	54	743	전라북도 장수군 번암면 유정리 산236-8	1
6	190701	16	299	17	대전광역시 동구 삼괴동 129	4
7	190701	6	82	4	충청북도 옥천군 이원면 용방리 570-2	1
8	190701	6	87	4	충청북도 옥천군 이원면 용방리 1203-1	2
9	190701	7	90	35	대전광역시 동구 구도동 111	2
10	190702	8	139	대불로	전라남도 영암군 삼호읍 난전리 1692-1	1
11	190702	4	50	나불로	전라남도 영암군 삼호읍 난전리 1715-2	1
12	190702	7	93	대불산단5로	전라남도 영암군 삼호읍 난전리 1712-2	2
13	190702	3	45	용앙로	전라남도 영암군 삼호읍 난전리 1702-1	2
14	190702	4	52	용앙로	전라남도 영암군 삼호읍 난전리 1705-3	14
15	190702	8	117	용앙로	전라남도 영암군 삼호읍 난전리 1707-9	1
16	190723	10	188	2	전라남도 순천시 야흥동 143-15	3
17	190723	5	66	2	전라남도 보성군 벌교읍 칠동리 1293-4	2
18	190723	9	125	빛고을대로	광주광역시 북구 용두동 1043-1	1
19	190723	4	60	주암호길	전라남도 순천시 주암면 광천리 516-1	1
20	190723	4	48	여순로	전라남도 순천시 해룡면 호두리 산195-6	1
21	190723	4	54	여순로	전라남도 순천시 해룡면 호두리 465-1	2
22	190723	4	47	여순로	전라남도 순천시 해룡면 호두리 772-5	1
23	190724	6	90	백운로	전라남도 광양시 중군동 산88-17	1
24	190724	4	54	백운로	전라남도 광양시 광양읍 죽림리 539-5	1
25	190724	5	68	백운로	전라남도 광양시 광양읍 사곡리 1123-29	1
26	190724	4	52	백운로	전라남도 광양시 광양읍 용강리 142-2	1
27	190724	5	69	15	전라남도 고흥군 점암면 연봉리 433-5	1
총계						61

- 항공사진 지상기준점 활용으로 포트홀 가시화 정확도 성능목표 달성(목표 1.0m, 달성 0.6m)

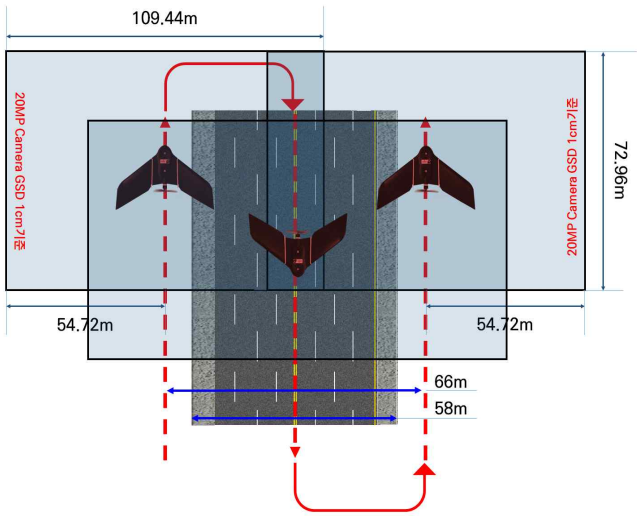


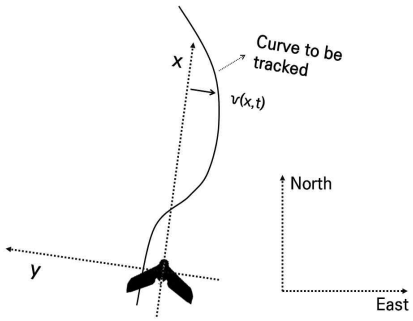
○ 안전고려, 접도구역 운항기법 도출
- 도로의 제원 및 차선별 폭에 대한 안전한 자동경로비행 도출
- 6차로 이하 2라인 경로비행, 6차로 초과 3라인 경로비행
- 6차로 초과 3라인 이상, 경로비행 중 비행모니터링 중요

1~8 차선도로 비행경로(2Line)



10~16 차선도로 비행경로(3line)





차선	도로 폭	길어깨 폭 포함	비행경로
1	3.25	없음	2
2	6.5	12.5	2
4	13	19	2
6	19.5	25.5	2
8	26	32	2
10	32.5	38.5	3
12	39	45	3
14	45.5	51.5	3
16	52	58	3

*길어깨 최소폭(도로의 구조,시설에 관한 규칙)

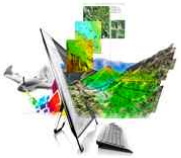
도로의 구분			차도 오른쪽 길어깨의 최소폭(m)		
			지방지역	도시지역	소형차도로
고속도로			3.00	2.00	2.00
일반도로	설계속도(km/h)	80이상	2.00	1.50	1.00
		60이상 80미만	1.50	1.00	0.75
		60이상	1.00	0.75	0.75

*차로의 최소폭 (도로의 구조,시설 에관한규칙)

도로의 구분			차로의 최소 폭(m)		
			지방지역	도시지역	소형차도로
고속도로			3.50	3.50	3.25
일반도로	설계속도(km/h)	80이상	3.50	3.25	3.25
		70이상	3.25	3.25	3.00
		60이상	3.25	3.00	3.00
		60미만	3.00	3.00	3.00

*터널,지하차도,교량,고가도로일 경우 0.5m로 할 수 있다.

○ 성과물 제작 표준화를 통한 영상처리 기술 고도화
- 대표적인 상용 드론 영상처리 S/W Pix4D사 Pix4D Mapper, Agisoft사 Metashape, Bentley사 Context Capture을 통해 데이터 처리시간, 품질 등 비교
- Pix4D Mapper의 경우 영상품질 보통, 영상처리시간과 압출율 우수, 사용성과 영상편집기능이 뛰어남
- Metashape의 경우 영상품질 우수, 영상처리시간과 압출율 보통, 사용성과 영상편집기능 보통
- ContextCapture의 경우 영상품질 매우우수, 영상처리시간과 압출율 나쁨, 사용성과 영상편집기능이 나쁨
- 모든 장단점을 고려, Pix4D Mapper가 포트홀 영상처리에 적합



테스트 컴퓨터 사양		테스트 데이터	
CPU	I7-5820K 3.3GHz	촬영면적	0.05㎢
GPU	GeForce GTX 980	카메라화소	20MP(5472*3648)
HDD	SSD : 256GB, HDD : 2TB	사진매수	299장
RAM	32GB	사진용량	2.33GB

구분		Pix4Dmapper	Metashape	Bentley CONTEXT CAPTURE
제조사		Pix4D	Agisoft	Bentley Systems
처리옵션		기본옵션	기본옵션	기본옵션
권장사양	CPU	쿼드코어 or i7이상	쿼드코어 or i7이상	Intel or AMD 1.0Ghz이상
	GPU	OpenGL 3.2 또는 2GB RAM이상	GTX980이상	NIVDIA or AMD OpenGL 3.2호환 그래픽카드
	HDD	SSD	1GB이상의 충분한 디스크 공간	2GB이상의 디스크 공간
	RAM	최소 8GB	최소 16GB	최소 4GB
처리속도		50분	1시간 40분	3시간 50분
정사영상 용량		1.26GB	2.78GB	3.98GB
프로젝트 용량		7.36GB 파일수 : 989	11GB 파일수 : 195	10.1GB 파일수 : 1,809

Pix4Dmapper



영상품질 : 적정
영상처리시간 : 빠름
영상압축률 : 높음

Metashape

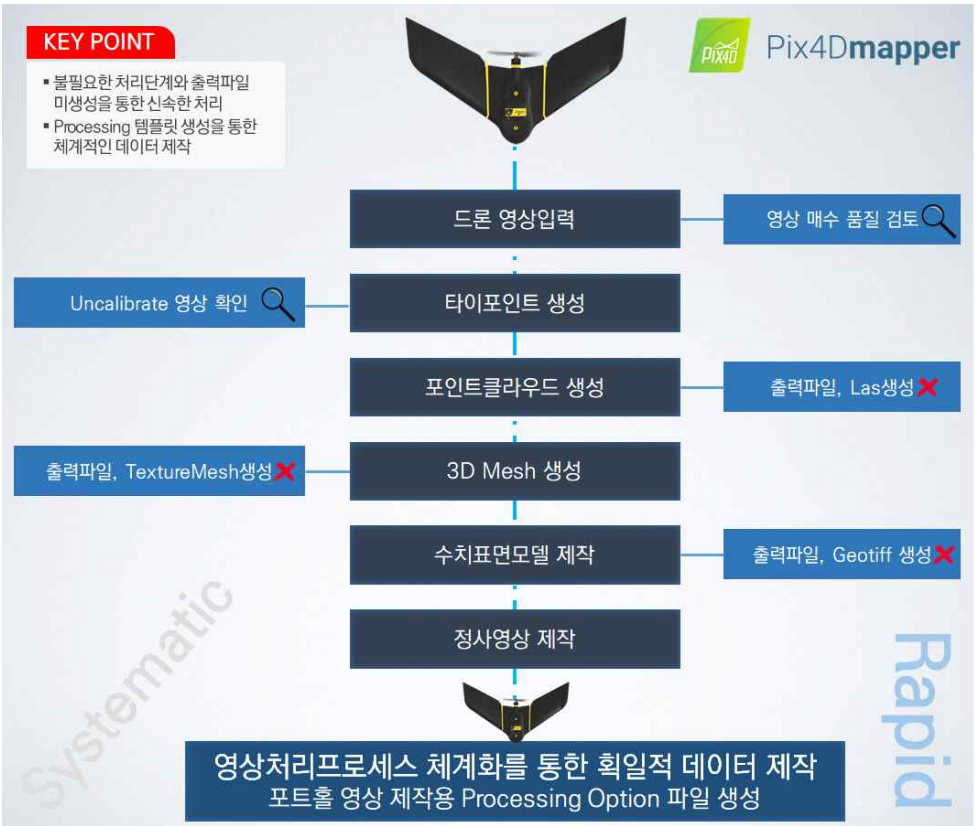


영상품질 : 높음
영상처리시간 : 보통
영상압축률 : 보통

Bentley CONTEXT CAPTURE



영상품질 : 높음
영상처리시간 : 느림
영상압축률 : 낮음



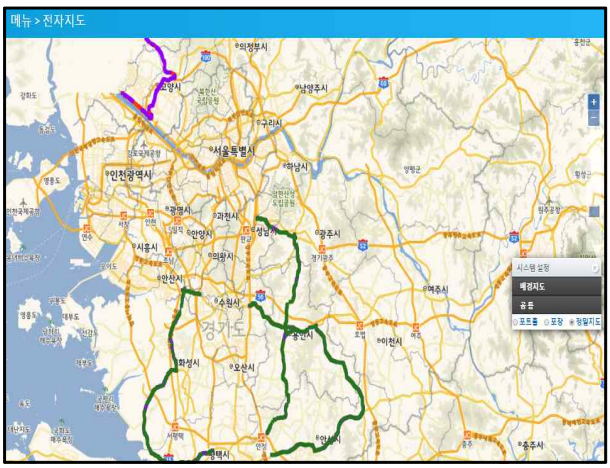
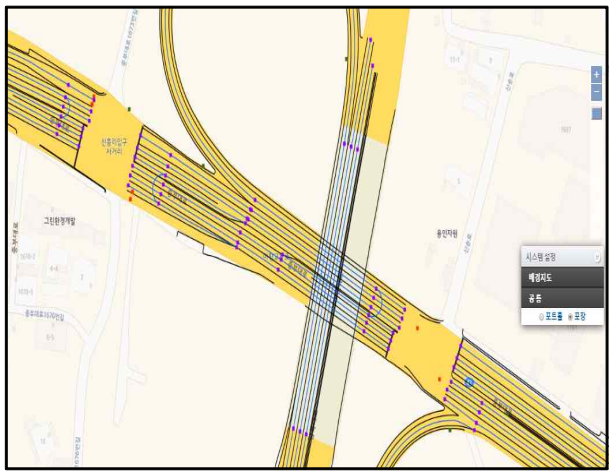
(3) 포트홀 탐지 결과 가시화(GIS/MMS맵)

○ 기술 개발 내용
- 포트홀 가시화 시스템 구축
- 도로 정밀지도 자료 가시화
- 서비스 운용 서버 구축, 운용편의성을 고려한 Devops환경 구성
- 포트홀 탐지결과 Json데이터처리 프로세스 설계/개발
- 포트홀 탐지결과 시스템 업로드 UI개발
- 포트홀 가시화 시스템 개발

○ 도로정밀지도 기반 포트홀 정보 가시화 연구
- 정밀지도 자료는 국토정보 지리원에서 다운로드하여 사용



· 국도1구간[국도 42호선 -39호선 - 77호선 -38호선(수원, 화성)] 약 6km
· 국도2구간[국도 42호선 -45호선(기흥, 용인)] 약 40km
· 국도3구간[국도 42호선 -17호선 - 38호선(용인, 안성)] 약 65km
· 국도5구간[국도 3호선 -45호선(광주, 용인, 성남)] 약 34km

정밀지도 (전체)	정밀지도 (확대)
	

· 정밀지도에서 차선, 중앙선, 정지선 정보를 표출
· 추가적으로 드론 촬영 지역의 정밀지도 자료를 확보

- 검지된 포트홀의 오차범위 1m내로 지도 API 연동
· 드론 촬영 영상의 중심 GPS좌표 정보를 가지고 계산된 포트홀위치에 대한 GPS좌표 정보와 정밀지도 자료를 연동
· 포트홀의 위치정보는 드론의 GPS좌표정보를 지도에 출력함

○ 포트홀 가시화시스템 Web Service 구축

- 서비스 운용을 위한 서버 구축

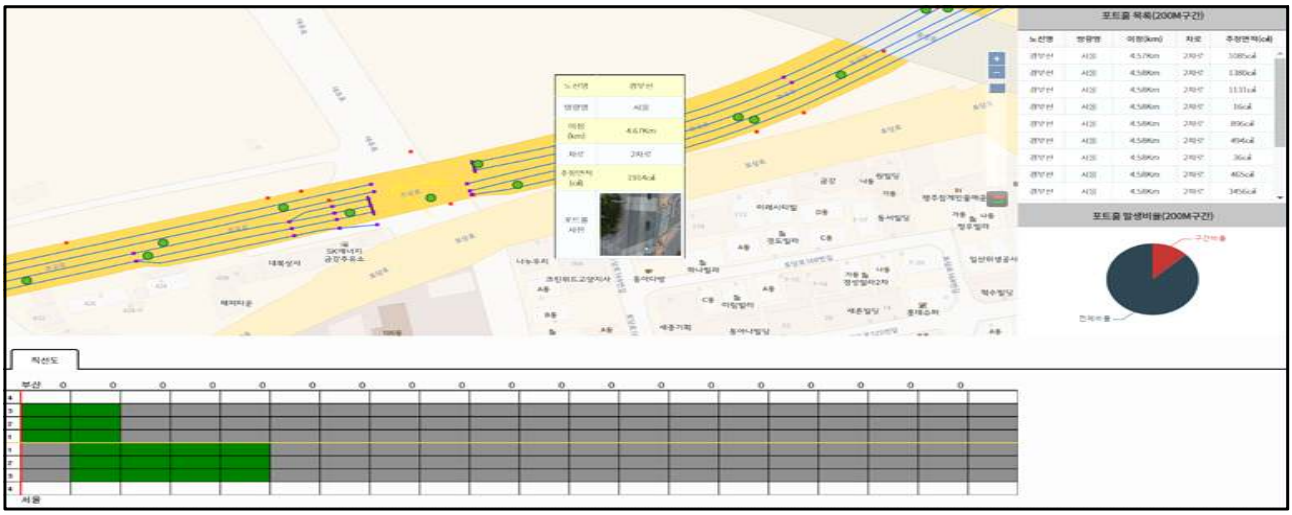
분류	프레임워크 & 라이브러리	비고
OS	CentOS	가시화 시스템 구축 OS
Framework	Spring Framework, Mybatis, Jquery	가시화 시스템 구성 Framework
WAS	Tomcat	Web application server
Database	Oracle xe 11g, Postgresql	
Devops	Jenkins	시스템 배포 자동화, 운영, 개발의 Devops시스템 구축
Etc	Geoserver	PCI 분석 Geotiff파일 가시화를 위한 웹서비스.

- 시스템 포트홀 데이터처리 프로세스 설계



1) 대구 가톨릭 대학교의 Json파일 시스템 연동 자료구조 분석
2) Json 데이터를 파싱하여 데이터 연동 모듈 개발
3) Json데이터 연동 모듈을 사용할 수 있는 User Interface설계
4) 연동 데이터 가시화 시스템 개발진행

- UI 구성



1) 포트홀 탐지 정보를 도로 구간별 포트홀 위치 표출 처리

- 지도 API에서 포트홀을 선택하면 도로 이정 정보를 중심기준으로 데이터 조회 범위 간격을 100m단위로 지정하여 포트홀정보 및 면적대비 발생 비율을 우측에 표출, 도로 노선모양의 직선도에 포트홀 발생빈도 및 크기 기준을 정하여 차선, 구간별로 상태정보를 하단에 표출함.
- 구간별 포출 시 지도 API의 도로정보에 대한 자료가 필요하며 드론 촬영은 위치 정보 이외 자료 생성의 어려운 문제가 발생함.

2) 포트홀 탐지 정보를 개별 포트홀 탐지 위치정보로 처리

- 기존 차량으로 진행하던 포트홀 조사방식의 경우 차량으로 조사 시 기록되는 도록 이정정보, 구간정보 등이 존재하지만 드론 포트홀조사의 경우 넓은 구간을 빠르게 탐지 가능하나 해당 정보가 존재하지 않기 때문에 드론탐지 포트홀 가시화시스템의 정보에서 제외됨.
- 드론 포트홀 탐지 데이터에서 추출 할 수 있는 드론의 촬영위치 GPS좌표를 기준으로 지도API에서 포트홀의 위치를 표출하는 기술 개발, 포트홀 탐지위치를 지도에서 선택 시 원본영상에 대한 포트홀 분석정보(우측), 드론 촬영영상 원본과 포트홀 분석영상(하단) 출력함.



다. 드론을 활용한 포장 노후화 상태(PCI) 측정 기술 개발

(1) 아스팔트 시편 및 코어 초분광 지수 통계 분석 모델 선정 및 개선

○ 초분광 실험결과의 분석을 위해 총 7종의 분석 모델식 선정
- PVISI는 1차년도 분석 모델을 개선한 PVIS2로 분석

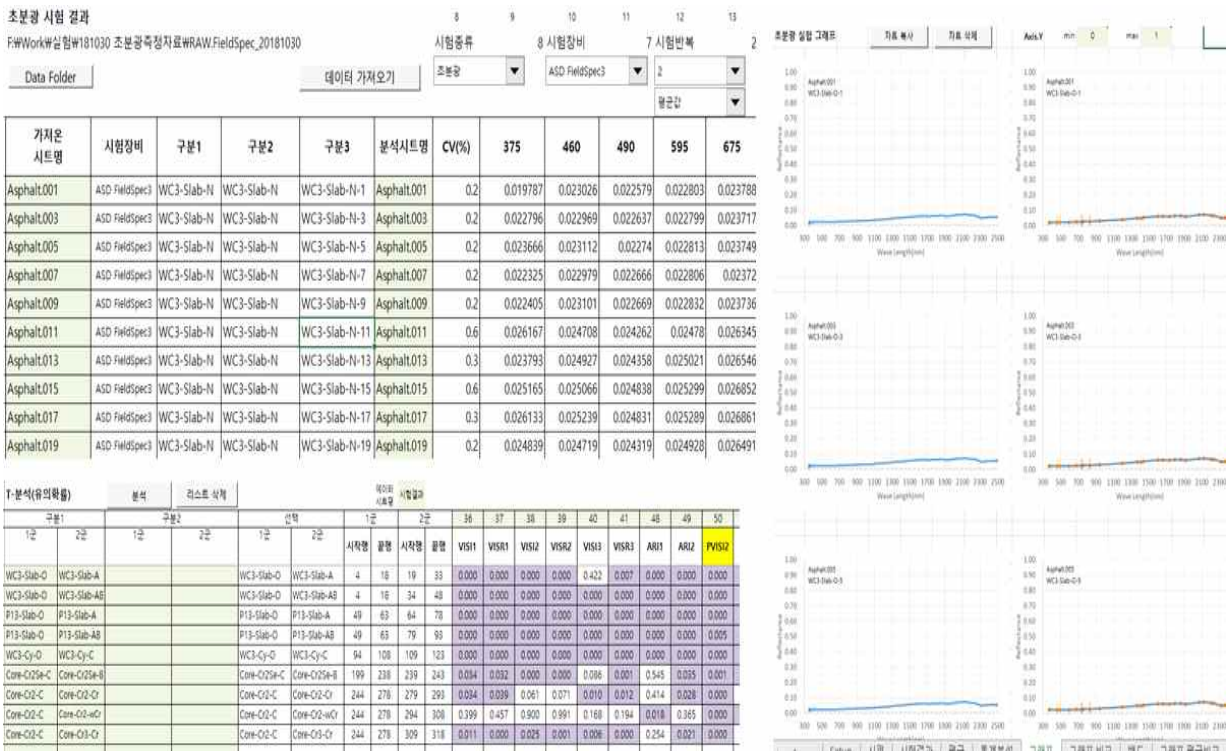
[초분광 분석 모델]

종류	모델식
VIS1(Visible Imaging Spectrometer 1)	$\mathrm{VISI1}=\left(\frac{\rho740-\rho460}{\rho740+\rho460}\right)\mathrm{VISR1}=\frac{\rho460}{\rho740}$
VIS2(Visible Imaging Spectrometer 2)	$\mathrm{VISI2}=\left(\frac{\rho830-\rho490}{\rho830+\rho490}\right)\mathrm{VISR2}=\frac{\rho490}{\rho830}$
VIS3(Visible Imaging Spectrometer 3)	$\mathrm{VISI3}=\left(\frac{\rho375-\rho595}{\rho375+\rho595}\right)\mathrm{VISR2}=\frac{\rho375}{\rho595}$
PVISI2(Pavement Visible Imaging Spectrometer Index 2)	$\mathrm{PVISI2}=\frac{\rho899}{\rho967}$

(2) 초분광 계측자료의 자동 분석을 위한 초분광 분석 엑셀 VBA 프로그램 개발

○ 프로그램 개요
- 초분광 자료 파일을 분석하여 대표 분광파장 대역, 분광지수(VIS1, VIS2) 등을 산출하고, 그래프를 출력

[초분광 분석 엑셀 VBA 프로그램]



○ 분석시트 설명
- [Setup] 시트 : 중요한 셋팅자료 (임의로 수정하면 안됨)
- [공시체물성] 시트 : 측정 파일을 불러올 경우 생성되는 시트명, 구분1, 구분2 등을 입력
- [시험결과] 시트 : 데이터 불러오기와 선택, 결과 출력 등을 위한 메인 시트
- [그래프] 시트 : 분석된 시트에 포함된 그래프를 모아서 출력

- [Form-HS3] 시트 : 초분광 자료 분석을 위한 샘플 시트 (샘플 시트 기준으로 자료가 분석됨)

○ 사용방법
- 분광계로 측정하여 자료파일(*.txt)을 임의폴더에 저장
- 엑셀 프로그램을 실행하며, 실행화면 상단의 '메크로 포함' 선택
- [시험결과]시트: 'Data Folder' 버튼을 클릭하여 분광계 측정 자료 파일이 있는 폴더 선택한 후 '확인' 클릭
- [시험결과]시트: 분광계로 반복측정했을 경우 반복횟수를 L열 콤보상자에서 선택 (예: 10회)
- [시험결과]시트: '데이터 불러오기' 버튼을 클릭하여 자료를 자동으로 엑셀로 가져옴 (신규 시트는 기존 시트의 끝에 자동으로 추가됨)
  ('Form-HS2'시트의 양식에 따라 자료의 파일명을 시트명으로 생성하여 데이터 저장)
- 자동 생성된 각 시트에 분광지수(VIS1, VIS2)가 자동으로 산출되며, 측정자료의 대표 분광 파장 대역의 반사율이 자동으로 산출됨. 또한, 같은 시료를 반복측정시 변동계수(CV)를 산출하여 분광계의 측정시 오류를 분석할 수 있고, 20%가 넘을 경우 붉은 색으로 표시됨
- [시험결과]시트: B열의 불러온 시트명을 참조하여 분석할 시트를 G열에 적거나 복사하여 붙여넣을 경우 자동으로 대표 분광 파장대역, 분광지수가 표시됨
- [공시체물성]시트의 시료명에 '시험결과' 시트 B열의 시트명과 구분내용을 적을 경우 '시험결과'시트의 '구분1'과 '구분2'에 반영됨
- [그래프]시트의 '차트복사' 버튼을 클릭시 '시험결과' 시트의 G열에 적은 시트에 포함된 그래프가 자동으로 표시됨
  (원본자료 그래프, 대표 분광파장+원본자료 그래프, 대표분광파장+변동계수(CV) 그래프 등 표시)

(3) 아스팔트 포장 노화 및 변색에 따른 유의확률 분석

○ 슬래브로 제작한 아스팔트 포장 시편의 초분광 측정값을 모델식에 따라 분석
○ 노화 전과, 노화 후 및 표면탈색 후의 초분광지수를 F-검정 후 이에 따라 등분산 또는 이분산 T-검정을 하여 유의확률을 평가
○ 분석 결과 VISI3를 제외하고 모두 분별력이 높음

[아스팔트 포장 시편의 노화에 따른 초분광 지수]

구분	VISI1	VISR1	VISI2	VISR2	VISI3	VISR3	PVISI2
WC3-Slab-N	0.059	0.888	0.120	0.786	0.024	0.996	0.926
WC3-Slab-A	0.022	0.957	0.073	0.864	0.031	1.061	0.949
WC3-Slab-AB	0.202	0.665	0.262	0.585	0.073	0.866	0.942
P13-Slab-N	0.015	0.970	0.064	0.880	0.094	1.210	0.929
P13-Slab-A	0.064	0.880	0.092	0.831	0.039	1.003	0.954
P13-Slab-AB	0.144	0.749	0.207	0.659	0.021	0.975	0.915

* WC3: 밀립도 아스팔트 혼합물, P13: 배수성 아스팔트 혼합물
  N: 일반, A: 강제노화, AB: 강제노화+표면탈색

[아스팔트 포장 시편의 노화에 따른 초분광 지수 유의확률]

구분		VISI1	VISR1	VISI2	VISR2	VISI3	VISR3	PVISI2
WC3-Slab-N	WC3-Slab-A	0	0	0	0	0.422	0.007	0
WC3-Slab-N	WC3-Slab-AB	0	0	0	0	0	0	0
P13-Slab-N	P13-Slab-A	0	0	0	0	0	0	0
P13-Slab-N	P13-Slab-AB	0	0	0	0	0	0	0.005

* WC3: 밀립도 아스팔트 혼합물, P13: 배수성 아스팔트 혼합물
N: 일반, A: 강제노화, AB: 강제노화+표면탈색

(4) 아스팔트 포장 균열에 따른 유의확률 분석

○ 도로에서 균열 발생 부분과 균열실링된 부분을 코어로 채취하여 측정한 초분광 측정값을 모델식에 따라 분석
- 균열이 없는 부분과 균열 발생부분 등을 비교하여 초분광지수를 F-검정 후 이에 따라 등분산 또는 이분산 T-검정을 하여 유의확률 평가
- 이 결과 가장 분별력이 높은 분석 방법은 PVISI2, VISR3 등

[아스팔트 포장 시편의 파손에 따른 초분광 지수]

구분	VISI1	VISR1	VISI2	VISR2	VISI3	VISR3	PVISI2
Core-Cr2Se-N	0.133	0.766	0.132	0.767	0.200	0.669	0.987
Core-Cr2Se-B	0.109	0.803	0.095	0.826	0.166	0.716	0.997
Core-Cr2-N	0.142	0.754	0.132	0.770	0.188	0.687	0.990
Core-Cr2-Cr	0.117	0.791	0.110	0.802	0.154	0.734	0.996
Core-Cr2-wCr	0.132	0.767	0.130	0.770	0.167	0.715	0.993
Core-Cr3-Cr	0.070	0.877	0.072	0.873	0.086	0.855	0.999

* Core: 현장채취코어, Cr2:2조각균열시편, Cr3: 3조각균열시편, Se:균열실링처리시편, N:양호부분, B:실링시편 검은밴드부, Cr: 균열, wCr: 큰균열

[아스팔트 포장 시편의 파손에 따른 초분광 지수 유의확률]

비교		VISI1	VISR1	VISI2	VISR2	VISI3	VISR3	PVISI2
Core-Cr2Se-N	Core-Cr2Se-B	0.034	0.032	0	0	0.086	0.001	0.001
Core-Cr2-N	Core-Cr2-Cr	0.034	0.039	0.061	0.071	0.010	0.012	0
Core-Cr2-N	Core-Cr2-wCr	0.399	0.457	0.900	0.991	0.168	0.194	0
Core-Cr2-N	Core-Cr3-Cr	0.011	0	0.025	0.001	0.006	0	0

* Core: 현장채취코어, Cr2:2조각균열시편, Cr3: 3조각균열시편, Se:균열실링처리시편, N:양호부분, B:실링시편 검은밴드부, Cr: 균열, wCr: 큰균열

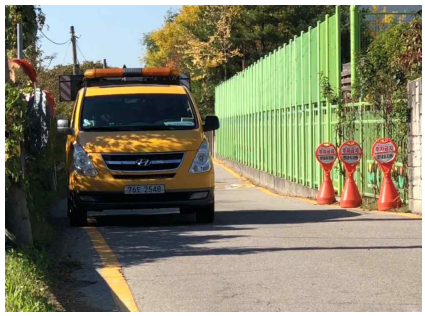
(5) 조사차량을 이용하여 산출한 PCI와 드론을 이용한 초분광 분석 자료 상관성 분석

○ 상관성 분석 방법

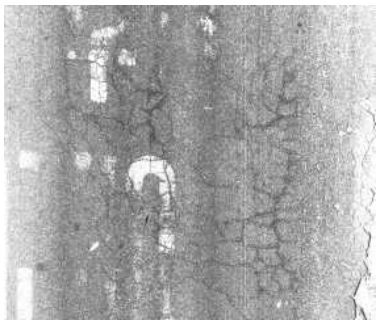
① 조사구간 선정 및 드론을 이용한 도로 포장 초분광 조사
② 아스팔트 포장 조사차량(PES)을 이용한 도로 조사
③ 조사구간을 10m 단위로 NHPCI 계산
④ 드론 조사구간 초분광 영상자료에서 분석영역 선정
⑤ 분석영역에 대한 평균 초분광 반사율 계산
⑥ 초분광 분석모델에 따른 초분광 지수 계산
⑦ 조사차량을 이용한 PCI(NHPCI)와 초분광 지수의 상관계수 분석



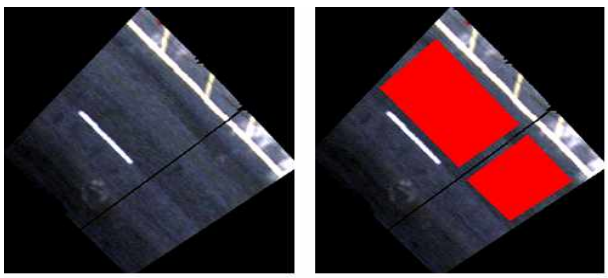
조사차량(PES)



도로조사(예)



조사결과(예)



[아스팔트 포장 분석 구간 선정(예)]

Band No.	Wavelength	SH1	SH2	SH3	SH4	SH5	SH6	SH7	SH8	SH9	SH10	SH11	SH12	SH13
Band 001	398.462	0.22683	0.21419	0.21296	0.21414	0.21633	0.21452	0.21305	0.19778	0.19451	0.19183	0.18741	0.1849	0.1809
Band 002	400.704	0.21204	0.20019	0.19885	0.19997	0.20218	0.20051	0.19914	0.18493	0.18195	0.17924	0.17526	0.17289	0.16919
Band 003	402.945	0.19839	0.18722	0.18577	0.18683	0.18905	0.18752	0.18623	0.17301	0.17029	0.16757	0.164	0.16175	0.15833
Band 004	405.186	0.18587	0.17529	0.17372	0.17472	0.17695	0.17553	0.17432	0.16201	0.15955	0.15681	0.1536	0.15148	0.14831
Band 005	407.428	0.17448	0.16441	0.1627	0.16364	0.16586	0.16455	0.16342	0.15194	0.1497	0.14696	0.14409	0.14208	0.13914
Band 006	409.669	0.16423	0.15456	0.15272	0.15358	0.1558	0.15457	0.15352	0.1428	0.14077	0.13802	0.13546	0.13354	0.13082
Band 007	411.91	0.1551	0.14575	0.14377	0.14456	0.14676	0.14561	0.14462	0.13459	0.13274	0.12999	0.1277	0.12587	0.12334
Band 008	414.151	0.14711	0.13799	0.13586	0.13656	0.13874	0.13765	0.13672	0.1273	0.12562	0.12287	0.12082	0.11907	0.11672
Band 009	416.393	0.14026	0.13126	0.12898	0.1296	0.13175	0.13071	0.12983	0.12095	0.11941	0.11667	0.11482	0.11314	0.11093
Band 010	418.634	0.13453	0.12558	0.12313	0.12366	0.12577	0.12477	0.12394	0.11552	0.1141	0.11138	0.1097	0.10807	0.106
Band 011	420.875	0.12994	0.12093	0.11832	0.11875	0.12082	0.11984	0.11905	0.11101	0.1097	0.10699	0.10545	0.10387	0.10191
Band 012	423.117	0.12141	0.11208	0.10933	0.10963	0.1118	0.11083	0.11025	0.10269	0.10134	0.09874	0.09766	0.09619	0.09412
Band 013	425.358	0.12549	0.11532	0.11246	0.11263	0.11471	0.1136	0.11302	0.10536	0.10408	0.10158	0.10011	0.09856	0.09667
Band 014	427.599	0.12617	0.11556	0.1125	0.11251	0.11439	0.11319	0.11261	0.10501	0.10383	0.10152	0.09981	0.0982	0.09646
Band 015	429.84	0.12995	0.11878	0.11543	0.11537	0.11719	0.11603	0.1154	0.10763	0.10641	0.10396	0.10229	0.10072	0.09888
Band 016	432.082	0.12969	0.1181	0.11452	0.11434	0.11605	0.11495	0.11428	0.10664	0.10539	0.10292	0.10135	0.09982	0.09799
Band 017	434.323	0.13428	0.12216	0.11824	0.11802	0.11973	0.11852	0.11785	0.11	0.10872	0.10617	0.10459	0.10298	0.10107
Band 018	436.564	0.13424	0.12185	0.11774	0.11739	0.11896	0.11776	0.11709	0.10935	0.1081	0.10557	0.10403	0.10241	0.10049
Band 019	438.805	0.13642	0.12364	0.11924	0.11878	0.12029	0.11902	0.11833	0.11055	0.10932	0.10678	0.10519	0.10355	0.10166
Band 020	441.047	0.13681	0.12384	0.11919	0.11859	0.12002	0.11872	0.11801	0.11031	0.10912	0.10662	0.10499	0.10333	0.10151

[PCI 분석 영역에 대한 평균 분광반사율 계산(예)]

(6) 조사차량을 이용한 PCI와 초분광 지수 상관관계 평가

○ 분석결과 PVISI2 모델로 분석한 결과 상관계수 0.83로써 목표인 0.6 이상을 초과 달성.
○ VISI1, VISI2, VISR2도 0.7 이상의 상관성 확보
○ 최적의 분석 모델은 PVISI2(ρ899/ρ967)가 적합한 것으로 평가

[도로 포장의 PCI와 조사결과의 상관성 분석]

구분	NHPCI	초분광 지수					
		VISI1	VISR1	VISI2	VISR2	VISI3	PVISI2
SH1	3.72	0.0362	1.2553	0.0653	1.4536	0.0822	0.7679
SH2	3.10	0.0344	1.2698	0.0534	1.4113	0.0881	1.0377
SH3	3.87	0.0181	1.1495	0.0326	1.2672	0.0340	1.1701
SH4	3.68	0.0141	1.1175	0.0279	1.2325	0.0157	1.2009
SH5	4.18	0.0076	1.0628	0.0219	1.1830	-0.0059	1.3617
SH6	4.40	0.0071	1.0595	0.0210	1.1774	-0.0085	1.3605
SH7	4.51	0.0076	1.0638	0.0214	1.1814	-0.0074	1.3574
SH8	4.61	0.0096	1.0863	0.0213	1.1924	0.0033	1.4331
SH9	5.20	0.0091	1.0824	0.0202	1.1843	0.0030	1.4861
SH10	5.45	0.0091	1.0842	0.0198	1.1847	0.0037	1.4789
SH11	5.65	0.0084	1.0788	0.0187	1.1768	0.0021	1.5471
SH12	5.98	0.0090	1.0858	0.0196	1.1881	0.0032	1.5041
SH13	5.73	0.0087	1.0848	0.0190	1.1861	0.0042	1.5489
SH14	5.52	0.0081	1.0789	0.0181	1.1776	0.0017	1.5737
SH15	5.56	0.0081	1.0791	0.0187	1.1845	-0.0032	1.5317
SH16	5.35	0.0086	1.0863	0.0182	1.1838	0.0017	1.5890
SH17	5.61	0.0083	1.0835	0.0184	1.1857	-0.0022	1.4774
SH18	5.59	0.0081	1.0815	0.0180	1.1816	-0.0010	1.5404
SH19	5.45	0.0080	1.0788	0.0185	1.1835	-0.0046	1.5336
SH20	5.36	0.0079	1.0796	0.0169	1.1700	0.0055	1.6294
SH21	5.06	0.0066	1.0650	0.0173	1.1721	-0.0109	1.5365
SH22	5.02	0.0049	1.0487	0.0146	1.1470	-0.0174	1.6599
SH23	5.29	0.0038	1.0379	0.0132	1.1349	-0.0238	1.6449
상관계수		-0.71	-0.67	-0.74	-0.71	-0.67	0.83

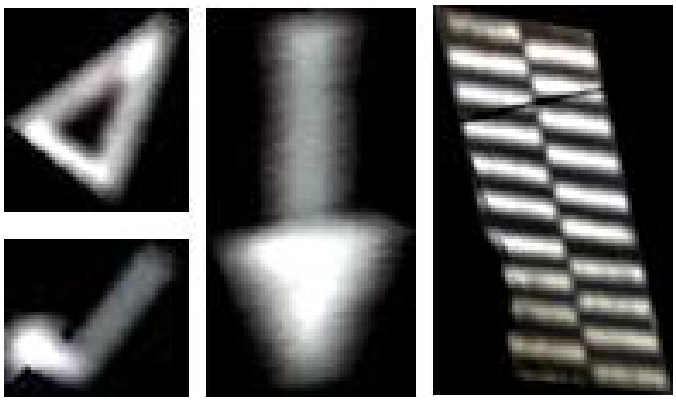
(7) 도로 포장 평가 제외 구간 자동 추출을 위한 라이브러리 구축

○ 도로 차선 및 표지 분류



[도로 포장 평가 제외 구간 분류]

○ 드론을 이용한 도로 표지 초분광 촬영



[초분광 사진]

○ 노면 표지 초분광 자료 구축

[노면 표지 초분광 자료(예)]

구분	반사율(nm)							
	460	490	595	675	740	760	830	940
흰색표지-1	0.6589	0.7009	0.8249	0.8945	0.9832	0.9366	1.1163	0.6304
흰색표지-2	0.3605	0.3739	0.4086	0.4253	0.4495	0.4193	0.4866	0.2490
흰색표지-3	0.4956	0.4811	0.4493	0.4642	0.4721	0.4482	0.5399	0.6873
흰색표지-4	0.3218	0.3426	0.3432	0.3398	0.3429	0.3438	0.3422	0.3762

(8) 도로포장상태(PCI) 측정을 위한 드론시스템 구축

○ 초분광 Platform 구축
- 드론 초분광센서는 Nano-Hyperspec 모델로 400nm ~ 1,000nm의 파장영역에서 영상 획득이 가능하며 최대 270개의 분광밴드를 취득할 수 있음
- 드론 기체는 중국 DJI의 회전익 드론인 Matrice 600 pro를 결합한 드론 플랫품을 사용



Hyperspectral Sensor

Specifications of Hyperspectral Sensor

Overview	Nano-Hyperspec
Wavelength Range	400 - 1,000 nm
number of Bands	270
Dispersion/Pixel(nm/pixel)	2.2
Storage capacity	480GB

○ 열적외선 Platform 구축
- 열적외선 센서를 통해 획득한 도로의 온도정보가 도로포장상태 미치는 영향을 판단할 수 있으며, 단일 센서로부터 취득된 데이터보다 더 많은 정보를 이용 가능함
- 드론 열화상센서는 FLIR Duo Pro R을 이용하였으며, 드론 기체는 중국 DJI의 회전익 드론 Matrice 600을 이용하여 드론 플랫폼을 사용함



FLIR Duo Pro R

Specifications of Thermal Sensor

Overview	Duo Pro R
Thermal Imager	Uncooled VOx Microbolometer
Spectral Band	7.5 - 13.5 μm
Thermal Sensitivity	[50 mK
Thermal Sensor Resolution Options	640 x 512
Thermal Lens Options	13mm : 45° x 37°
Thermal Frame Rate	30Hz

○ 도로포장상태 평가를 위한 드론 초분광시스템 영상 취득은 촬영계획, 촬영허가 신청, 초분광센서 촬영, 촬영성과 확인 순서로 진행함
- 촬영계획 단계에서는 한국건설기술연구원에서 운영 중인 PES장비를 토대로 취득된 도로포장상태지수(PCI)가 존재하는 지역을 우선으로 사전답사 및 주변 환경에 따른 드론 운영 유무를 판단하여 촬영 대상지를 선정하는 단계임, 대상지 선정 시 PCI이 존재하는 대상지를 선정하는 이유는 드론 초분광시스템으로 산출된 PCI을 검증하는데 목적이 있음
- 촬영허가 신청 단계에서는 해당 지역에 대하여 보안 및 안전에 관련된 허가를 미리 승인받는 단계로써 과업의 수행 시 상공에서 발생될 수 있는 안전문제와 보안지역에 대한 촬영에 대하여 차질이 없도록 함에 목적이 있음
- 초분광영상 취득은 센서 특성상 태양 고도에 따른 입사각 변화에 대해 아주 민감하게 반응하며, 이는 직접적으로 초분광영상의 신뢰도에 영향을 미치므로 탄력적인 운행이 필요함에 따라 영상촬영 시간은 오전 9시 ~ 오전 11시, 오후 2시 ~ 오후 4시에 진행하였으며, 태양의 조도 및 입사각의 변화가 발생하지 않게 직선도로를 기준으로 단일경로 비행을 수행함
- 초분광영상 보정 및 전처리 단계에서는 기존의 다중분광영상자료와 달리 수많은 분광밴드를 가지고 있기 때문에 획득된 자료를 분석에 활용하기 위해 오차를 보정하는 방사보정, 기하보정 등 전처리 과정을 반드시 수행되어야 함

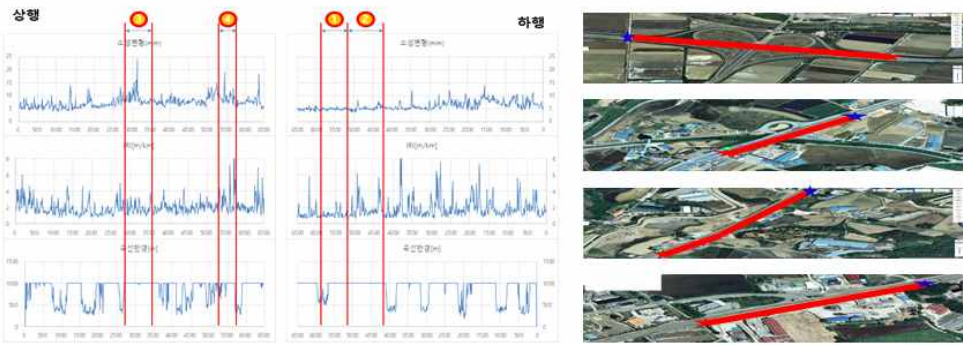


[드론 초분광센서 영상 취득 절차]

○ 1차 대상지 선정

- 드론 탑재형 초분광센서를 통해 취득한 PCI의 활용성 평가를 위해서는 도로의 상태를 객관적으로 나타낼 수 있는 기준값이 필요하며, 또한 다양한 도로환경에 따른 도로 분광라이브러리 특성을 파악하기 위해서는 도로포장상태가 다양한 지역을 대상지로 선정하는 것이 중요함
- 이에 위 조건을 반영할 수 있는 국도 38호선(안성종합버스터미널~일죽IC)를 대상지로 선정하여 한국건설기술연구원 PES 장비로 PCI를 산출하였으며 결과값(소성변형, IRI)을 기준으로 도로상태가 좋은 지역, 안 좋은 지역을 선택하여 1차 대상지로 선정함

	Case 1	Case 2	Case 3	Case 4
소성변형(mm)	Good	Good	Bad	Bad
IRI(m/km)	Good	Bad	Good	Bad



[초분광센서를 이용한 도로조사 구간 선정(1차)]

○ 2차 대상지 선정

- 도로포장상태지수(PCI)를 기준으로 선정된 1차 대상지에 대해 현장답사를 통하여 드론운영의 적합성 및 비행금지구역 등 환경적 요인을 파악하여 최종 도로조사 구간을 선정함

위치도	
소재지	경기도 용인시 처인구 양지면 추계리 53-2
	
대상도로	드론 이·착륙지
대상현지조사사진	
	
입지환경 분석	
대상지 장점	- 대형차량의 교통량이 많아 PCI(도로포장지수)를 기반한 '하' 지역별로 선정 - 약 250m 직선도로 존재, 송전탑 및 고층 건물 없음
대상지 단점	- 신호체계에 따른 교통량 및 신호정체가 많음
선정사유	- 건설기술연구원 PES 장비를 이용한 PCI 결과값 활용 및 연계

위치도	
소재지	경기도 여주시 하동 3-2
	
대상도로	드론 이·착륙지
대상현지조사사진	
	
입지환경 분석	
대상지 장점	- 대형차량의 교통량이 많아 PCI(도로포장지수)를 기반한 '중' 지역별로 선정 - 약 500m 직선도로 존재, 송전탑 및 고층 건물 없음
대상지 단점	- 영상 후처리 시 대교의 DEM 부재로 인한 영상정합의 오차 발생
선정사유	- 건설연 PES 장비를 이용한 PCI 결과값 활용 및 연계

위치도	
소재지	경기도 여주시 가남읍 심석리 100-1

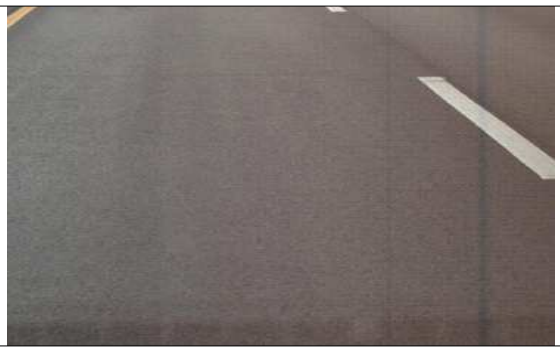




대상도로	드론 이·착륙지

대상현지조사사진	





입지환경 분석	
대상지 장점	- 대형차량의 교통량이 많아 PCI(도로포장지수)를 기반한 '상' 지역별로 선정 - 약 300m 직선도로 존재, 송전탑 및 고층 건물 없음
대상지 단점	- 도로 양쪽에 다수의 둔턱(언덕)이 존재하여 도로에 그림자가 짐
선정사유	- 건설연 PES 장비를 이용한 PCI 결과값 활용 및 연계

위치도	
소재지	경기도 고양시 일산서구 대화동 882





대상도로	드론 이·착륙지

대상현지조사사진	





입지환경 분석	
대상지 장점	- 건설연 인근 지방도로로 접근이 용이하고 도로포장 조사에 호조건
대상지 단점	- 아파트 단지 및 초등학교가 있어 유동인구가 많음
선정사유	- 지속적인 연구 수행 위해 건설연 PES팀과 협의하여 PCI 재산출 및 결과값 활용

위치도	
소재지	경기도 시흥시 정왕동 2506





대상도로	드론 이·착륙지

대상현지조사사진	





입지환경 분석	
대상지 장점	- 신도시 개발로 교통량이 적으며 드론 운영에 방해가 되는 고층건물이 없음
대상지 단점	- 대규모 공원이 있어 유동인구가 많음
선정사유	- 접근 용이하고, 도로포장상태 양호한 도로 초분광 라이브러리 구축을 목적

위치도	
소재지	경기도 안성시 일죽면 월정리 719





대상도로	드론 이·착륙지

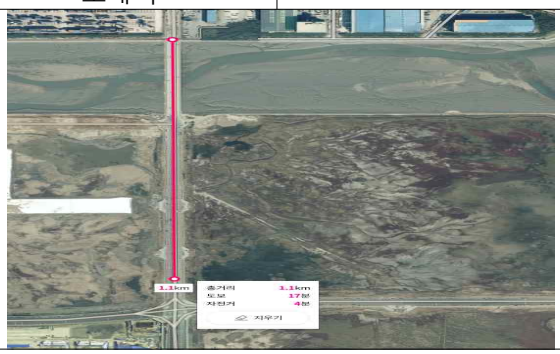
대상현지조사사진	





입지환경 분석	
대상지 장점	- 교통량이 적으며, 중장기 장비(트렉터 등) 이동으로 도로포장상태가 좋지 않음
대상지 단점	- 농가 밀접지역으로 도로표피에 부산물(흙, 퇴비 등)이 많음
선정사유	- 도로포장이 불량한 도로의 초분광 라이브러리 구축을 목적

위치도	
소재지	경기도 연천 37호선 국도 “자장사거리” 일대 선정
	
대상도로	드론 이·착륙지
대상현지조사사진	
	
입지환경 분석	
대상지 장점	- 건설연 PES 장비를 이용한 PCI 결과값 활용 및 연계
대상지 단점	- 휴전선 인근 초경량비행승인의 어려움
선정사유	- 콘크리트 포장상태 상반된 2곳 선정하여 도로파손뱔 분광 라이브러리 DB 구축

위치도	
소재지	인천광역시 중구 신흥동3가 80
	
대상도로	드론 이·착륙지
대상현지조사사진	
	
입지환경 분석	
대상지 장점	- 아스팔트/콘크리트 공존 및 도로상태 양호하여 도로 분광라이브러리 구축
대상지 단점	- 군부대 인근에 위치하여 초경량비행신고가 요구됨
선정사유	- 접근이 용이하고 아스팔트/콘크리트 공존하여 복합 분광라이브러리 구축 용이

위치도	
소재지	인천광역시 동구 송현동 146





대상도로	드론 이·착륙지

대상현지조사사진	





입지환경 분석	
대상지 장점	- 아스팔트/콘크리트 공존하며, 화물차량 이동으로 불량 도로포장 다수
대상지 단점	- 군부대 인근에 위치하여 초경량비행신고가 요구됨
선정사유	- 접근이 용이하고 아스팔트/콘크리트 도로 공존하여 분광라이브러리 구축 용이

위치도	
소재지	경기도 안성시 삼죽면 진촌리 307-3





대상도로	드론 이·착륙지

대상현지조사사진	





입지환경 분석	
대상지 장점	- 드론 운영 시 방해가 될 수 있는 송전탑 및 고층건물이 존재하지 않음
대상지 단점	- 교통량이 많아 모든 대상지역의 도로표피 데이터 구축에 한계 발생
선정사유	- 건설연 PES 장비를 이용한 PCI 결과값 활용 및 연계

위치도	
소재지	경기도 안성시 죽산면 두현리 109-1





대상도로	드론 이·착륙지

대상현지조사사진	





입지환경 분석	
대상지 장점	- 드론 운영 시 방해가 될 수 있는 송전탑 및 고층건물이 존재하지 않음
대상지 단점	- 교통량이 많아 모든 대상지역의 도로표피 데이터 구축에 한계 발생
선정사유	- 건설연 PES 장비를 이용한 PCI 결과값 활용 및 연계

위치도	
소재지	경기도 평택시 포승읍 도곡리 170-76

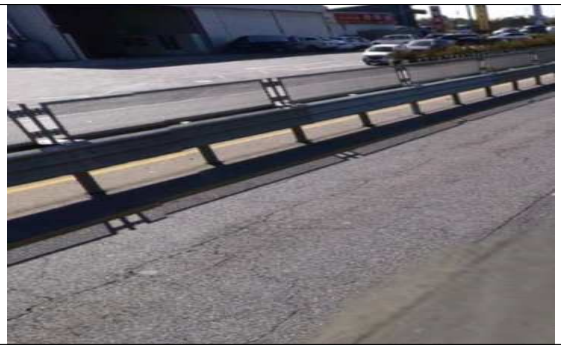




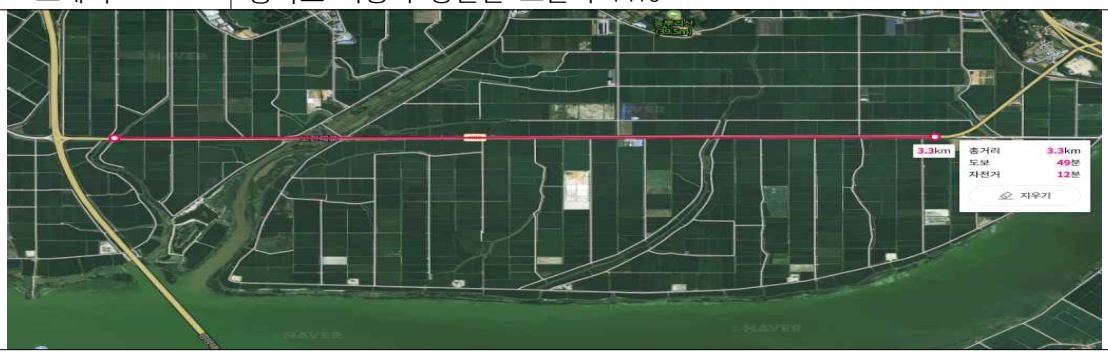
대상도로	드론 이·착륙지

대상현지조사사진	





입지환경 분석	
대상지 장점	- 약 500m 정도의 도로포장상태가 불량한 직진도로
대상지 단점	- 인근 공장단지 존재로 인한 교통량이 많음
선정사유	- 도로의 초분광 라이브러리 구축을 목적

위치도	
소재지	경기도 화성시 장안면 노진리 1416
	
대상도로	
대상현지조사사진	
	
입지환경 분석	
대상지 장점	- 약 3.3km 정도의 직진도로
대상지 단점	- 왕복 2차선 도로
선정사유	- 도로의 초분광 라이브러리 구축을 목적

- 드론 탑재형 초분광시스템 기반 산출된 PCI의 정확도 분석을 위해 사용되는 기존 PCI는 10m 단위로 지수값이 생성되기 때문에 PES 장비와 드론의 촬영 도로의 정확한 위치 일치가 이루어져야함으로, 대상지 선정 시 시점(촬영 시작점)과 종점(촬영 종료지점)을 정확히 명시하여 두 장비를 통해 데이터를 구축함

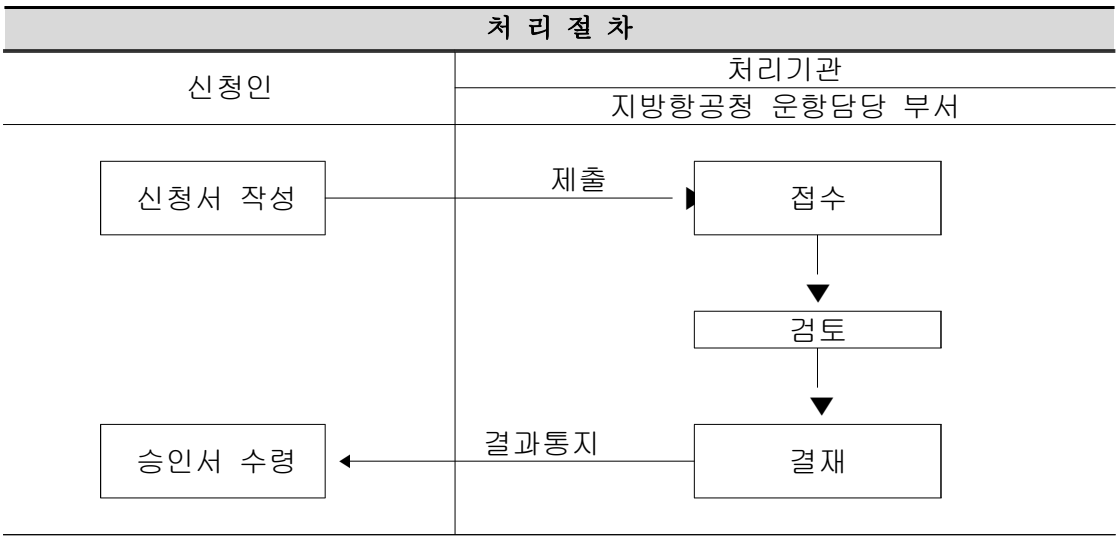


[초분광센서를 이용한 도로조사 구간 선정]

- 드론 초분광센서 촬영 허가

· 영리 목적으로 드론을 운행 시 반드시 초경량비행장치(드론) 비행을 항공안전법 제 127조제2항 및 같은 법 시행규칙 제308조제2항 규정에 의거한 승인신청 절차를 걸쳐야하며, 드론사업을 수행하는 업체는 드론 운영 전에 드론보험가입 증명서, 초경량비행장치 신고증명서, 초경량비행장치 사용사업 등록증 등 비행승인 신청자료를 준비하여 항공기 운항스케줄 '원스탑' 민원 처리시스템을 통해 비행승인 신청을 해야 함

[초경량비행장치(드론) 비행승인 신청절차]



- 현장답사 및 비행승인 유무에 따라 최종 대상지를 선정하였으며, 대상지역 중 파주에 위치한 '자장사거리' 일대를 제외한 다른 지력은 경기도 일대로 비행금지대상이 위치해 있지 않아 관할 군부대의 보안성검토 절차 없이 촬영이 가능한 지역으로 안전한 드론운행을 위한 조정자 안전의무사항을 숙지하여 안전하게 드론촬영을 수행함

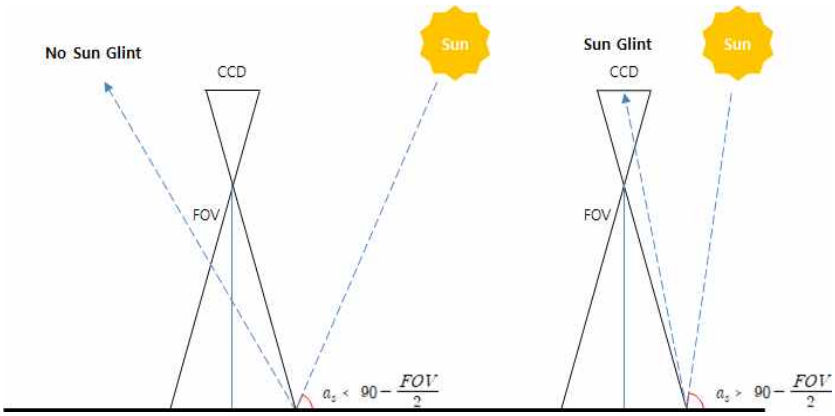
[도로조사를 위한 초분광 시스템 촬영 대상지]

구분	대상지역	비고
1	경기 용인 양지면 추계리 53-2	
2	경기 여주 하동 3-2	
3	경기 여주 가남읍 심석리 100-1	
4	경기도 고양시 일산서구 대화동 882	
5	경기 시흥 정왕동	
6	경기 안성 일죽면 월정리 719	
7	경기 연천 37호선 국도 자장사거리 일대	휴전선 인근 지역에 따른 비행승인 불가
8	인천 중구 신흥동3가 80	
9	인천 동구 송현동 146	
10	경기도 안성시 삼죽면 진촌리 307-3(상행)	
11	경기도 안성시 삼죽면 진촌리 307-3(하행)	
12	경기도 안성시 죽산면 두현리 109-1	
13	경기도 평택시 포승읍 도곡리 170-76	
14	경기도 화성시 장안면 노진리 1416	

○ 드론 초분광시스템 최적의 비행 방법 정립

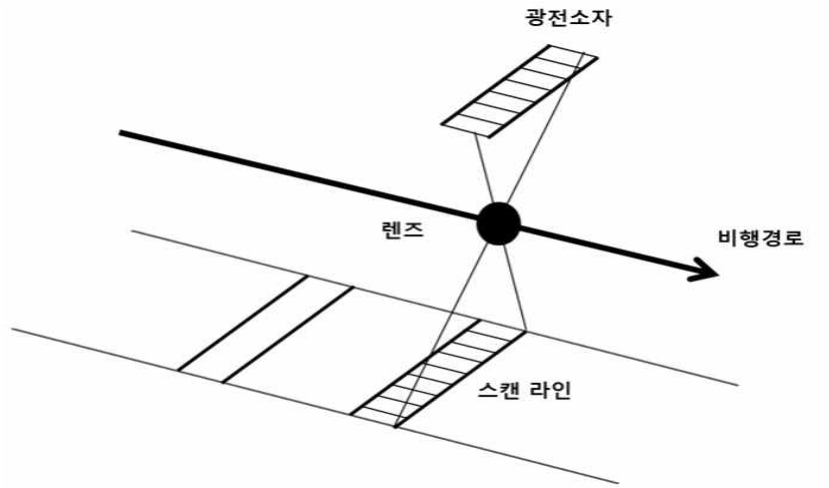
- 초분광영상의 품질은 드론 운영 방법에 따라 많은 영향을 미침으로 대상지에 따른 적절한 촬영계획을 선정하여 촬영해야 하며, 본 과업에서는 촬영 계획 수립 시 태양고도각, 라인스캔(Push Broom)방식에 따른 해상도를 고려함

· 태양 고도각$(a_s) > 90^\circ - \frac{FOV}{2}$이면 반사광이 센서 저장 영역(CCD) 안으로 빛이 직접적으로 들어와 Sun Glint현상이 발생하며, 이는 초분광영상에 빛 번짐, 분광라이브러리 오류 등으로 인한 영상 품질저하를 초래함으로 가급적으로 태양고도가 45~65° 에 해당하는 오전 9시 ~ 오전 11시, 오후 2시 ~ 오후 4시에 집중적으로 촬영함



[태양 고도각에 따른 Sun Glint 발생 유무]

· 라인 스캔(Push Broom)방식은 프레임 방식 DMC와 달리 1회 촬영 시 비행경로의 횡방향 1라인 데이터만을 취득하므로, 횡방향 공간 해상도와 종방향 공간해상도가 일치하지 않는다. 전자의 경우 기준면으로부터 비행 고도, 초점 거리, 인접 광전 소자사이 이격거리를 기준으로 계산할 수 있으며, 후자는 플랫폼의 수평 속도와 초당 촬영 수에 따라 달라진다. 따라서 본 과업에서는 도로포장상태 평가에 가장 적절한 공간해상도 구축을 위한 비행고도 및 비행속도를 수립하여 촬영함



[라인스캔(Push Broom) 방식]

· 도로상태에 따른 최적의 도로분광라이브러리 구축을 위하여 촬영고도, 촬영속도 등 다양한 경우의 촬영을 진행하였으며, 그 결과 비행고도 70m, 비행속도 5m/s의 촬영제원으로 진행하였을 때 가장 좋은 성과를 취득하였음

○ 드론 초분광시스템 최적의 촬영계획 설립

[경기도 용인시 양지면 추계1리 촬영제원]

촬영제원	
촬영고도	70m
비행속도	9m/s
중복도	20% (Pushbroom Type Scanner)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



[경기도 여주시 하동 3-2 촬영제원]

촬영제원	
촬영고도	70m
비행속도	7m/s
중복도	20% (Pushbroom Type Scanner)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



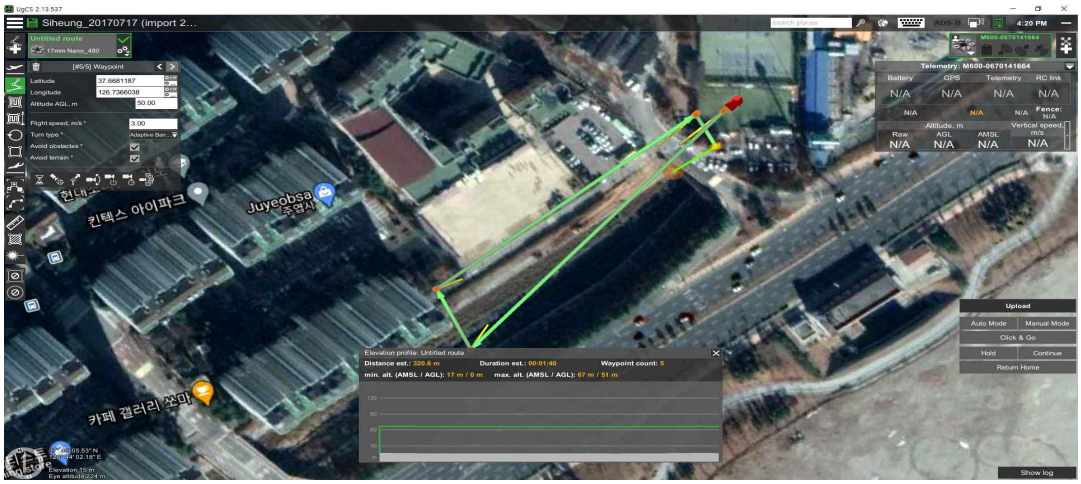
[경기도 여주 가남읍 심석리 100-1 촬영제원]

촬영제원	
촬영고도	70m
비행속도	7m/s
중복도	20% (Pushbroom Type Scanner)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



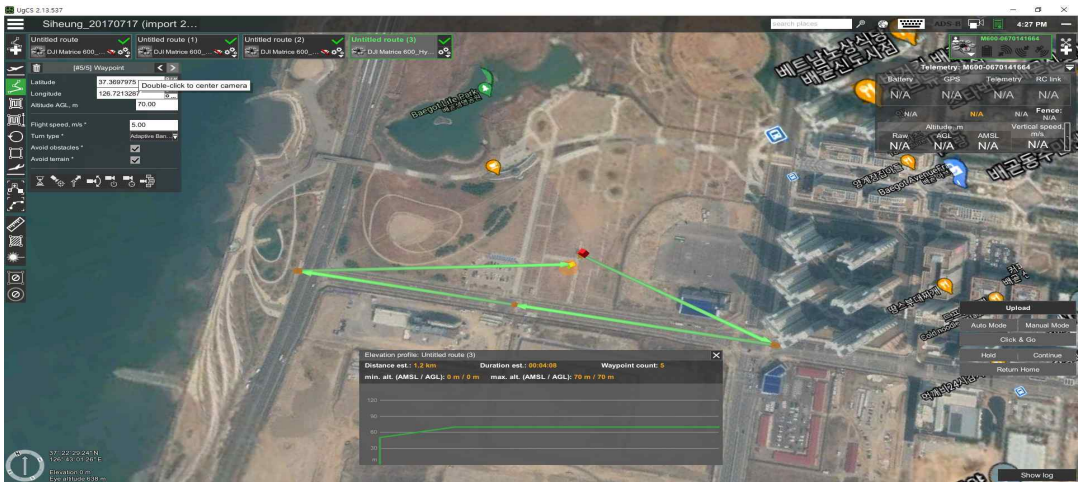
[경기도 고양시 일산서구 대화동 882 촬영제원]

촬영제원	
촬영고도	100m * 주변 아파트로 인한 드론 고도↑
비행속도	6m/s
중복도	단일비행경로 (PushBroom에 인한 Forward Overlap 1%)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



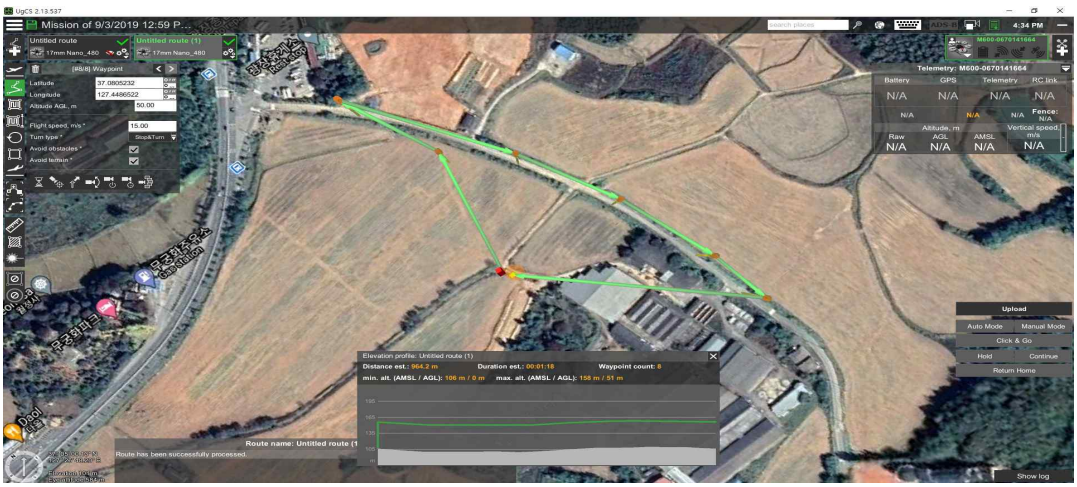
[경기도 시흥시 정왕동 2506 촬영제원]

촬영제원	
촬영고도	70m
비행속도	6m/s
중복도	단일비행경로(PushBroom에 인한 Forward Overlap 1%)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



[경기도 안성시 일죽면 월정리 719 촬영제원]

촬영제원	
촬영고도	100m (주변 장애물로 인한 드론 고도↑)
비행속도	9m/s (강풍으로인한 드론 비행속도↑)
중복도	단일비행경로 (PushBroom에 인한 Forward Overlap 1%)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



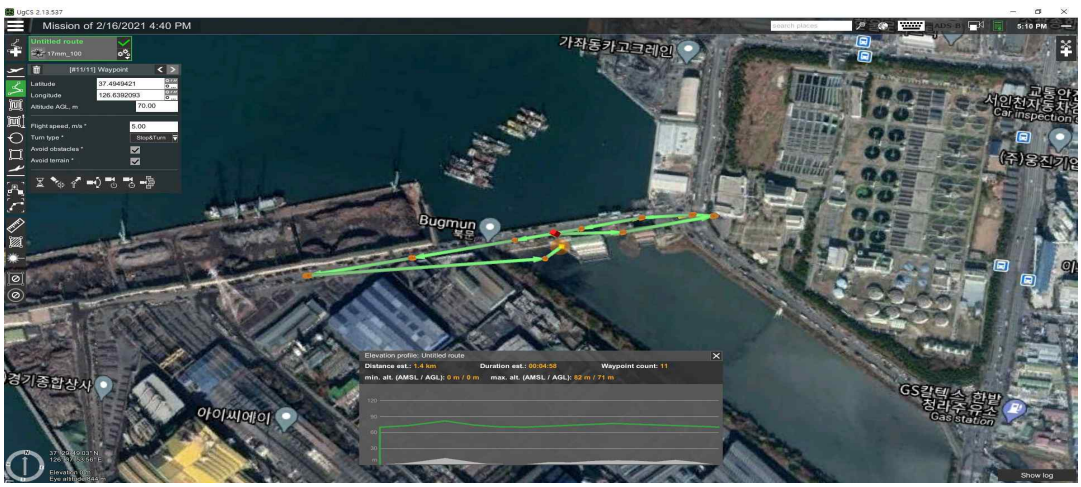
[인천광역시 중구 신흥동3가 80 촬영제원]

촬영제원	
촬영고도	50m
비행속도	5m/s
중복도	단일비행경로(PushBroom에 인한 Forward Overlap 1%)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



[인천광역시 동구 송현동 146 촬영제원]

촬영제원	
촬영고도	50m
비행속도	5m/s
중복도	단일비행경로 (PushBroom에 인한 Forward Overlap 1%)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



[경기도 안성시 삼죽면 진촌리 307-3 촬영제원]

촬영제원	
촬영고도	70m
비행속도	5m/s
중복도	단일비행경로 (PushBroom에 인한 Forward Overlap 1%)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



[경기도 안성시 죽산면 두현리 109-1 촬영제원]

촬영제원	
촬영고도	70m
비행속도	5m/s
중복도	단일비행경로(PushBroom에 인한 Forward Overlap 1%)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



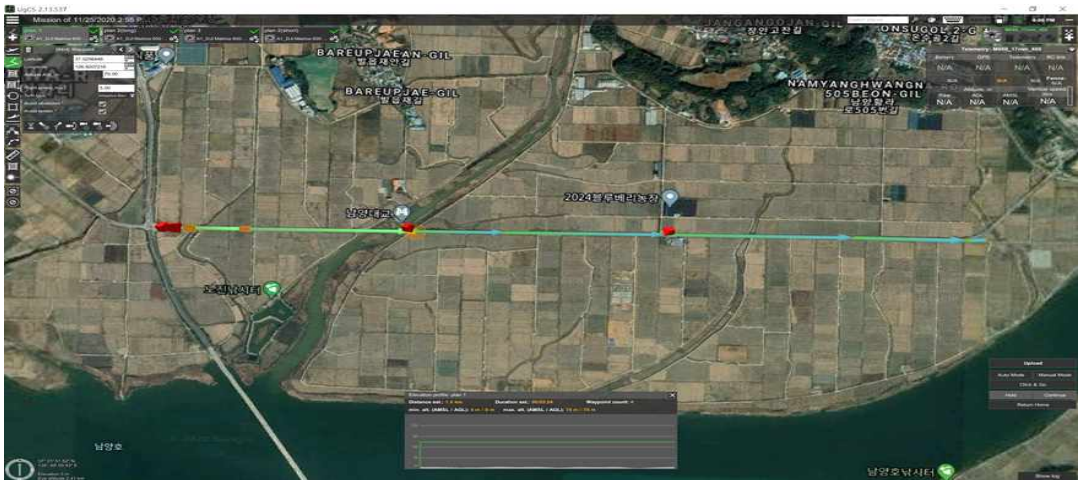
[경기도 평택시 포승읍 도곡리 170-76 촬영제원]

촬영제원	
촬영고도	70m
비행속도	5m/s
중복도	단일비행경로(PushBroom에 인한 Forward Overlap 1%)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



[경기도 화성시 장안면 노진리 1416 촬영제원]

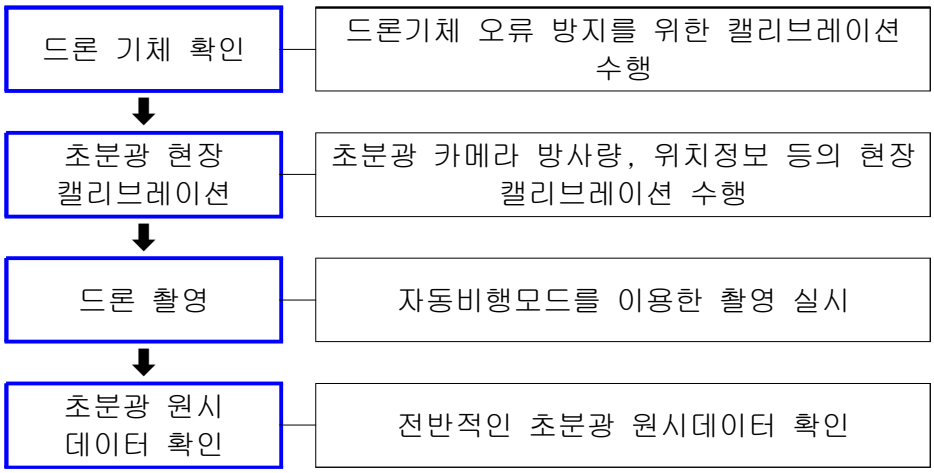
촬영제원	
촬영고도	70m
비행속도	5m/s
중복도	단일비행경로(PushBroom에 인한 Forward Overlap 1%)
촬영모드	자동비행모드
자동비행 SW	Headwall 사의 전용 SW(Ugcs)



○ 드론 탑재형 초분광 시스템 촬영

- 드론 탑재형 초분광시스템을 통해 대상지의 정확한 분광라이브러리를 구축을 위해 촬영공정에 따라 촬영이 진행 함

[드론 초분광시스템 촬영공정]



· 드론 기체 확인은 드론 운영 시 조정자의 과실이 아닌 드론기체의 오작동으로 인해 발생하는 사고를 방지하고 높은 품질의 성과를 취득하기 위해 현장에서 수행하는 캘리브레이션 단계임
· 특히 드론기체가 처음 운영하는 대상지역에서는 현장에서 드론 짐벌, IMU, 콤파스 캘리브레이션을 수행 후 촬영을 진행함
· 초분광 현장 캘리브레이션은 센서 특성상 다른 RGB, 열화상 센서와 달리 현장 캘리브레이션을 실시하여 초분광센서의 영상처리 과정에서 방사보정과 기하보정을 위한 필수 과정으로 영상 품질과 직결되는 문제이므로 철저히 실시해야함

[초분광센서 현장 캘리브레이션 항목]

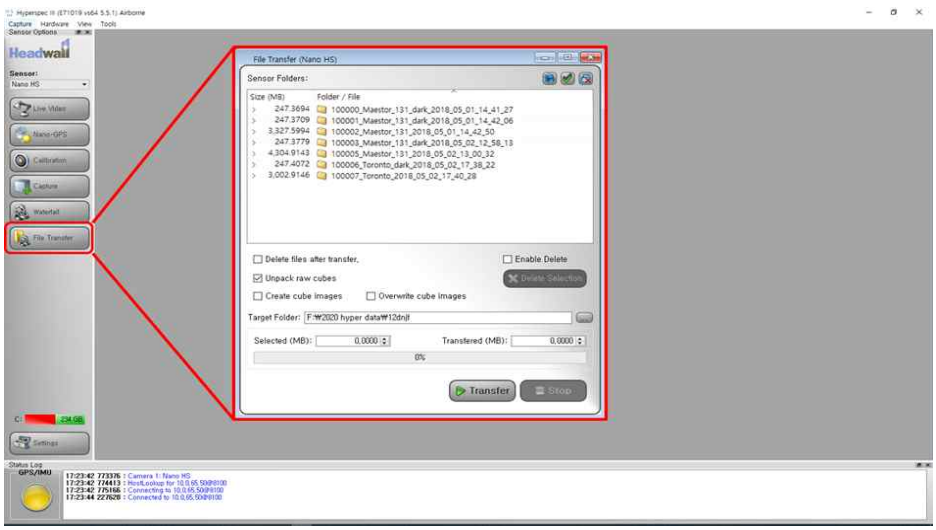
항목	점검사항	비고
하이퍼 전용 GPS 캘리브레이션	전용 SW(Spectral III)를 이용한 GPS 연결상태 확인	Spectral III 이용
하이퍼 전용 IMU 캘리브레이션	수평축, 수직축 캘리브레이션 실시	Spectral III 이용
Black 밸런스 점검	카메라 렌즈 덮개를 막고 Black 밸런스 점검	Spectral III 이용
White 밸런스 점검	카메라 렌즈를 풀고, 반사도 95% 순백지를 이용하여 점검 실시	Spectral III 이용

· 드론 촬영 단계에서는 양질의 초분광영상 취득을 위해 비행고도, 촬영시간, GPS 수신 등 다양한 요소를 파악하여 최적의 비행방안을 도출해야 함
· 비행고도는 약 70m일 때 20m의 촬영 폭으로 영상을 취득할 수 있으며 촬영에 활용되는 Nano-Hyperspec 센서의 분광범위는 400nm ~ 1,000nm이며 라인스캔방식으로 촬영을 진행함
· 초분광센서로 취득되는 분광데이터는 반사 및 방사와 같은 빛에너지를 취득하게 되는데, 이러한 과정에서 태양고도각에 의한 경면반사와 같이 대기 및 분광패턴에 영향을 미치는 요소를 최소화하기 위하여 촬영 시 태양고도각이 65° 이하가 되는 오전 9시 ~ 오전 11시, 오후 2시 ~ 오후 4시에 촬영을 수행함
· GPS 관측위성은 수신지역의 환경 및 시간의 변화에 따라 유동적으로 변화하여 GPSS/IMU데티어 품질에 많은 영향을 미치기 때문에 초분광 촬영을 단일코스로 계획하여 드론기체가 안정적으로 유지하여 양호한 위성 신호를 수신하며, 위성 고도 각은 15° 이상, 위성개수는 7개 이상이 되도록 함

[드론 탑재형 초분광시스템 촬영 현장사진]

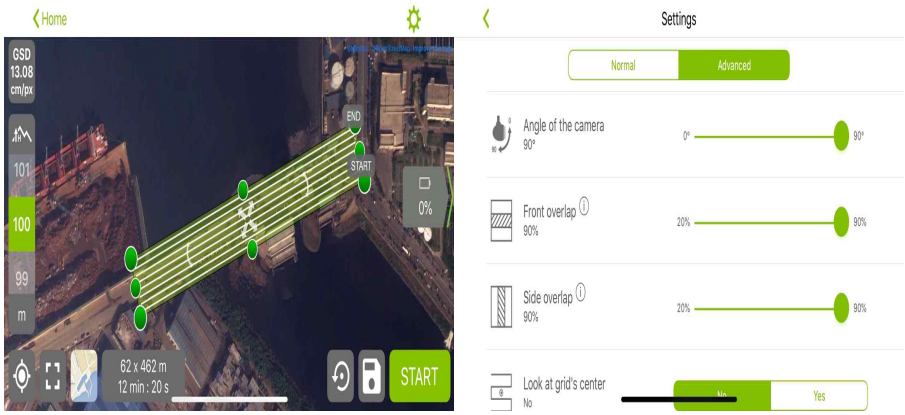


· 초분광 원시데이터 확인은 촬영 완료 후 HyperspecIII SW를 활용하여 센서 자체 스토리지에 있는 영상을 유선네트워크 연결을 통해 영상처리가 가능한 컴퓨터로 데이터를 옮긴 후 영상의 품질을 육안으로 검토하여 촬영계획과 동일하게 데이터가 취득되었는지를 확인하는 단계임



○ 드론 탑재형 열화상 시스템 촬영

- 도로의 온도정보가 포장상태 미치는 영향분석을 위한 드론 열적외선 운영 방안
- 대상지의 주변 환경을 고려한 촬영고도 및 노선 계획 수립
- 열적외선 영상품질은 영상 외곽으로 갈수록 온도정보 누락으로 노이즈 현상이 발생
- 따라서, 영상품질 향상을 위한 높은 중복도 및 여유로운 촬영면적 선정이 요구

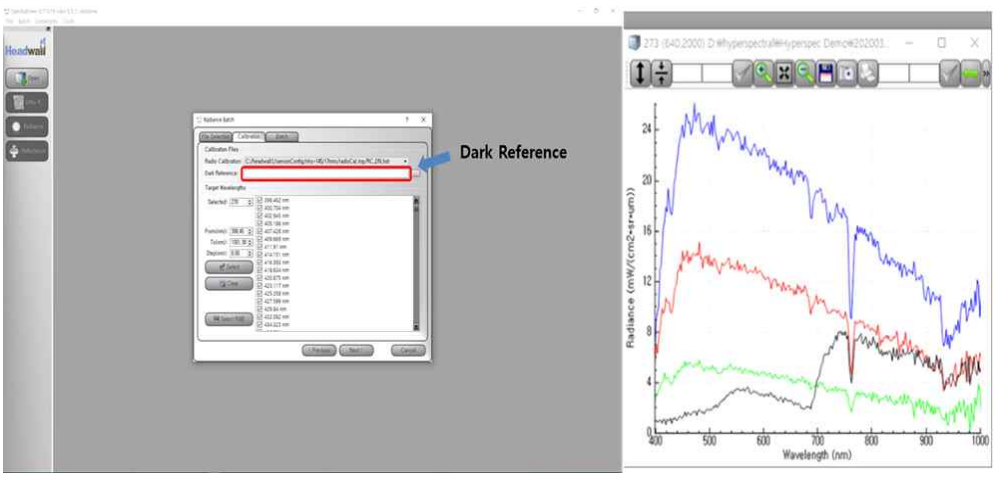


○ 드론 초분광시스템 전처리 구현

- 초분광 영상자료는 기존의 다중분광영상자료와 달리 수많은 분광밴드를 가지고 있기 때문에 보정을 거치지 않은 RAW 데이터의 경우 의미가 없는 Digital Number로 저장되어 있어, 이러한 무의미한 데이터를 분석에 활용하기 위해서 오차를 보정하는 방사보정, 기하보정, 대기보정 등 전처리 과정을 통해 의미있는 수치로 바꿔주어야 함



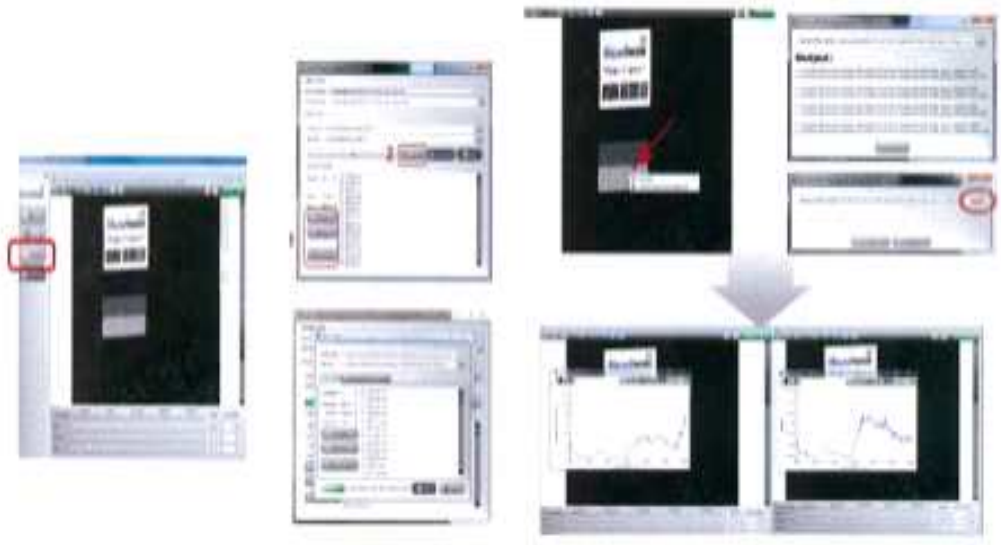
· **(방사보정)** 태양의 고도각, 탐측기의 응답 특성, 대기의 상태, 지형적 용인에 의한 영상의 밝기값 변화로 나타나는 왜곡을 보정하기 위해 실시하는 과정으로 Headwall社 소프웨어 Spectral View를 통해 초분광 원시데이터가 복사된 에너지의 양인 Radiance 값을 가지게 됨
· 방사보정은 현장에서 센서 캘리브레이션 수행 시 현장에서 White Reference 및 Dark Reference를 수행하게 되는데 이러한 현장의 환경이 적용된 데이터를 통해 방사 보정을 수행하였음



[방사보정을 통한 Radiance값 취득]

· **(대기보정)** 대기보정은 영상 내에 고정된 반사율을 가지는 물체 TRAF를 현장에 설치하고, 반사판이 영상에서 가지는 반사율을 기준으로 전체 영상을 정규화하여 반사율을 얻은 후 수행함

· 대기보정 반사율 식은 $\frac{(Raw\,data - Dark\,Reference)}{(White\,Reference - Dark\,Reference)}$ 를 이용하여 적용하였음



[초분광 영상 대기보정 과정]

· **(기하보정)** 초분광 데이터를 취득 할 때 지구와 드론 기체 및 센서의 특징으로 인한 기기적 한계로 인해 초분광데이터 영상의 공간적 왜곡이 발생하게 되며 이를 기하오차라고 한다. 이런 기하오차를 보정하기 위해서 위성 영상의 픽셀좌표와 지상의 지리좌표를 대응하여 초분광 데이터영상을 적걸한 실제 위치에 위치 할 수 있도록 프로세스를 진행해줘야 하는데 이를 기하보정이라고 함

· 기하보정 시 기복변위 및 기복으로 인한 반사광 변화를 고려해야 하므로 반드시 DEM을 이용해야하며 DEM의 정밀도가 높을수록 초분광영상의 품질에 좋은 영향을 미친다. 따라서 본 연구에서는 5m 해상도를 가지는 DEM을 적용하여 기하보정을 수행하였다. 또한 Headwall社의 Nano-Spectral 제품의 기하보정 시 GPSS/IMU Offsets 대상지역에 맞는 매개변수를 입력하여 영상 품질을 향상 시킬 수 있음



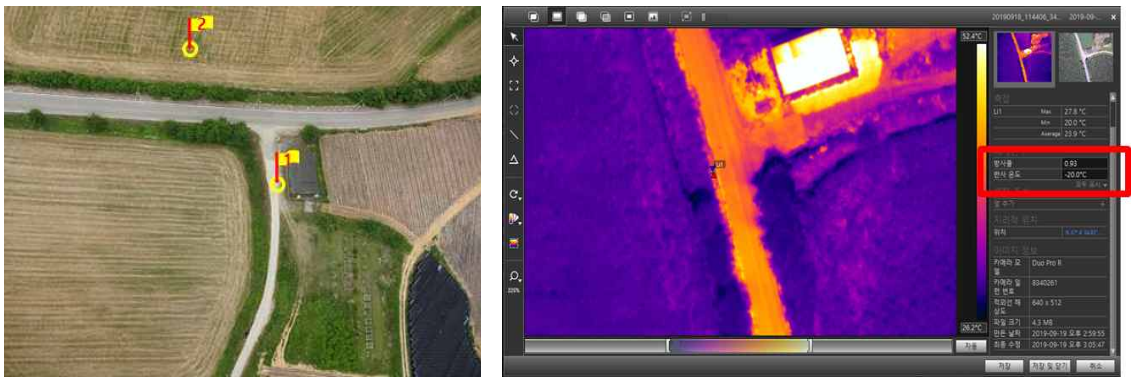
[초분광 영상 기하보정]

○ 드론 열화상시스템 전처리 구현

- 열화상 전용 SW인 Pix4D mapper Pro를 활용한 열적외선 영상 산출함

· 열화상 데이터는 상대적인 온도 분포로 나타남으로 도로의 실제 온도를 산출하기 위해서는 현장에서 직접 취득한 온도를 기준으로 변환해주는 작업이 추가적으로 발생함

· 실제온도로 변환하기 위해 농경지, 주변도로에서 촬영시점의 온도를 측정하였으며, 표시된 2지점의 온도는 21~40.5° C의 범위에 분포한 것으로 나타났으며, 온도값 보정을 위해 FLIR 社의 FLIR tools SW를 활용하여 취득한 대상지의 도로 온도를 절대값으로 변환하여 산출하였음



[열화상시스템으로 취득한 도로 온도를 절대온도로 변환]

○ 초분광영상 DB 구축

- 데이터 취득을 위한 드론운영은 태양고도각에 따른 초분광영상 품질 저하를 최소화 시키기 위한 단일비행코스로 촬영을 수행하였으며, 영상의 해상도를 고려한 최적의 비행고도 및 속도를 고려하여 드론 운행을 수행하였음
- 드론 탑재용 초분광시스템을 통해 취득된 도로포장상태지수의 검증 및 활용성을 평가하기 위해 한국건설기술연구원에서 PES 기반으로 취득된 PCI 기준값이 존재하는 도로(약 100km)를 기준으로 드론 운영이 용이하며 촬영 가능한 지역을 선정하여 초분광영상을 구축함

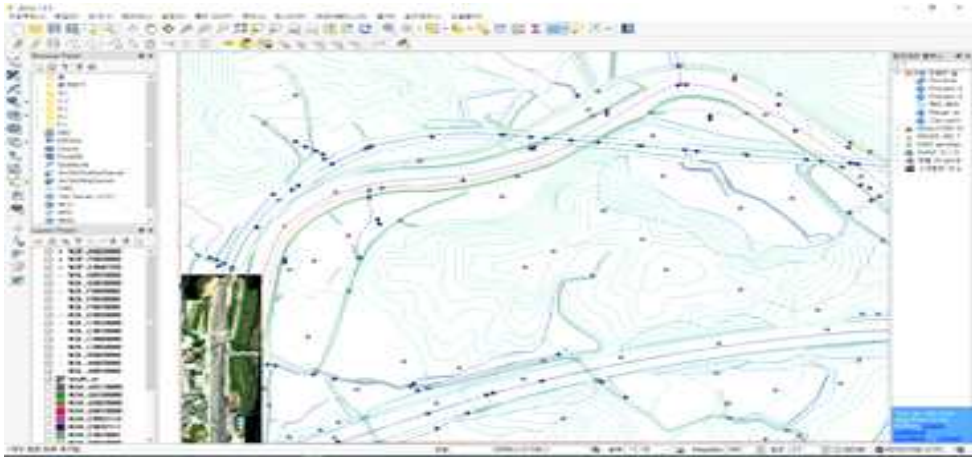
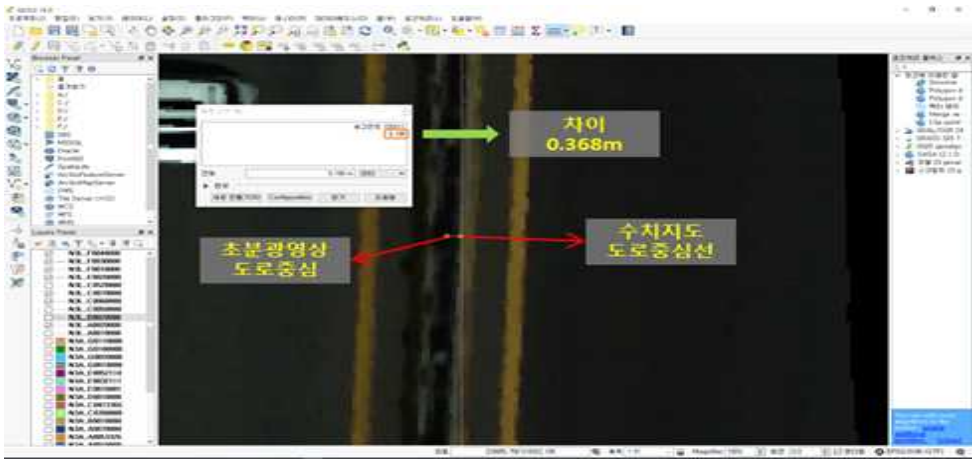
[드론시스템을 활용한 도로포장 초분광데이터 구축 현황]

구분	대상지역	기존 PES DB현황	드론 초분광DB	촬영횟수	비고
1	경기 용인 양지면 추계리 53-2	20.0 km	0.4 km	2회	PES DB 중 최적 구간 선정
2	경기 여주 하동 3-2	20.0 km	1.0 km	2회	상동
3	경기 여주 가남읍 심석리 100-1	20.0 km	0.5 km	2회	상동
4	경기 고양 일산서구 대화동 882	0.2 km	0.2 km	2회	상동
5	경기 시흥 정왕동	0.6 km	0.6 km	2회	상동
6	경기 안성 일죽면 월정리 719	10.0 km	0.4 km	2회	상동
7	경기 연천 국도 37호선 자장사거리	10.0 km	-	2회	비행승인 불가
8	인천 중구 신흥동3가 80	0.7 km	0.7 km	2회	PES DB 중 최적 구간 선정
9	인천 동구 송현동 146	0.5 km	0.5 km	2회	상동
10	경기 안성 삼죽면 진촌리 307-3(상행)	20.0 km	0.5 km	2회	상동
11	경기 안성 삼죽면 진촌리 307-3(하행)		0.5 km	2회	상동
12	경기 안성 죽산면 두현리 109-1		0.5 km	2회	상동
13	경기 평택 포승읍 도곡리 170-76	-	1.5 km	2회	상동
14	경기 화성 장안면 노진리 1416	-	3.3 km	1회	상동
합계		100.0 km	10.6 km	27회	도로 1차선 기준

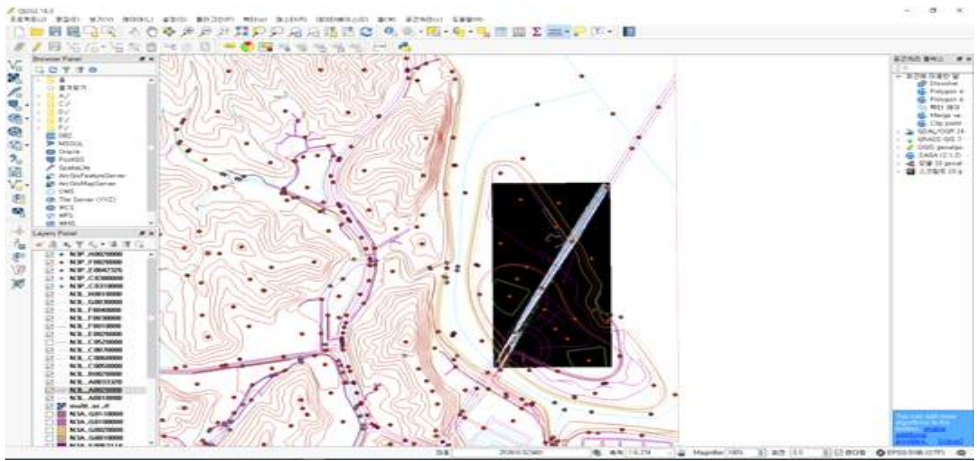
○ 초분광영상 위치 정확도 검증

- 드론초분광시스템 기반으로 취득한 초분광영상의 PCI 측정을 위한 위치정확도 검증을 위해 국토정보플랫폼에서 제공되어지는 국가기본도(1:5,000)를 취득하였으며, 이를 통해 초분광영상의 위치 정확도는 평가함
- 평가는 수치지도의 도로중심선(N3L_A0020000), 인도(도로-인도 경계) 경계선(N3L_A0033320) 레이어를 통해 위치 정확도 검증을 수행함
- 이를 통해 초분광영상의 위치정확도는 **약 0.361m**의 수준으로 본 과제 PCI 측정을 위한 위치정확도 최종목표치인 0.5m 이내로 목표를 달성하였음

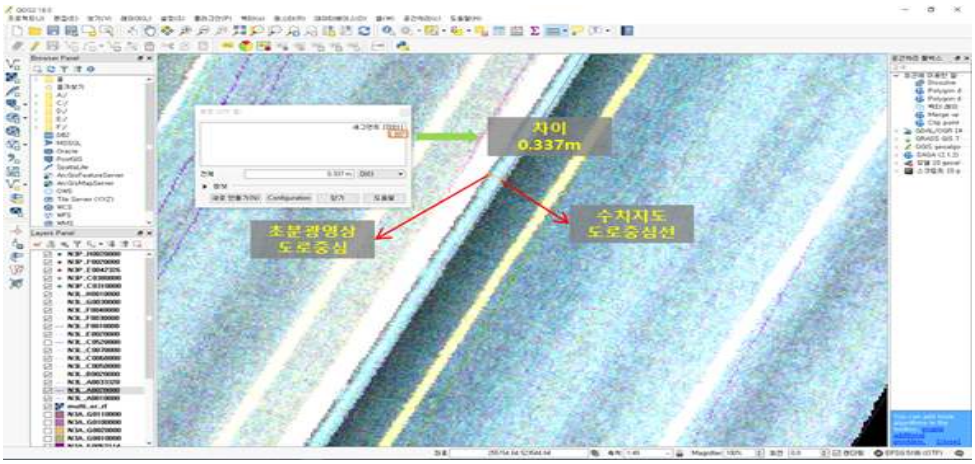
[경기도 용인 양지면 추계리 53-2 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 용인 양지면 추계리 53-2
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준으로 초분광영상의 위치정확도는 0.368m 수준임	

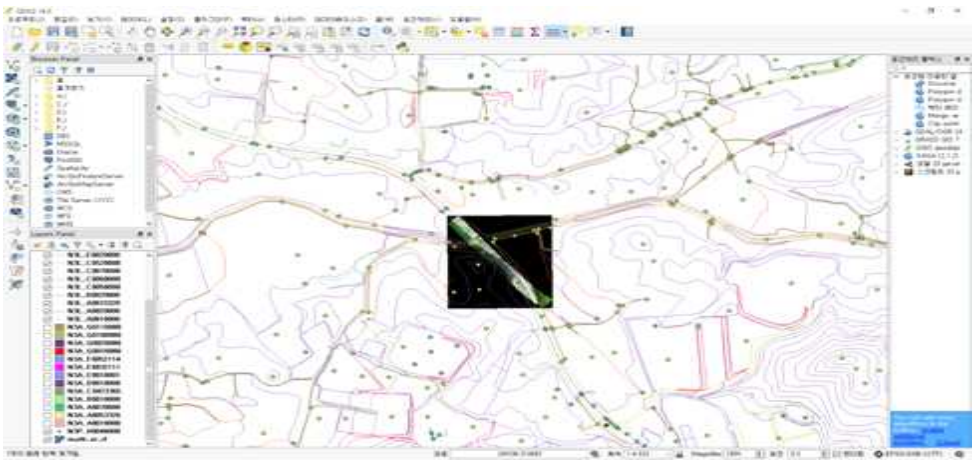
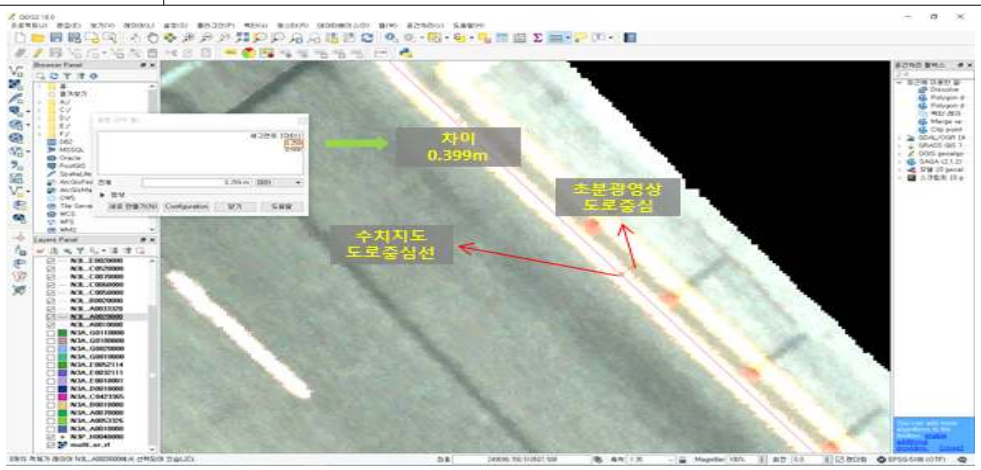
[경기도 여주시 하동 3-2 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 여주시 하동 3-2
	

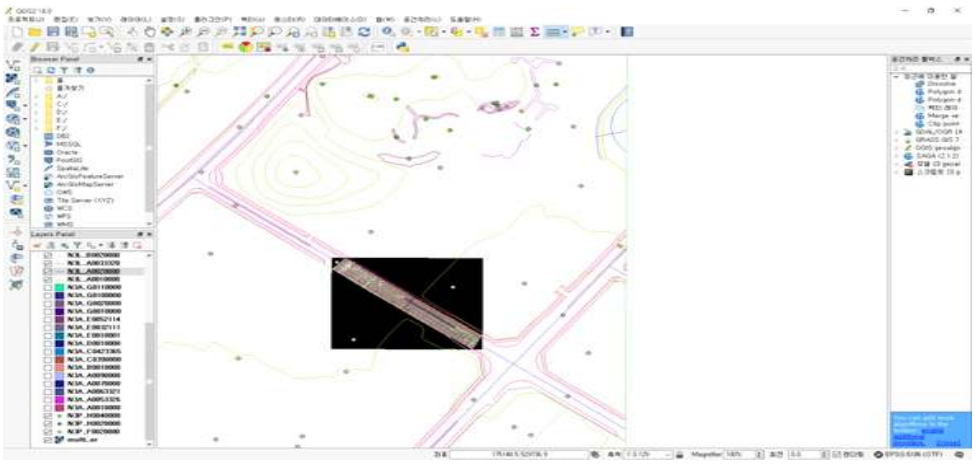
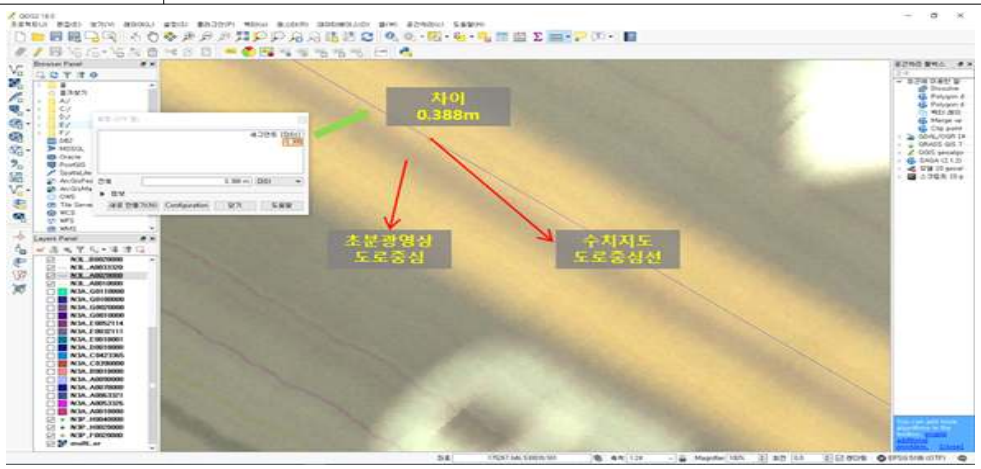
[경기도 여주시 하동 3-2 위치 정확도 검증]

위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준으로 초분광영상의 위치정확도는 0.337m 수준임	

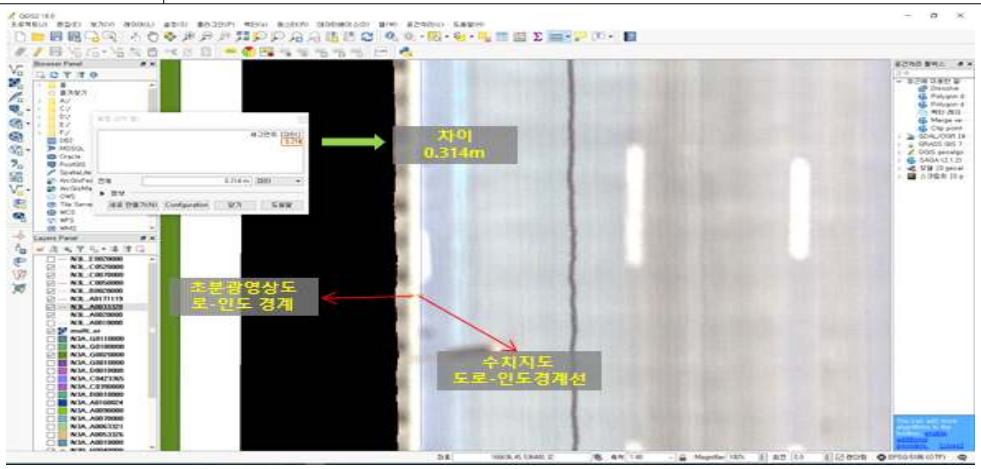
[경기도 여주시 가남읍 심석리 100-1 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 여주시 가남읍 심석리 100-1
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준으로 초분광영상의 위치정확도는 0.399m 수준임	

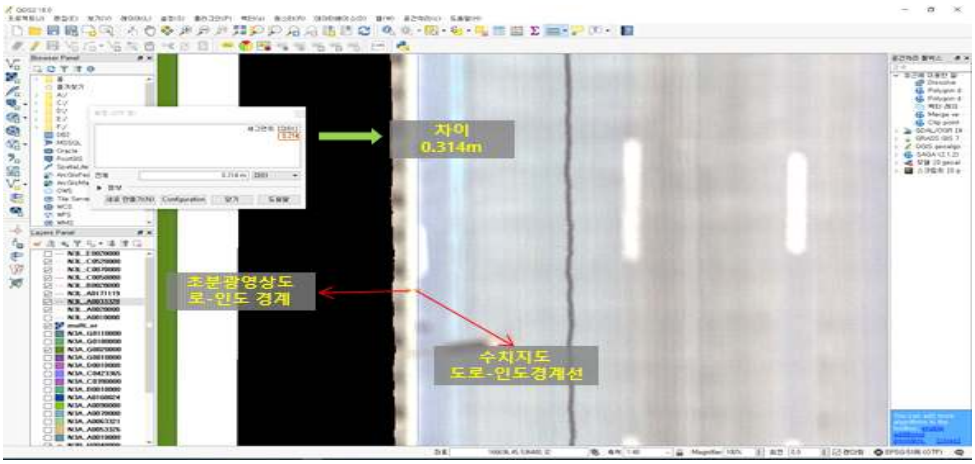
[경기도 시흥시 정왕동 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 시흥시 정왕동
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준으로 초분광영상의 위치정확도는 **0.388m 수준**임	

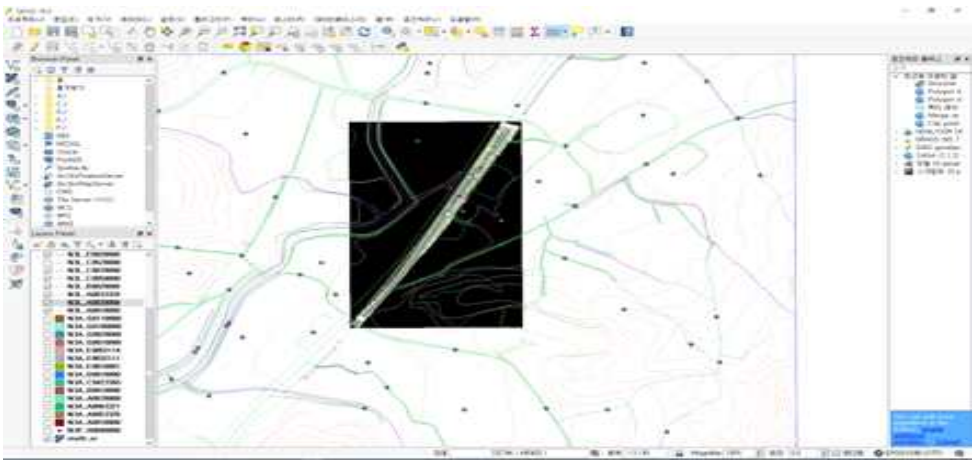
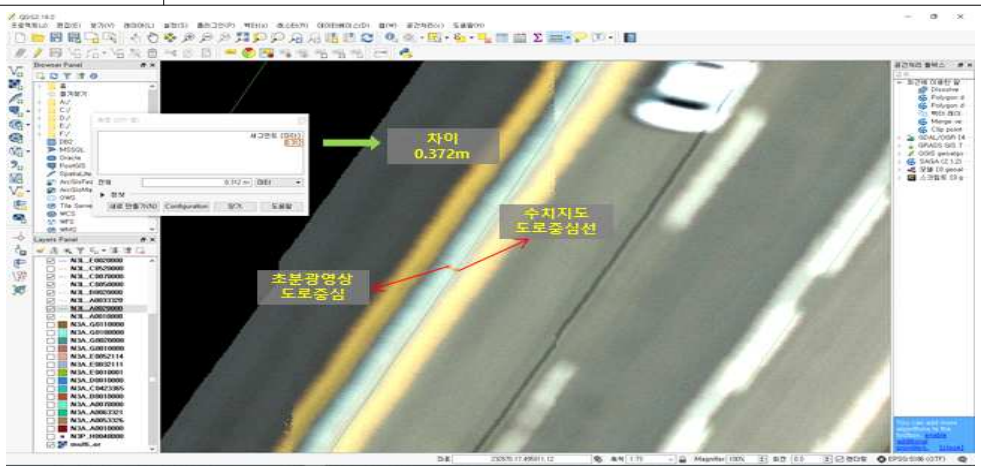
[인천광역시 중구 신흥동3가 80 위치 정확도 검증]

위치정확도 검증	
	
도로-인도 경계선(N3L_A0033320) 기준으로 초분광영상의 위치정확도는 0.314m 수준임	

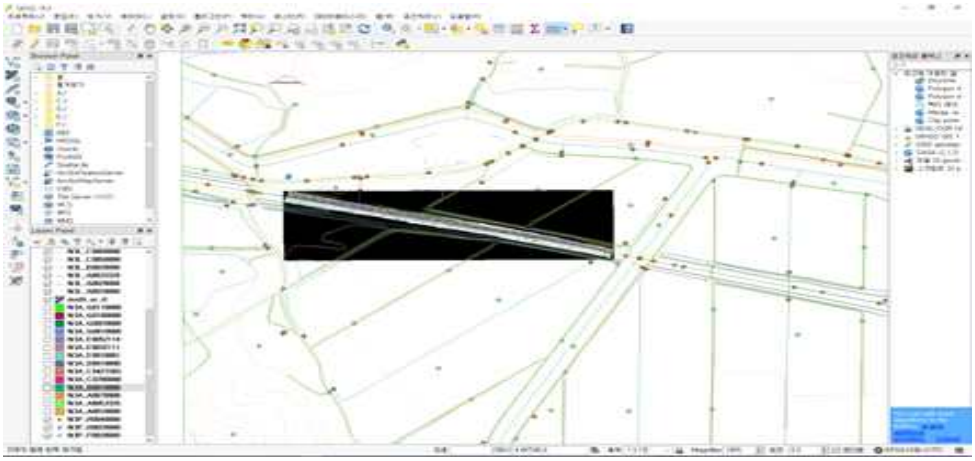
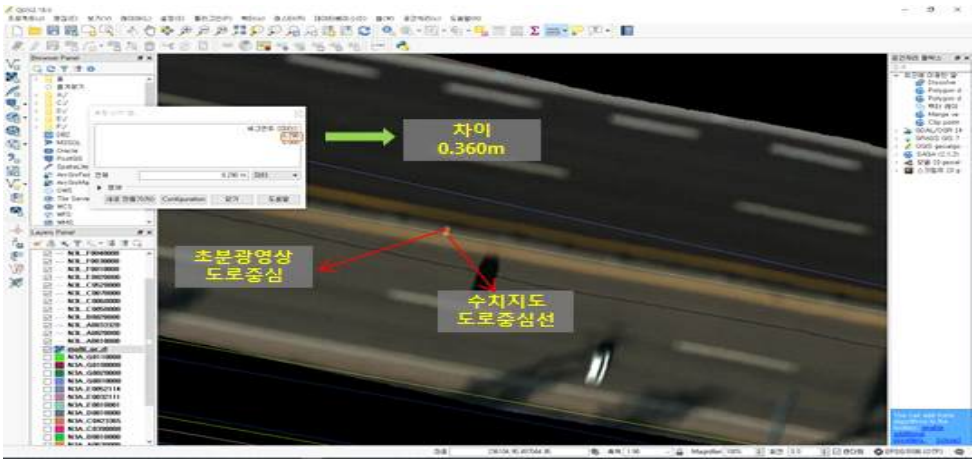
[인천광역시 중구 신흥동3가 80 위치 정확도 검증]

위치정확도 검증	
	
도로-인도 경계선(N3L_A0033320) 기준으로 초분광영상의 위치정확도는 0.314m 수준임	

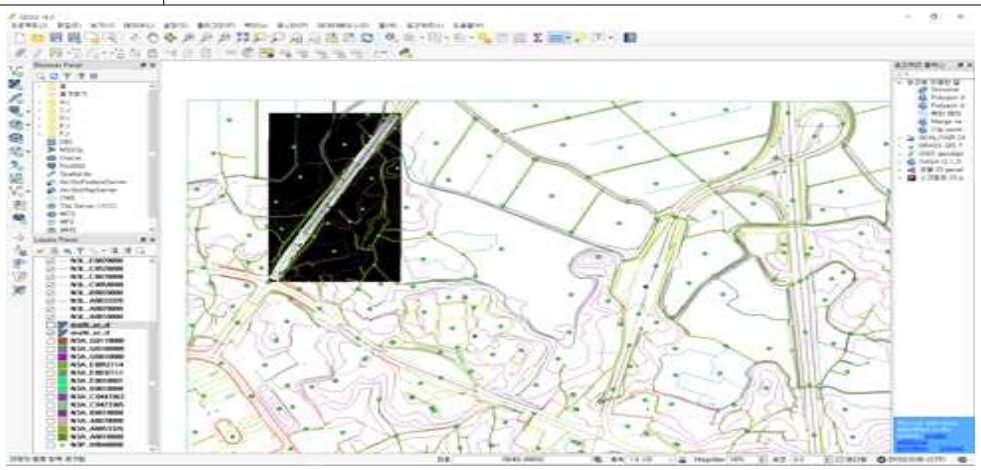
[경기도 안성시 삼죽면 진촌리 307-3 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 안성시 삼죽면 진촌리 307-3
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준으로 초분광영상의 위치정확도는 0.372m 수준임	

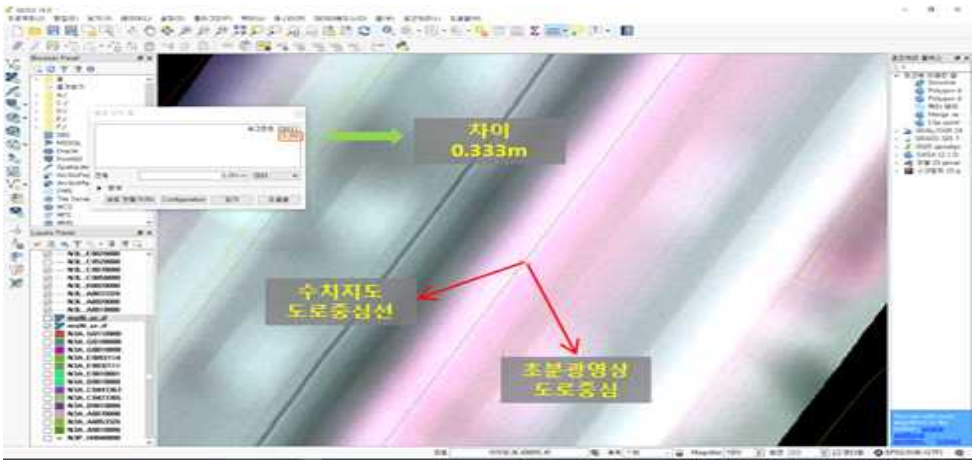
[경기도 안성시 죽산면 두현리 109-1 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 안성시 죽산면 두현리 109-1
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준으로 초분광영상의 위치정확도는 0.360m 수준임	

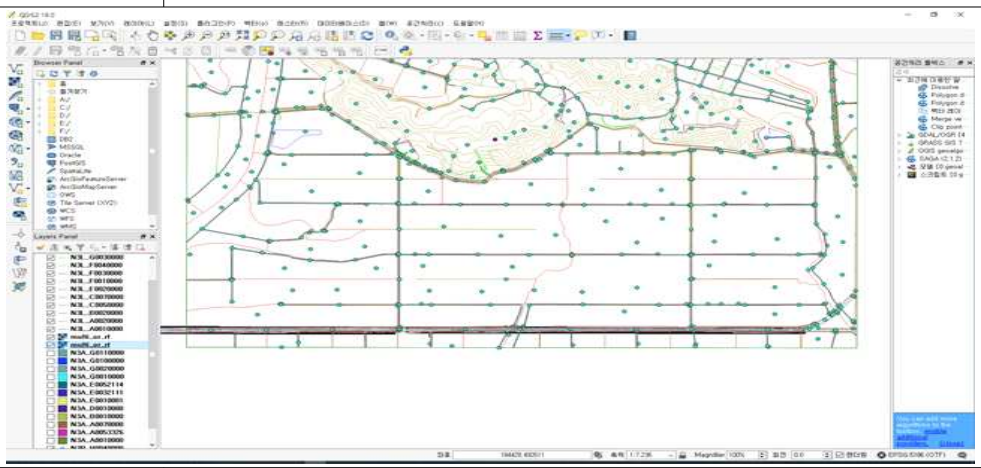
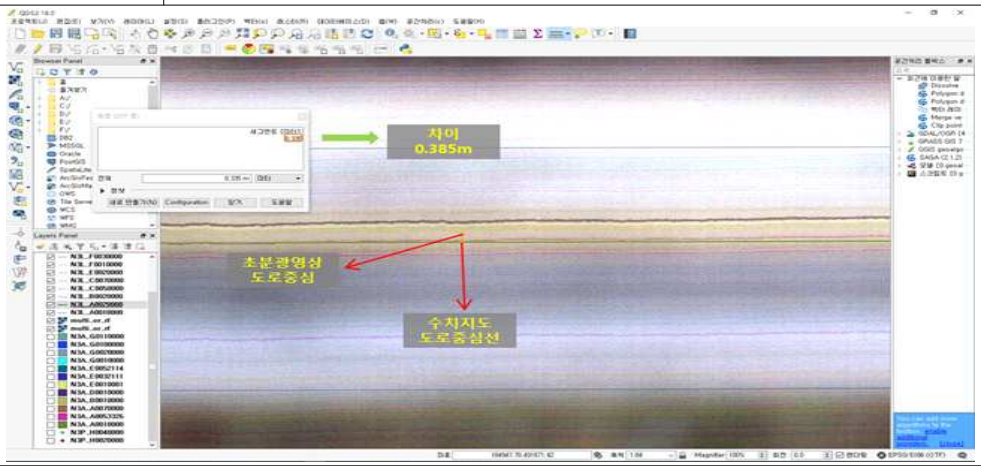
[경기도 평택시 포승읍 도곡리 170-76 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 평택시 포승읍 도곡리 170-76
	

[경기도 평택시 포승읍 도곡리 170-76 위치 정확도 검증]

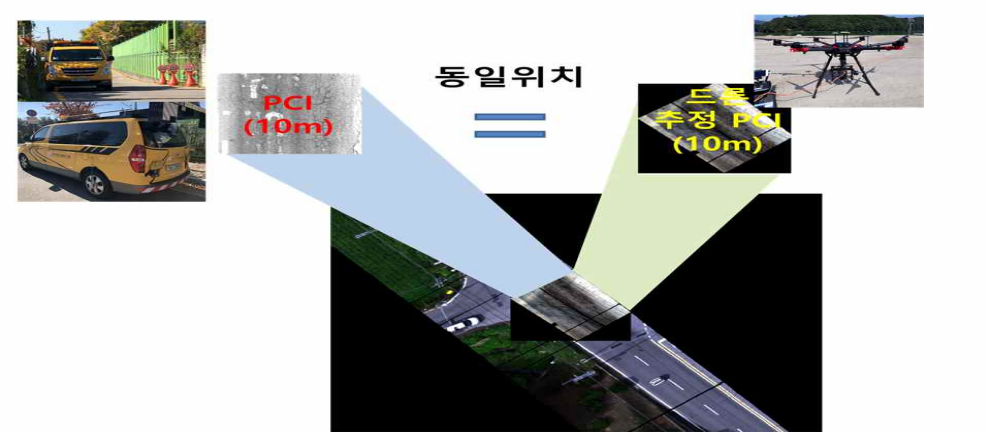
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준으로 초분광영상의 위치정확도는 0.333m 수준임	

[경기도 화성시 장안면 노신리 1416 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 화성시 장안면 노신리 1416
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준으로 초분광영상의 위치정확도는 0.385m 수준임	

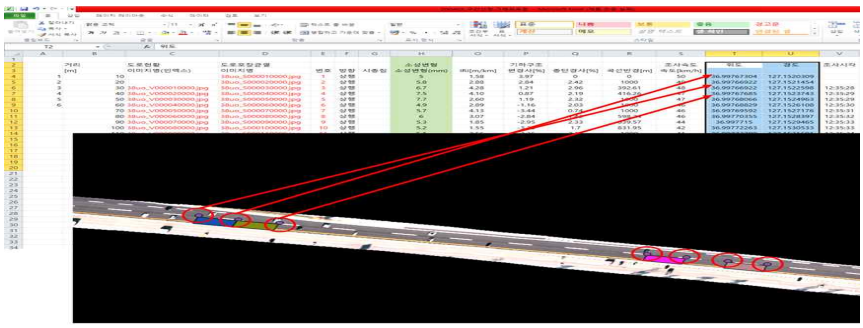
○ 초분광영상 셀단위(10m) 데이터 가공

- 기존 도로포장상태를 정량적으로 나타내는 지수인 PCI는 도로의 10m 단위로 수치값이 주어지고 있으며, 이를 바탕으로 도로포장 유지보수 여부를 판단하고 있음
- 이에 개발된 드론 초분광시스템으로 추정된 PCI의 신뢰성을 판단하기 위해서는 기존의 PCI와 동일한 위치에 도로 분광라이브러리를 제공되어야 함



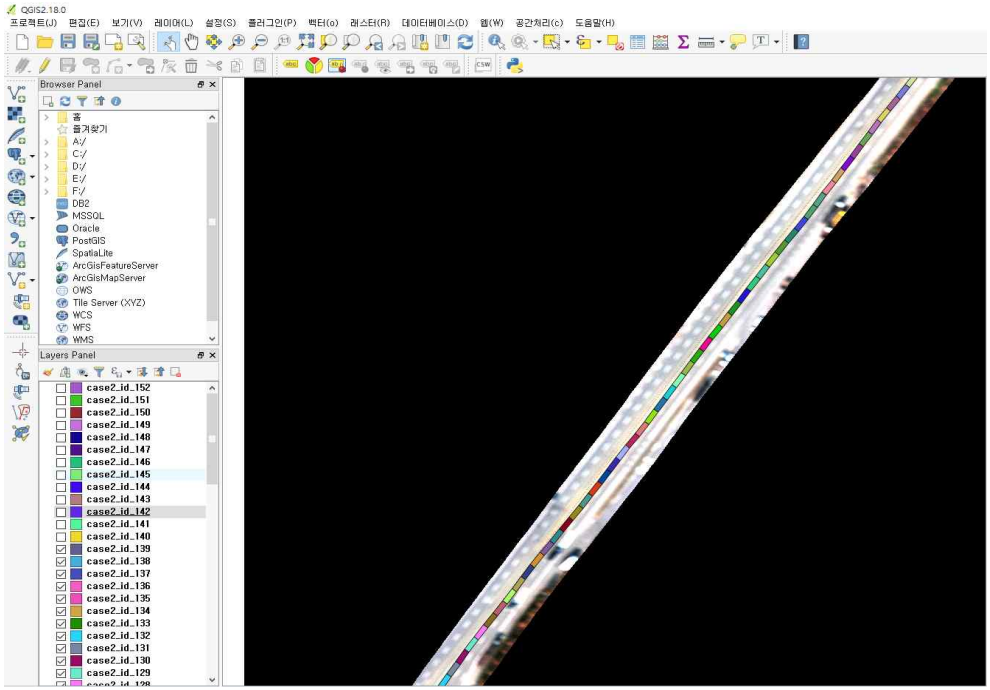
[기존 PCI vs. 드론 기반 추정된 PCI 비교검증]

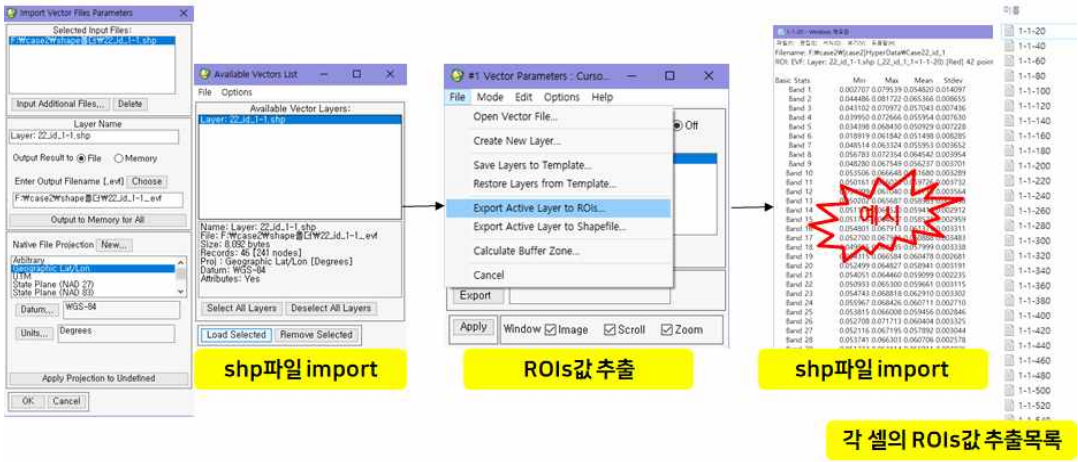
- 도로조사구간에서 기존 PCI의 위치를 알기 위해서는 PES 장비가 도로 데이터 취득 시 차량에 장착 된 GPS를 통해 위치(10m씩)를 아래와 같이 취득하게 됨, 이러한 위치를 기준으로 초분광영상에서 동일한 위치의 도로 분광데이터를 추출함



[PES에 장착된 GPS를 통한 기존 PCI 위치 파악]

- 초분광영상 데이터 가공의 절차는 취득한 초분광영상자료(.hrd) 파일을 백터데이터(.shp) 파일로 변환하여 QGIS SW를 통해 조사된 도로를 10m의 일정한 간격으로 대상지 내 구간분리를 수행한 후 분리한 shp파일을 이용하여 envi 프로그램 내에서 각 셀의 분광영역대 ROI값을 min, max, mean 등 으로 추출하여 기존 PCI값과 비교 분석하는데 활용하였음





[초분광영상 데이터 가공의 절차]

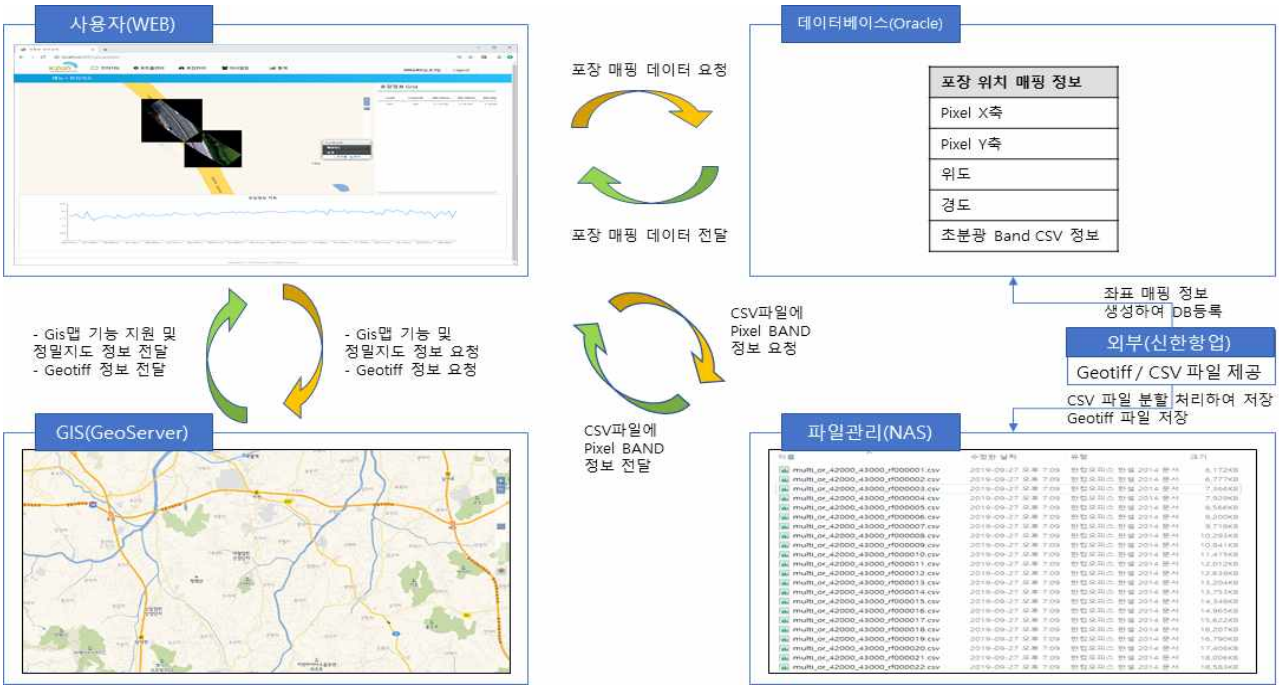
(9) PCI 측정 결과 가시화(GIS맵)

○ 기술 개발 내용
- 포장상태(PCI)가시화 시스템 구축
- 도로정밀지도 기반 포장상태(PCI) 정보 가시화
- 차로별 포장상태(PCI) 위치 오차 1m이내의 지도 API표출
- 초분광 영상데이터 분석방안 연구

○ 서비스 운용을 위한 서버 구축

분류	프레임워크 & 라이브러리	비고
OS	CentOS	가시화 시스템 구축 OS
Framework	Spring Framework, Mybatis, Jquery	가시화 시스템 구성 Framework
WAS	Tomcat	Web application server
Database	Oracle xe 11g, Postgresql	
Devops	Jenkins	시스템 배포 자동화, 운영, 개발의 Devops시스템 구축
Etc	Geoserver	PCI 분석 Geotiff파일 가시화를 위한 웹서비스.

○ 포장상태(PCI)가시화 시스템 구축



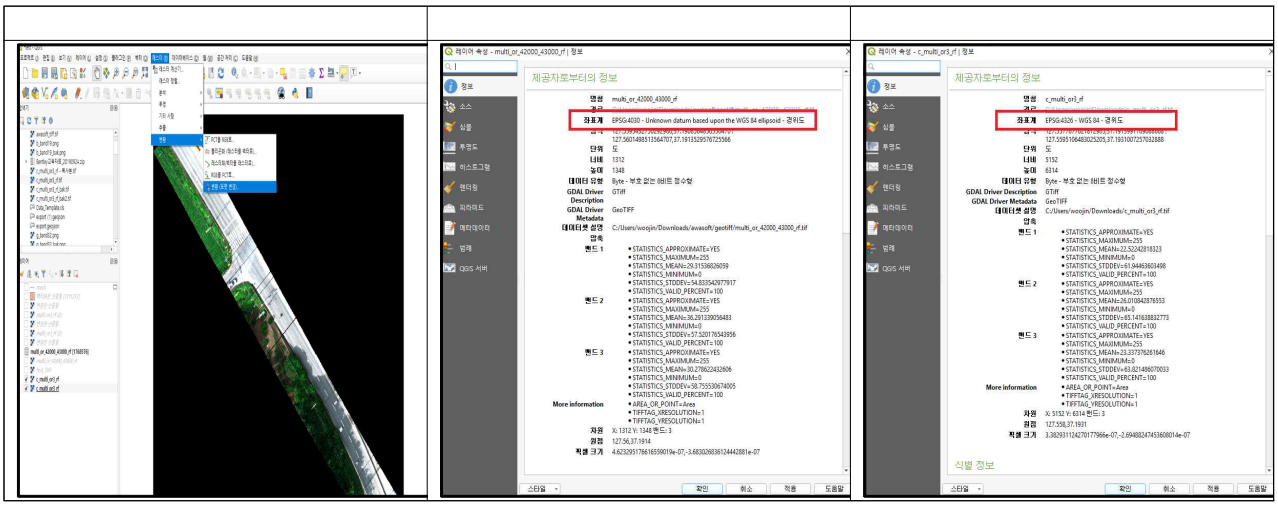
○ 초분광 영상정보 분석 방안
- 초분광 영상자료(.tiff)의 데이터 분석 및 처리방안에 대한 연구 : 신항항업에서 제공된 초분광 영상 원시영상데이터와 Band정보 CSV파일 자료에서 Band정보와 GPS위치정보를 이미지 픽셀좌표정보와 매핑시켜 DB화작업 (Pixel X, Pixel Y, 픽셀의 위도좌표, 픽셀의 경도좌표, RGBA값, Band 파장수치)
- 초분광 영상 DB화
· 1구간 촬영영상의 데이터사이즈 : 5152(width) * 6313(height) * 112(band) = 약 36억건
· DB화한 1구간의 모든픽셀의 Band정보를 출력할 경우 가시화 시스템 웹브라우저의 메모리 Overflow가 발생함, 이를 개선하기 위하여 PCI정보를 압축하여 보여줄 수 있는 다른방안을 연구함.



[Pixel 데이터 약 3만8천건 로딩이미지]

○ 초분광 영상정보 분석 방안 2

- 초분광 영상 정보를 위치 및 데이터 파일형태로 처리
  · 신한항업에서 위치 및 이미지 정보 Geotiff파일과 초분광 Band정보 CSV파일자료를 제공
  · Geotiff파일의 좌표가 EPSG:4030으로 지정되어 있어 QGIS Tool을 사용하여 EPSG:4326좌표계로 변환

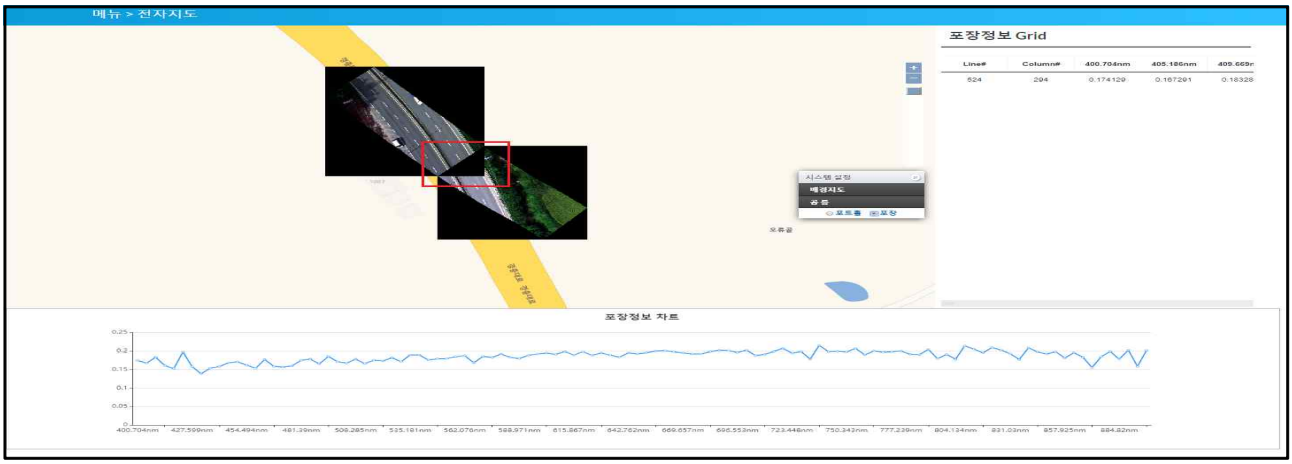


- CSV파일은 Band정보 용량이 Gb단위로 파일 로딩 속도에 문제가 발생, 약 10Mb이하 기준으로 분할하여 파일단위로 인덱싱처리, Pixel단위로 위/경도와 112밴드정보를 추출하여 DB에 등록하여 파일단위 인덱스된 210만건의 매핑데이터를 조회하여 36억건데이터의 저장 및 표출방안을 개선

○ 초분광 영상데이터의 지도레이어 동기화 검증

- QGIS + Geoserver를 활용하영 tiff파일의 지도 레이어 직접 import작업 수행
  · Geoserver에 의한 이미지 좌표정보 매핑 후 출력 시 지도레이어에서 이미지가 겹치는 이슈를 확인함. 또한 매핑된 좌표정보의 왜곡(고도, 렌즈 등등의 영향으로 인한)현상도 발생하여 차선과의 싱크의 오차가 발생함을 확인함
  · 대용량 고해상도 이미지를 웹 브라우저에서 랜더링하려다보니 하드웨어 & 소프트웨어 성능적인 이슈가 발생함. PCI분석데이터에 필요한 것은 구간에대한 GPS좌표 4건과 해당위치의 분석결과이므로 대용량 이미지에서 분석된 결과값는 받아 정보를 시각화하는 방향으로 연구를 진행함.

○ 초분광영상의 구간에대한 GPS좌표를 Polygon 형태로 가공하여 서버 요청 데이터의 최적화, 지도레이어와의 오차를 최소화 하며 구간별 PCI분석결과에 대한 직관적인 UI를 구성함.





[초분광 데이터 가공을 통한 PCI영상분석 구간에 대한 Polygon 레이어 생성]

[2차년도 연구성과 요약]

□ 시계열 SAR자료 수집/처리 분석기술(알고리즘) 및 정밀 변위 해석기술 개발

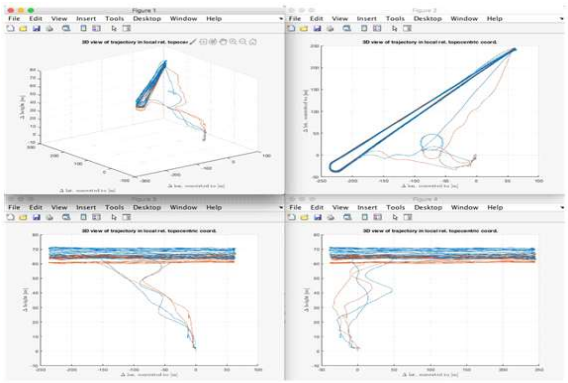
○ 위상간섭기법을 이용한 비탈면 정밀 변위 해석기술 개발

○ band SAR 드론 시스템 시험 운영 및 데이터 분석

○ 정밀 변위 측정 소프트웨어 적용 기술 개발 구현



[VTOL UAV를 이용한 L-밴드 SAR 드론시스템 시험 운영(19.2.13, 19.3.28, 2회 수행)]

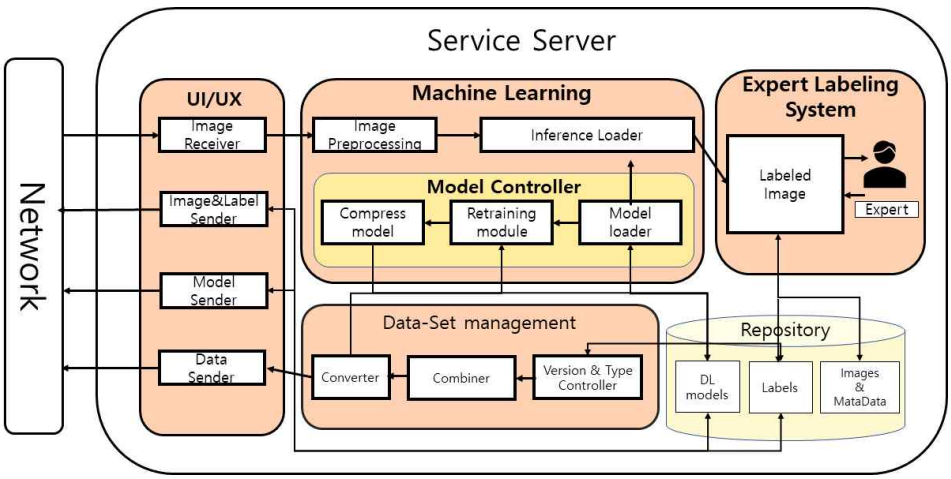


[관성항법장치/GPS 시스템을 이용 정밀 측정한 드론시스템 비행 궤적]

□ 드론 영상 데이터 기반 포트홀 탐지 소프트웨어 개발

○ 포트홀 자동탐지 결과 관리시스템 관련 연구

○ 도로 항공 영상에서 포트홀 탐지 결과 관리시스템 구현



[관리시스템 아키텍처]

□ 드론을 활용한 포장 노후화 상태(PCI) 측정 기술 개발

○ 아스팔트 시편 및 코어 초분광 지수 통계 분석 모델 선정 및 개선

○ 초분광 실험결과의 분석을 위해 총 7종의 분석 모델식 선정

종류	모델식
VIS1(Visible Imaging Spectrometer 1)	$\mathrm{VISI1}=\left(\frac{\rho740-\rho460}{\rho740+\rho460}\right)\mathrm{VISR1}=\frac{\rho460}{\rho740}$
VIS2(Visible Imaging Spectrometer 2)	$\mathrm{VISI2}=\left(\frac{\rho830-\rho490}{\rho830+\rho490}\right)\mathrm{VISR2}=\frac{\rho490}{\rho830}$
VIS3(Visible Imaging Spectrometer 3)	$\mathrm{VISI3}=\left(\frac{\rho375-\rho595}{\rho375+\rho595}\right)\mathrm{VISR2}=\frac{\rho375}{\rho595}$
PVISI2(Pavement Visible Imaging Spectrometer Index 2)	$\mathrm{PVISI2}=\frac{\rho899}{\rho967}$

[3차 년도]

가. 비탈면 변위측정 기술

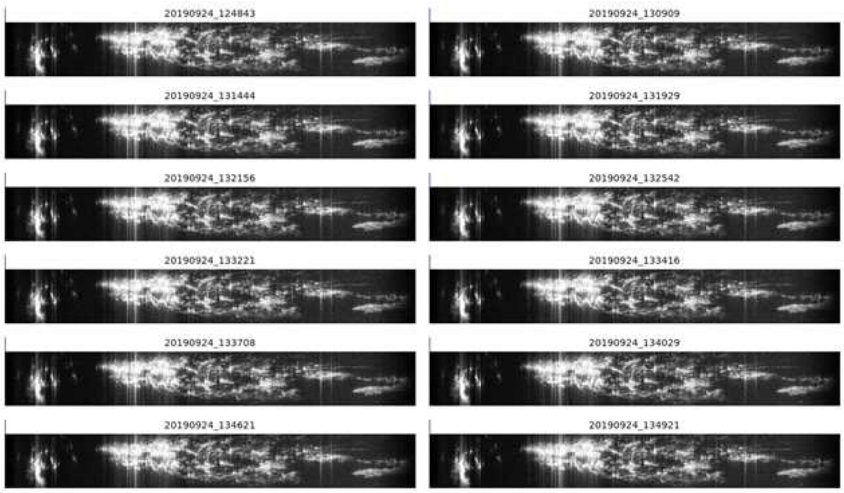
(1) 드론 SAR 데이터 기반 비탈면 변위 측정 기술

○ 시계열 L-band RailSAR 자료 수집
- GAMMA에서 제공하는 Chutz, Belpberg near Muensigen, Bern Switzerland에서 2019년 9월 24일에 획득한 RailSAR 원시 데이터셋을 획득



[데이터셋을 획득한 스위스 베른 근처의
Chutz, Belpberg]
(출처: Google Earth)

- GAMMA L-band RailSAR 시스템으로 획득한 원시 데이터셋은 Deramped radar data를 포함하는 이진 데이터와 레이더 데이터에 대한 추가 정보가 있는 매개변수 파일로 구성



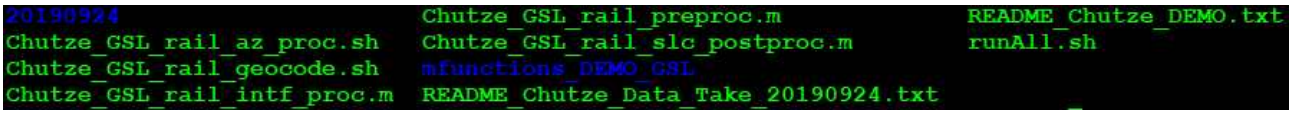
[수집된 RailSAR SLC 영상]

○ 시계열 L-band RailSAR 원시 데이터 처리

- 원시 데이터의 지정된 채널을 추출하여 힐버트 변환 및 range spectral windowing을 수행
- 힐버트 변환은 선형 필터로써 모든 주파수에서 크기는 변하지 않고 (all-pass filter) 위상만 음의 주파수 일 때는 π/2 , 양의 주파수 일 때는 -π/2 만큼 바꿔줌. 이를 통해 실수 신호를 복소수 차원으로 확장시킬 수 있게 되어 진폭과 위상의 분석이 쉬워지게 됨
- 레일을 따라 측정된 각 펄스에서 위치와 속도를 계산하고 ASCII 파일로 위치와 속도를 작성
- 극좌표 재구성 그리드에 초점을 맞춘 레일 기반 GAMMA L-band SAR 데이터에 대한 극좌표 재구성 하여 그리드를 생성함
- Time-Domain Back Projection (TDBP) SAR processor을 사용하여 SLC SAR 영상 포커싱을 수행
- Back projection 알고리즘은 SAR 포커싱을 수행하기 위해 SAR 데이터의 시간 영역 컨볼루션을 사용하는 SAR 처리 방법으로서 TDBP azimuth focusing의 핵심 처리 단계는 다음 단계로 구성 (Othmar Frey, 2014)
  1) 센서 위치 및 속도 데이터를 읽음
  2) 이미지 재구성 그리드의 좌표를 읽음
  3) SLC 이미지 블록을 위해 GPU 장치에 메모리를 할당
  4) 레이더 에코 데이터, 도플러 중심 값 및 에코 당 위상 보정 값을 위해 GPU 장치에 메모리를 할당
  5) Back projection을 수행하기 위해 레이더 에코를 반복
     · 루프의 현재 레이더 에코와 일치하도록 range-azimuth 변화 도플러 중심 값을 업데이트
     · 각 azimuth 에코에 대해 업데이트 되는 다양한 위상 보정 값을 업데이트
     · 레이더 에코를 장치 메모리에 복사
     · cuFFT 라이브러리를 사용하여 GPU 장치에서 레이더 에코의 FFT 기반 보간
     · 업샘플링된 에코의 베이스 밴드에서 대역 통과 변환 (시간 지연 도입)
     · 현재 위치와 재구성 그리드의 좌표 사이의 범위 거리를 계산
     · 보간된 에코를 중간 SLC 이미지 그리드로 back projection
     · 중간 이미지 그리드에 저장된 현재 back projection 데이터로 SLC 이미지를 업데이트
  6) Back projection 루프의 끝
  7) SLC 이미지의 긴밀 베이스밴드 변환
  8) GPU 장치 메모리에서 호스트 메모리로 집중된 SLC 이미지를 복사
- SLC 파라미터 파일을 기반으로 제작 후 azimuth spectral windowing 처리

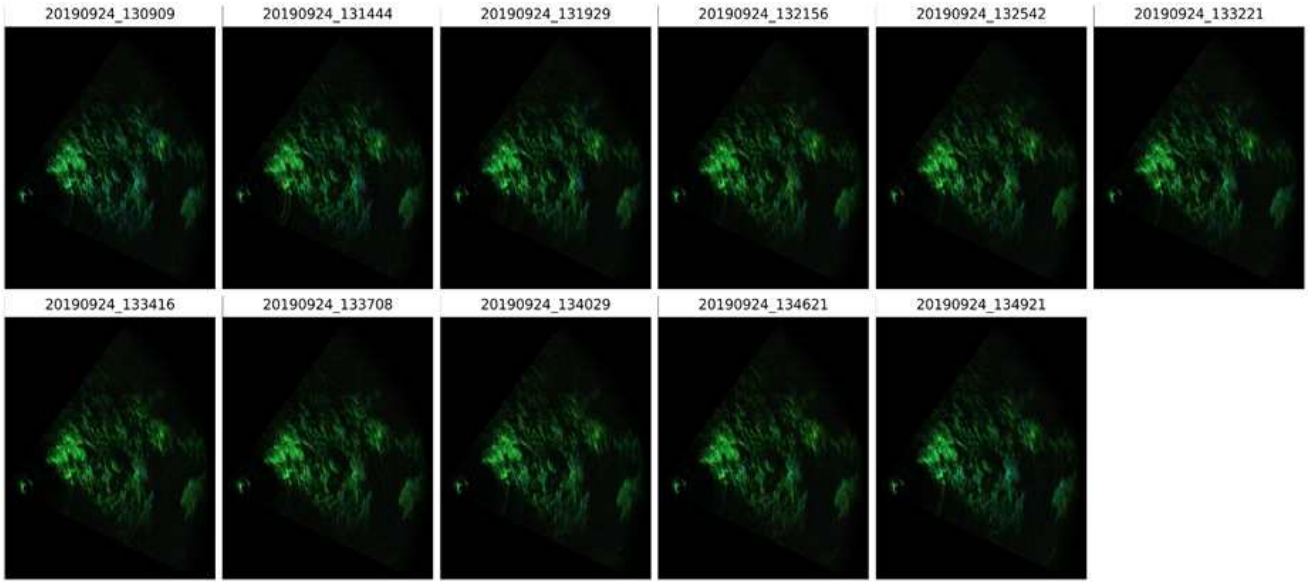
○ 시계열 L-band RailSAR 원시 데이터 정밀 변위 해석 기술

- slc 파일을 처리하고 간섭도를 제작



[데모 데이터 처리에 사용되는 명령어]

- 총 12개의 SLC 영상으로부터 계산한 11개의 위상간섭도를 제작함
- 동일한 날 관측한 영상을 테스트한 것으로 특이한 지표 변위가 발견되지 않으며 영상레이더 신호를 통해 지표변위를 효과적으로 측정할 수 있음을 간접적으로 확인할 수 있음. 또한 지표변위가 없음을 신호를 통해 확인함으로써 영상레이더 센서의 품질 정확도를 평가할 수 있음



[데모 데이터를 처리하여 획득한 간섭도]

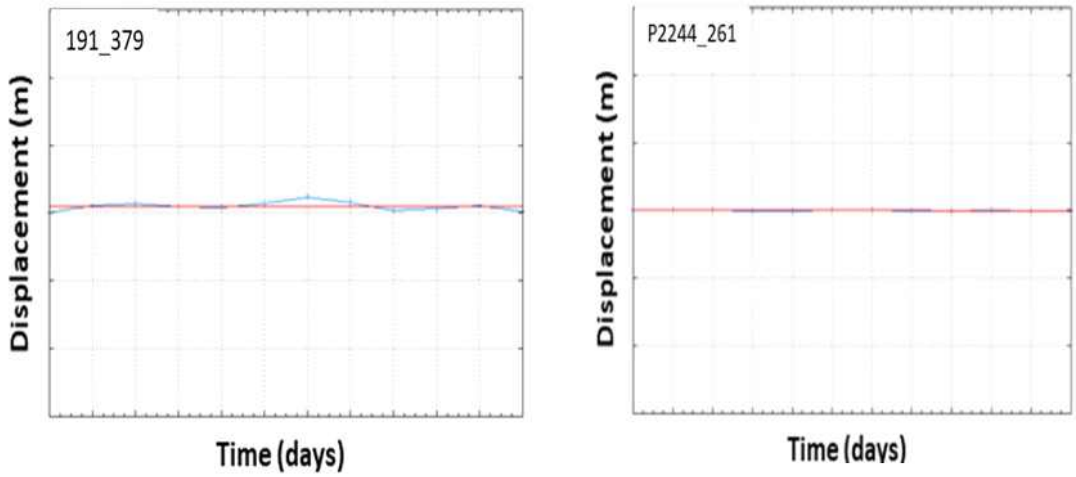
- Small Baseline Subset (SBAS) 시계열 지표변위 분석을 위한 위상 간섭도 작성
- 테스트를 위해 가능한 모든 기선거리 조합에 대해 간섭도를 제작
- 동일한 궤도에서 관측한 영상이므로 시간에 대한 베이스라인만 조정하여 사용하였음



[SBAS 시계열 분석을 위한 위상간섭도 제작]

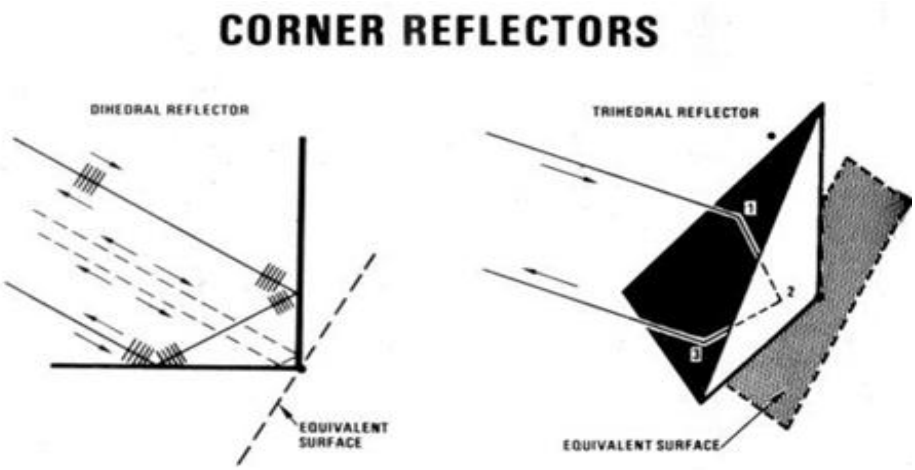
- 총 12장 영상레이더로부터 지표면 시계열 변화 관측 연구 수행
- GSL GAMMA 파일 포맷을 이용한 pseudo-script 작성 완료하였으며 다음은 영상 처리 순서와 설명을 정리한 것임
- mk_tab을 이용하여 slc 영상 및 파라미터 파일에 대한 목록 작성 (RSLC_tab)
- base_calc를 이용하여 각 영상들에 대한 수직 기선거리 및 delta_T 값을 계산하고 각 간섭도에 대한 SLC를 지정하는 간섭도 표를 작성 (itab_mb)
- 각 영상에 multi-look을 진행시키고 MLI 평균 영상을 생성함 (rmli_1_1.ave)
- mk_base_calc 명령어와 제작된 RSLC_tab과 itab_mb를 이용하여 궤도 기선거리를 계산하고 수직 기선거리의 리스트를 나타냄 (ibtab_mb.bperp)
- mk_diff_2d_railsar를 통해 2D 차분 간섭도를 계산
- mk_adf_2d를 통해 위에서 제작한 2D 차분 간섭도 세트에 대한 적응형 필터링 (adaptive filtering)을 수행 (adf_cc)

- 제작된 적응형 correlation coefficient 영상에 대한 평균 영상을 생성 (adf_cc.ave)
- correlation coefficient 영상 (adf_cc.ave)과 위에서 제작한 MLI 평균 영상 (rmli_1_1.ave)을 이용하여 8비트 래스터 그래픽 이미지를 생성
- Minimum Cost Flow (MCF) 알고리즘을 이용하여 위상간섭도 스택에 대한 unwraping을 수행
- mb를 통해 다중 기선거리 (multi-baseline) 연속 간섭도의 세트로부터 위상 시계열을 계산
- 계산된 시계열을 래스터 영상으로 바꾼 뒤 vu_disp2d를 이용하여 변형 시계열을 래스터 영상을 이용하여 나타냄
- 드론 영상레이더의 경우 지리정보를 가지고 있으므로 geocoding 없이 기하 보정된 시계열 변화 분석 지도 제작 가능
- 동일한 지역에서 동일한 시기에 관측된 영상레이더 자료를 활용하였으므로 시계열 영상에서 거의 변화가 없는 모습을 보여주고 있음



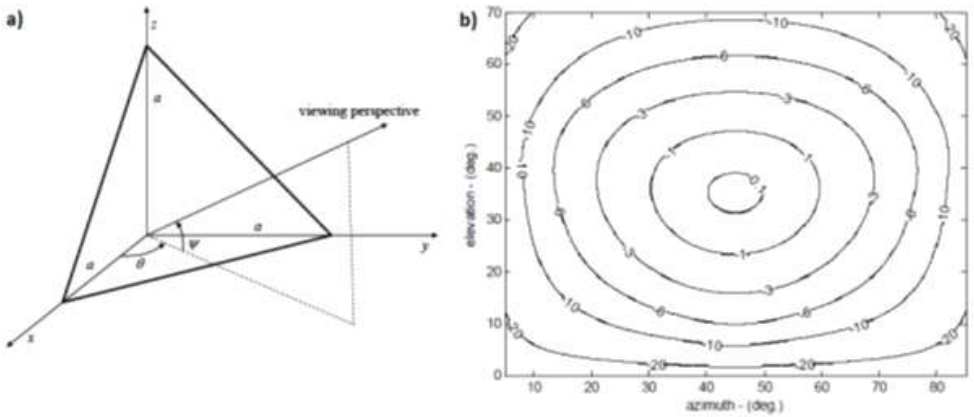
[12장의 영상레이더를 이용한 지표면 시계열 변화 관측 결과]

○ 정밀위치자료와 고해상도 DEM을 이용한 영상레이더 정밀 보정 정사영상 자료 생성
- 영상레이더 안테나로부터 전송된 레이더 신호는 표면 상태, 표면 형태, 그리고 산란 메커니즘에 따라 다양한 방향으로 산란되고 전송된 레이더 신호 중 오직 작은 부분만 안테나로 산란되어 돌아감
- 코너리플렉터는 수신된 전자기파 에너지를 직접 반사하는 수동 장치로 영상 내 낮은 반사 신호를 향상시키거나 SAR 영상의 기하/방사 보정에 적합하고 위상간섭기법을 적용하여 지면 변형을 정확하게 측정하는 데 활용 가능



[이면체/삼면체 코너리플렉터의 형태와 신호 산란 형태 도식]
(Shimon, 2016)

- 최대 반사 산란 신호를 얻기 위해서 코너리플렉터는 안테나 송신 신호를 지향하도록 설계하고 설치되어야 함

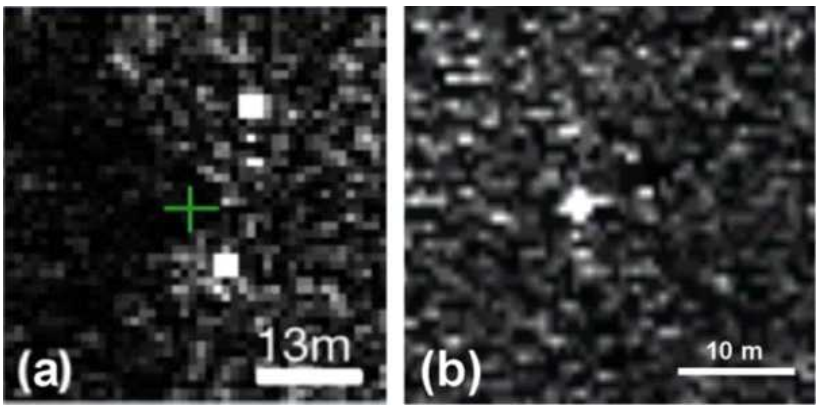


[(a) 기준 방향에서의 삼각 삼면체 코너리플렉터의 관측 형태. a는 inner leg dimension을 의미함. (b) 피크 RCS에 대한 azimuth (Θ) 및 고도 (ψ) 각도의 함수로서 삼각 코너리플렉터의 RCS 반응]
(Shimon, 2016)

- Clutter은 전자 시스템, 특히 레이더에서 원하지 않는 신호 성분을 말하며, 땅, 바다, 비, 동물/곤충, 대기 등에 의해 발생하는 잡음 현상임
- 코너리플렉터는 레이더 관측물의 진폭과 위상 모두에게 영향을 주며, 진폭 영상에서 코너리플렉터는 아주 작고 밝은 물체로 나타남
- 진폭 강화 수준은 보통 Signal-to-Clutter Ratio (SCR; Freeman, 1992)로 정의됨

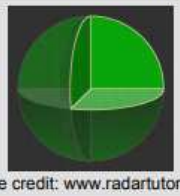
$$SCR = \frac{\sigma_T}{<\sigma_c>} = \frac{\sigma_T}{<\sigma^0> A}$$

- σT는 점 표적의 radar cross section (RCS)이고, <σc>는 점 표적 근처의 clutter RCS의 앙상블 평균
- 이론적으로 코너리플렉터의 가장자리에서의 신호 산란 길이는 코너리플렉터 중심부에서의 산란 길이와 같아 안정화된 위상 신호를 관측할 수 있으나 현실에서는 위상 측정에 코너리플렉터 신호 길이의 미세 변화 차이로 인해 SCR과 반비례하는 오차가 발생



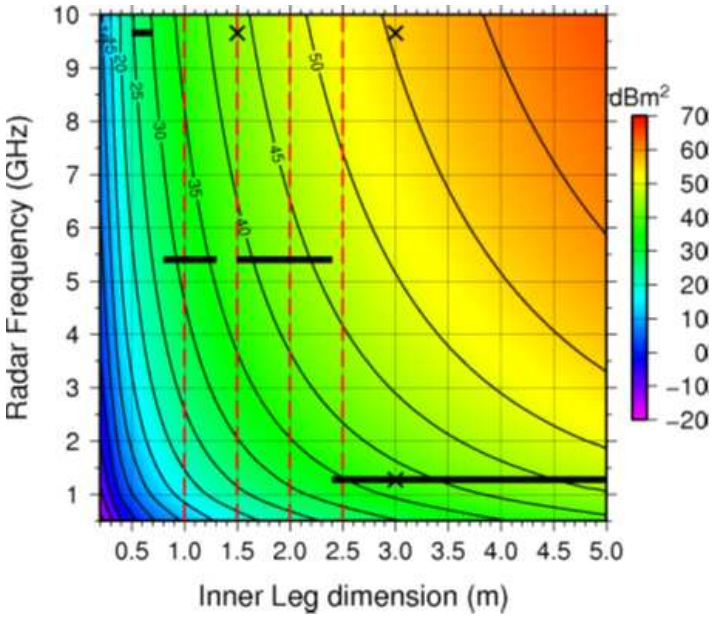
[(a) Sand Hill Scout reservation에 있는 연구지역 코너리플렉터의 TSX Spotlight 진폭 영상, (b) 콜로라도 코너리플렉터 테스트 지역의 TSX spotlight 영상]
(Shimon, 2016)

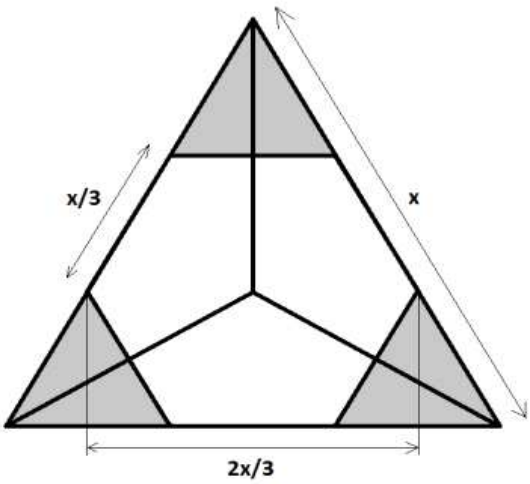
○ 정밀위치자료와 고해상도 DEM을 이용한 영상레이더 정밀 보정 정사영상 자료 생성을 위한 코너리플렉터 제작

Target	Example	Maximum Theoretical RCS (dBm^2)	RCS magnitude factor
Triangular Trihedral	 Image credit: Geoscience Australia	$\sigma_T = \frac{4\pi a^4}{3\lambda^2}$	4.19
Flat square plate	 Image credit: *Drake and Hatty* [2013]	$\sigma_T = \frac{4\pi a^4}{\lambda^2}$	12.57
Circular Trihedral	 Image credit: www.radartutorial.eu	$\sigma_T = \frac{0.507\pi^3 a^4}{\lambda^2}$	15.92
Dihedral	 Image credit: *Ferretti et al.* [2007]	$\sigma_T = \frac{8\pi a^4}{\lambda^2}$	25.13
Square Trihedral	 Image credit: *Qin et al.* [2013]	$\sigma_T = \frac{12\pi a^4}{\lambda^2}$	37.70

[코너리플렉터 디자인에 대한 예시와 이론적 최대 RCS]
(Garthwaite et al., 2015)

- 코너리플렉터의 효율성은 디자인, 면적, SAR 신호의 파장, 그리고 공간 해상도에 의존
- 높은 공간 해상도는 코너리플렉터의 신호를 강화할 수 있는 반면, 낮은 공간 해상도는 코너리플렉터의 효과를 감소시킴
- 평평한 판이나 이면체 코너 리플렉터는 높은 RCS 값을 보이지만 azimuth 방향으로의 아주 좁은 산란 패턴으로 신호가 분산될 가능성이 있어 매우 정확한 정렬이 요구됨
- 삼면체 코너리플렉터의 경우 azimuth와 고도 방향에서 모두 유연성을 제공하나 전송하는 레이더 에너지 소스를 향해 15도 이내일 때 가장 잘 작동하며, 최대 산란은 삼면체가 정확하게 소스를 향했을 때 일어남
- 삼면체 코너리플렉터의 경우 다른 형태의 코너리플렉터에 비해 정렬에 구애를 덜 받아 보다 자주 사용되며 본 제작에서도 사용됨
- 삼면체 코너리플렉터에서 판의 모양은 RCS 크기에 큰 영향을 줌
- 주어진 크기에서 밝기는 사각형 > 쿼터서클 > 삼각형 순으로 밝음
- 주어진 크기에서 내구도와 수명은 삼각형 > 쿼터서클 > 사각형 순으로 높음
- 코너리플렉터의 수명이 가장 길고 제작 재료가 적게 드는 삼각 삼면체 코너리플렉터를 선택
- RCS는 코너리플렉터의 크기에 의존하며 동일한 SCR 유지를 위해 신호의 파장과 코너리플렉터의 크기를 고려할 필요가 있음





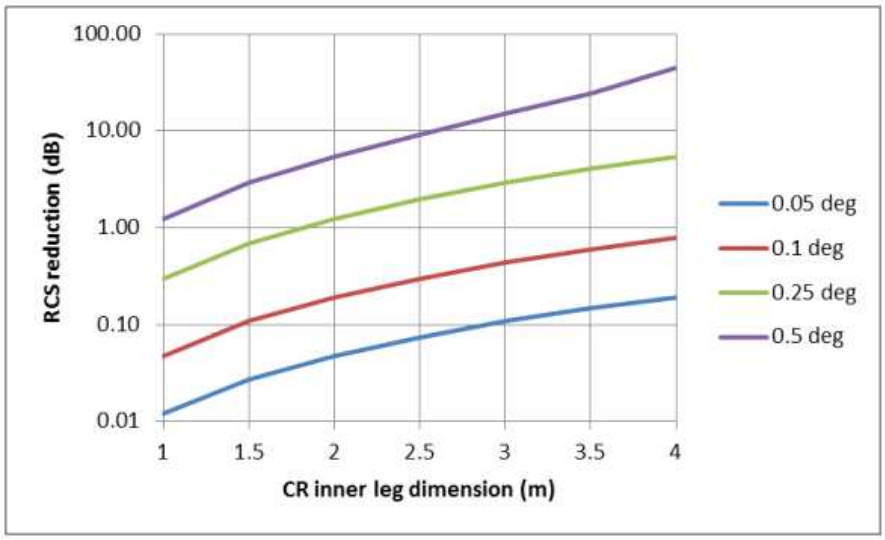
[삼각 삼면체 코너리플렉터의 유효 영역]
(Garthwaite et al., 2015)

- 위 그림에서 검은 수평선은 각각 X-, C-, 그리고 L-band 신호에 대한 반사도 요구사항을 충족하는데 필요한 삼각 삼면체 코너리플렉터의 크기를 표시한 것
- 드론 영상레이더의 L-band 주파수에 대한 코너리플렉터의 RCS의 평가를 위해 다양한 크기의 코너리플렉터를 제작



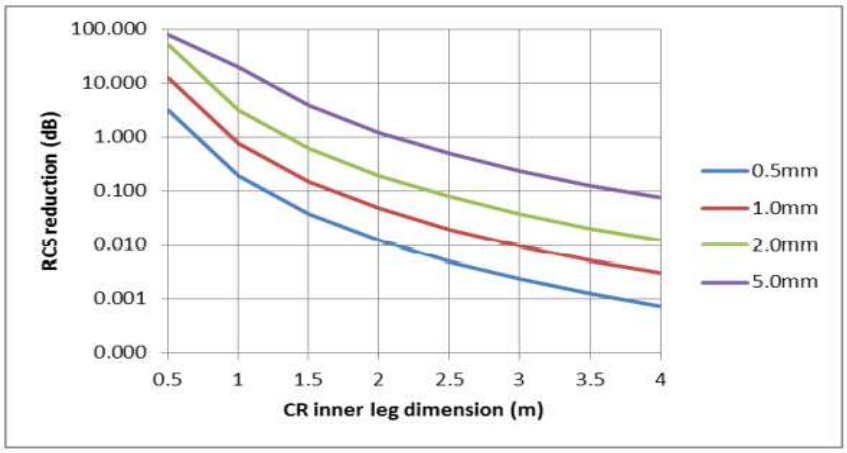
[제작한 0.5m, 1.0m, 1.5m 삼각 삼면체 코너리플렉터 판]

- 코너리플렉터 제작 과정에서 RCS에 영향을 줄 수 있는 요인에 대한 관리가 필요
- 개별 판 사이의 직교성을 유지하지 않을 경우 코너리플렉터의 RCS가 급격하게 감소할 수 있음



[X-band 에서의 삼면체 코너리플렉터 판 사이 직교 오류에 대한 RCS 감소] (Garthwaite et al., 2015)

- 판 곡률은 전체 길이를 따라 완전히 평평한 평면에서 판이 변형되는 정도를 나타내는 것으로 판 곡률에 의한 RCS 감소는 파장 및 코너리플렉터의 크기에 의존함

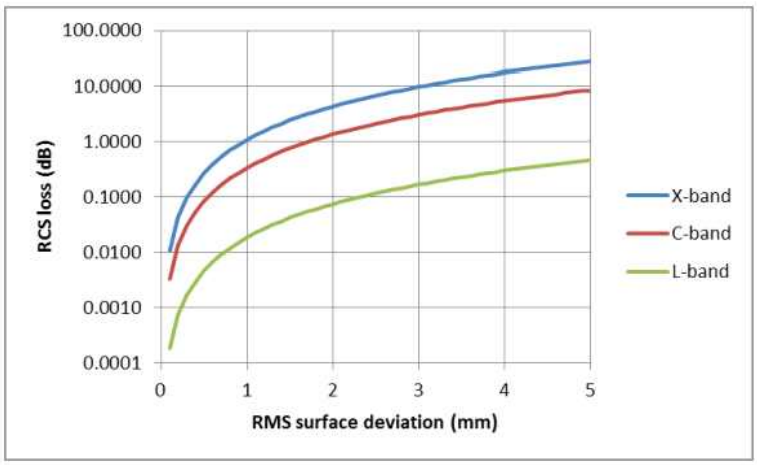


[X-band 에서의 삼면체 코너리플렉터 판 곡률 편차에 의한 RCS 감소] (Garthwaite et al., 2015)



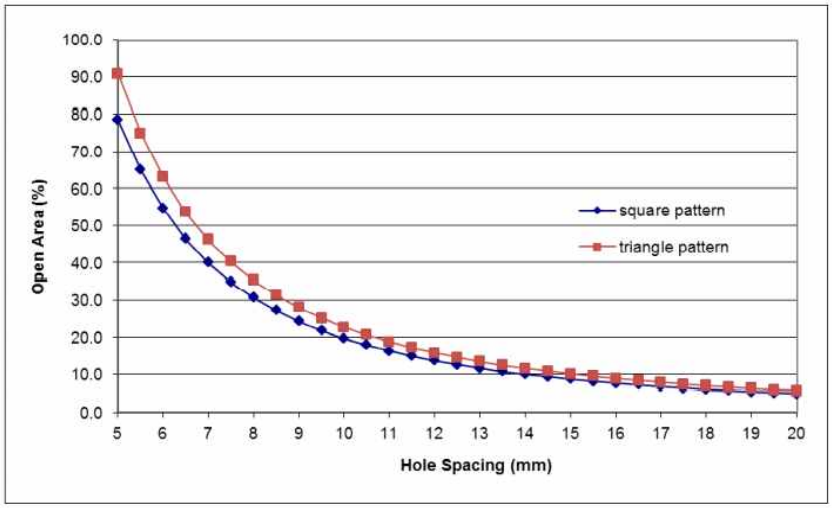
[판의 평평함을 유지하기 위해 판의
뒷면에 부착한 부착물]

- 판 표면의 요철 등의 불규칙성으로 인한 RCS 감소는 파장에 의존하고 타겟의 크기와 상관없음



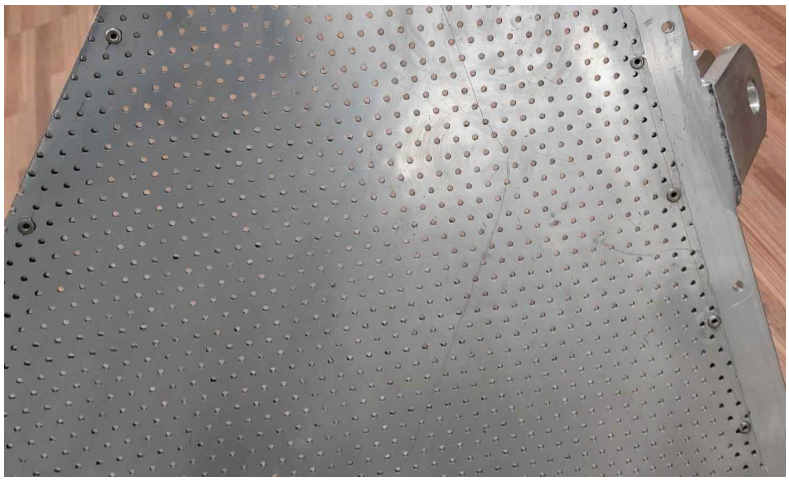
[판 표면 불규칙성으로 인한 RCS 감소]
(Garthwaite et al., 2015)

- 코너리플렉터 판을 제작하는 데에는 일반적으로 알루미늄이 사용됨
- 메쉬 천공은 바람에 의해 코너리플렉터 구조물에 가해지는 힘의 일부를 완화하고 강우로 인한 빠른 배수가 가능하며, 먼지 및 기타 바람에 의한 침전물의 자체 청소를 위해 제작됨
- 구멍의 직경이 커질수록, 구멍 사이의 간격으로 인한 개방 면적이 커질수록 RCS가 감소함
- 코너리플렉터의 RCS에 큰 영향을 주지 않기 위해 구멍의 직경이 파장의 1/6 보다 작아야 함
- X-, C-, L-band 모두에서 사용 가능한 코너리플렉터 제작을 위해 X-band의 약 1/10, C-band의 약 1/20에 해당하는 2.8 mm 직경, 12 mm 간격, 개방 영역 4.9 %로 천공함



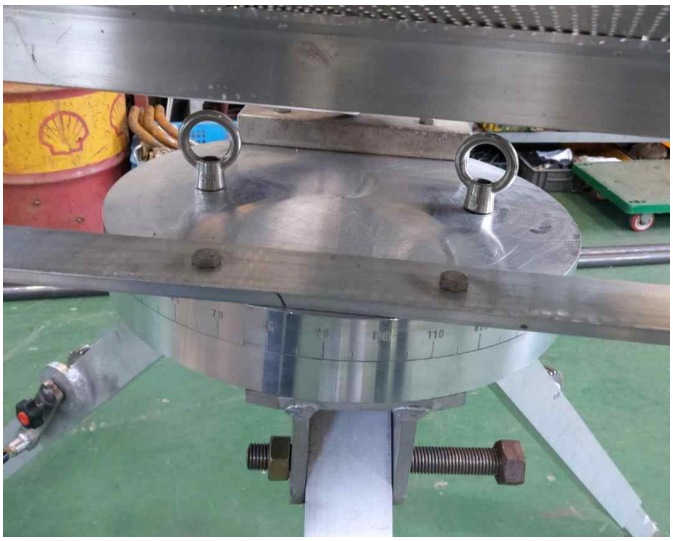
[정삼각형 및 정사각형 메쉬에 대한 천공 간격 대 개방 면적 비율]

(Garthwaite et al., 2015)



[2.8 mm 직경, 12 mm 간격, 개방 영역 4.9 %, 정삼각형 그리드로 천공한 코너리플렉터 판]

- 정확도 높은 코너리플렉터 정렬을 위하여 좌우 180 도 이상, 상하 45도 각도 이상으로 회전이 가능하도록 제작
- 손잡이를 돌림으로써 상하 각도 조절이 가능하고 더 큰 각도를 조절하고 싶을 경우 다리의 높이 조절을 통해 해결



[좌우 360도 회전판]



[상하 각도 조절 장치]

- 삼각대를 제작하여 코너리플렉터를 설치하고 mm 단위의 정밀 높이 조절 및 정확한 높이 측정이 가능할 수 있도록 제작



- 운반의 편리성을 위해 조립 및 분해가 가능하도록 함



[제작된 코너리플렉터 전체 모습]

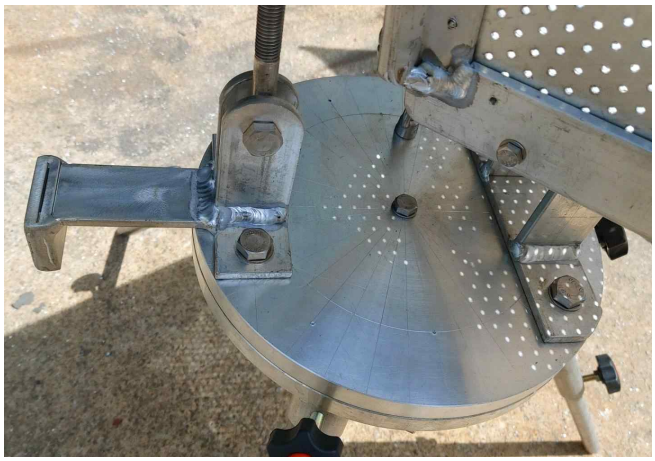
○ 경량 코너리플렉터 제작

- 레일 또는 드론 기반 레이더 보정 위하여 운반 및 설치가 간편한 50 cm 경량 코너리플렉터 제작
- 이전 제작한 코너리플렉터와 같은 이론을 통해 제작하되 돌림판/삼각대 무게의 경량화를 진행



[제작한 경량 코너리플렉터]

- X-, C-, L-band 모두에서 사용 가능한 코너리플렉터 제작을 위해 X-band의 약 1/10, C-band의 약 1/20에 해당하는 2.8 mm 직경, 12 mm 간격, 개방 영역 4.9 %로 천공함
- 정확도 높은 코너리플렉터 정렬을 위하여 좌우 180 도 이상, 상하 45도 각도 이상으로 회전이 가능하도록 제작



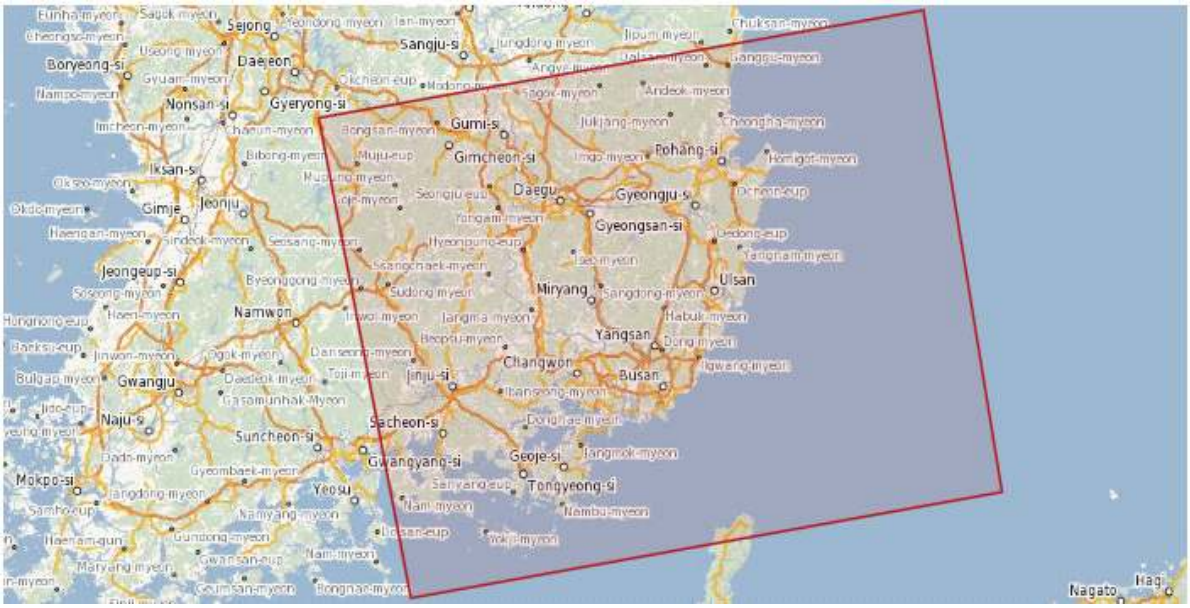
[경량 코너리플렉터 좌우 각도 조절 장치]

- 너트를 통해 상하 각도 조절이 가능하고 더 큰 각도를 조절하고 싶을 경우 다리의 높이 조절을 통해 해결



[경량 코너리플렉터 상하 각도 조절 장치]

○ Sentinel-1 위성 영상을 이용한 코너리플렉터 실험
- 코너리플렉터 실험을 위해 C-band Sentinel-1 위성이 사용됨
- Sentinel-1은 C-band 영상레이더를 탑재하고 있으며 재방문 주기가 12일로 짧고 오픈 엑세스가 가능하므로 코너리플렉터 실험에 적합하다 판단하였음
- 또한 L-band, X-band 위성의 경우 주기적인 관측이 용이하지 않아 유럽의 Sentinel-1 위성을 사용하였음
- 대한민국 부산 영도에 위치한 한국해양과학기술원 (KIOST)의 협조를 구하여 연구원 내 공터에서 실험 진행
- Sentinel-1 위성영상 검색을 통해 설치 지역을 커버하는 영상 촬영 계획을 확인하였으며 54 path 상승궤도 영상을 선택하였음



[Sentinel-1 54 path 상승궤도 영상, 촬영시간 18시 23분, 촬영 간격 12일]
(Copernicus Open Access hub, https://scihub.copernicus.eu/dhus/#/home)



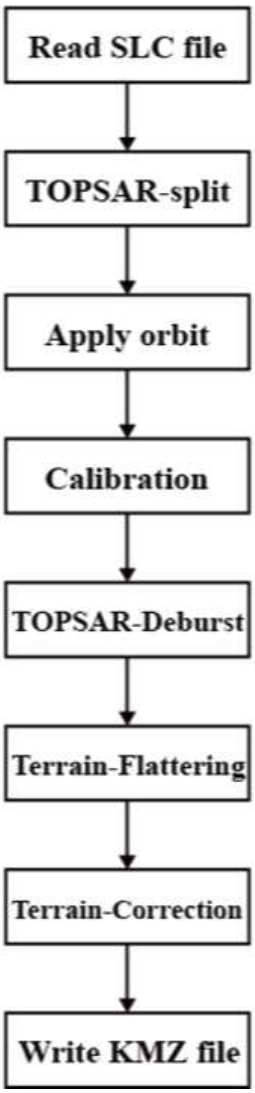
[Sentinel-1 54 path 상승궤도 영상, 촬영시간 18시 22분, 촬영 간격 12일]
(Copernicus Open Access hub, https://scihub.copernicus.eu/dhus/#/home)

- 위 두 영상 모두 한국해양과학기술원 본원을 포함함을 확인하였으며 영상 간의 촬영 간격이 6일이므로 6일마다 실험을 진행하였음 (2021년 3월 24일, 3월 30일, 4월 5일: 총 3번 실험)
- 위성의 촬영 시간은 약 18시 22분, 날아가는 방향은 -13.05도, incidence angle은 약 38도로 확인
- 코너리플렉터가 최대 RCS 값을 보이는 각도는 입사 레이더가 바닥판에서 35도를 이루었을 때이므로 코너리플렉터의 바닥면이 수평을 기준으로 17도 상승할 경우 최대 RCS를 보일 것으로 판단하였으며 각 실험에서 동일한 위치에 동일한 각도로 설치하고자 함
- 상시 관측을 위해서는 바람 등으로 인한 파손 위험을 줄이기 위해 전가지파 차폐를 고려한 레이돔(radome) 설치가 반드시 요구됨
- 3월 12일, 3월 18일 진폭 영상과의 비교를 통해 코너리플렉터의 위성 영상에 대한 적용 확인



[2021년 3월 24일 실험 모습]

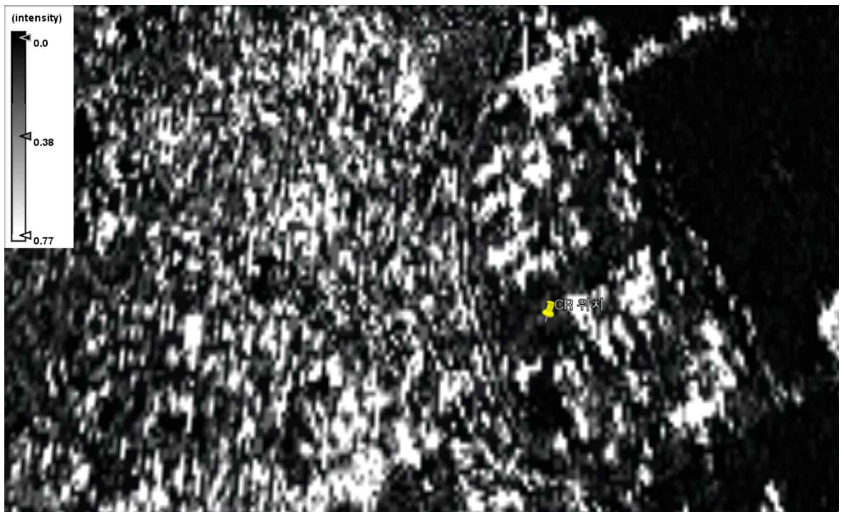
- Copernicus Open Access hub (https://scihub.copernicus.eu/dhus/#/home)에서 영상레이더 획득 후 Sentinel Application Platform (SNAP)을 이용하여 진폭 영상으로 변환
- 획득한 SLC 영상에서 연구 지역이 포함된 subswath의 burst를 사용함 (TOPSAR-split)
- SAR 제품의 메타 데이터에서 제공되는 궤도 상태 벡터는 일반적으로 정확하지 않기 때문에 정확한 궤도 파일을 통해 제품의 메타 데이터에 있는 궤도 상태 벡터를 업데이트 해줌
- 일반적인 SAR 데이터 처리에는 방사성 보정이 포함되지 않으며 상당한 방사성 편향이 남아있어 SAR 이미지의 픽셀 값이 반사 표면의 레이더 후방 산란을 나타내고 SAR 데이터를 정량적으로 사용할 수 있도록 방사 보정을 적용함
- 지형 변화가 지구 표면에서 표적의 위치뿐만 아니라 돌아오는 레이더의 밝기에도 영향을 미치기 때문에 이를 처리하지 않으면 지형 변화로 인한 방사성 편향이 개입되므로 지형 평탄화 알고리즘을 적용하여 지형과 관련된 방사성 가변성을 제거하고 토지 피복과 관련된 방사성 가변성을 남김
- 지형적 변화와 위성 센서의 기울기로 인해 SAR 영상에서 거리가 왜곡될 수 있고, 센서의 nadir 위치에 직접 있지 않은 영상 데이터에는 약간의 왜곡이 존재하므로 지형 보정을 통해 영상의 기하학적 표현이 현실에 최대한 가깝도록 왜곡을 보정함
- 변환한 영상을 kmz 형식으로 바꾸어 구글어스를 이용해 표시함



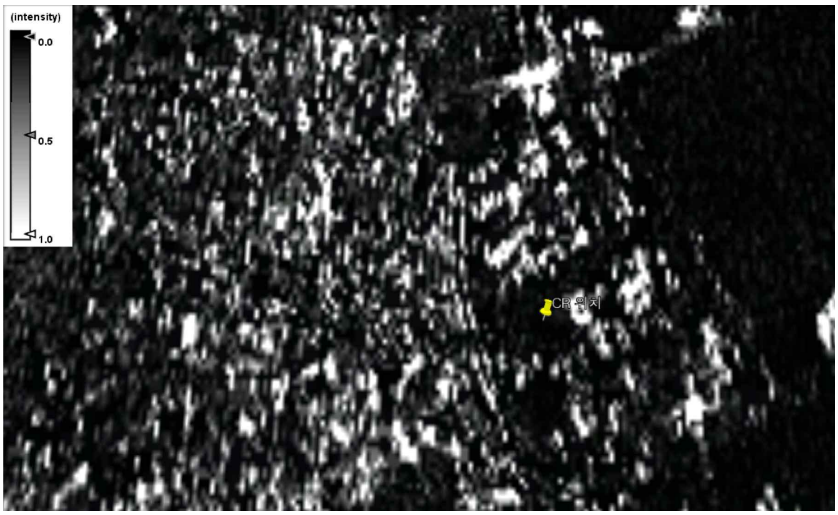
[SNAP을 이용한 Sentinel-1 진폭영상 처리 플로우차트]



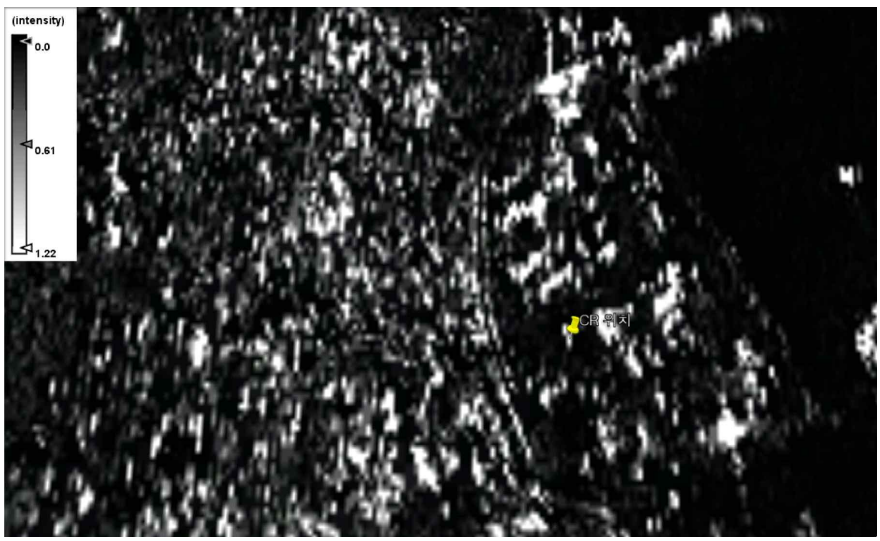
[코너리플렉터 위치, 출처: Google Earth]



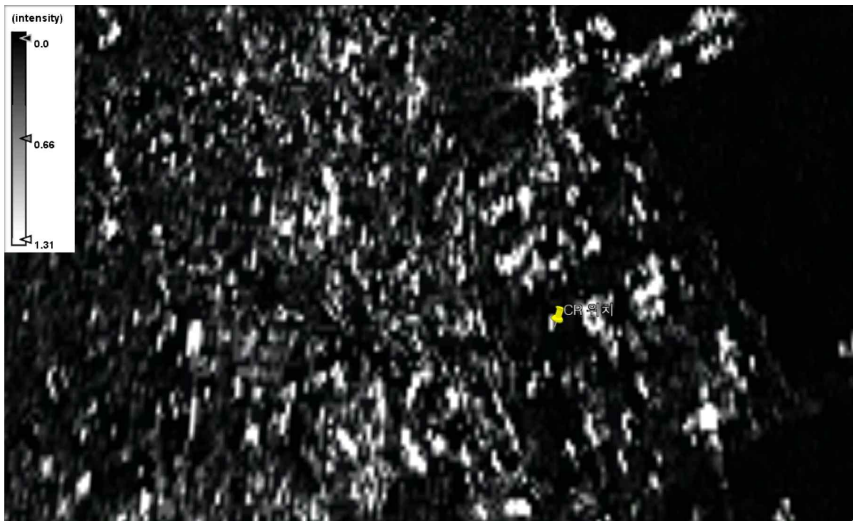
[3월 12일 Sentinel-1 진폭 영상]



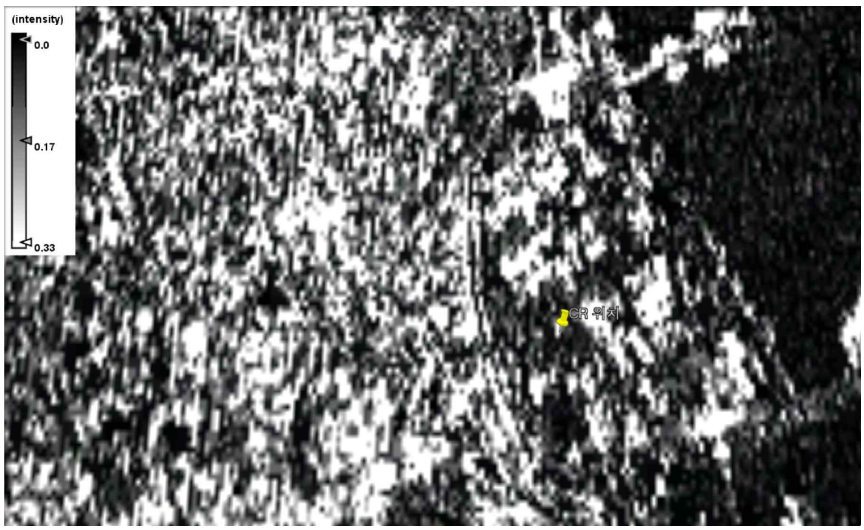
[3월 18일 Sentinel-1 진폭 영상]



[3월 24일 1차 실험 Sentinel-1 진폭 영상]



[3월 30일 2차 실험 Sentinel-1 진폭 영상]



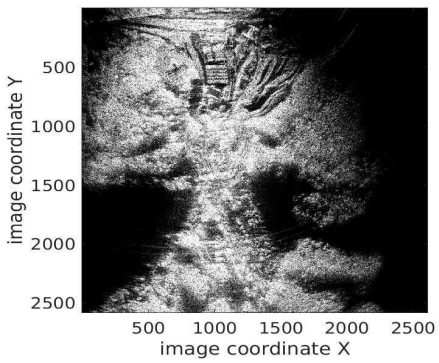
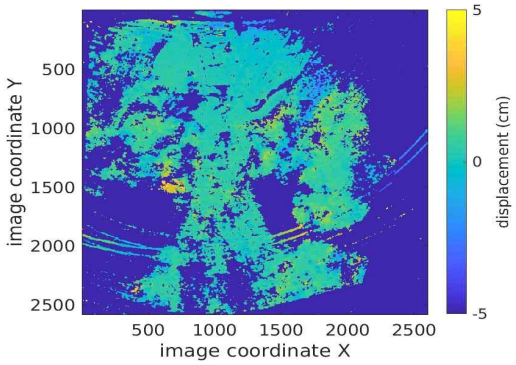
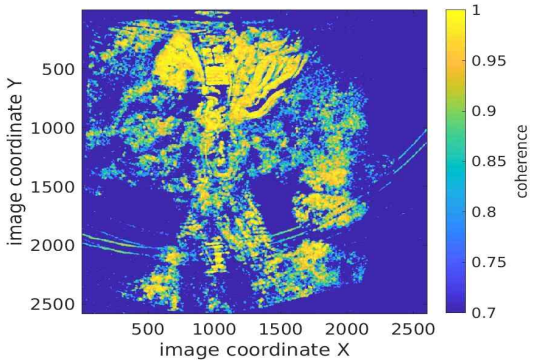
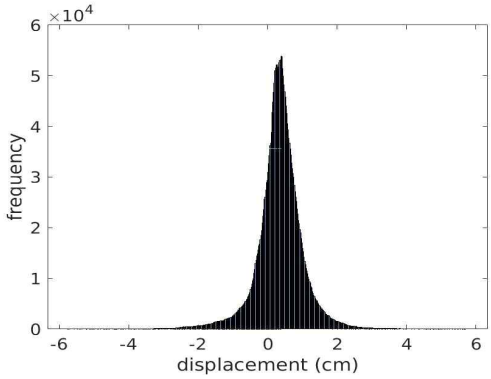
[4월 5일 3차 실험 Sentinel-1 진폭 영상]

- 3월 12일과 18일에 나타나지 않았던 후방산란이 실험 영상에서 나타나는 것을 확인
- 1차 실험 (3월 24일)의 경우 최초 실험 경험 부족으로 인한 오류 발생으로 인접 픽셀에 비해 약 30배 큰 후방산란 값을 획득
- 2차 실험 (3월 30일)과 3차 실험 (4월 5일)의 경우 인접 픽셀에 비해 약 360배에서 800배 큰 후방 산란 값을 획득할 수 있었음
- 코너리플렉터를 통한 후방 산란값 증폭을 확인할 수 있었으며 지상기준점 오차 확인에 충분히 활용할 수 있을 것으로 판단됨

○ 드론 측정 데이터와 지상 라이더 데이터 비교 분석

- 2021년 5월 31일 경기도 남양주 사이트에서 실시한 두 장의 드론 영상레이더 관측 시간은 약 7분 차이로 해당 시간 동안 비탈면 변위는 없다고 가정
- 비탈면 변위 분석 결과값은 드론 SAR 관측 시스템의 정확도로 판단 가능함

※ 아래 그림에서 상대적으로 밝은 부분이 관심 비탈면 지역임

<table>
<tr><th>드론 SAR 시스템 후방산란지도</th><th>드론시스템으로 관측된 비탈면 변위 지도</th></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><th>영상레이더 자료 사이의 긴밀도</th><th>관측 변위량의 히스토그램</th></tr>
<tr><td> 
</td><td> 
</td></tr>
</table>

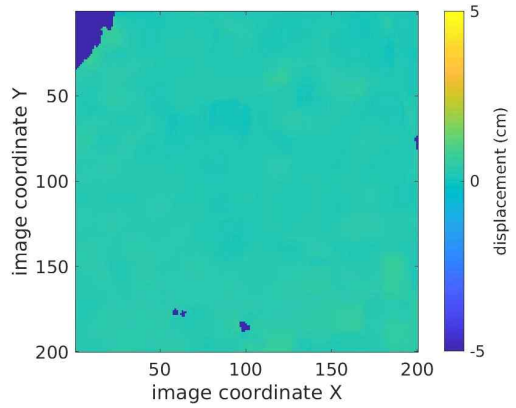
- 비탈면 변위 평가 분석 결과 과제목표인 5 cm 이하에 해당하는 화소는 전체 유효화소수(2,888,826개)의 99.95 %에 해당하는 결과를 보임

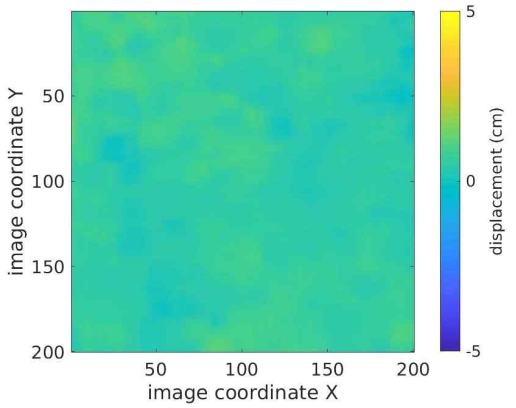
변위 정확도 기준	해당 화소 수 (개)	비율 (%)
5 cm 이하	2,887,478	99.95
4 cm 이하	2,885,892	99.90
3 cm 이하	2,876,332	99.57
2 cm 이하	2,827,286	97.87
1 cm 이하	2,462,789	85.25
0.5 cm 이하	1,597,292	55.29

- 200 × 200 화소 영역 두 지역에 대한 변위 측정 추가 분석 수행

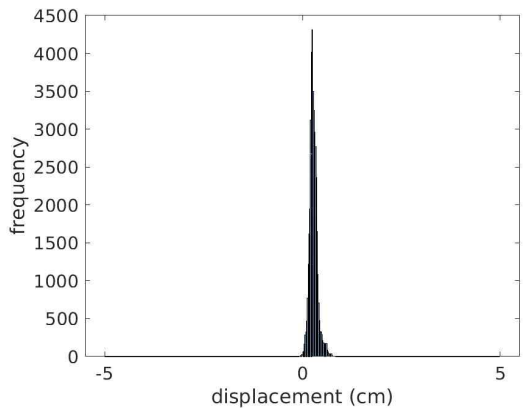
	최소변위값 (cm)	최대변위값 (cm)	평균 (cm)	표준편차 (cm)
상세 지역 1	-0.09	0.83	0.28	0.10
상세 지역 2	-0.16	1.26	0.56	0.28

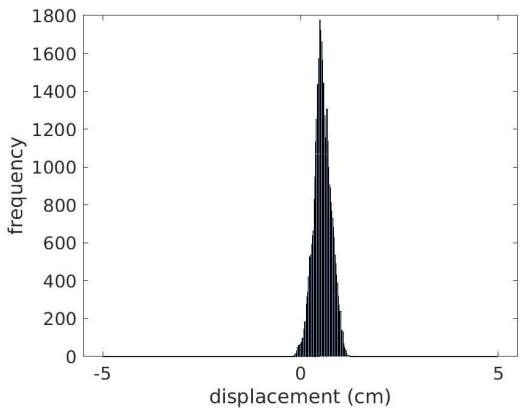
드론시스템으로 관측된 비탈면 변위 지도 (상세지역 1, 2)



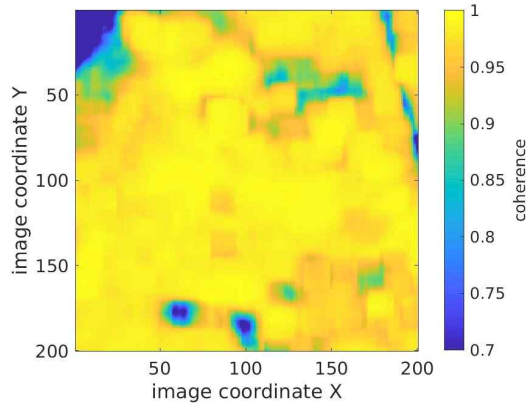


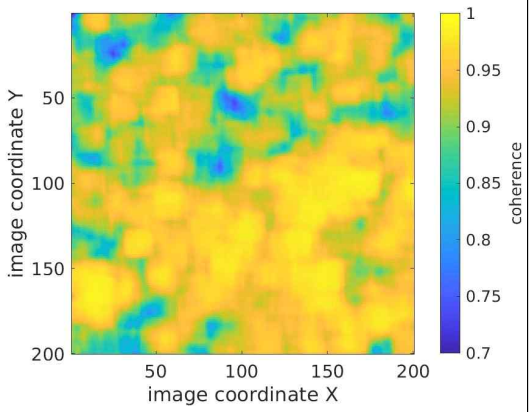
관측 변위량의 히스토그램



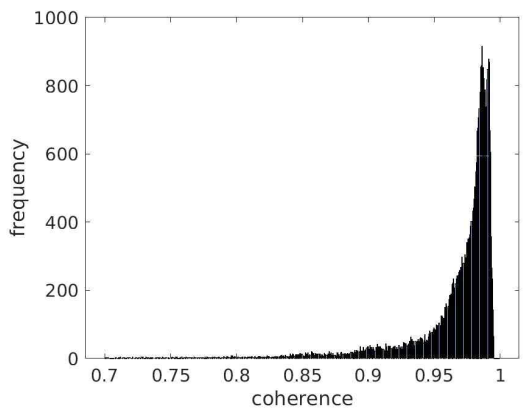


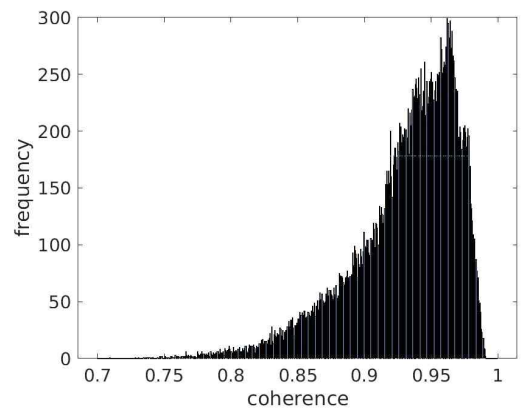
영상레이더 자료 사이의 긴밀도





관측 긴밀도의 히스토그램





- 대상 지역 변위 탐지 평가를 위해 corner reflector (CR) 3기를 설치하여 총 7회의 자료 획득을 수행하였으며 해당 비행 차수에 따른 인공적 변위 높이는 다음과 같음

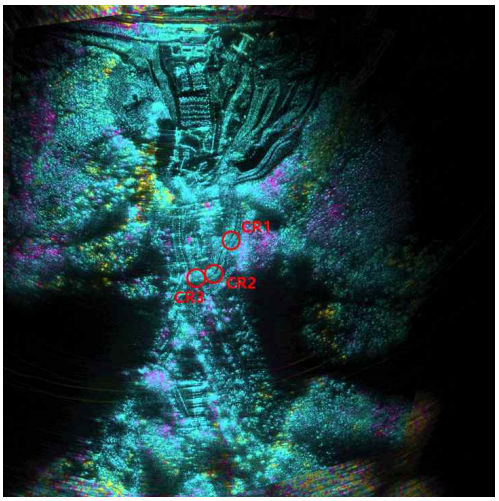
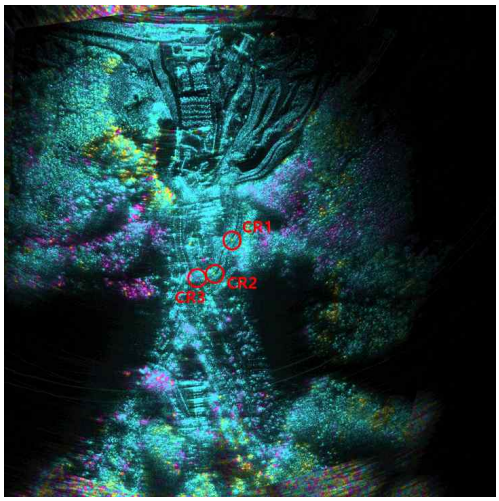
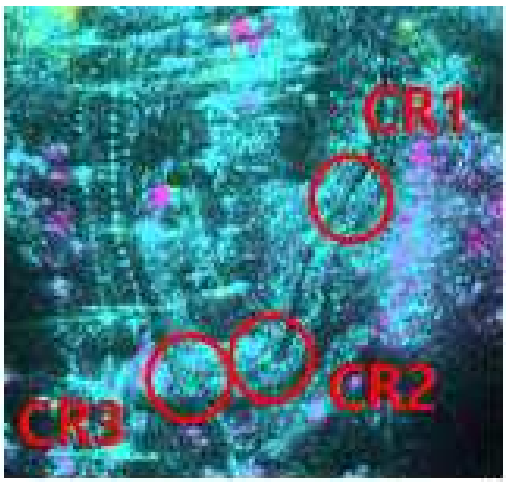
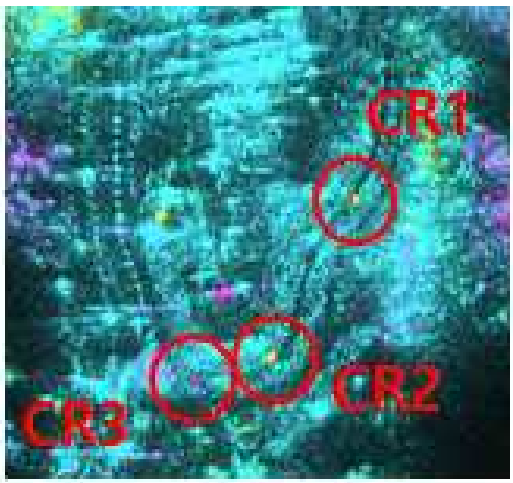
변위 탐지 평가용 CR 설치 위치



변위 탐지 평가용 CR의 비행 차수에 따른 임의적 변위량

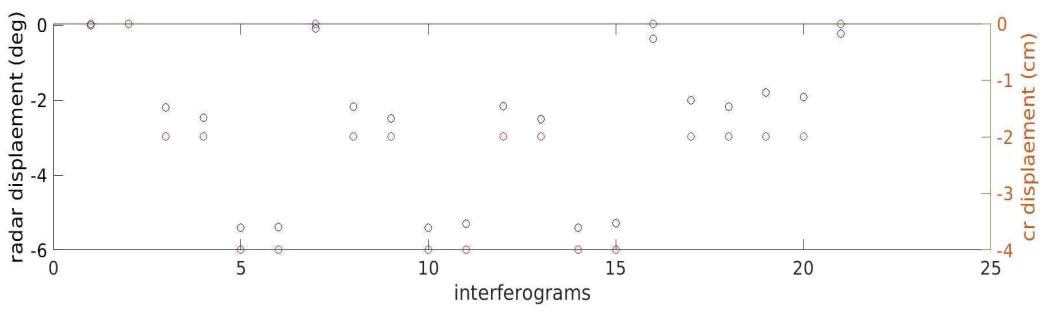
비행 차수	CR1	CR2	CR3
1차	0 cm	0 cm	0 cm
2차	0 cm	0 cm	0 cm
3차	0 cm	0 cm	0 cm
4차	2 cm	2 cm	0 cm
5차	2 cm	2 cm	0 cm
6차	4 cm	4 cm	0 cm
7차	4 cm	4 cm	0 cm

- 각 CR 점에서 2 ~ 4 cm의 수직 변위를 인위적으로 조정하기 전과 후의 위상간섭도에서 변위를 탐지할 수 있음을 확인할 수 있음

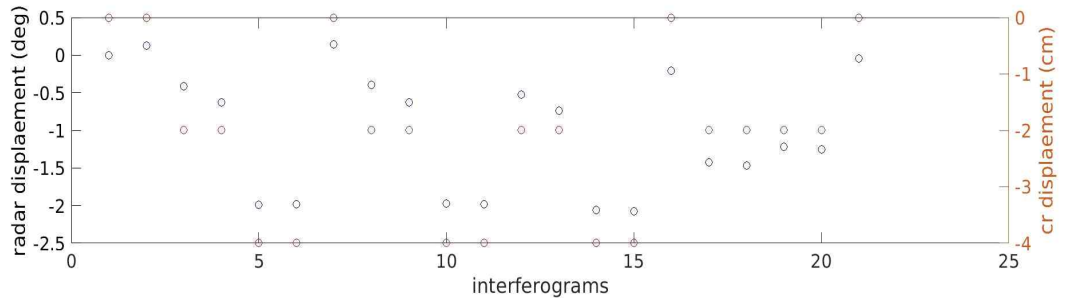
CR 수직 변위가 없는 경우 간섭도	CR1, CR2의 수직 변위가 있는 경우 간섭도
	
CR 수직 변위가 없는 경우 간섭도 (확대)	CR1, CR2 수직 변위시 간섭도 (확대)
	

- 각 CR 점에 해당하는 변위의 차이와 간섭도 위상의 경향성을 아래 그림과 같이 비교하였으며 변위량에 비례하여 변화하는 것을 알 수 있음
- 단, CR 지역 주위의 산란체 특성으로 인해 위상간섭도의 위상에 영향을 받았을 것으로 추정되며, 보다 정밀한 위상 변위 추정을 위해서는 드론 비행 관측의 정밀 기하구조 분석과 평활하고 고르게 분포한 산란체 지역에서 추가 실험이 요구됨

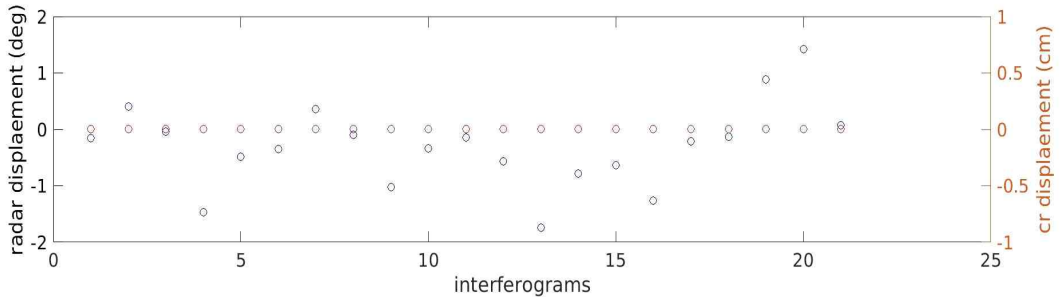
CR 1에서의 수직변위(붉은색)와 위상간섭도 위상(파란색)과의 비교



CR 2에서의 수직변위(붉은색)와 위상간섭도 위상(파란색)과의 비교

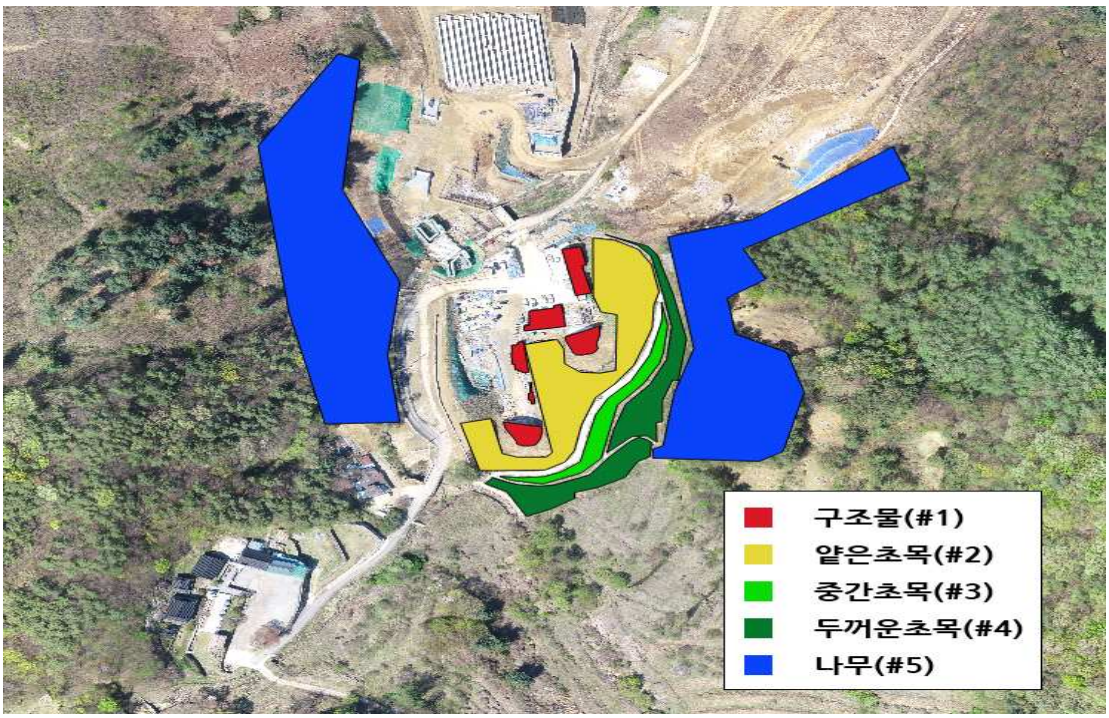


CR 3에서의 수직변위(붉은색)와 위상간섭도 위상(파란색)과의 비교



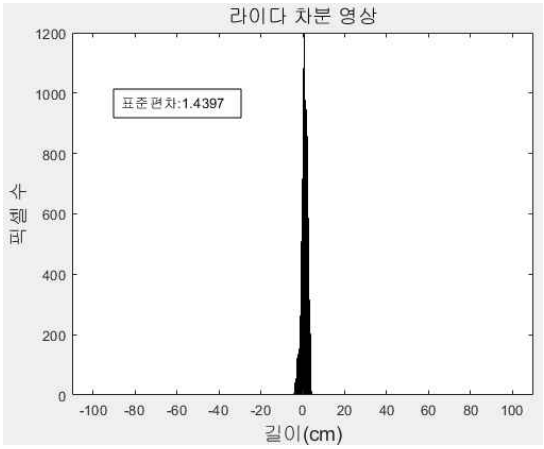
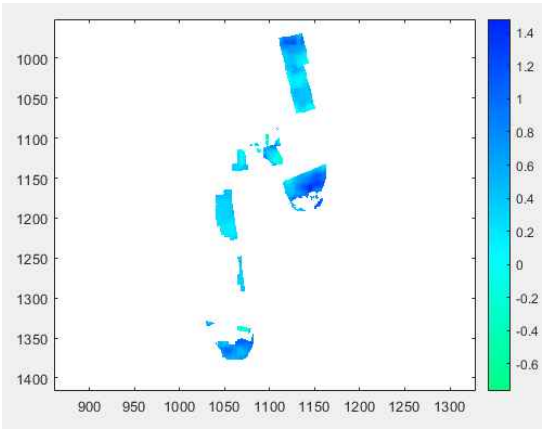
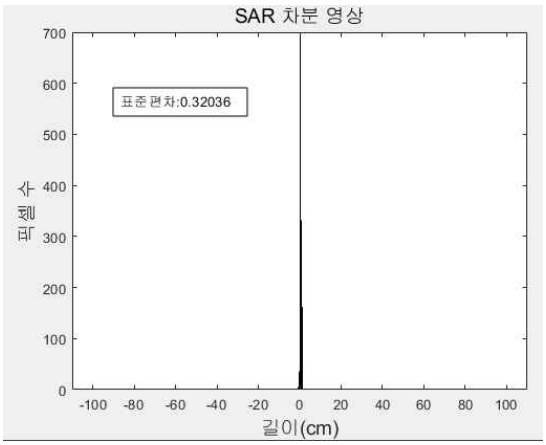
○ 비탈면 변위측정 정밀도

- 드론 SAR 영상의 변위측정 정밀도 검증은 동일 지역에서 획득한 LiDAR DSM 데이터와 비교하여 수행함
- LiDAR 데이터는 2021년 4월 28일, 2021년 5월 31일 총 2개의 데이터를 획득하였으며 각각 DSM을 생성한 후, 차분하여 검증에 사용하였음
- SAR 영상은 5월 31일에 촬영한 7개의 영상 중, 2개 영상으로 간섭영상(차분)을 생성하였음
- 비교에 사용된 LiDAR 차분영상과 SAR 간섭영상은 현장에서 드론으로 촬영한 사진을 육안분석을 통해 구조물(#1), 얕은초목(#2), 중간초목(#3), 두꺼운초목(#4), 나무(#5) 총 5개 지역으로 구분하여 비교검증을 수행하였음
- 데이터 획득 당시 현장의 변위가 발생하지 않았으므로 구분된 5개 지역의 표준편차는 0에 가까울수록 정밀함을 나타냄



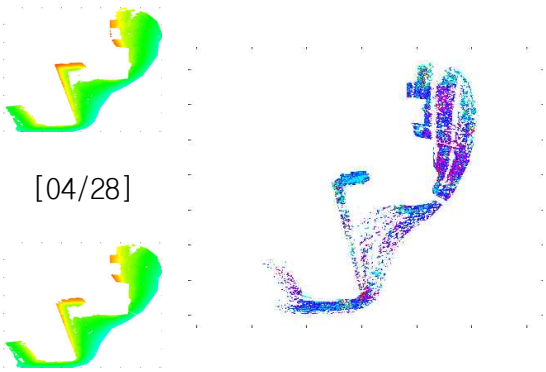
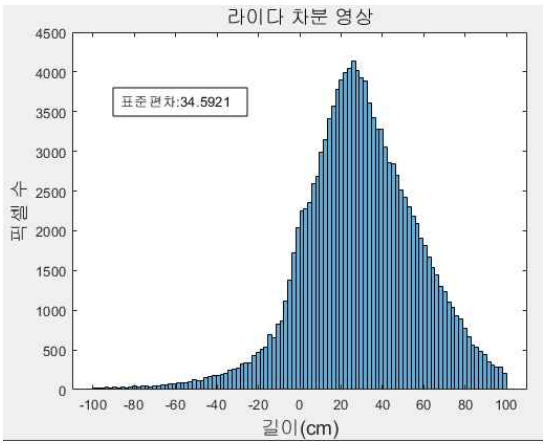
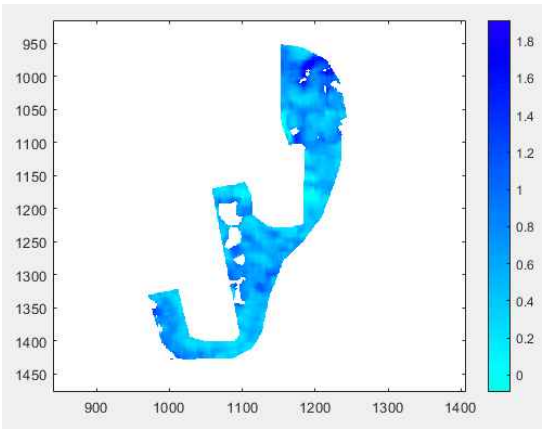
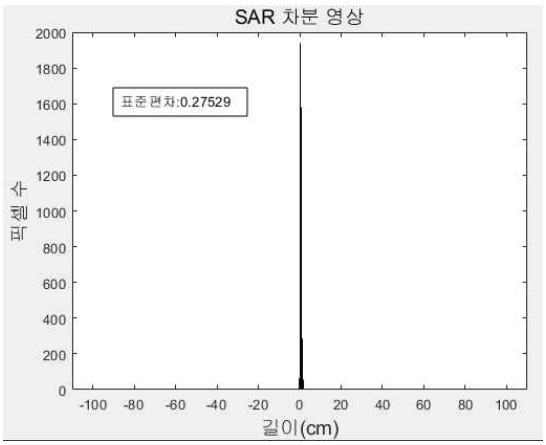
구조물 (#1)

- 구조물에 해당하는 영역 중 LiDAR 차분영상에서 나타나는 구조물은 주로 터널임. 터널을 제외한 다른 구조물의 경우, 점군이 많지 않아 분석에 사용하지 않았음
- SAR 간섭영상에서는 터널 구조물 뿐만 아니라 현장에 설치된 컨테이너 건물 등이 분석에 포함되었음
- 분석결과로 LiDAR 차분영상의 표준편차는 1.4397, SAR 간섭영상의 표준편차는 0.32036으로 나타났음. 두 결과모두 낮은 값으로 구조물에서는 정밀한 것으로 나타났음
- SAR 간섭영상에 비해 LiDAR 차분영상의 표준편차가 더 높게 나타났으나, LiDAR 측정당시 측점의 위치, GPS 보정등의 오차가 포함될수 있음

<table>
<tr><th colspan="4">#1 구조물</th></tr>
<tr><td rowspan="2">LiDAR</td><td>[04/28]</td><td rowspan="2">[05/31 - 04/28]</td><td rowspan="2"> 
</td></tr>
<tr><td>[05/31]</td></tr>
<tr><td colspan="2">LiDAR 차분 DSM 영상</td><td>LiDAR 히스토그램</td></tr>
<tr><td rowspan="2">SAR</td><td colspan="2">  </td><td> 
</td></tr>
<tr><td colspan="2">드론 SAR 차분(InSAR) 영상</td><td>SAR 히스토그램</td></tr>
</table>

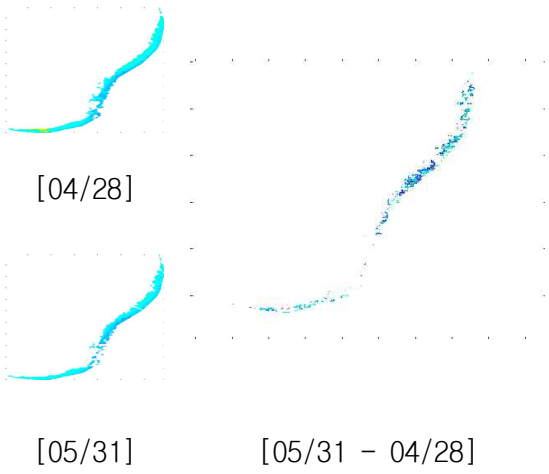
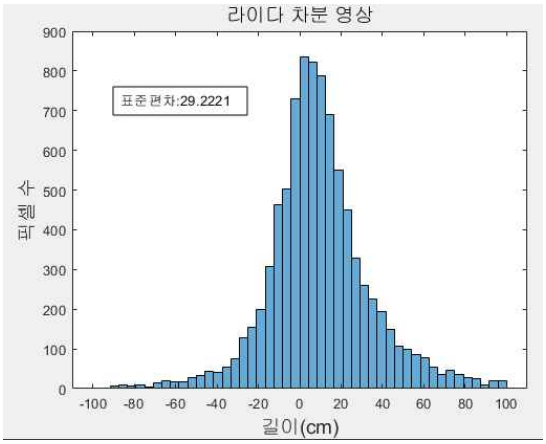
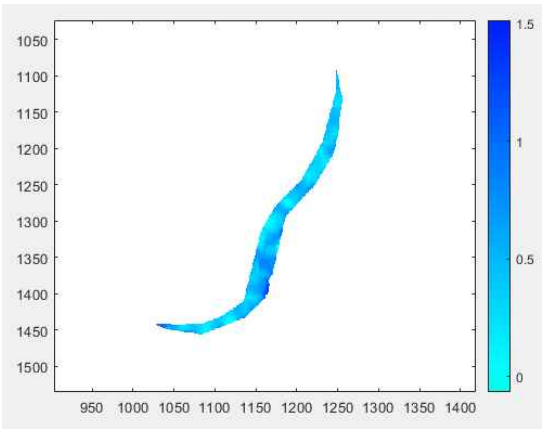
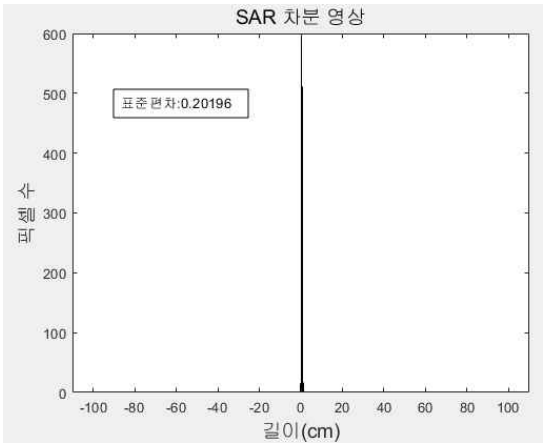
얕은초목 (#2)

- 얕은초목에 해당하는 영역은 터널구조물 주변부로 약 3~10cm 정도의 풀로 덮혀있는 지역임
- 분석결과로 LiDAR 차분영상의 표준편차는 34.5921, SAR 간섭영상의 표준편차는 0.27529으로 나타났음. SAR 간섭영상의 정밀도는 구조물 지역과 비슷한 값을 보였음
- LiDAR 차분영상의 평균값은 약 25cm인데, 이는 차분에 사용한 두 데이터의 시간 간격이 약 1달로 식생의 생장이 반영된 것으로 보임. 그러나 LiDAR 파장의 경우, 투과력이 없기 때문에 정밀도는 낮게 나타났음. 반면 SAR 간섭영상의 경우, 긴파장으로 인한 투과력으로 초목에 관계없이 높은 정밀도를 보임

<table>
<tr><th colspan="3">#2 얕은 초목</th></tr>
<tr><td rowspan="2">LiDAR</td><td> 
</td><td> 
</td></tr>
<tr><td>LiDAR 차분 DSM 영상</td><td>LiDAR 히스토그램</td></tr>
<tr><td rowspan="2">SAR</td><td>  </td><td> 
</td></tr>
<tr><td>드론 SAR 차분(InSAR) 영상</td><td>SAR 히스토그램</td></tr>
</table>

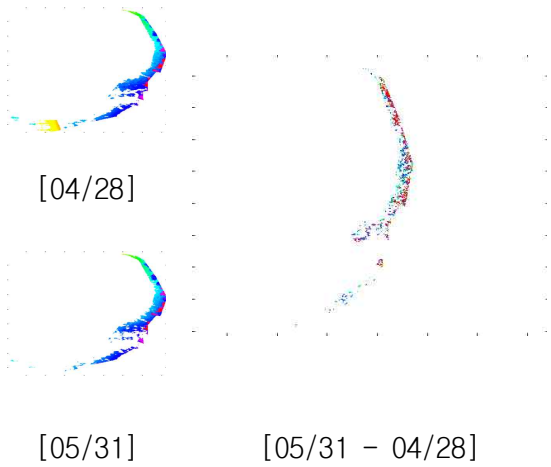
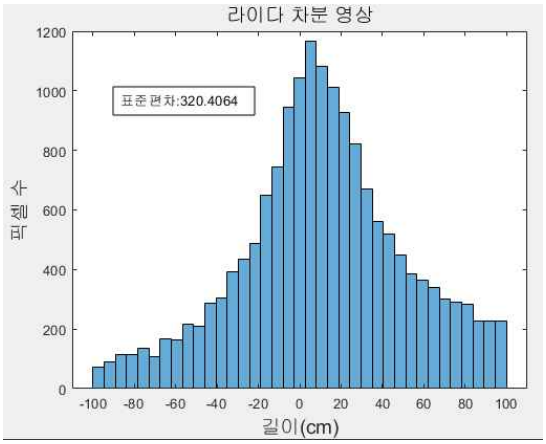
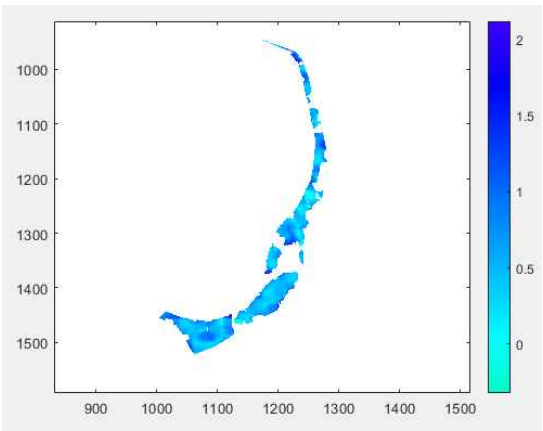
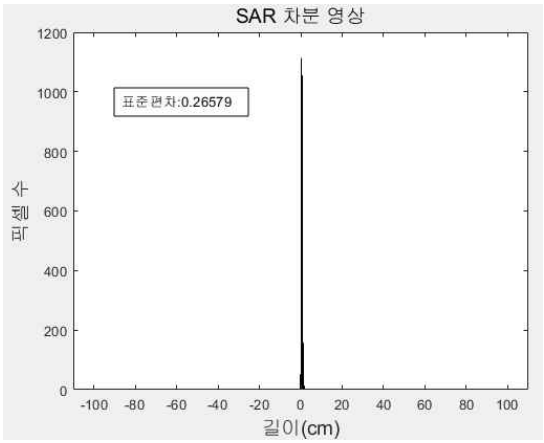
중간초목 (#3)

- 중간초목에 해당하는 영역은 #2 얕은초목의 상부 지역으로 약 10cm 정도의 풀로 덮혀있는 지역으로 얕은초목 지역과 비슷하거나 조금 긴 특징이 있음
- 분석결과로 LiDAR 차분영상의 표준편차는 29.2221, SAR 간섭영상의 표준편차는 0.20196으로 나타났음. SAR 간섭영상의 정밀도는 구조물, 얕은초목 지역보다 높게 나타났음
- 중간초목부 또한 LiDAR 차분영상의 평균값이 약 15cm로 나타났으며 식생의 생장이 반영된 것으로 분석됨. SAR 간섭영상의 경우 매우 정밀한 값을 보임

<table>
<tr><th colspan="3">#3 중간 초목</th></tr>
<tr><td rowspan="2">LiDAR</td><td> 
</td><td> 
</td></tr>
<tr><td>LiDAR 차분 DSM 영상</td><td>LiDAR 히스토그램</td></tr>
<tr><td rowspan="2">SAR</td><td> 
</td><td> 
</td></tr>
<tr><td>드론 SAR 차분(InSAR) 영상</td><td>SAR 히스토그램</td></tr>
</table>

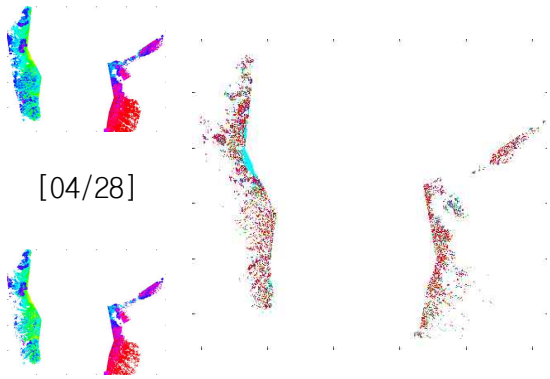
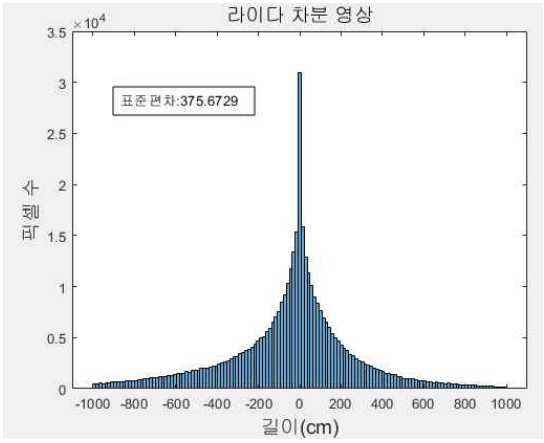
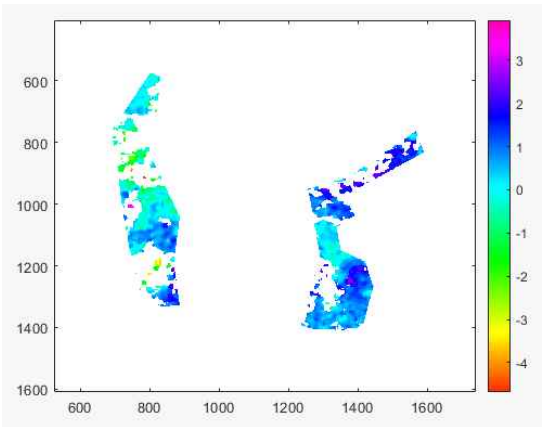
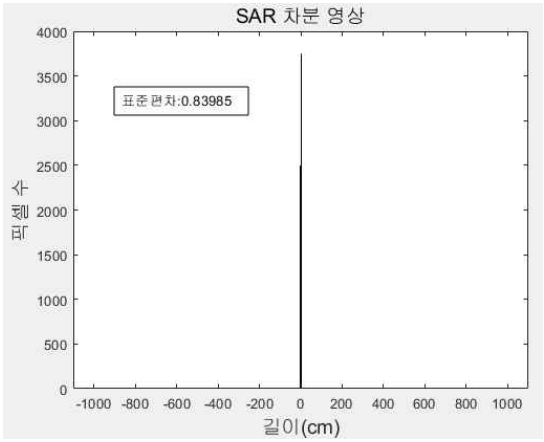
두꺼운 초목 (#4)

- 두꺼운 초목에 해당하는 영역은 #3 중간초목의 상부 지역으로 수풀, 작은 관목으로 이루어짐
- 분석결과로 LiDAR 차분영상의 표준편차는 320.4064, SAR 간섭영상의 표준편차는 0.26579으로 나타났음. SAR 간섭영상의 정밀도는 다른 지역과 마찬가지로 높게 나타났음
- 두꺼운 초목지역은 풀의 두께가 두껍기 때문에 특히 LiDAR에서 정밀도가 현저하게 낮게 나타났음. 반면, SAR 간섭영상의 경우 이론적으로 약 12cm의 투과력을 가지기 때문에 수풀, 작은 관목에도 불구하고 높은 정밀도를 보였음

<table>
<tr><th colspan="3">#4 두꺼운 초목</th></tr>
<tr><td rowspan="2">LiDAR</td><td> 
</td><td> 
</td></tr>
<tr><td>LiDAR 차분 DSM 영상</td><td>LiDAR 히스토그램</td></tr>
<tr><td rowspan="2">SAR</td><td> 
</td><td> 
</td></tr>
<tr><td>드론 SAR 차분(InSAR) 영상</td><td>SAR 히스토그램</td></tr>
</table>

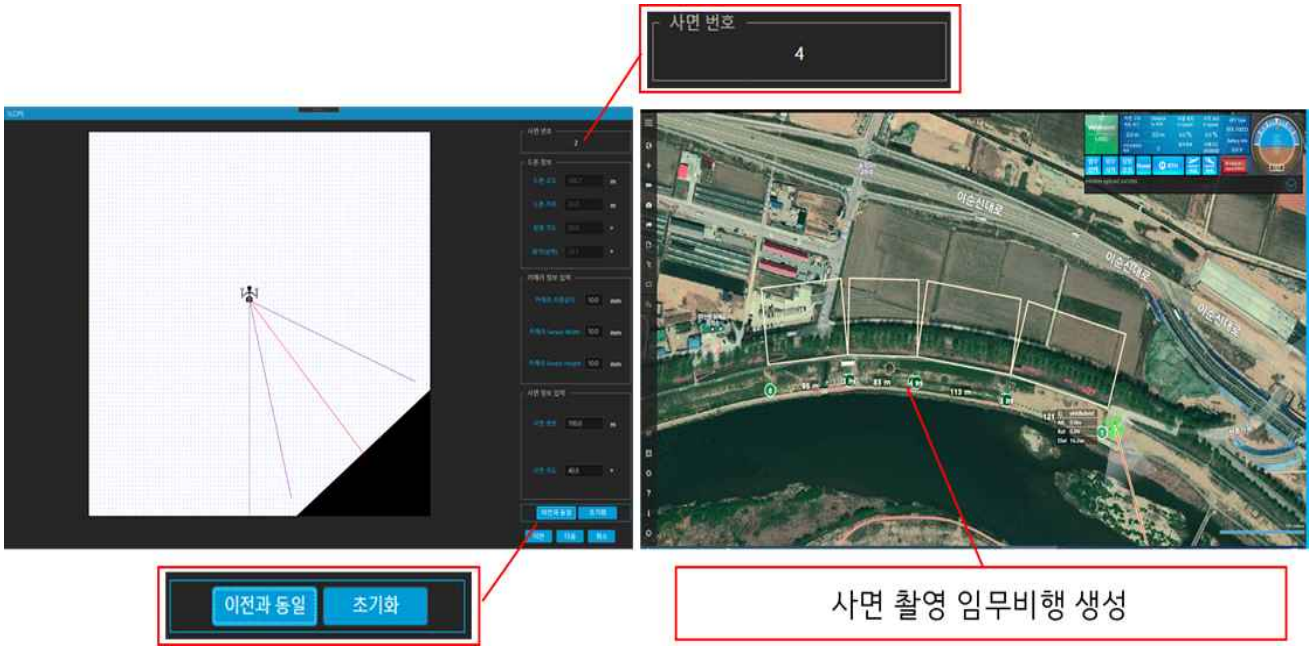
나무 (#5)

- 마지막으로 분석한 영역은 나무로 구성된 영역으로 주로 중간크기의 관목으로 이루어짐
- 분석결과로 LiDAR 차분영상의 표준편차는 375.6729, SAR 간섭영상의 표준편차는 0.83985으로 나타났음. LiDAR 차분영상의 표준편차는 두꺼운 초목지역 보다도 정밀도가 낮게 나타났으며 SAR 간섭영상은 다른 지역보다는 낮지만 1이하의 높은 정밀도를 보였음
- 나무로 구성된 영역의 경우, 가지가 많고 잎이 두꺼워 두 영상모두 다른지역보다 표준편차가 낮게 나타났음. 그러나 SAR 간섭영상에서는 약 0.8로 높은 정밀도를 보였는데, 이는 파장이 얕은 가지를 관통하고 나무의 몸통에서 획득된 신호로 분석됨
- LiDAR 차분영상과 비교하여 전반적으로 SAR 간섭영상이 매우 정밀한 결과를 보였음

<table>
<tr><th colspan="3">#5 나무</th></tr>
<tr><td rowspan="2">LiDAR</td><td> 
</td><td> 
</td></tr>
<tr><td>LiDAR 차분 DSM 영상</td><td>LiDAR 히스토그램</td></tr>
<tr><td rowspan="2">SAR</td><td>  </td><td> 
</td></tr>
<tr><td>드론 SAR 차분(InSAR) 영상</td><td>SAR 히스토그램</td></tr>
</table>

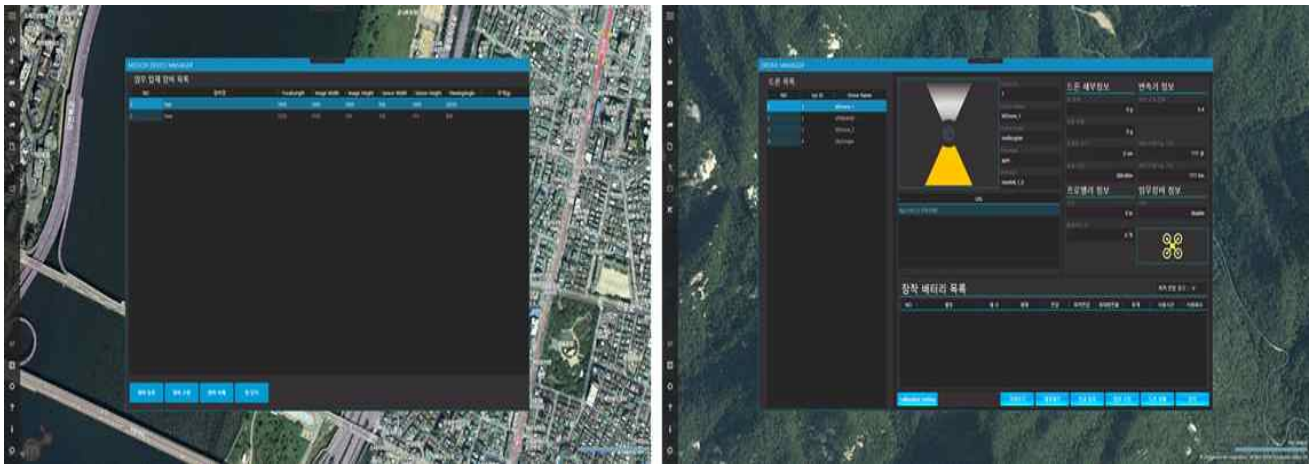
(2) 자동운항 기반 임무비행계획 소프트웨어 고도화

○ 다중 사면 촬영 기능
- 사면의 형태가 곡면의 형태인 경우 촬영을 위한 복수의 임무계획서
- 곡면의 사면을 다수의 직선임무비행계획서 작성하여 사면 촬영
- GIS상에 촬영을 위한 사면을 표시하고 임무비행계획서 분할 생성 기능
- 사면의 촬영을 위한 드론의 운영고도 및 카메라 각도 복수 생성 기능
- 복수 사면의 특정 영역에 대한 개별 정보 수정 및 생성 기능



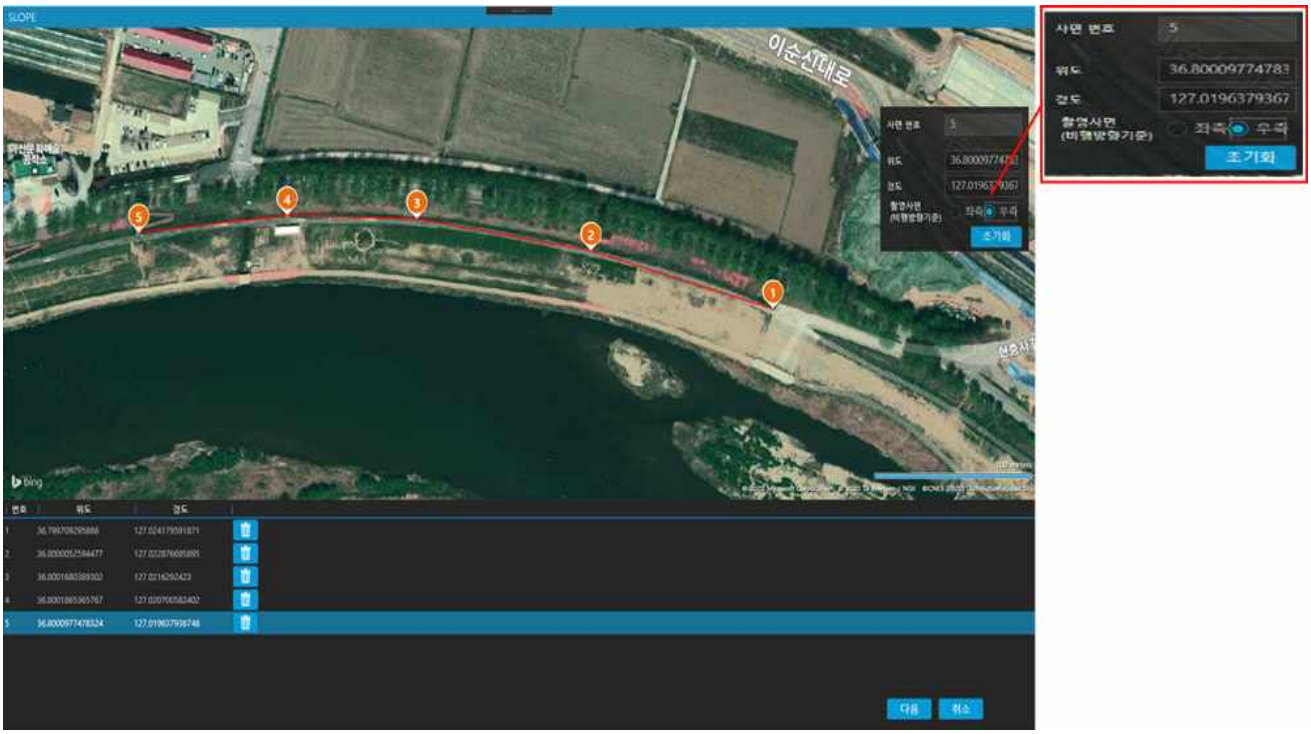
[드론의 임무고도 및 카메라 각도 입력 및 복수의 비행계획서 생성]

○ 사면의 위치에 따른 드론 카메라 촬영 방향 제어
- 드론 등록 및 케메라 정보 관리 기능



[(좌)카메라정보 등록, (우) 드론 등록]

- 촬영하고자 하는 사면의 위치에 따라 드론의 카메라 방향을 설정 기능
- 동일한 사면의 양쪽을 드론 카메라의 방향을 조정하여 촬영하는 기능
- 드론의 카메라 인터페이스에 따른 제어 메시지 전송 기능

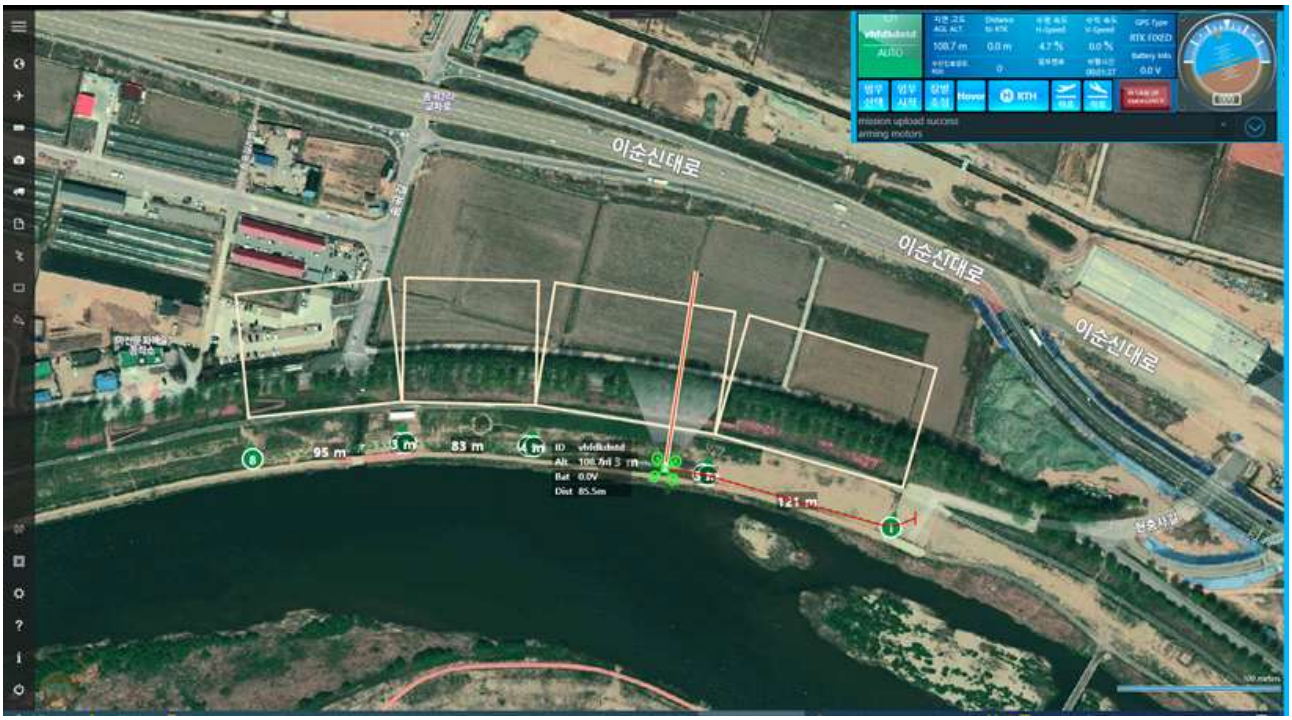


[사면의 촬영방향에 따른 카메라 방향 설정 기능]

- 드론의 카메라 방향을 사면에 맞게 조정하여 드론 임무비행계획서 생성 기능
- 드론의 카메라 방향과 각도 표시하는 기능



[드론 카메라 각도에 따른 임무비행계획서 및 임무 수행]



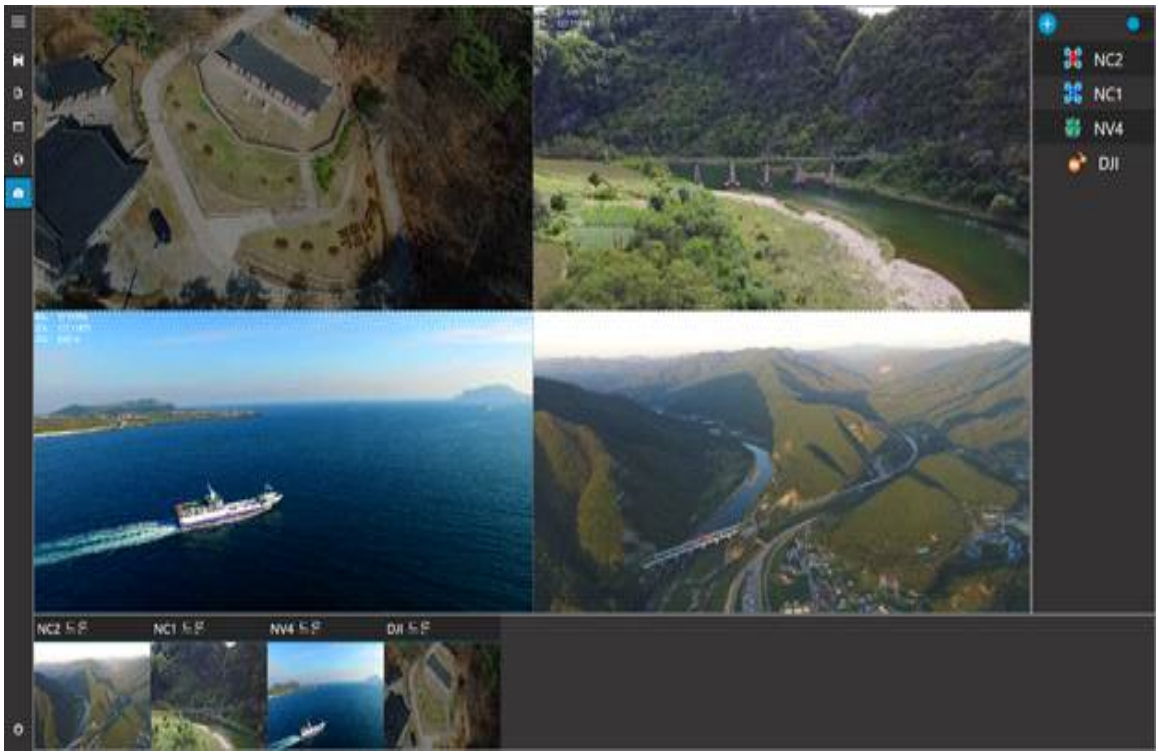
[드론 카메라 각도에 따른 임무비행계획서 및 임무 수행]

○ 임무비행계획서



[다중운용 기반 임무비행계획 및 관제]

○ 실시간 영상수신 및 현시



[다중운용 기반 실시간 임무영상 현시 화면]

(3) 테스트베드 운영

○ 비탈면 현장 테스트베드에서 지상 장비 이용 측정 기준값과 L-SAR 센서 기반 드론 변위 데이터 비교분석

- 화도-양평 깎기 비탈면 현장에 실시간 변위 측정 센서 장치(스마트폴) 설치로 실시간 깎기 비탈면 상부 자연비탈면 변화량 경향 측정
- 지상 레이저 스캐너로 식생부/미식생부 혼재 구간 절대 기준값 측정
- 미식생부 주변의 식생부 L-SAR센서 기반 드론 변위 데이터과 지상 레이저 스캐너와 비교분석



깎기 완료 영역



깎기 진행 영역

[테스트베드 현장 (화도-양평 o공구)]

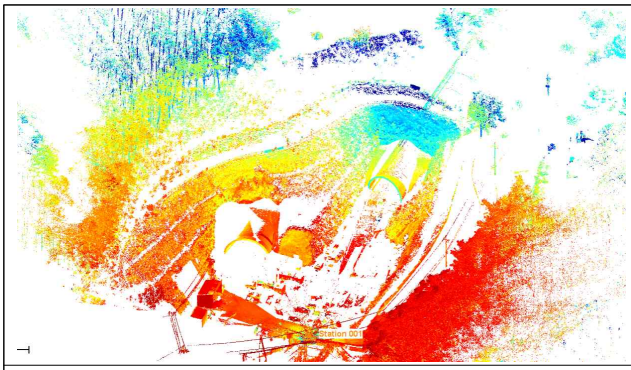


지상 계측점 위치에서 바라본 현장



식생(두께 20~40㎝)과 미식생부 혼재영역

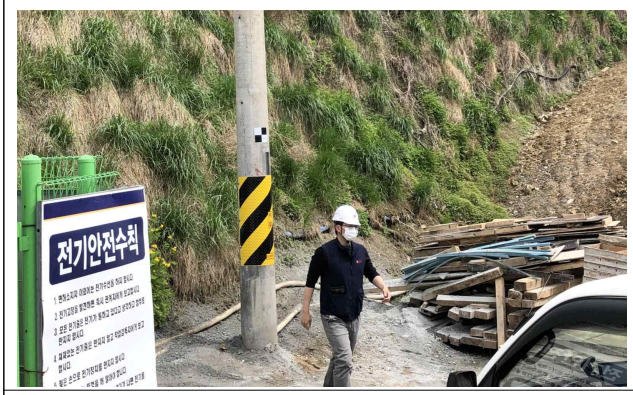
[테스트베드 현장 (화도-양평 o공구)]



#1



#2



#3



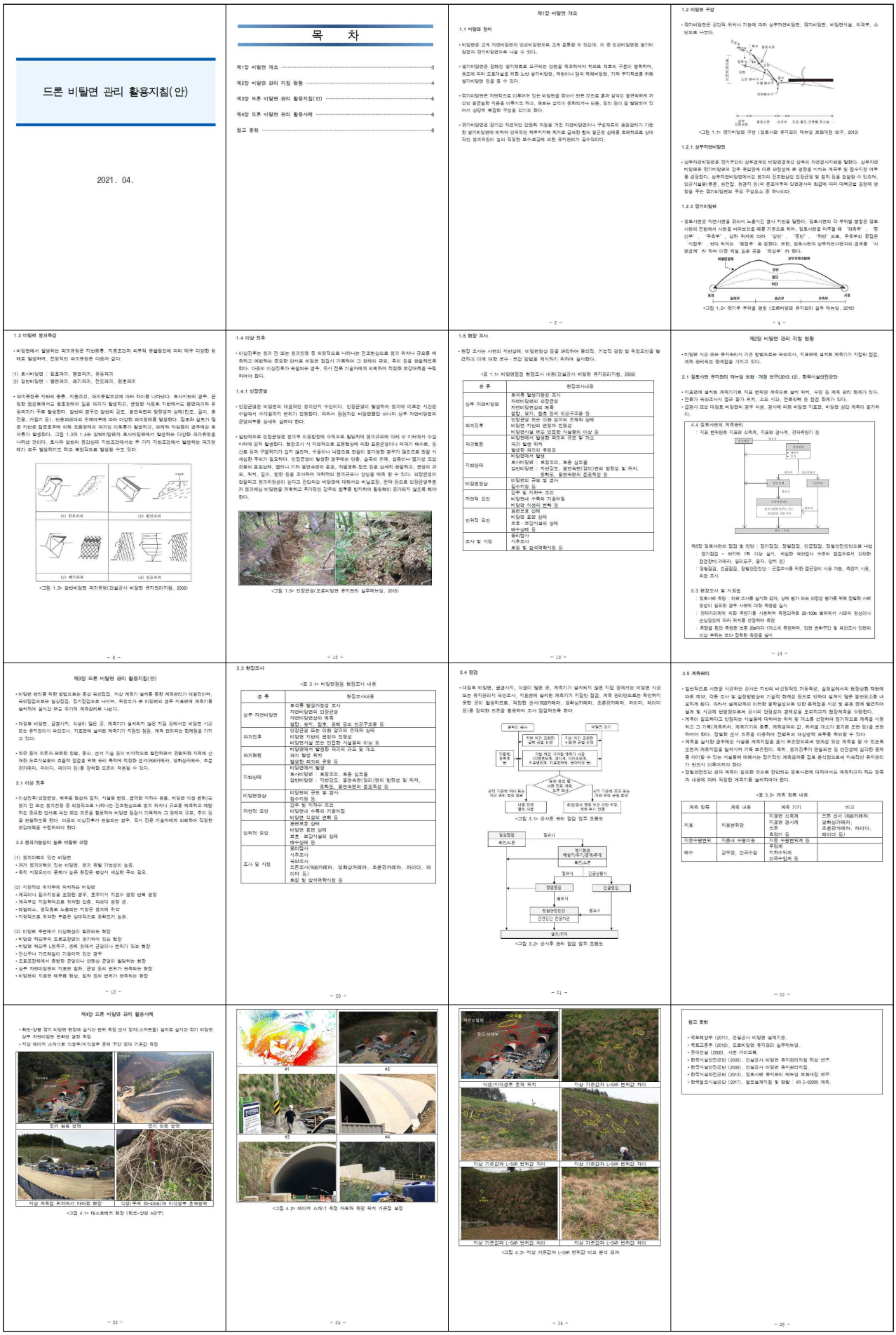
#4





[레이저 스캐너 측정 자료와 측량 위치 기준점 설정]

○ L-SAR 센서 기반 드론 비탈면 변위 가시화 개발시스템 활용 지침 작성



드론 비탈면 관리 활용지침(안)

2021. 04.

목 차

제1장 비탈면 개요 ····· 3
제2장 비탈면 관리 지침 현황 ····· 4
제3장 드론 비탈면 관리 활용지침(안) ····· 5
제4장 드론 비탈면 관리 활용사례 ····· 6
참고 문헌 ····· 6

제1장 비탈면 개요

1.1 비탈면 정의

1.2 비탈면 구성

<그림 1.1> 관기비탈면 구성 (정보사관 무지원리 매뉴얼 보완개정 연구, 2013)

1.2.1 상부자연비탈면

1.2.2 관기비탈면

<그림 1.2> 관기부 무허별 명칭 (도로비탈면 유지관리 실무 매뉴얼, 2018)

1.3 비탈면 붕괴특성

<그림 1.3> 암반비탈면 파기유형(건설공사 비탈면 유지관리지침, 2006)

1.4 이상 징후

<그림 1.5> 인장균열(도로비탈면 유지관리 실무매뉴얼, 2018)

1.5 현장 조사

<표 1.1> 비탈면점검 현장조사 내용(건설공사 비탈면 유지관리지침, 2006)

분 류	현장조사내용
상부 자연비탈면	토석류 발생가능성 조사 자연비탈면의 인장균열 자연비탈면상의 폭포 절단, 굽힘, 함몰, 융력 등의 암공구조물 등
파기전후	인장균열 또는 이와 같거나 유사한 상태 비탈면 지상의 변화측 진행성 비탈면사용 또는 인접한 시설물의 이상 등
파기현황	비탈면에서 발생한 파기의 규모 및 개소 파기 발생 위치 발생한 파기의 유형 등
지반상태	비탈면에서 발생 토사비탈면 : 토질조건, 흙층 상조물 암반비탈면 : 지반암도, 불연속면(절리)면의 방향성 및 위치, 충화도, 불연속면의 공극특성 등
비탈면정상	비탈면의 규모 및 경사 집수지형 등
자연적 요인	강우 및 지하수 조건 비탈면내 수목의 기울어짐
인위적 요인	비탈면 식생의 변화 등 표면보호 상태 비탈면 표면 상태 보호 · 보강시설의 상태 배수상태 등
조사 및 시험	물리탐사 시추조사 육안조사 스프로사(RQD커레카, 일회상커레카, 초분광카메라, 라이다, 허이더 등) 토질 및 강석역학시험 등

제2장 비탈면 관리 지침 현황

2.1 절토사면 유지관리 매뉴얼 보완 · 개정 연구(2013.12), 한국시설안전공단

<그림 2.1> 절토사면의 계측관리

제2장 절토사면의 점검 및 진단 : 정기점검, 정밀점검, 긴급점검, 정밀안전진단으로 나눔

5.3 현장조사 및 시험법

제3장 드론 비탈면 관리 활용지침(안)

3.1 이상 징후

3.2 평가개념의 높은 비탈면 선정

3.3 현장조사

<표 3.1> 비탈면점검 현장조사 내용

분 류	현장조사내용
상부 자연비탈면	토석류 발생가능성 조사 자연비탈면의 인장균열 자연비탈면상의 폭포 절단, 굽힘, 함몰, 융력 등의 암공구조물 등
파기전후	인장균열 또는 이와 같거나 유사한 상태 비탈면 지상의 변화측 진행성 비탈면사용 또는 인접한 시설물의 이상 등
파기현황	비탈면에서 발생한 파기의 규모 및 개소 파기 발생 위치 발생한 파기의 유형 등
지반상태	비탈면에서 발생 토사비탈면 : 토질조건, 흙층 상조물 암반비탈면 : 지반암도, 불연속면(절리)면의 방향성 및 위치, 충화도, 불연속면의 공극특성 등
비탈면정상	비탈면의 규모 및 경사 집수지형 등
자연적 요인	강우 및 지하수 조건 비탈면내 수목의 기울어짐
인위적 요인	표면보호 상태 비탈면 표면 상태 보호 · 보강시설의 상태 배수상태 등
조사 및 시험	물리탐사 시추조사 육안조사 스프로사(RQD커레카, 일회상커레카, 초분광카메라, 라이다, 허이더 등) 토질 및 강석역학시험 등

3.4 점검

<그림 3.1> 공사중 관리 점검 업무 흐름도

<그림 3.2> 공사후 관리 점검 업무 흐름도

3.5 계측관리

<표 3.2> 계측 항목 내용

계측 항목	계측 내용	계측 기기	비고
지표	지표변위량	지표면 신축계 지표면 경사계 트콘 측량기 등	트콘 센서 (RQD커레카, 일회상커레카, 초분광카메라, 라이다, 허이더 등)
지중수평변위	지중내 수평이동	지중 수평변위계 등	
배수	강우량, 간극수압	우량계 지하수위계 간극수압계 등	

제4장 드론 비탈면 관리 활용사례

<그림 4.1> 테스트베드 현장 (화도-양평 6공구)

<그림 4.2> 레이저 스캐너 측정 자료와 측정 위치 기준점 설정

<그림 4.3> 지상 기준값과 L-SAR 변위값 비교 분석 결과

참고 문헌

- 국토해양부 (2011), 건설공사 비탈면 설계기준.
- 국토교통부 (2018), 도로비탈면 유지관리 실무매뉴얼.
- 롯데건설 (2008), 사면 기이드북.
- 한국시설안전공단 (2008), 건설공사 비탈면 유지관리지침 작성 연구.
- 한국시설안전공단 (2006), 건설공사 비탈면 유지관리지침.
- 한국시설안전공단 (2013), 절토사면 유지관리 매뉴얼 보완개정 연구.
- 한국철도시설공단 (2017), 철도설계지침 및 편람 : KR C-02005 계측.

나. 드론을 활용한 포트홀 탐지 기술 개발

(1) 드론을 활용한 포트홀 탐지 기술 개발

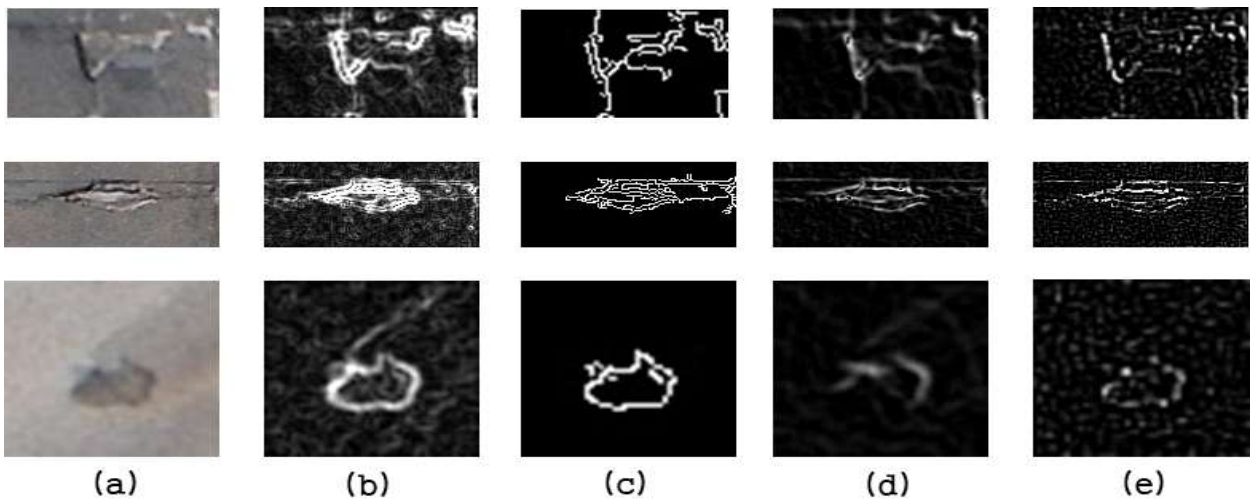
○ 포트홀 탐지기술 관련 연구

– 영상기반 포트홀 탐지기술 관련 연구

탐지 방법		정확도	처리 시간	탐지 방법 분석	
				장점	단점
온도 측정 카메라를 이용한 딥러닝 기반 포트홀 탐지(Bhatia, Y., 2019).		Accuracy:97 %	–	온도 측정 영상기반 포트홀 탐지로 정확한 탐지 가능	안개, 비가 있는 환경에서 이미지처리 불가능
블랙박스 카메라를 이용한 딥러닝 기반 포트홀 탐지(Jo, Y., 2015)		Sensitivity: 71% Precision: 88%	–	데이터베이스에 포트홀 위치 저장	통신 오류 문제 발생 가능
객체 인식 및 외곽선 검출 기반 포트홀 탐지(Baek, J. W., 2020)		Precision: 83% Accuracy: 71%	–	도로 영상 내 포트홀을 제외한 객체 제거 후 포트홀 탐지 진행으로 오탐지율을 낮춤	테스트 데이터가 적어 실제 서비스에 적용할 만큼 실험이 진행된 것이 아님
Fast R-CNN 기반 항공 영상에서 포트홀 탐지(Becker, Y., 2019)		Accuracy: 70.4%	–	이미지 증폭 기법을 이용한 학습 및 검증 데이터셋 증폭	낮은 정확도 및 느린 처리 속도
이진화 기반 도로 영상 내 포트홀 탐지(Koch, C., 2011)		Accuracy: 85.9% Precision: 81.6% Recall: 86.1%	–	포트홀 이미지의 히스토그램에서 나타나는 특징을 이용하여 포트홀 탐지	낮은 정확도 포트홀과 유사한 특징의 물체 오탐지 가능성 존재
Mask R-CNN 기반 도로 균열 검출(이민혜, 2019)		Accuracy:73 %	–	항공 영상 내 균열의 위치 파악 가능	낮은 정확도로 오탐지 가능성 존재, 처리 시간 증가
YOLO를 이용한 포트홀 탐지(Sulistyaningrum, D., 2020)		IoU: 75.1% Accuracy: 99%	0.83s (s/image)	빠른 처리속도와 높은 정확성	오탐지에 대한 성능지표가 존재하지 않아 실제 서비스 적용 시 문제 발생 가능
포트홀 탐지 방법 비교 (Vigneshwar, K., 2016)	이미지 임계처리	Accuracy: 80.6% Sensitivity: 60.4% Specificity: 83.0%	0.102s (s/image)	각 포트홀 탐지 방법별 정확도 및 처리 시간 측정으로 최적의 탐지 방법 도출	실제 테스트 샘플의 수가 20장 정도이기 때문에 실제 서비스에 적용할 만큼의 실험이 진행되지 않음
	외곽선 추출	Accuracy: 90.1% Sensitivity: 67.3% Specificity: 93.1%	0.02s (s/image)		
	K-Means 클러스터링	Accuracy: 82.4% Sensitivity: 87.1% Specificity: 82.2%	0.01s (s/image)		
	퍼지 C-Means 클러스터링	Accuracy: 82.4% Sensitivity: 71.3% Specificity: 83.6%	0.05s (s/image)		

○ 마스크 내 포트홀 영역 추출 알고리즘을 위한 관련 연구

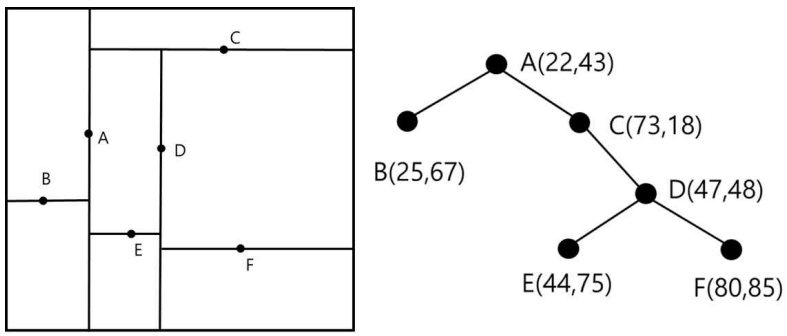
- 외곽선 검출 알고리즘
  · 개념 : 물체의 내부나 배경은 변화가 없거나 작고, 물체의 경계는 변화가 크다. 이 원리에 따라 외곽선 검출 알고리즘은 명암, 컬러 또는 텍스쳐의 변화량을 측정하고 변화량이 큰 곳을 외곽선이라고 검출
  · 활용 : 딥러닝 추론 결과 마스크 영역 내 포트홀 영역 추출 알고리즘을 위한 외곽선 추출
  · 연구 및 실험 내용 : 소벨, 프리윗, 라플라시안, 케니 외곽선 검출 알고리즘 성능 비교 진행
  · 결과 : 케니 에지가 가장 잡음이 적고 외곽선을 선명하게 검출



[외곽선 검출 알고리즘 결과((a)원본, (b)소벨, (c)케니, (d)프리윗, (e)라플라시안)]

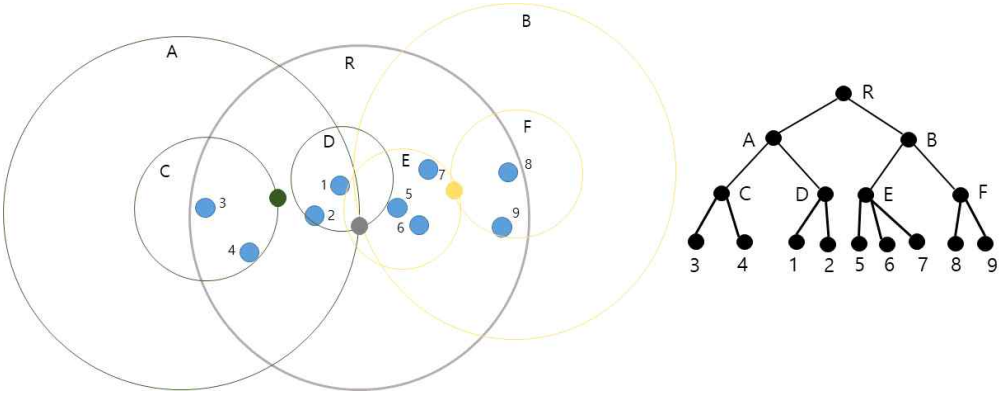
- 소벨(Sobel) 외곽선 검출
  · 개념 : 마스크와 밝기에 대한 1차 미분연산을 통해 물체의 경계 검출, 모든 방향에 외곽선을 추출하고 수직, 수평 외곽선보다 대각선 방향 외곽선에 민감하게 반응
  · 활용 : 딥러닝 추론 결과 마스크 영역 내 포트홀 영역 추출 알고리즘을 위한 외곽선 추출
  · 연구 및 실험 내용 : 소벨, 프리윗, 라플라시안, 케니 외곽선 검출 알고리즘 성능 비교 진행
  · 결과 : 잡음이 많아 포트홀 외곽선 외의 영역탐지
- 프리윗(Prewitt) 외곽선 검출
  · 개념 : 마스크와 밝기에 대한 1차 미분연산을 통해 물체의 경계 검출, 소벨에 비해 처리 시간이 빠르지만, 외곽선이 덜 드러나는 특징
  · 활용 : 딥러닝 추론 결과 마스크 영역 내 포트홀 영역 추출 알고리즘을 위한 외곽선 추출
  · 연구 및 실험 내용 : 소벨, 프리윗, 라플라시안, 케니 외곽선 검출 알고리즘 성능 비교 진행
  · 결과 : 잡음은 적으나 외곽선이 덜 드러남
- 라플라시안(Laplacian) 외곽선 검출
  · 개념 : 1차 미분의 방식은 밝기가 점진적으로 변화하는 부분까지 외곽선을 탐지하여 불필요한 외곽선 검출, 라플라시안 외곽선 검출은 2차 미분 연산자를 통해 마스크 영역의 중심에 있는 외곽선 검출
  · 활용 : 딥러닝 추론 결과 마스크 영역 내 포트홀 영역 추출 알고리즘을 위한 외곽선 추출
  · 연구 및 실험 내용 : 소벨, 프리윗, 라플라시안, 케니 외곽선 검출 알고리즘 성능 비교 진행
  · 결과 : 잡음은 적으나 외곽선이 뚜렷하게 드러나지 않음
- 케니(Canny) 외곽선 검출
  · 개념 : 기존 1차 미분의 방식에 블러링, 비 최댓값(Non-maximum value)제거, 히스테리시스 임계값(hysteresis Thresholding)을 추가하는 방식이며 소벨, 프리윗 외곽선 추출 결과보다 성능이 좋다고 알려짐
  · 활용 : 딥러닝 추론 결과 마스크 영역 내 포트홀 영역 추출 알고리즘을 위한 외곽선 추출
  · 연구 및 실험 내용 : 소벨, 프리윗, 라플라시안, 케니 외곽선 검출 알고리즘 성능 비교 진행
  · 결과 : 가장 잡음 적고 외곽선을 선명하게 검출, 본 연구에서 사용되는 외곽선 검출 알고리즘

- kd tree
  · 개념 : 각 노드의 데이터가 특정 수의 차원 공간의 점들을 구조화하는 공간 분할 트리
  · 활용 : 외곽선 검출 결과들을 최단거리로 연결하여 포트홀 추론 결과 마스크 내 포트홀 영역 검출
  · 연구 및 실험 내용 : kd tree, ball tree 성능 비교 실험 진행
  · 결과 : 본 연구의 외곽선 검출 결과 최단거리 연결 알고리즘으로 사용됨



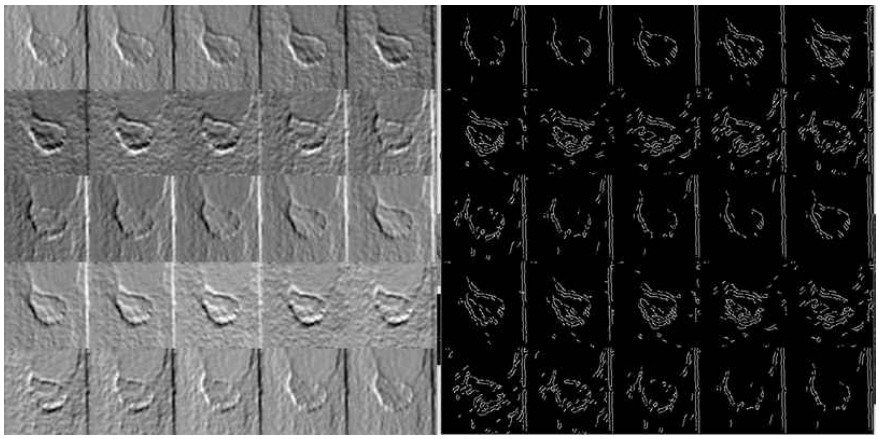
[kd tree 예시]

- ball tree
  · 개념 : ball tree는 특정 수의 차원 공간 안에 점 중 무작위로 점을 선정하고 해당 점에서 가장 멀리 있는 두 개의 점을 선정한다. 그 두 점 사이의 중점을 사이에 두고 두점이 중점인 원을 두 개 만든다. 그리고 생긴 원안에서 새로운 원을 계속 만들어 최근접한 점을 찾아내는 알고리즘
  · 활용 : 외곽선 검출 결과들을 최단거리로 연결하여 포트홀 추론 결과 마스크 내 포트홀 영역 검출
  · 연구 및 실험 내용 : kd tree, ball tree 성능 비교 실험 진행
  · 결과 : 본 연구의 외곽선 검출 결과 최단거리 연결 알고리즘으로 사용됨



[ball tree 예시]

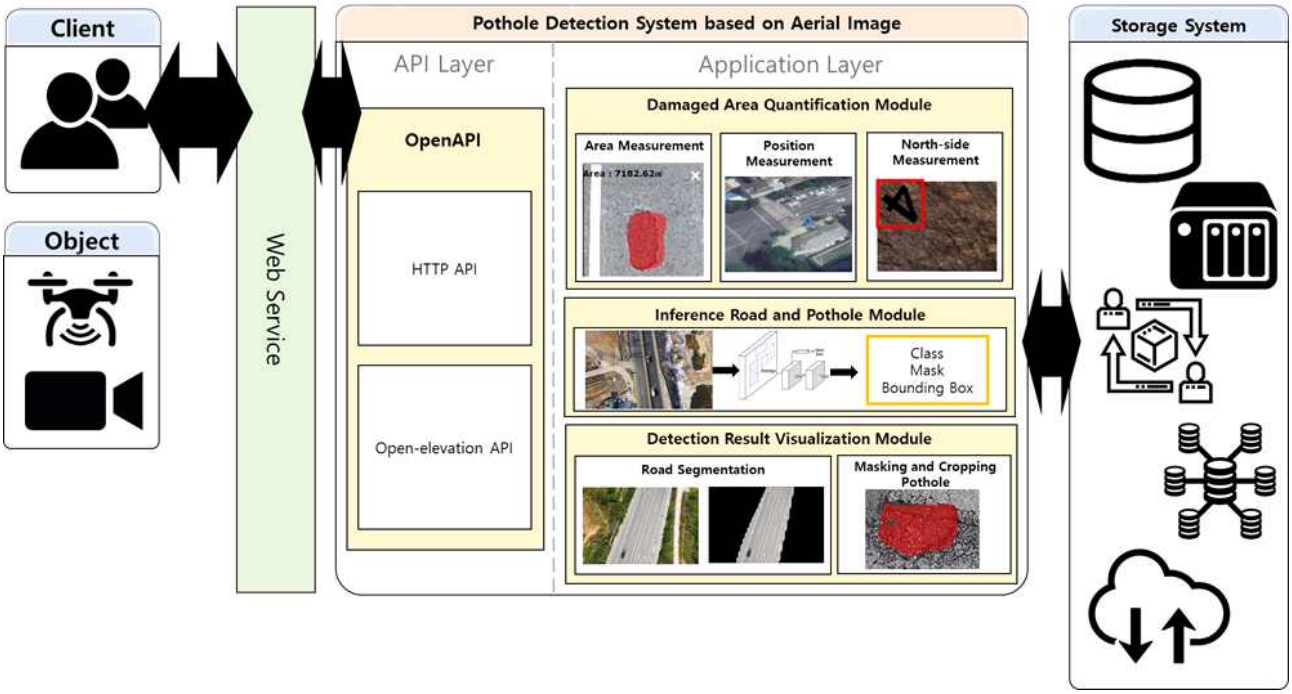
- steerable filter
  · 개념 : 이미지 특징 추출에 사용되는 orientation-selective convolution kernel로, 회전된 작은 버전의 선형 조합을 통해 표현될 수 있다.
  · 활용 : 딥러닝 추론 결과 마스크 영역 내 포트홀 영역 추출 알고리즘을 위한 외곽선 추출
  · 연구 및 실험 내용 : 외곽선 영역 추출
  · 결과 : 잡음이 적고 외곽선을 선명하게 검출, 본 연구에서 사용되는 외곽선 검출 알고리즘



[steerable filter 예시]

- 밀도 기반 클러스터링(DBSCAN)
  · 개념 : 밀도 기반의 클러스터링은 점이 세밀하게 몰려 있어서 밀도가 높은 부분을 클러스터링 하는 방식
  · 활용 : steerable filter의 결과에서 가장 큰 군집 탐색
  · 결과 : steerable filter의 결과에서 가장 큰 군집이 포트홀로 검출

○ 드론 영상 데이터 기반 포트홀 탐지 소프트웨어 추가 개발



[포트홀 자동탐지 관리시스템 아키텍쳐]

- 항공 영상 내 포트홀 자동탐지 결과 관리시스템 구현
  · Inference Road and Pothole Module : 딥러닝 기반 포트홀 탐지 인공신경망과 도로 탐지 인공신경망으로 구성
  · Detection Result Visualization Module : 도로 탐지 결과를 이용한 도로 영역 분할 및 포트홀 탐지 결과를 이용한 포트홀 영역 추출
  · Damaged Area Quantification Module : 포트홀 면적 및 가로, 세로 측정, 항공 영상 내 포트홀 좌표 측정, 도로의 상행 하행 구분을 위한 정북 방향 표시 등의 기능
  · Storage System : 각 결과 및 데이터를 파일 형태별로 저장, 데이터베이스 내 포트홀 자동탐지 결과데이터 저장
  · Web Service : 포트홀 자동탐지 결과 관리시스템 웹서비스로 제공

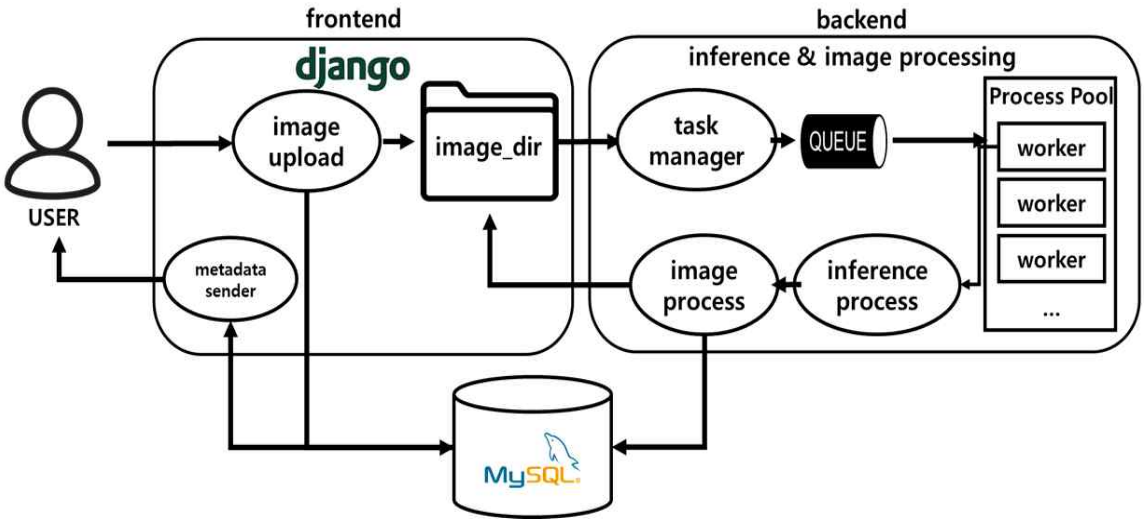


[포트홀 자동탐지 결과 관리시스템 메인페이지]



[포트홀 자동탐지 결과 시각화 페이지]

- 항공 영상 내 포트홀 자동탐지 속도 향상 연구 진행
  · 멀티프로세싱 풀(Multiprocessing Pool)을 이용한 포트홀 병렬 추론 연구
  · 동시에 여러 장의 항공 영상을 추론하여 항공 영상 내 포트홀 탐지 속도 향상



[포트홀 자동탐지 시스템의 데이터 처리 흐름도]

○ 딥러닝 기반 항공 영상 이미지 내 포트홀 탐지 알고리즘 구현

- Mask R-CNN 기반 포트홀 탐지 알고리즘 개발
  · 영상처리 및 Inception v3 기반 포트홀 탐지 알고리즘보다 높은 검지율 낮은 오탐율
  · RPN(Region Proposal Network)를 통해 영상 내 객체 추출 후 포트홀 탐지
  · 추론 결과로 추출된 마스크 영역을 통해 포트홀의 면적 추론 가능₩



〈원본 항공 영상〉 〈포트홀 자동 탐지결과〉

[포트홀 자동 탐지 결과 마스크 영역]

- Mask R-CNN 기반 도로 추출 알고리즘 개발
  · 도로 내 포트홀만 탐지하기 위한 도로 추출 알고리즘 개발
  · 도로 외 오탐지 영역 제거로 인한 오탐지율 감소



〈원본 항공 영상〉 〈도로만 추출된 항공 영상〉

[항공 영상 내 도로 추출 결과]

○ 항공 영상 및 도로 영상 기반 딥러닝용 데이터 셋 구축

- 약 730,134장 데이터 셋 구축
  · Mask R-CNN 기반 포트홀 탐지 인공신경망 학습을 위한 항공 영상 내 포트홀 라벨링 진행
  · 신규 항공 영상 추가 및 데이터증강을 통한 데이터셋 증폭

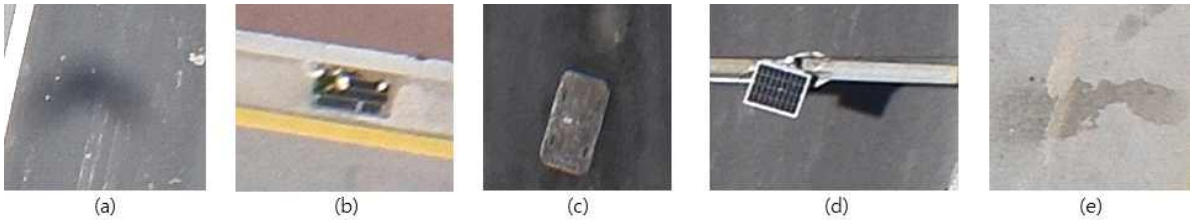
구분	데이터 셋 명	증강 데이터 수 (기존 데이터 수)	총 이미지 장 수
오픈 데이터 셋	크랜필드 데이터셋(Alzoubi, 2018)	86,992장 (10,874장)	약 730,134장
	Pothole 600(Fan, R. 2018)	4,800장 (600장)	
자체 구축 데이터 셋	도로 주행 영상 데이터 셋	약 371,190장	
	전이 학습 용 데이터셋	161,888장(20,236)장	
	포트홀 이진 분류 데이터 셋	99,264장(12,408장)	
	객체 분할 데이터 셋	6,000장(1,500장)	



〈도로 라벨링 수행〉 〈포트홀 라벨링 수행〉

[포트홀 및 도로 탐지 모델망을 위한 데이터셋 구축]

○ 딥러닝 기반 항공 영상 이미지 내 포트홀 탐지 결과 분석
- 광주 인근 국도의 특정 구간을 약 70m 고도에서 촬영한 항공 영상 79장 기준(포트홀 총 3개)
- 항공 영상 내 포트홀 탐지 결과(포트홀 탐지)
  · 포트홀 검지율 : 탐지 포트홀 수 / 포트홀 수 [TP/(TP+FN)]
  · 포트홀 오검지율 : 오탐지 횟수 / 포트홀 탐지 횟수 [FP/(TP+FP)]
  · 포트홀 검지율 : 100%
  · 포트홀 오검지율 : 54.24%
- 항공 영상 내 도로 추출 결과(도로 추출)
  · 도로 추출 영역 평균 IoU(Intersection over Union) : (정답 도로 영역∩도로 추출 영역)/(정답 도로 영역∪도로 추출 영역)
  · 도로 검지율 : 탐지 도로 수 / 도로 수 [TP/(TP+FN)]
  · 도로 추출 영역 평균 IoU : 약 0.7
  · 도로 검지율 : 98%
- 항공 영상 내 포트홀 탐지 결과(포트홀 탐지 + 도로 추출)
  · 포트홀 검지율 : 탐지 포트홀 수 / 포트홀 수 [TP/(TP+FN)]
  · 포트홀 오검지율 : 오탐지 횟수 / 포트홀 탐지 횟수 [FP/(TP+FP)]
  · 포트홀 검지율 : 97.2%
  · 포트홀 오검지율 : 9.8%
- 2차년도 탐지결과와 3차년도 탐지결과 비교
  · 2차년도 : 78.78%(정탐율), 18.30%(오탐율)
  · 2차년도 대비 정탐율 18.42%p 상승 및 오탐율 8.50%p 감소
  · 학습 및 검증 데이터 셋 추가 구축, 모델망 변경을 통한 오탐율 감소 확인
- 3차년도 오탐 케이스
  · a 오탐 : 드론 그림자를 포트홀로 오탐
  · b, c 오탐 : 길 위의 맨홀과 하수구를 포트홀로 오탐
  · d 오탐 : 도로 내 신호등, 속도 단속 카메라와 같은 구조물을 포트홀로 오탐
  · e 오탐 : 물 자국, 기름 자국을 포트홀로 오탐



[포트홀 오탐지 예시]

○ 포트홀 정확한 면적 측정을 위한 마스크 내 포트홀 영역 추출 알고리즘 개발
- 포트홀의 면적을 정확하게 측정하기 위해 포트홀 탐지의 결과로 나온 포트홀 마스크 영역 내에서 포트홀 영역 추출

- 캐니 외곽선 검출(Canny Edge)과 kd tree 기반 외곽선 연결을 이용한 포트홀 영역 추출
  · 탐지 결과와 캐니 외곽선 검출을 AND 연산 후, kd tree 기반 외곽선 연결 적용



[케니 외곽선 검출과 kd tree 기반 외곽선 연결을 이용한 포트홀 영역 추출 결과]

- Steerable filter와 DBSCAN, kd tree기반 외곽선 연결을 이용한 포트홀 영역 추출
  · steerable filter의 결과에서 DBSCAN을 이용하여 가장 큰 군집 판별
  · 가장 큰 군집과 탐지결과 AND 연산
  · kd tree 기반 외곽선 연결 수행 후 Floodfill 연산



[Steerable filter, DBSCAN, kd tree기반 외곽선 연결 이용 포트홀영역추출 결과]

- 추출 결과 비교
  · 중복 오차율 : $\frac{(Ground\ Truth \cap Prediction)^C}{Ground\ Truth \cup Prediction}$ = 1-IoU(Intersection over Union)
  · 데이터 셋 : 포트홀 영상 74장
  · Mask R-CNN 포트홀 영역 추출 결과 : 58.7%
  · Steerable filter와 DBSCAN,kd tree을 이용한 포트홀 영역 추출 결과 : 47.9%
  · 캐니 외곽선 검출과 kdtree를 이용한 포트홀 영역 추출 결과 : 51.4%

○ 항공 영상 내 도로의 상행 하행 판단을 위한 정북 방향 표시 기능 개발
- 이미지의 XMP(Extensible Metadata Platform) 데이터 내 yaw 값을 이용하여 이미지의 정북 방향 계산



[정북 방향 회전 영상 관련 그림]

○ 항공 영상 내 포트홀 좌표 계산 알고리즘 개발
- 항공 영상 내에서 정확한 포트홀 좌표를 계산하여 실무자에게 편의성 제공
- 포트홀 좌표 계산 공식

[계산 공식에 대한 기호 설명]

약호, 기호	설명	단위
x	위도 반지름	km
y	적도 반지름	km
z	극 반지름	km
n	적도 기준에 따른 값(북위 = 1, 남위 = −1)	−
L	위도	도
K	경도	도

· EXIF 데이터 내 위도와 경도 값을 식(3),(4)를 이용해 도 단위 값으로 변환

$$L = \sqrt{y \times cos(x) - y \times cos(x+n))^2 + (z \times sin(x) - z \times sin(x+n))^2} \quad (3)$$

$$K = \cos(x) \times 2\pi \times y/360 \quad (4)$$

· 위도와 경도를 1초 단위 값으로 변환한 후, 영상의 중점에서 포트홀까지의 가로 거리와 세로거리를 위도와 경도로 나누어 거리를 초단위로 변환

· 영상의 중점좌표에 초단위로 변환된 포트홀 까지의 가로, 세로거리를 더하여 포트홀 위치 좌표 계산

○ 포트홀 발생원인, 파손유형에 대한 조사·분석

– 아스팔트 포장의 포트홀은 ① 아스팔트 포장의 균열에 빗물이 침투, ② 포장이 수분으로 포화 ③ 아스팔트 박리, 동결/융해에 따른 포장 지지력 저하 등으로 인해 포장 파손의 순서로 진행되며 발생함

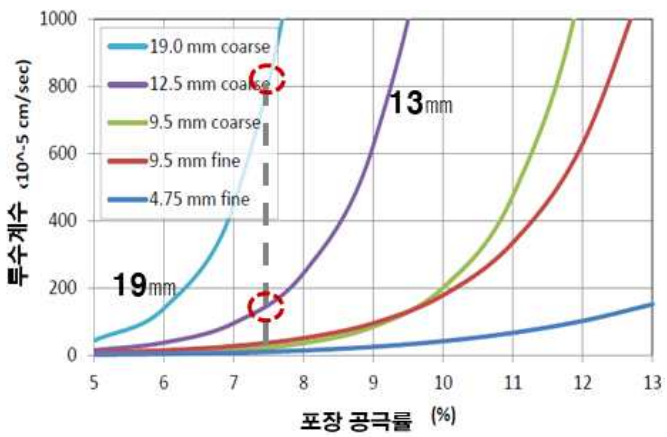
① 아스팔트 포장의 균열에 빗물이 침투

- 아스팔트 혼합물 불량 또는 다짐부족으로 아스팔트 포장의 공극률이 높을 경우 투수계수가 높아져서 빗물의 침투가 많아짐

· 아스팔트 포장의 공극률이 동일해도 아스팔트 혼합물의 최대 골재 크기가 클수록 투수계수가 크게 높아짐. 이는 폐쇄형 공극이 줄고 연결되어 수분이 침투될 수 있는 공극이 증가함에 따름

- 포장 노후시에 발생하는 거북등 균열, 종방향 조인트의 다짐부족시 발생하는 종방향 균열 등을 조기에 보수하지 않을 경우 수분 침투가 높아짐

➜ 주요원인: **포장 균열, 국부적인 포장 공극률 불량**



아스팔트 포장의 종류에 따른 투수계수

② 포장이 수분으로 포화

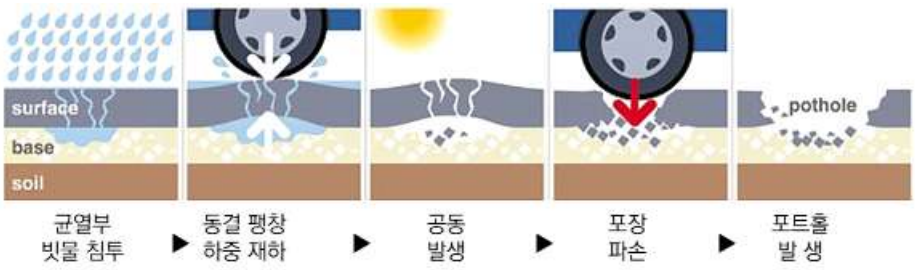
- 노상 또는 보조기층의 배수 불량 또는 동상방지층 설계/재료 불량으로 포장이 수분으로 포화
- 여름철 장마기간이 증가할 경우 아스팔트 포장에 침투된 수분이 조기에 배수되지 못하여 포장이 수분으로 포화됨
- 겨울철 폭설이 지속될 경우 포장에 제설제를 사용하고, 이에 따라 융해된 물이 포장에 침투하여 포장이 포화됨

➜ 주요원인: **기후 환경, 배수 불량**

③ 아스팔트 박리, 동결/융해에 따른 포장 지지력 저하 등으로 인해 포장 파손

- 노상 또는 보조기층에 있는 수분이 겨울철 동결로 인해 팽창하여 아스팔트 포장을 융기 시키고, 봄철 해빙에 따라 노상 또는 보조기층의 지지력이 낮아져 교통량에 의해 침하되어 포트홀 발생
- 차량이 통행할 때마다 포장 내부에 포화된 수분으로 인해 간극수압이 발생/해소가 반복되어 골재 표면의 아스팔트가 박리되어 강도가 저하되고, 수분침투가 증가하여 급격하게 파손되며 포트홀 발생
- 아스팔트 혼합물이 수분에 취약하여 박리가 쉽게 발생할수록 포트홀 발생이 높아짐

➜ 주요원인: **아스팔트 혼합물 박리저항성 불량, 동상방지층 불량, 배수 불량**



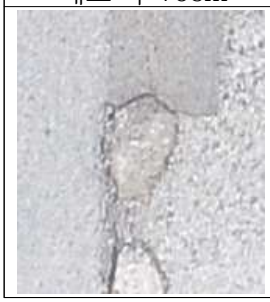
[배수불량으로 인한 포트홀 발생 과정]

– 포트홀 발생 형태

· 균열 + 포트홀

가로:약 110cm 세로:약 170cm	가로:약 38cm 세로:약 54cm	가로:약 63cm 세로:약 62cm	가로:약 49cm 세로:약 72cm
			

- 포트홀

가로:약 52cm 세로:약 78cm	가로:약 52cm 세로:약 78cm	가로:약 45cm 세로:약 53cm	가로:약 52cm 세로:약 78cm
			

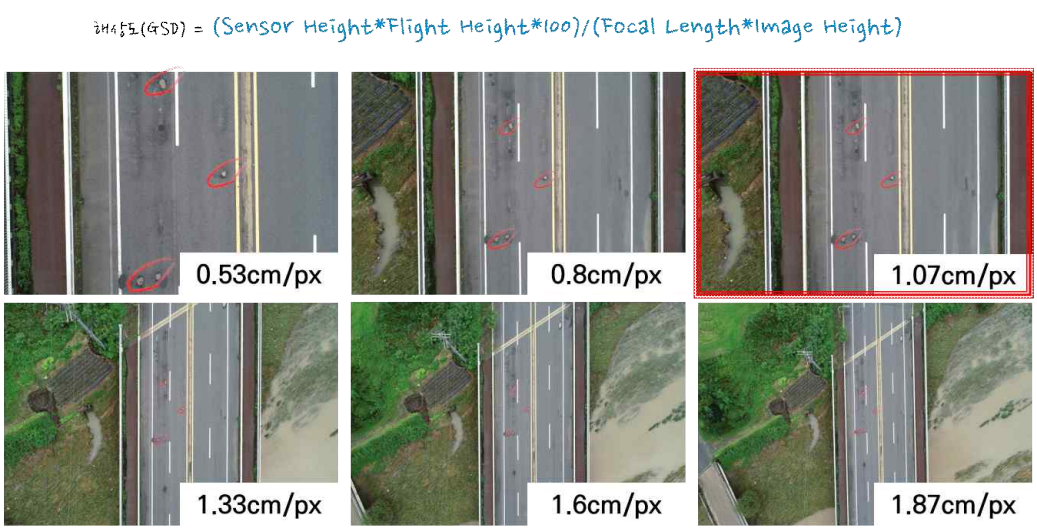
· 포트홀 자료 분석 결과 균열이 진전하여 포트홀이 발생한 경우는 약 60%, 주변부 균열 없이 갑자기 포트홀이 발생한 경우는 40%인 것으로 파악되었음

(2) 포트홀 탐지를 위한 드론시스템 기술 개발

○ 고정밀 GPS(RTK/PPK) 탑재 드론 포트홀 AI학습용 상시 데이터 수집

- 기존 정사영상 판독에서 원본영상 판독으로 학습 알고리즘 변화에 대한 포트홀 촬영 해상도 재선정
- 해상도별 영상 촬영결과 1㎝/px 이하일 때, 포트홀 육안식별이 용이한 것으로 판정

고도 (m)	해상도(cm/px) 회전익 (20mp)	고정익 (24mp)
20	0.53	
30	0.8	저고도 비행불가
40	1.07	
47	1.25	1.0
50	1.33	1.05
60	1.6	1.27
70	1.87	1.48



- 국도 약 111.4㎞에 대한 드론촬영 수행

No	촬영일	노선번호	지역	투입드론	해상도	사진매수	노선길이(km)
1	02월 03일	38번 국도	안성	Phantom4 Pro ebee X	1.0cm/px 1.5cm/px	73	1
2	02월 04일	38번 국도	음성, 이천	Phantom4 Pro	1.0cm/px	64	1
3	02월 05일	42번국도	원주, 여주, 이천	ebee X	1.5cm/px	713	11
4	02월 06일	38번국도	이천, 안성	ebee X	1.5cm/px	2,982	42.3
5	05월 11일	42번국도	여주, 이천	eBee Plus	1.8cm/px	297	16.5
6	05월 20일	38번 국도	평택	ebee X	1.5cm/px	512	7.4
7	06월 09일	하남진곡산단로	광주, 담양, 장성	ebee X	1.5cm/px	505	6.8
8	07월 22일	26번국도	군산	ebee X	1.5cm/px	1,274	16.2
9	08월 04일	29번국도	화순	ebee X	1.5cm/px	867	8.11
10	08월 11일	빛고을대로, 13번, 15번 국도	광주, 담양, 장성	Phantom4 Pro	0.53~1.87/px	93	1
합계						**7,385**	**111.39**
평균						**738**	**11.14**

- 서울지방국토관리청 산하 수원국토관리사무소 관리국도 40km 실증







- 일 시 : 2020.2.6
- 장 소 : 국도 38호선(안성터미널~감곡 IC) 약 40km
- 내 용 : 도로 포트홀 탐지기술 현장조사, 자료처리, 결과협의 등 기관간 기술개발 현장실험

담당자 요구사항

- 영상에서 육안으로 포트홀 식별이 어려움
  (화전익 드론 활용 고도별 영상을 취득하여 식별 가능한 해상도 재선정)
- 조사 및 분석시간 단축 필요
  (상습 포트홀 발생구간 사전답사를 통한 신속 DB구축체계 마련)



○ 포트홀 탐지 시스템 운용 체계 구축

- 신속, 체계적인 프로세스를 통한 신속 DB구축 체계 마련

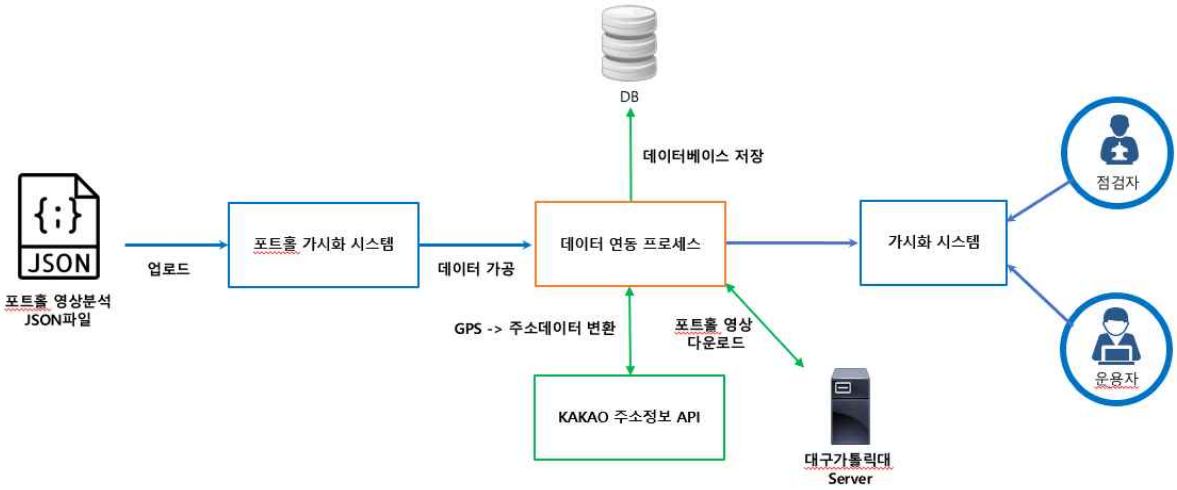


- 포트홀 탐지 드론 운용 매뉴얼 제작
  - 드론의 분류, 장치신고, 보험가입과 같은 드론운용 사전 준비사항
  - 비행전 대상지선정, 공역확인, 촬영허가 및 비행승인신청 등 사전 준비항목
  - 드론 운용 및 안전, 단계별 주의사항, 사고 발생 시 조취사항
  - 포트홀 자동 탐지시스템 및 가시화 시스템 운용법

(3) 포트홀 위치 가시화

○ 포트홀 가시화시스템

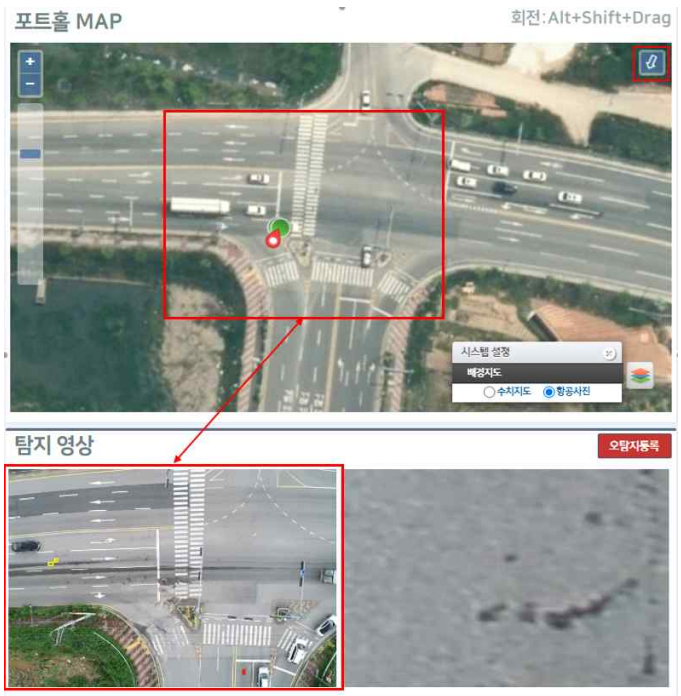
- 드론의 촬영 영상의 포트홀 탐지 영상분석 결과데이터를 가시화하여 운용할 수 있는 웹 서비스 시스템개발. 운용자는 포트홀 영상분석 결과데이터를 실시간 업로드하여 Vworld기반 지도 레이어에 가시화 할 수 있으며, 작업자는 시스템을 통해 포트홀 발생위치를 인지하고 레포팅 하여 현장조치 할 수 있는 시스템으로 구성됨.



[포트홀 가시화시스템 아키텍쳐]

○ UI 설계와 고도화

- 시스템 인터페이스 개선을 위하여 연구원들의 요구사항을 반영한 UI/UX 아키텍처의 재구성을 수행하였음. Interaction Design을 고려하여 레이아웃과 Frontend Architecture를 재개발함.
- 포트홀 영상분석 전용 웹 뷰어를 개발하여 포트홀의 탐지위치의 직관적인 확인과 원본 데이터 다운로드 등의 운용편의성 기능 개발하였음. HTML5 Canvas, Asynchronous Javascript And XML기술을 적용하여 포트홀 탐지위치에 대한 동적랜더링 기술을 Library형태로 개발/적용함



[정북방향 동기화 UI]

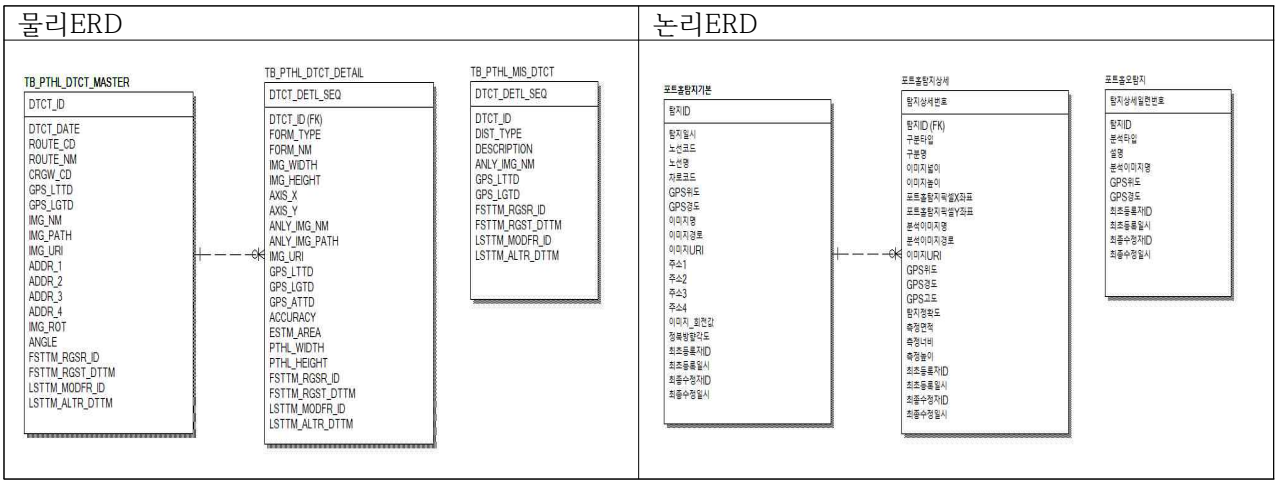
- 웹브라우저의 HTML Canvas형태로 구성된 지도 API데이터를 PDF이미지로 변환하는 Javascript module을 개발하였음.
- 포트홀 정북방향과의 오차각도값을 지도 API에 연동하여 드론촬영이미지와 지도 API의 정북방향의 동기화 로직을 구현함. 이로 인해 드론의 포트홀 촬영영상과 지도API를 확인하여 운용자의 직관적인 포트홀 위치탐지가 가능함.

○ 데이터 연동 프로토콜 추가 반영
- 포트홀 분석데이터 연동 프로토콜을 추가 반영하였으며 기존 데이터에대해 전체적인 재수집 작업을 진행함. 연동데이터의 자료구조를 분석하여 데이터모델 Class를 설계 데이터 가공 시 자료형을 정의함(소수점자릿수, 문자열제한 등).
- 정북방향기준 각도오차값, 탐지면적, 포트홀 너비, 포트홀 높이, 포트홀 위도, 포트홀 경도 데이터를 추가 연동함

[시스템 데이터연동 정의]

key	value	data type (Oracl Database)	데이터 정의 년도
name	드론 촬영이미지명	VARCHAR2(300)	2차년도
type	영상분석 결과 유형	VARCHAR2(200)	2차년도
image_uri	영상분석 결과 이미지 URL	VARCHAR2(2000)	2차년도
latitude	드론위도	NUMBER(12,9)	2차년도
longitude	드론경도	NUMBER(12,9)	2차년도
altitude	고도	NUMBER	2차년도
angle	정북방향 각도오차값	NUMBER	3차년도
area	탐지면적	NUMBER	3차년도
axis_x	드론 촬영이미지 기준 X축 탐지위치 픽셀	NUMBER	2차년도
axis_y	드론 촬영이미지 기준 Y축 탐지위치 픽셀	NUMBER	2차년도
axis_w	드론 촬영이미지 기준 X축 탐지Box 넓이	NUMBER	2차년도
axis_h	드론 촬영이미지 기준 Y축 탐지Box 높이	NUMBER	2차년도
pothole_width	포트홀 너비	NUMBER	3차년도
pothole_height	포트홀 높이	NUMBER	3차년도
pothole_lat	포트홀 위도	NUMBER(12,9)	3차년도
pothole_long	포트홀 경도	NUMBER(12,9)	3차년도

○ Database 설계 개선
- 포트홀 분석 시스템의 추가된 데이터 요소를 포트홀 가시화 시스템에 연동시키기 위하여 Database 아키텍처를 추가 개선함.
- Database ER-Diagram은 Logical Data Model과 Physical DataModel을 구분하여 작성하였으며 각 Entity 항목별로 DB Table을 작성함.
- 포트홀 가시화 시스템에 대한 Database ER-Diagram(Logical, Physical Data Model) 설계는 포트홀 탐지 원시영상(드론의 촬영영상)을 기본정보로 하며, 탐지한 포트홀의 좌표정보, 포트홀 너비, 면적, GPS Coordinate를 하위정보로 하여 각 Table에 대하여 Primary Key, Foreign key등 Relation구성을 보다 더 고도화 하여 재설계하였음.



[포트홀 가시화시스템 ER-Diagram]

○ KAKAO GPS API를 연동한 시스템 주소정보 수집 자동화

Response


```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
  "meta": {
    "total_count": 2
  },
  "documents": [
    {
      "region_type": "B",
      "address_name": "경기도 성남시 분당구 삼평동",
      "region_1depth_name": "경기도",
      "region_2depth_name": "성남시 분당구",
      "region_3depth_name": "삼평동",
      "region_4depth_name": "",
      "code": "4113510900",
      "x": 127.10459896729914,
      "y": 37.40269721785548
    },
    {
      "region_type": "H",
      "address_name": "경기도 성남시 분당구 삼평동",
      "region_1depth_name": "경기도",
      "region_2depth_name": "성남시 분당구",
      "region_3depth_name": "삼평동",
      "region_4depth_name": "",
      "code": "4113565500",
      "x": 127.1163593869371,
      "y": 37.40612091848614
    }
  ]
}
```


[KAKAO API 응답 데이터]

- 가톨릭대학교의 포트홀 분석 데이터의 GPS좌표값을 Java HttpsURLConnection을 활용하여 주소값 연동 모듈을 설계 및 구현함.
- 위도, 경도 좌표정보를 Get Request하여 Response받은 Json데이터를 포트홀 가시화시스템 데이터모델에 파싱 -> DB화 로직을 적용하여 주소데이터의 자동화 연동이 가능해짐.
- 시스템 연동된 포트홀 영상분석 이미지 데이터의 보관을 위한 Storege 환경을 구축하였음.
- 포트홀 면적에대한 위험도 임계치 가시화를 위하여 차량별 타이어 규격정보 806건 조사하여 위험도 상,중,하 별 임계면적값을 판정하여 시스템 적용하였음.



추정면적(㎡)	위험도
1290	하
980.4	하
7940.4	상
5452.8	상
2707.2	중
2565.6	하

[차량별 타이어면적 데이터조사(좌), 타이어 면적데이터 기반 시스템 임계치 적용(우)]

- 데이터 연동 이외 오탐지 분류 기능을 시스템에 추가하여 7115건에 대한 오탐지 처리작업 수행하였음.

	DTCT_ID	DTCT_DETL_SEQ	DIST_TYPE	DESCRIPTION	FSTTM_RGSR_ID	FSTTM_RGST_DTTM	LSTTM_MODFR_ID	LSTTM_ALTR_DTTM	ANLY_IN
1	26	202000008450	02	워크샵 오탐지	SYSTEM	2020-05-11 16:59:17	SYSTEM	2020-05-11 16:59:17	881_929_2
2	292	202000009247	02	[NULL]	SYSTEM	2020-05-11 17:01:54	SYSTEM	2020-05-11 17:01:54	1666_1719
3	203	202000009000	02	[NULL]	SYSTEM	2020-05-13 18:29:48	SYSTEM	2020-05-13 18:29:48	2541_2617
4	45	202000005877	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:30:39	SYSTEM	2020-05-13 18:30:39	1979_2014
5	36	202000005862	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:30:39	SYSTEM	2020-05-13 18:30:39	872_919_3
6	218	202000009033	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	2553_2601
7	218	202000009032	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	48_101_81
8	218	202000009035	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	1046_1100
9	296	202000009263	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	2462_2504
10	262	202000009146	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	1874_1917
11	35	202000008482	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	133_181_5
12	81	202000008649	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	2472_2521
13	81	202000008647	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	945_993_5
14	81	202000008650	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	1356_1399
15	81	202000008648	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	2579_2627
16	58	202000005905	02	도로를 벗어난 포트홀	SYSTEM	2020-02-17 18:09:43	SYSTEM	2020-02-17 18:09:43	2155_2272
17	145	202000008840	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	3316_3359
18	145	202000008839	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	523_566_3
19	145	202000008841	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	2798_2841
20	145	202000008838	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	1004_1046
21	145	202000008842	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	347_384_4
22	267	202000009155	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	2585_2622
23	59	202000008539	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:36:14	SYSTEM	2020-05-13 18:36:14	1986_2045
24	292	202000009246	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	2334_2376
25	236	202000009080	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	1767_1826
26	75	202000008611	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	3012_3065
27	75	202000008610	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	2259_2318
28	205	202000009002	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	218_251_3
29	251	202000009117	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	1874_1933
30	298	202000009267	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	181_240_2
31	108	202000008743	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	2467_2521
32	17	202000008426	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	2649_2713
33	313	202000009330	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	3268_3316
34	313	202000009331	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	1778_1826
35	313	202000009326	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	747_801_3
36	313	202000009328	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	1629_1682
37	313	202000009324	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	924_956_3
38	140	202000008821	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	3124_3177
39	140	202000008820	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	3327_3380
40	140	202000008816	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	2008_2067
41	140	202000008814	02	오탐지 일괄 처리	SYSTEM	2020-05-13 18:38:25	SYSTEM	2020-05-13 18:38:25	2617_2665

[오탐지 처리 이력]

- 수원 국토관리사무소를 방문하여 시스템에 개선사항과 실무자 추가 요구사항을 반영하여 HTML5, JSPDF기술을 활용한 현장점검 Report출력 기능 개발하였음.



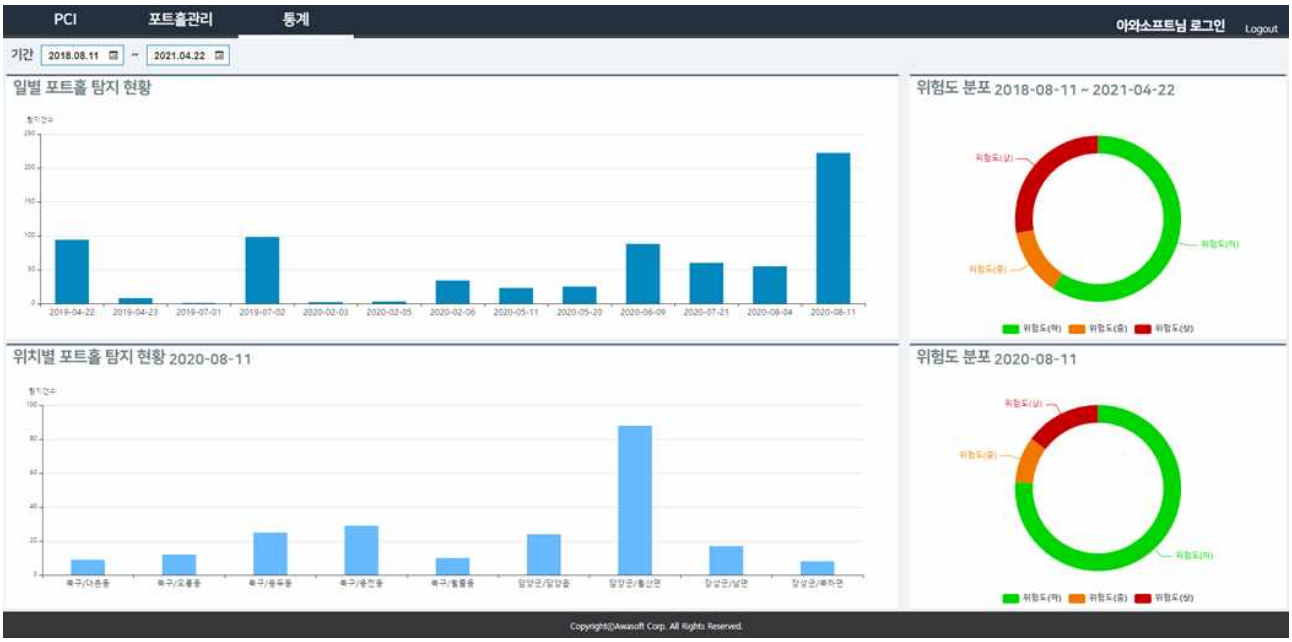
[포트홀 작업 Report 기능]

- 대구 가톨릭대 연구진과 협력하여 지속적인 포트홀 데이터 연동작업 수행하였음.

촬영일자	위치
2020년 2월 3일 ~ 2월 7일	국도 38호선(안성터미널 ~ 감곡IC)
2020년 05월 11일	이천시
2020년 05월 20일	평택 18번 국도
2020년 06월 09일	광주광역시
2020년 07월 21일	군산시
2020년 08월 04일	화순군
2020년 08월 11일	광주광역시

[2차년도 데이터 연동 이력]

- 시스템 연동 포트홀정보 통계화면을 개발하여 시스템 내 수집된 포트홀 정보데이터를 탐지건수와 위험도 분포 2가지 분류로 시각화작업을 수행. 탐지일자, 지역별 데이터를 간단하게 조회할 수 있는 Chart간 상호작용 UI를 개발하였음.



[시스템 연동 포트홀정보 통계 화면]

다. 드론을 활용한 포장 노후화 상태(PCI) 측정 기술 개발

(1) 포장 노후화 상태(PCI) 측정을 위한 초분광 분석 모델 고도화 방안 개발

○ 도로포장 조사 차량을 이용한 PCI 측정 방법

- 포장평가지수인 PCI는 일반국도, 고속국도, 서울시 등에 따라 계산식의 차이가 있으나, 아래와 같이 균열율, 소성변형, 종단평탄성 등 3가지 변수를 이용하여 계산

· 일반국도 NHPCI (National Highway Pavement Condition Index)

$$NHPCI = \frac{1}{(0.33 + 0.003 \times X_{CR} + 0.004 \times X_{RD} + 0.0183 \times X_{IRI})^2}$$

여기서, X_{CR} : 균열율(%)

X_{CR} : 소성변형 깊이(㎜)

X_{IRI} : 종단평탄성(m/㎞))

· 고속국도 HPCI (Highway Pavement Condition Index)

$$HPCI = 5 - 0.9 \times \log(1 + SD) - 0.75 \times RD^{1.2} - 0.54 \times IRI^{0.8}$$

여기서, SD: 균열, 소파보수 등 표면손상 면적(㎡)

RD: 소성변형 깊이(㎜)

IRI: 종단평탄성(m/㎞)

· 서울시 SPI (Seoul Pavement Index)

$$SPI = 10 - [(10 - SPI_1)^5 + (10 - SPI_2)^5 + (10 - SPI_2)^5]^{\frac{1}{5}}$$

여기서, $SPI_1 = 10 - 2.23 \times C^{0.3}$ (C: 균열율(%))

$SPI_2 = 10 - 0.2 \times RD$ (RD: 소성변형 깊이(㎜))

$SPI_3 = 10 - 0.667 \times IRI$ (IRI: 종단평탄성(m/㎞))

- 조사차량을 이용한 균열율, 소성변형, 종단평탄성은 10m당 다음의 방법에 따라 측정함

· 균열율: 도로 전체에서 발생한 균열을 검출하며, 선형균열일 경우 균열길이에 일정한 폭을 곱하여 면적을 산출하며, 균열면적의 전체 균열면적에 대한 비율을 계산

· 소성변형: 차량 통행으로 인한 도로의 변형깊이로써 좌우 휠패스 영역의 깊이 중 최대깊이를 계산

· 종단평탄성: 국제평탄성지수인 IRI를 산출하며, 차량이 80km로 주행시 현가장치를 통해 측정되는 상하 변위 프로파일. 일반적으로 25cm 간격의 종단 프로파일을 측정하고, 쿼터카 시뮬레이션(80km 주행시 현가장치를 고려한 1개 바퀴의 상하 이동 모사)으로 계산한 프로파일의 합을 프로파일 길이로 나누어 1km당 상하이동거리(m/km)로 계산

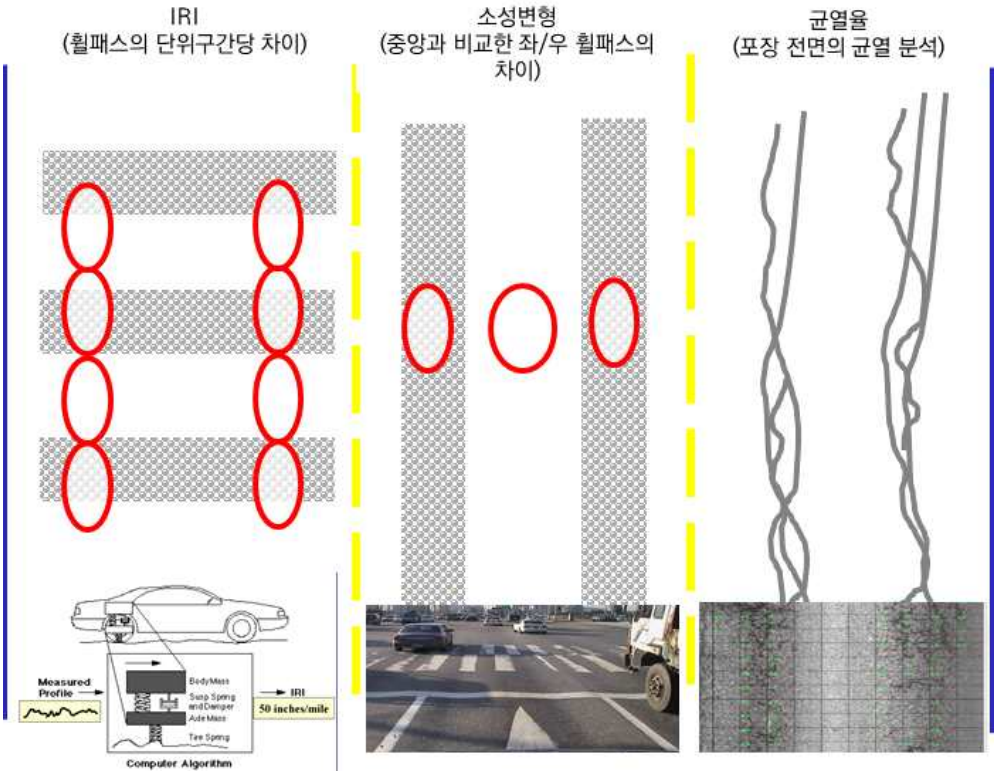
○ PCI 고도화를 위한 균열율, 소성변형, 종단평탄성 분석 방법 개발

- 1,2차년도는 드론 초분광 모델과 PCI를 비교 분석하였음
- 3차년도에는 드론 초분광 모델을 이용하여 PCI를 계산시 적용하는 균열율, 소성변형 깊이, 종단평탄성 등의 3가지 변수 대한 세부적인 평가 가능성을 평가
- 드론으로 측정한 초분광 데이터를 상기 도로 포장 조사차량의 측정방법을 모사하기 위해 아래와 같은 분석 방법을 개발

· 균열율: 도로 전체에서 발생한 균열로써 분석 영역의 전체 데이터를 비교하여 데이터 검출

· 소성변형은 좌·우 휠패스 영역과 변형이 거의 일어나지 않는 도로의 중심부의 데이터를 비교하여 분석

· 종단평탄성은 휠패스 영역을 25cm 단위구간으로 분할하여 단위구간 이전 영역의 데이터와 비교하여 분석



[도로포장 조사항목 및 특성]

○ 드론 초분광 데이터를 이용한 PCI 고도화 방안

- 드론으로 측정한 초분광 데이터를 이용한 분석을 자동화하기 위해 다음 방법으로 연구 추진
  · 정사영상으로 구현된 초분광 데이터 이미지에서 도로 영역 추출 기술 개발
  · 검출된 도로 영역에서 차선, 차량, 노면표지를 분류하여 분석 영역에서 제외
  · 맵 GPS 데이터를 이용하여 종방향으로 10m 단위로 분할
  · 10m 단위구간을 좌·우 휠패스와 중앙부로 구분하여 25cm 세부 단위구간으로 분할
  · 세부 단위구간의 초분광 데이터를 아래의 초분광 모델에 적용하여 계산

[초분광 분석 모델]

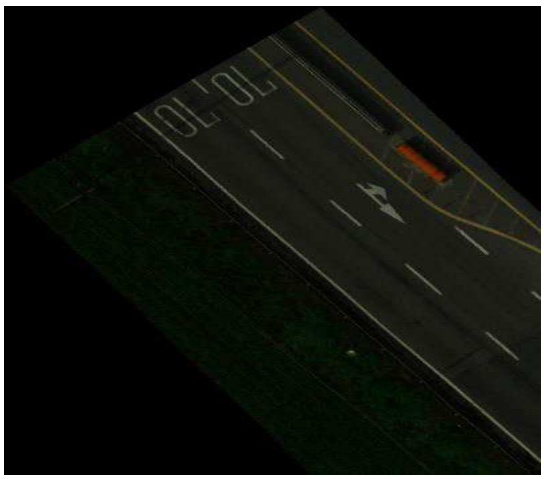
종류	모델식
VIS1(Visible Imaging Spectrometer 1)	$\mathrm{VISI1} = \left(\frac{\rho740 - \rho460}{\rho740 + \rho460}\right) \mathrm{VISR1} = \frac{\rho460}{\rho740}$
VIS2(Visible Imaging Spectrometer 2)	$\mathrm{VISI2} = \left(\frac{\rho830 - \rho490}{\rho830 + \rho490}\right) \mathrm{VISR2} = \frac{\rho490}{\rho830}$
VIS3(Visible Imaging Spectrometer 3)	$\mathrm{VISI3} = \left(\frac{\rho375 - \rho595}{\rho375 + \rho595}\right) \mathrm{VISR2} = \frac{\rho375}{\rho595}$
NDVI(Normalized Delta)	$\mathrm{NDVI} = \left(\frac{\rho802 - \rho638}{\rho802 + \rho638}\right)$
PVISI1(Pavement Visible Imaging Spectrometer Index 1)	$\mathrm{PVISI2} = \frac{\rho688}{\rho836}$
PVISI2(Pavement Visible Imaging Spectrometer Index 2)	$\mathrm{PVISI2} = \frac{\rho899}{\rho967}$

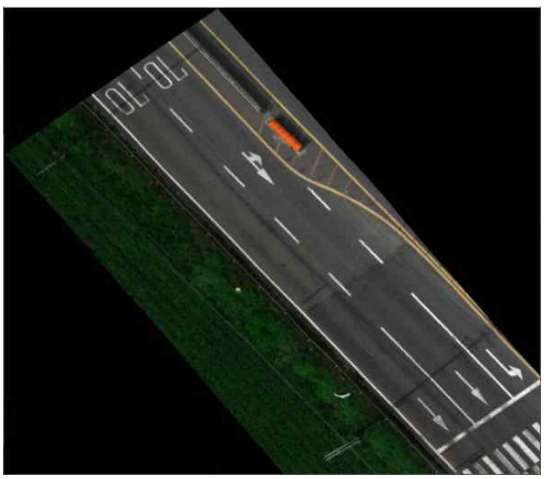
· 초분광 모델별 산출 데이터를 균열율, 소성변형, 종단평탄성 분석방법에 따라 계산하고 상관계수가 높은 최적 모델 결정
· 초분광 분석결과를 조사차량으로 측정한 값과 회귀분석하여 초분광 분석결과를 변수로 균열율, 소성변형, 종단평탄성 산출

(2) 초분광 분석 고도화를 위한 초분광 데이터 기반 도로 이미지 추출 기술 개발

○ 초분광데이터에서 RGB 이미지 데이터 추출
- Red(669.657nm), Green(544.146nm), Blue(485.872nm) 파장 데이터를 추출하여 PNG 데이터로 변환

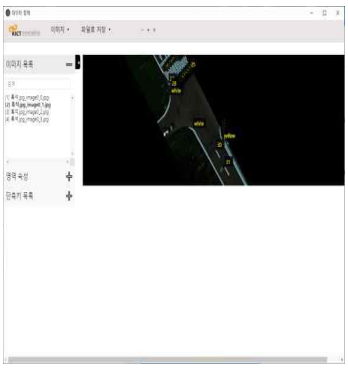
○ 인공지능 모델 학습을 위해 image normalization 방법으로 육안 구분 가능 이미지 생성
- 데이터 최댓값, 최솟값 증폭변환: csv 데이터의 RGB 파장이 내포하고 있는 데이터의 최소값을 0으로, 최댓값을 255로 지정하여 각 데이터 값을 증폭하는 방법
- 데이터 log 변환을 이용한 최댓값, 최솟값 증폭: 데이터의 편차가 매우 낮아 단순 데이터 증폭 시 벌어지는 데이터의 간격을 보정

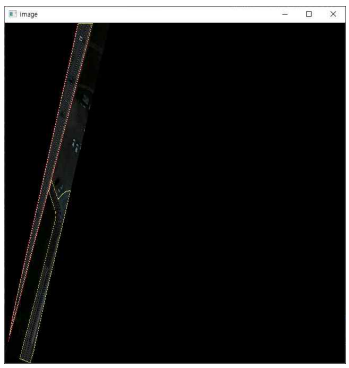

데이터 증폭 image normalization

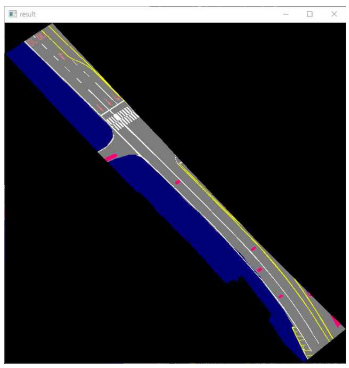

log 변환 image normalization

=> Log 변환 image normalization 적용

○ Deep Lab V3+를 이용한 도로 Segmentation
- 이미지의 검출 객체를 배경, 도로, 차선(흰색), 차선(노란색) 차(장애물), 노면표지로 분류
- 각 객체의 형상 및 색상에 따라 polygon 형식(여러개의 점을 가진 선을 활용하여 특정 영역을 검출 객체의 모양에 알맞게 세밀하게 라벨링 하는 가공 방식)의 라벨링 방법 적용


데이터 정제

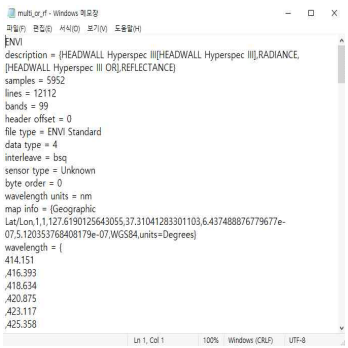

모델 학습과정


모델 학습결과

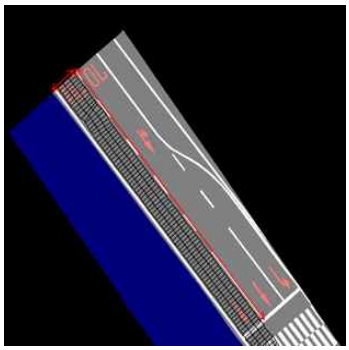
○ 목표 영역 추출 알고리즘 개발
- 분석 구간의 도로는 교차로, 차선병합, 신규차선 생성 등 다양한 변수로 인해 영역 전체를 자동화 분할 어려움

- 차선 기반한 도로 영역을 추출하되 교차로 등의 직선 차선이 이어지지 않을 경우 차선 내 포인트를 지정하고, 해당 포인트가 중심인 정사각형을 차선의 기울기 방향으로 중복하여 생성하며 차선여부를 인식하여 차선으로 인식될 경우 차선 영역을 연결하는 선택 지점 기반 차선 인식 알고리즘 개발
- 맵 GPS 데이터 기반하여 픽셀 이미지에서 실제 거리를 측정하는 알고리즘을 개발
- 도로 차선별로 종단방향으로 10m 연장으로 끊어서 영역 추출

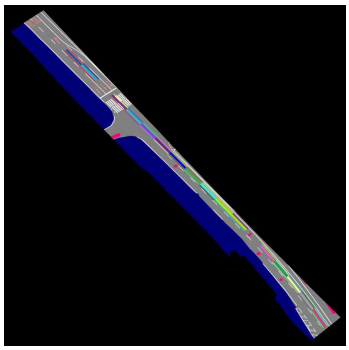
○ 맵 GPS 데이터를 이용한 10m 단위거리 분할 알고리즘 개발
- 초분광 데이터와 실제 위성거리 정보를 내포하고 있는 map info 데이터를 이용하여 이미지 데이터의 픽셀당 거리를 계산하여 실제 거리를 계산하는 알고리즘 개발
- 맵 GPS 데이터는 초분광 데이터의 최초 지점의 위성좌표를 내포하고 1픽셀당 가로, 세로 길이의 실제 거리를 위성좌표 기준으로 제공
- Map info data로 이미지 변환 시 시작 픽셀 위성좌표와, 1픽셀당 가로길이와 세로길이를 위성좌표를 통해 계산
- Map GPS 위성좌표 거리를 m 단위로 변환하여 차선의 실제거리를 dx, dy 거리를 계산하여 피타고라스 공식을 이용하여 분석 도로의 거리를 계산하는 알고리즘 사용
- 최종적으로 10m 단위로 분할된 도로의 이미지 제작



(a) map info data



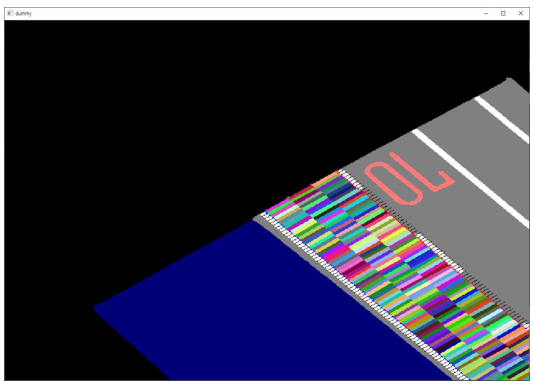
(b) 픽셀 거리 환산



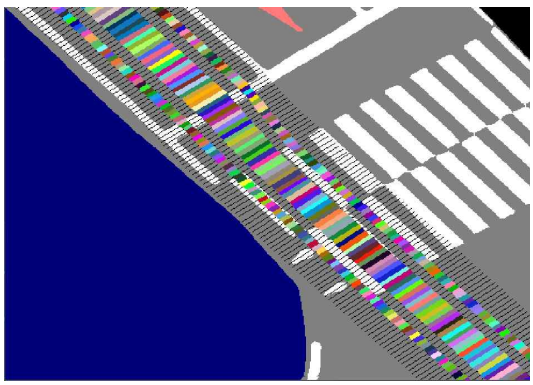
(c) 10m 단위 분할

[초분광 데이터 거리 환산 및 단위구간 분할]

○ 초분광 데이터 분석용 도로 내부 영역 분할 방법 개발
- 10m 단위로 검출한 도로의 영역을 다시 세밀 분할하는 알고리즘 개발
- 종방향 길이 25cm로 50개의 영역으로 분할, 횡방향은 도로 중심부 및 휠패스(도로의 ⅓,⅔지점을 중심으로 가로 40cm로 분할)구간을 분할하는 알고리즘 개발



(a) 10m 내부 영역 분할



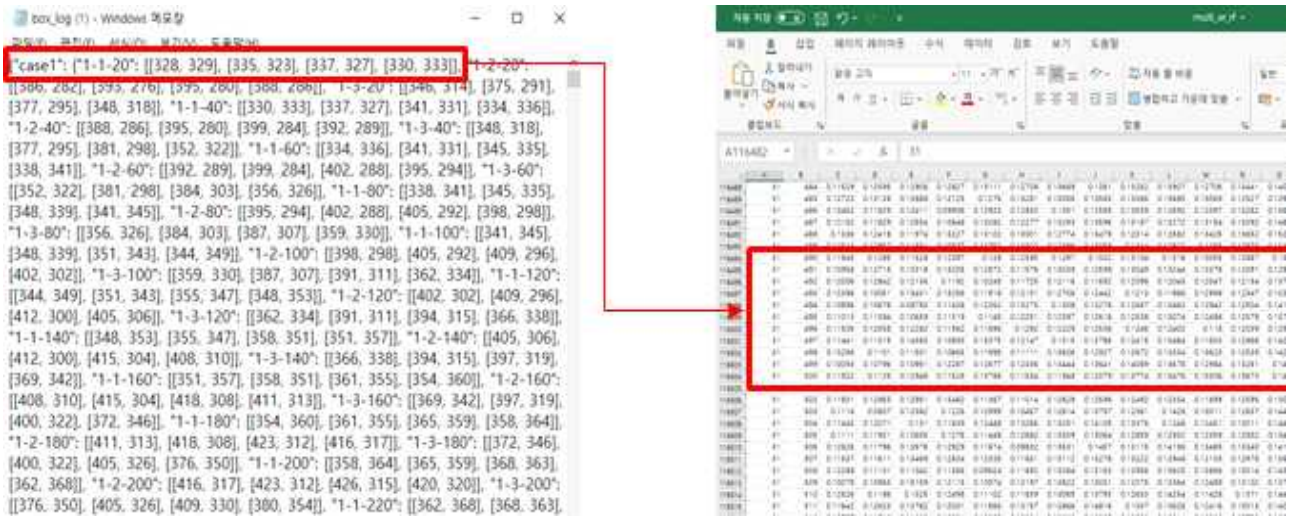
(b) 휠패스 고려 영역 분할

[단위구간의 내부 영역 분할]

(3) 초분광 데이터 기반 도로 상태 평가 알고리즘 개발

○ 검출 영역 좌표와 초분광 데이터 매칭 알고리즘 개발

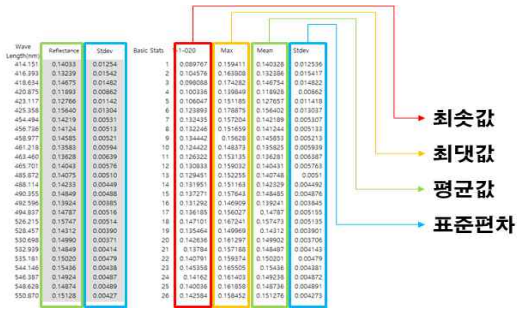
- 2차원 위치좌표로 구성되어 있는 이미지 데이터와 1차원 위치좌표로 구성되어있는 초분광 csv 데이터를 매치하는 알고리즘 개발
- csv 데이터의 위치좌표가 열을 기준으로 1차원으로 저장되어 있으므로 이미지 데이터를 기반으로 추출된 사각형 영역의 좌표에 해당하는 csv 데이터를 스캔하여 csv 데이터에서 불러올 수 있는 1차원 좌표로 변환
- 아래 그림은 분할된 영역에 할당되어 있는 point 좌표를 추출한 위치좌표 데이터와 2차원 영역좌표에 매치되는 1차원 csv 데이터 이미지



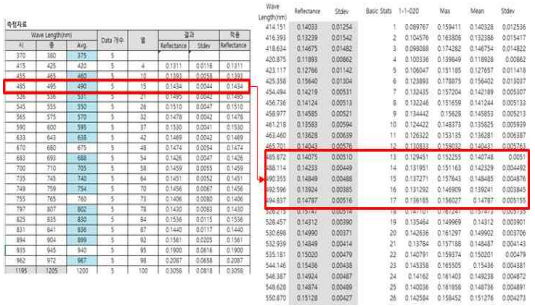
[영역 분할 좌표 추출 및 csv 데이터 매칭 알고리즘]

○ CSV 데이터 기반 수치해석 알고리즘 개발

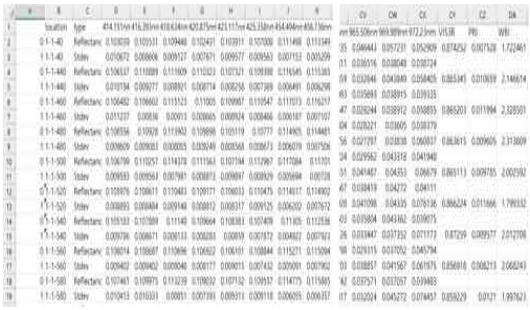
- CSV 데이터에서 추출된 영역의 초분광 데이터를 기반으로 IRI, rutting, crack을 분석하기 위한 수치해석 모델 적용
- 각 지수는 검출된 영역의 전체 csv 데이터를 분석에 필요한 22개의 파장영역으로 분할하는 1차 데이터 가공을 진행하고, 1차 가공 데이터를 기반으로 실제 지수 산출에 필요한 산출식을 적용하여 25㎝ 영역의 지수 값을 검출하는 2차 데이터 계산
- 최종적으로 10m 도로를 분석하기 위해 25㎝ 단위로 분석된 2차 가공 데이터를 기반으로 각 영역 데이터 비교 분석 및 최종 상관관계를 도출하는 3차 데이터 계산
- 데이터 해석 과정을 자동화 하는 알고리즘을 개발하여 최종적으로 도로 상태에 따른 결과값을 추출하는 알고리즘 개발
- 최종 산출 데이터의 품질 확인을 위해 각 데이터 가공 과정의 결과 csv 파일을 추출하여 초분광 데이터를 통한 실제 분석에 따른 결과 데이터와 차량을 통해 분석된 도로 분석 데이터의 비교분석
- 분석 가공 이전 각 파장별로 가용되는 특징 값을 분석한 데이터로써 데이터 분석 이미지를 기반으로 수치해석에서 사용되는 특정 파장군의 영역에 맞춰 1차 데이터 분석가공
- 실제 도로 상태지수를 판별하기 위해 사용되는 16가지 산출식으로 20㎝ 단위에서의 파장으로 최종 데이터 분석 결과 이미지로 10m 단위 도로의 상태지수를 비교분석



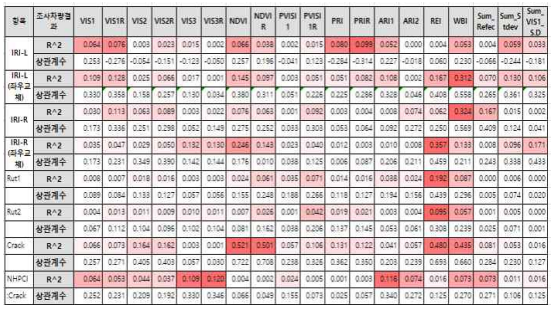
(a) csv 데이터 분석



(b) 1차 데이터 분석 가공



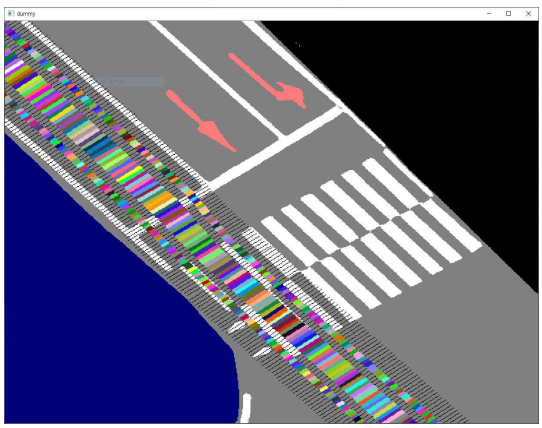
(c) 2차 데이터 분석 가공



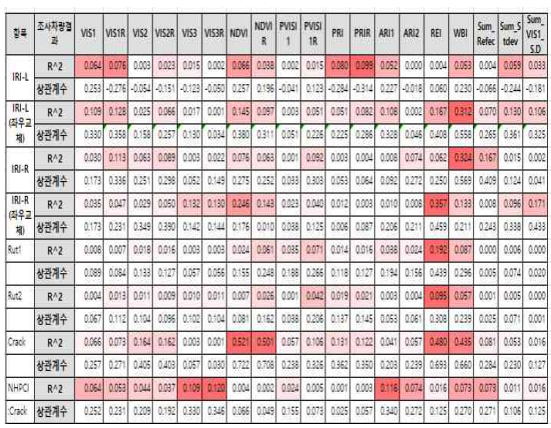
(d) 3차 데이터 분석 가공

[초분광 데이터 분석]

- 기술개발 결과 드론을 통해 수집된 초분광 데이터를 기반으로 도로의 상태지수인 IRI, rutting, crack를 추출하고 최종적으로 PCI지수를 추출
- DeepLab V3+ 인공지능 모델을 통해 추출된 도로 및 차선 객체는 기울기를 기반으로 차선을 인식하는 알고리즘을 통해 분석이 필요한 목표 차선 추출
- 인식 차선의 기울기 보정을 통해 교차로, 곡선구간 등 도로 내 변수에 대응
- 위성좌표와 픽셀거리를 이용하여 실제 도로거리를 검출하고 데이터 분석을 위하 10m 단위 분할 및 10m 내부 20cm 길이와 횔패스 구간을 정밀하게 분석하여 위치좌표 도출
- 도출된 위치좌표를 기반으로 자동화 수치해석 알고리즘을 개발하여 최종적으로 도로의 상태지수인 IRI, rutting, crack을 추출하고 10m 단위 도로의 PCI 지수를 추출하는 알고리즘 모델 개발



(a) 차선 인식 및 영역 분할 이미지



(b) 데이터 분석 결과 이미지

[초분광 데이터를 이용한 PCI 산출]

(4) 드론 초분광 데이터 기반 도로 상태 평가 결과

○ 초분광 분석모델을 이용한 포장 평가 결과

- 초분광 분석모델을 이용하여 IRI, 소성변형, 균열율 등을 계산하고 조사차량을 이용한 조사결과와 비교 평가한 결과 소성변형은 비교적 높은 상관성을 나타냈으나 IRI, 균열 모두 상관성이 낮았음

· IRI: 초분광 분석모델에서 VIS3가 가장 높은 상관성을 보이나 0.178로 낮음
· 소성변형: PVISI2가 0.704로 가장 높은 상관성을 보임
· 균열: NDVI가 가장 높은 상관성을 보이나 0.452로 낮음

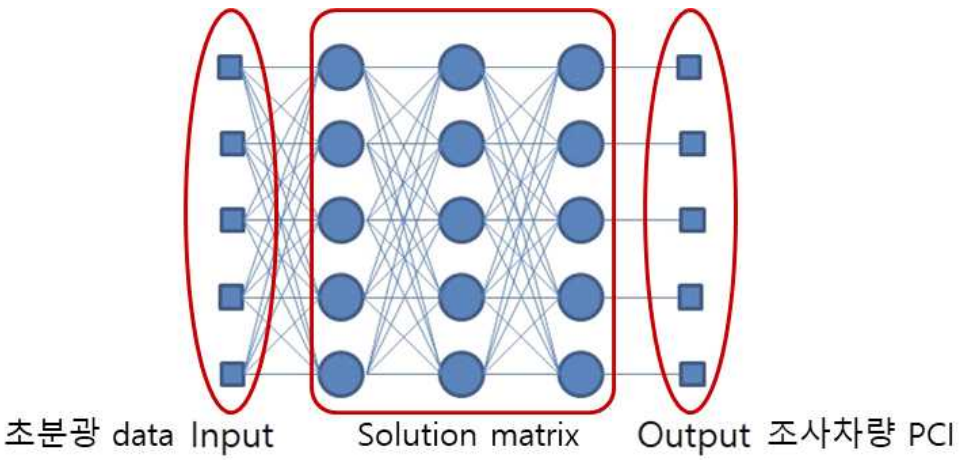
[초분광 분석모델을 이용한 분석결과의 상관계수]

항목	VIS1	VIS2	VIS3	NDVI	PVISI1	PVISI2
IRI	0.179	0.060	0.178	0.171	0.116	0.159
소성변형	0.262	0.417	0.472	0.393	0.487	**0.704**
균열율	0.228	0.235	0.326	0.452	0.414	0.279

○ 딥러닝을 이용한 포장 평가 기술 설계

- 도로상의 차량구간, 노면표시, 구조물 그림자 등 변수 데이터를 제외하는 초분광 분석모델은 도로 노면표시, 과속방지턱 등 변수 데이터를 포함하는 조사차량을 이용한 분석결과와 비교하였을 때 상관성이 낮음
- 상관성을 확보하기 위해 딥러닝 분석 방법을 통해 조사차량 PCI 데이터와 초분광 데이터를 매칭하여 산출수식을 추출하는 역추적 방식을 검토함
- 변수 데이터 구간을 포함하는 초분광 분석 모델 결과의 데이터 보완을 위해 변수 구간 데이터를 예측 할 수 있는 딥러닝 기술을 활용한 구조를 설계함
- 균열, iri, rutting, pci 지수 오차율 감소를 위한 다중 인공신경망인 DNN(Deep Neural Network)구조로 수식 산출 모델을 설계함

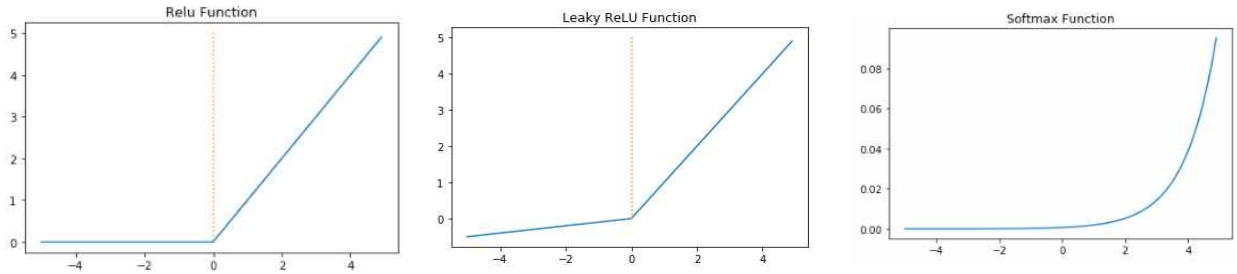
· 정확성 향상을 위한 7중 인공신경망 구조 설계
matrix [3000x4096][4096x1024][1024x512] [512x128][128x32][32x16][16x1]
· 비선형 활성화 함수(ReLU, Softmax 등)기반 출력값 왜곡 보정 기반 정확성 향상



[DNN(Deep Neural Network) Architecture]

[DNN 기반 초분광 데이터 산출 수식]

종류	산출 수식
Inference (추론과정)	$\hat{y} = w_6 \times ReLU(w_5 \times ReLU(w_4 \times (ReLU(w_3 \times ReLU(w_2 \times ReLU(w_1 \times x))))))$ $\because (x \in \mathbb{R}^{x}, \quad w_1 \in \mathbb{R}^{w_1 \times x}, \quad w_2 \in \mathbb{R}^{w_2 \times w_1},$ $w_3 \in \mathbb{R}^{w_3 \times w_2}, w_4 \in \mathbb{R}^{w_4 \times w_3}, \quad w_5 \in \mathbb{R}^{w_5 \times w_4}, \quad w_6 \in \mathbb{R}^{o \times w_5})$ $w = [w_1, w_2, w_3, w_4, w_5, w_6] \because weight$ $x = \in put\,dimension$ $\hat{y} = prediction\,(result)$ $ReLU = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$
Training (학습과정)	$minimize\,\mathcal{L} = \sum_{n} \|y_n - \hat{y}_n\|_2 \quad (\because y_n = \Pr(y\|x=n), \hat{y}_n = \Pr(\hat{y}\|x=n))$ $y = Real\ Value, \hat{y} = Predict\ Value$



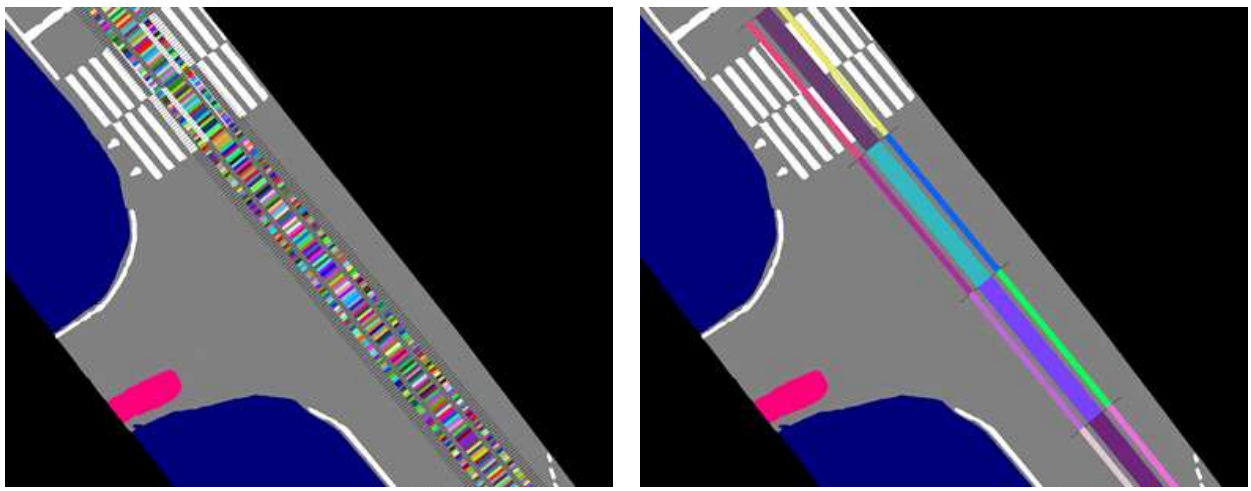
(a) 렐루 함수 (b) 리키 렐루 함수 (c) 소프트맥스 함수

[activation function]

○ 딥러닝을 이용한 포장 평가 기술 개발 및 학습

- 도로상의 차량, 노면표시 등 변수 데이터를 제외하는 전처리 알고리즘을 개발함
- 각 데이터의 gps를 기반으로 초분광 데이터와 조사차량의 포장분석 데이터를 맵핑함
- 예측값과 실제값의 오차 감소를 위해 역전파 학습 방법을 적용함
- 20cm 단위 구간의 초분광 데이터를 기반으로 10M 구간의 포장분석 예측 결과 값을 추출하는 Matrix 행렬 곱 인공신경망 구조를 개발함

· 10M 영역 내 초분광 데이터(3,000), 포장분석 예측 결과 데이터(1) 행렬 곱 연산 학습
· 흑석, 연천, 안성의 초분광 데이터, 포장분석 데이터 학습
· 초분광 데이터기반 포장 분석 규칙성 검출

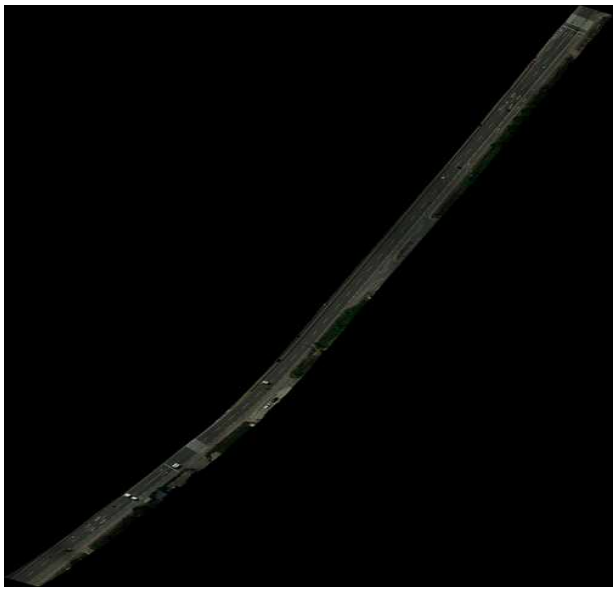


(a) 20cm 단위 영역 분할 (b) 10M 단위 영역 분할

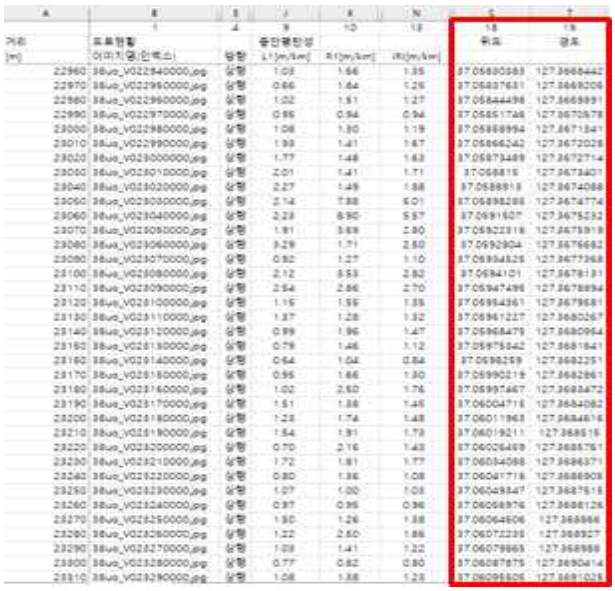
[도로포장 분석을 위한 영역 분할]

- 초분광 데이터 분석용 도로 내부 영역 분할 자료 이용한 딥러닝 학습
· 도로 데이터 셋에 대해 Back-propagation을 활용하여 Weight matrix(w) 최적화 학습
· 계산된 Loss를 Back-propagation을 통하여, 해당 Loss를 연산하기 위해 이용되었던 모든 matrix(w)의 weight를 update하는 방식으로 학습하였음

○ 딥러닝을 이용한 포장 평가 결과
- 안성 지역 550M 구간에 대한 포장 평가를 진행함
- 드론을 이용한 초분광 조사 결과를 이용하여 단위 영역을 분할하고, 10m 단위로 나누어진 포장 평가 결과 도출
- 드론을 이용한 초분광 조사 결과와 전용 차량을 이용한 차량조사 결과를 비교 분석함

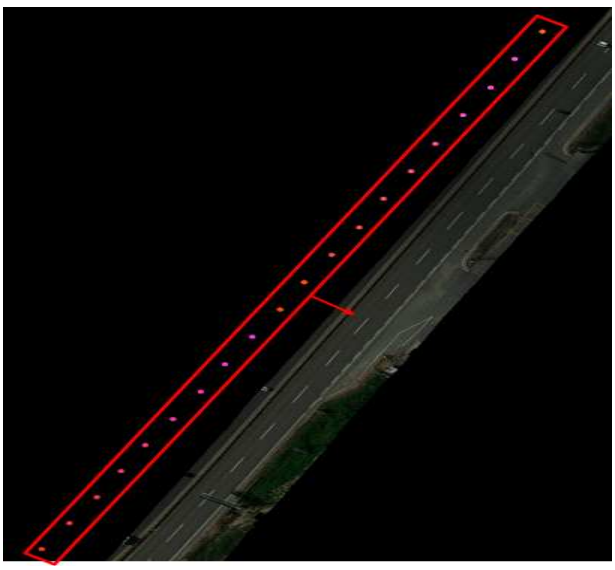


(a) 초분광 조사 시각화 이미지

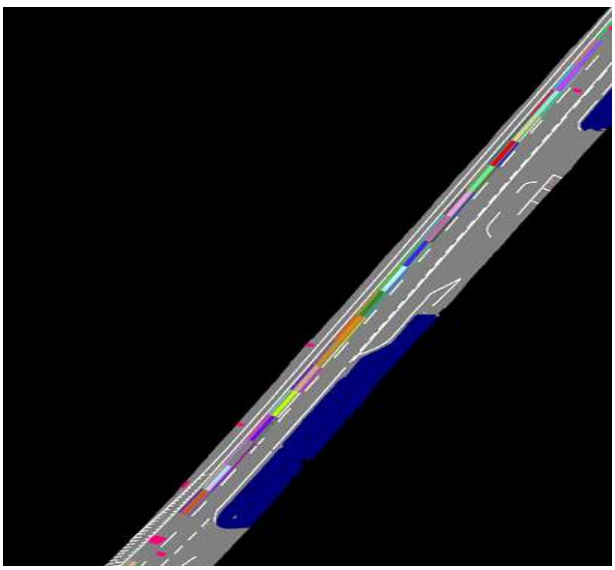


(b) 조사차량 결과 및 위성 좌표

[포장조사 구간(안성)의 초분광 이미지와 조사차량 결과]



(c) 조사 차량 위성 좌표 이미지 매핑



(d) 분석 구간 도로 포장 분석

[포장조사 구간(안성)의 초분광 포장 분석 과정 및 상관성 분석 과정]

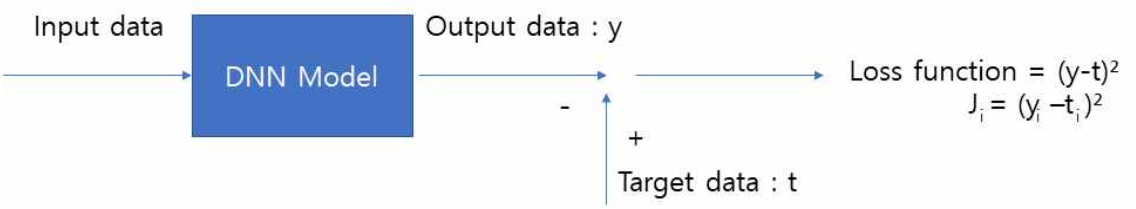
- 딥러닝 모델로 산출된 도로포장 평가 결과는 조사차량으로 평가된 포장평가지수(PCI)와 상관계수가 0.769이었으며, 개발 목표인 0.7 이상을 만족하였음

[조사차량 및 드론 초분광 아스팔트 포장 평가 결과]

No.	IRI (m/km)			rutting (mm)			균열 (%)			PCI		
	조사차량	초분광	편차	조사차량	초분광	편차	조사차량	초분광	편차	조사차량	초분광	편차
1	0.780	1.044	0.264	4.400	4.686	0.286	0.000	0.000	0.000	0.890	1.133	0.243
2	1.200	1.040	0.160	4.200	4.778	0.578	0.000	0.000	0.000	1.380	1.243	0.137
3	1.840	1.108	0.732	4.500	4.643	0.143	0.000	0.000	0.000	2.270	1.378	0.892
4	1.370	1.164	0.206	3.800	4.344	0.544	0.000	0.000	0.000	1.470	1.300	0.170
5	0.970	1.060	0.090	3.800	4.375	0.575	0.000	0.000	0.000	1.000	1.045	0.045
6	0.840	1.091	0.251	4.000	4.382	0.382	0.000	0.000	0.000	0.840	1.053	0.213
7	0.940	1.018	0.078	5.200	5.136	0.064	0.000	0.000	0.000	1.020	1.086	0.066
8	1.040	1.275	0.235	4.200	5.903	1.703	0.000	1.779	1.779	1.330	1.415	0.085
9	1.070	1.239	0.169	4.100	6.218	2.118	0.000	2.979	2.979	1.080	1.388	0.308
10	1.510	1.484	0.026	4.600	4.940	0.340	0.000	0.000	0.000	1.700	1.324	0.376
11	1.620	1.299	0.321	4.800	4.361	0.439	0.000	0.000	0.000	1.630	1.305	0.325
12	1.110	1.094	0.016	4.000	4.100	0.100	0.000	0.000	0.000	1.190	1.179	0.011
13	1.440	1.088	0.352	4.300	4.702	0.402	0.000	0.000	0.000	1.990	1.265	0.725
14	1.390	1.425	0.035	4.300	4.396	0.096	0.000	0.000	0.000	1.720	1.552	0.168
15	0.930	1.054	0.124	5.700	4.920	0.780	0.000	0.000	0.000	1.080	1.344	0.264
16	1.540	1.362	0.178	4.700	4.791	0.091	0.000	0.000	0.000	1.580	1.552	0.028
17	1.100	1.201	0.101	4.200	4.603	0.403	0.000	0.000	0.000	1.320	1.434	0.114
18	1.460	1.522	0.062	4.800	4.613	0.187	0.000	0.000	0.000	1.960	2.015	0.055
19	1.020	1.011	0.009	4.400	4.411	0.011	0.000	0.000	0.000	1.300	1.308	0.008
20	1.630	1.394	0.236	4.700	4.613	0.087	0.000	0.000	0.000	1.950	1.753	0.197
21	1.590	1.576	0.014	4.300	4.552	0.252	0.000	0.000	0.000	2.000	2.006	0.006
22	2.390	1.779	0.611	4.200	4.399	0.199	0.000	0.000	0.000	2.560	2.559	0.001
23	1.430	1.301	0.129	4.500	4.416	0.084	0.000	0.000	0.000	1.510	1.337	0.173
24	1.100	1.142	0.042	4.700	4.743	0.043	0.000	0.000	0.000	1.140	1.222	0.082
25	0.820	0.800	0.020	4.300	4.335	0.035	0.000	0.000	0.000	0.890	0.897	0.007
26	0.700	0.642	0.058	5.100	5.330	0.230	0.000	0.000	0.000	0.890	0.836	0.054
27	0.810	0.781	0.029	4.000	4.088	0.088	0.000	0.000	0.000	0.940	0.926	0.014
28	0.870	0.842	0.028	5.000	4.990	0.010	0.000	0.000	0.000	1.350	1.382	0.032
29	1.210	1.490	0.280	5.100	6.127	1.027	0.000	0.000	0.000	1.420	2.110	0.690
30	1.390	1.662	0.272	5.400	5.561	0.161	0.000	0.000	0.000	1.640	1.543	0.097
31	1.320	1.362	0.042	4.000	5.158	1.158	0.000	0.000	0.000	1.590	1.932	0.342
32	1.240	1.403	0.163	4.600	4.992	0.392	0.000	0.000	0.000	1.490	1.829	0.339
33	1.560	1.263	0.297	4.300	4.852	0.552	0.000	0.000	0.000	1.570	1.591	0.021
34	1.600	1.601	0.001	4.900	4.830	0.070	0.000	0.000	0.000	2.250	2.253	0.003
35	1.010	1.041	0.031	4.500	4.690	0.190	0.000	0.000	0.000	1.120	1.391	0.271
36	0.930	1.128	0.198	5.000	4.723	0.277	0.000	0.000	0.000	1.300	1.420	0.120
37	0.720	0.935	0.215	4.700	4.726	0.026	0.000	0.000	0.000	0.720	0.863	0.143
38	0.580	0.619	0.039	4.300	4.171	0.129	0.000	0.000	0.000	0.610	0.520	0.090
39	0.790	0.969	0.179	4.900	4.586	0.314	0.000	0.000	0.000	0.850	0.917	0.067
40	1.010	1.001	0.009	5.600	5.563	0.037	0.000	0.000	0.000	1.120	1.106	0.014
41	1.120	1.122	0.002	5.400	5.387	0.013	0.000	0.000	0.000	1.260	1.260	0.000
42	0.800	0.977	0.177	4.800	4.704	0.096	0.000	0.000	0.000	0.970	1.220	0.250
43	0.710	1.052	0.342	4.800	4.684	0.116	0.000	0.000	0.000	0.790	1.299	0.509
44	0.840	0.871	0.031	5.500	5.161	0.339	0.000	0.000	0.000	0.870	0.968	0.098
45	1.160	1.216	0.056	5.300	5.017	0.283	0.000	0.000	0.000	1.400	1.369	0.031
46	1.220	1.121	0.099	5.800	5.037	0.763	0.000	0.000	0.000	1.230	1.359	0.129
47	0.660	1.024	0.364	4.400	4.588	0.188	0.000	0.000	0.000	0.700	1.219	0.519
48	0.960	1.074	0.114	5.200	4.722	0.478	0.000	0.000	0.000	1.260	1.261	0.001
49	0.970	1.162	0.192	4.900	4.416	0.484	0.000	0.000	0.000	1.040	1.221	0.181
50	1.090	1.229	0.139	4.700	4.417	0.283	0.000	0.000	0.000	1.290	1.306	0.016
51	1.330	1.249	0.081	4.700	4.682	0.018	0.000	0.000	0.000	1.430	1.379	0.051
52	1.320	1.147	0.173	5.200	5.056	0.144	0.000	0.000	0.000	1.430	1.390	0.040
53	0.780	1.234	0.454	6.100	5.144	0.956	0.000	0.000	0.000	1.000	1.420	0.420
54	1.570	1.226	0.344	6.400	5.141	1.259	0.000	0.000	0.000	1.870	1.508	0.362
55	1.960	1.319	0.641	5.600	5.593	0.007	0.000	0.000	0.000	2.110	1.601	0.509
56	1.610	1.311	0.299	7.600	6.328	1.272	0.000	0.000	0.000	1.850	1.318	0.532
상관계수	0.760			0.568			-			0.769		

○ PCI 지수의 산출 특성을 고려한 딥러닝 기반 PCI 추정 알고리즘

· 딥러닝 모델의 출력값과 정답(차량 PCI 조사 값)의 오차를 판별하는 Loss Function을 추출



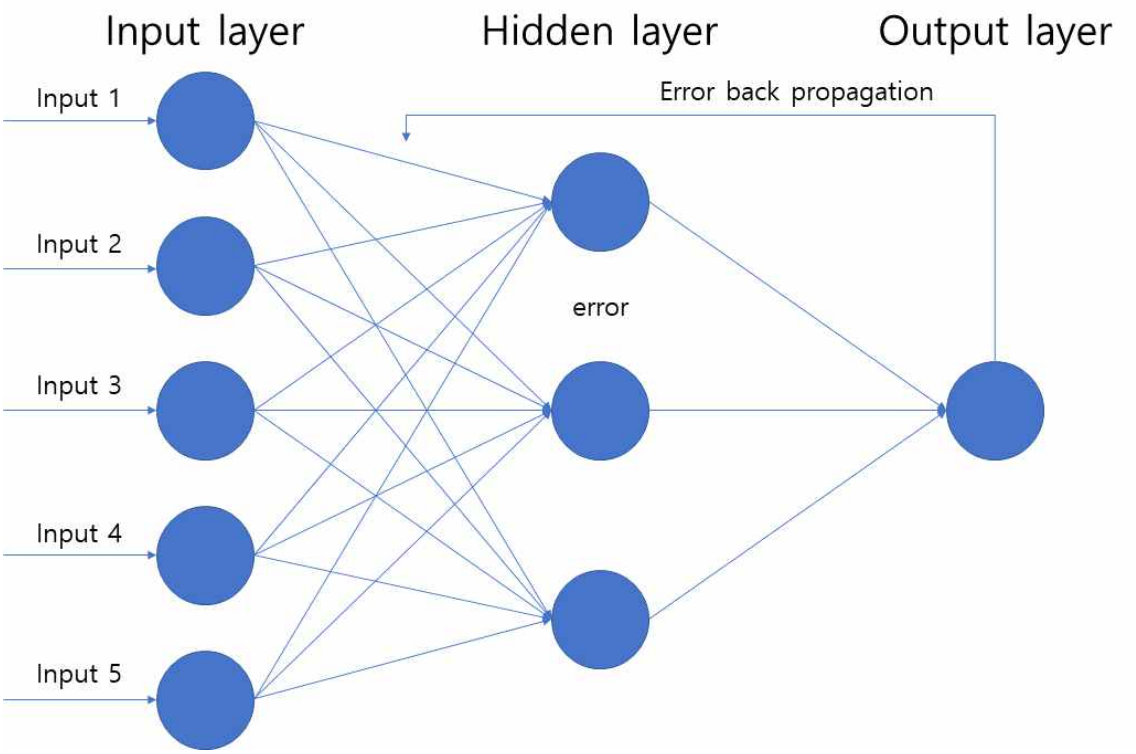
[Loss Function 추출 process]

· Optimizer(최적화)기법을 통한 하이퍼 파라미터 조절로 Loss(오차)의 편미분 값 추출

$$W_{k+1} = W_k - \alpha \cdot 2 \cdot (y_k - t) \cdot x_k$$

$$b_{k+1} = b_k - \alpha \cdot \frac{\delta J}{\delta b} = b_k - \alpha \cdot 2 \cdot (y_k - t) \cdot x_k$$

· Back-propagation(오차역전파)를 통해 인공지능 DNN 구조의 파라미터의 편 미분값을 추출하여 W(가중치)와 b(바이어스)에 대한 Loss의 편미분 값을 추출하고 파라미터를 업데이트하는 딥러닝 학습 진행



[Back-propagation Architecture]

· 도로 데이터 셋에 대해 통계적 기법 기반의 다층 퍼셉트론 학습방법을 활용하여 Weight matrix(w) 최적화 학습

· 계산된 Loss를 Back-propagation을 통하여, 해당 Loss를 연산하기 위해 이용되었던 모든 matrix(w)의 weight를 update하는 방식으로 학습

· PCI 조사 환경변수(차량, 노면표시 등)의 가중치를 낮게 조정하고, PCI 상관변수의 가중치를 높게 조정하여 PCI 검출 정확성 향상

○ 실용화를 위한 딥러닝 기법을 활용한 ‘드론 초분광 아스팔트 포장상태 조사 매뉴얼’ 개발
- 매뉴얼은 총칙, 사용 장비 및 프로그램 등의 기준, 촬영 계획 수립, 드론 운용, 드론 초분광 영상 분석 및 가시화 등으로 구성되어 있음
- 특히, 초분광 영상의 방사보정, 대기보정, 기하보정 등 전처리 과정 후 딥러닝을 이용한 포장 상태 분석을 위한 프로그램을 개발하였으며, 매뉴얼에 다음과 같이 세부적인 설명 포함

· 초분광 포장상태분석 프로그램(HSPCA 1.0)을 실행

· ‘초분광 원자료 열기’ 버튼을 클릭하여 초분광 영상(CSV) 파일과 데이터속성(hdr) 파일 업로드



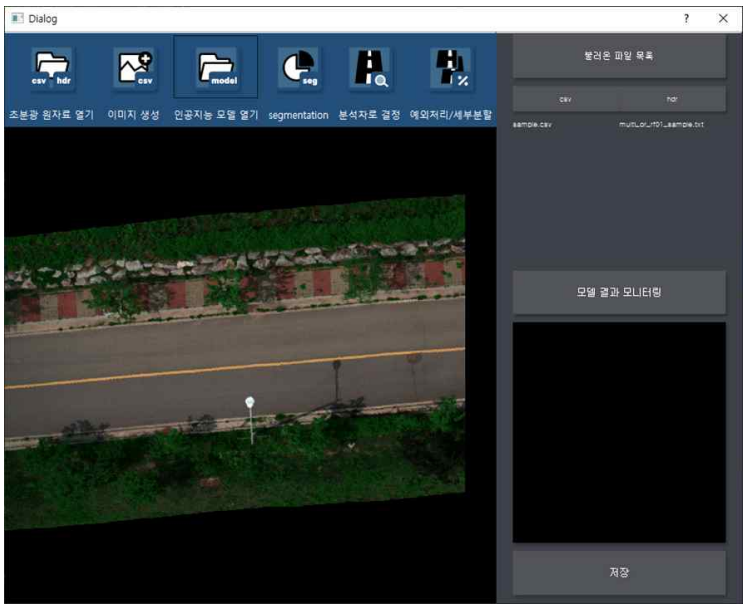
[HSPCA의 분석용 버튼]

· 창 오른 쪽의 불러온 파일 목록에 파일 정보가 표시되는지 확인



[HSPCA의 초분광 원자료 정보 표시]

· ‘이미지 생성’ 버튼을 클릭하여 시각화된 이미지를 생성

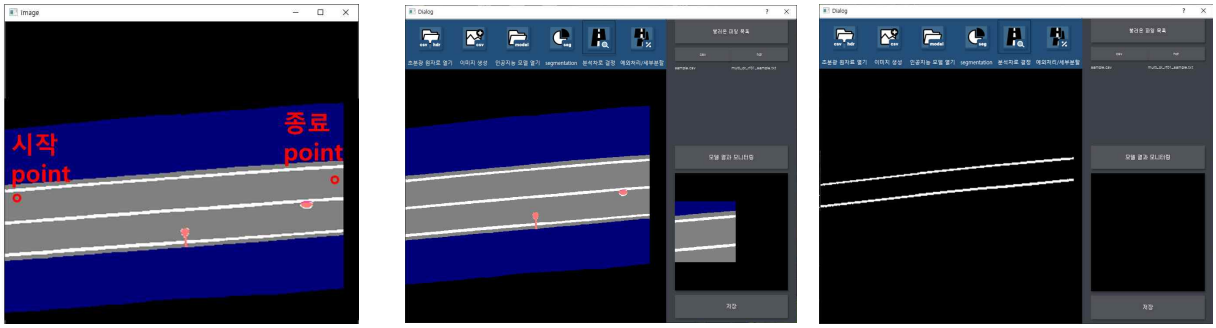


[HSPCA의 초분광 원자료를 이용한 시각화된 이미지 표시]

· ‘인공지능 모델 열기’ 버튼을 클릭하여 이미지분석 모델 파일을 업로드

· ‘segmentation’ 버튼을 클릭하여 인공지능 모델을 이용하여 차선을 분할 시작

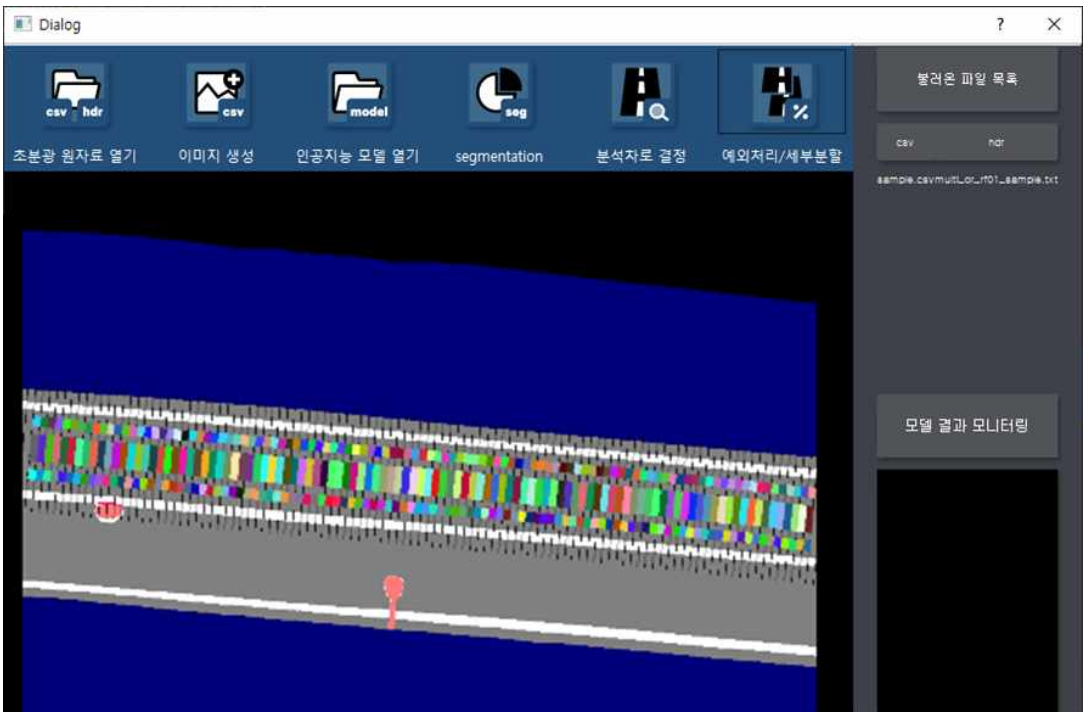
· ‘분석차로 결정’ 버튼을 클릭하여 분석된 이미지에서 분석할 차선의 두 지점을 선택하여 분석 차선을 검출



(a) 분석 차선 내부 2점 선택 (b) 차선 검출 인공지능 모델 (c) 검출된 차선 결과 이미지

[HSPCA의 인공지능 모델을 이용한 차선 검출]

· ‘예외처리/세부분할’ 버튼을 클릭하여 검출된 차선 영역에서 20cm 단위로 중앙부/휠패스부로 영역 분할 진행. 또한, 노면표시, 횡단보도, 차량 등 데이터 품질 저하 변수 요소 영역을 제외함



[HSPCA의 검출 차선의 분할]

· ‘저장’ 버튼을 클릭하여 분할된 영역으로 산출한 소성변형, 균열율, 평탄성 등을 데이터 파일(CSV) 과 데이터 속성 파일(json)으로 저장



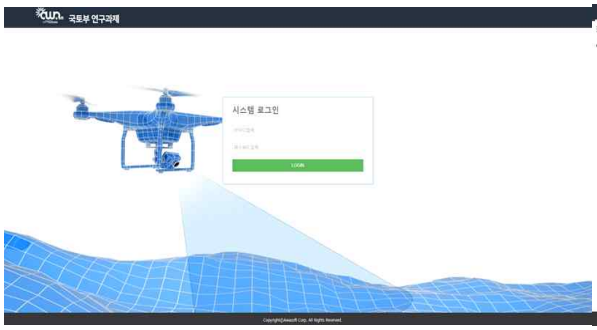
	A	B	C	D	
1		0	1	2	
2	prediction_crack	0.068574	0.446197	-0.00378	0.
3	prediction_iri	0.131422	0.224215	0.102562	0.
4	prediction_pci	0.487694	0.525665	0.686994	0.
5	prediction_rutting	1.117223	1.018642	0.512186	0.

{"meta": "ENVI₩ndescription = {HEADWALL Hyperspec III[HEADWA
= nm₩nmap info = {Geographic Lat/Lon,1,1,127.3667560289648,3
{₩n415.119₩n,417.361₩n,419.602₩n,421.844₩n,424.086₩n,426.32
96₩n,572.037₩n,574.279₩n,576.521₩n,592.212₩n,594.454₩n,596.
922₩n,740.164₩n,742.405₩n,744.647₩n,746.889₩n,749.131₩n,751
6,049₩n,939.674₩n,941.916₩n,946.399₩n,966.574₩n,971.058₩n,9
= 0₩n;Ortho DEM File = C:₩₩headwall1₩₩dem₩₩case12_dem.tif
(ms) = 0₩n;Ortho GPS Offset Yaw = 0₩n;Ortho Lems EFL (mm) =
Running Avg. of Raw = 0₩n;Ortho Running Avg. of Roll = 0₩n;Or
Number = nHS-145₩n;Units = mW/(cm2*sr*um)₩n;averagePixelDi
86], [10109, 359], [10036, 322]], "prediction_crack": 11.709249019
0.6011898070573807, "prediction_pci": 0.458267405629158, "pred

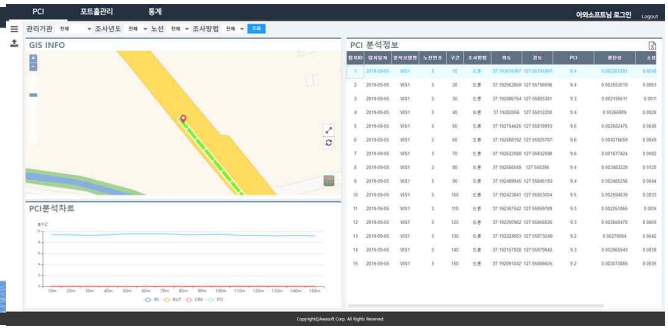
(a) json, csv 파일 (b) 소성변형, 균열율, 평탄성 등 데이터 파일 (c) 데이터 속성 파일

[HSPCA의 최종 결과물 데이터 파일 형식]

· 드론 포장상태 분석 결과 가시화



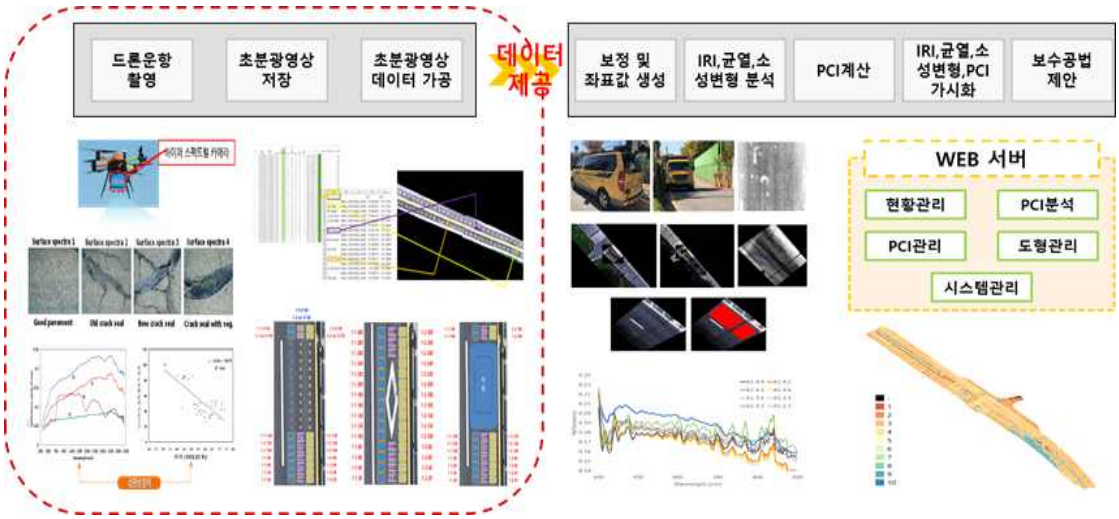
[가시화 시스템 로그인]



[PCI 데이터 입력/조회]

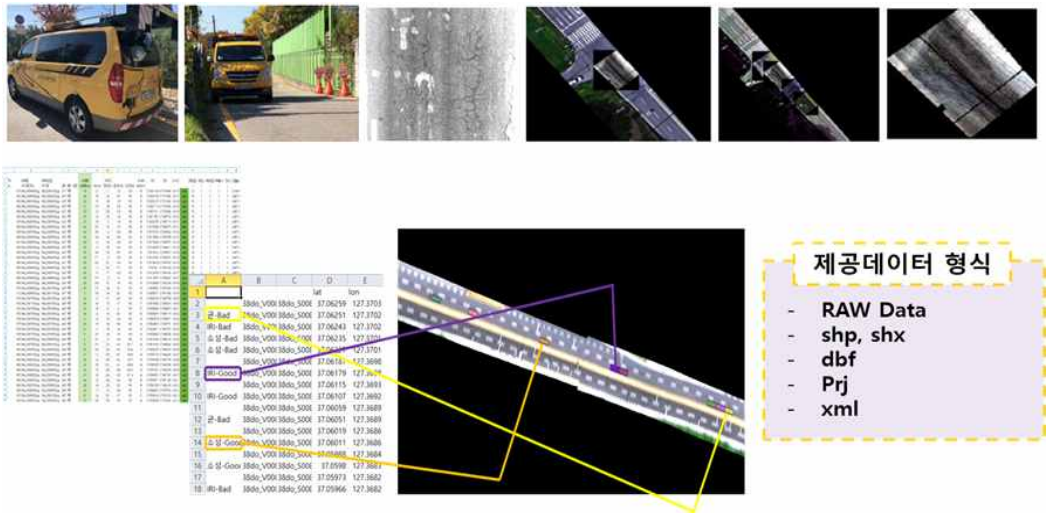
(5) PCI 측정을 위한 초분광 드론 운영체계

○ 도로포장상태 탐지를 위한 드론 초분광시스템 운영체계
- 드론 수집 초분광 데이터와 PCI 기준값 매칭 라이브러리 구축



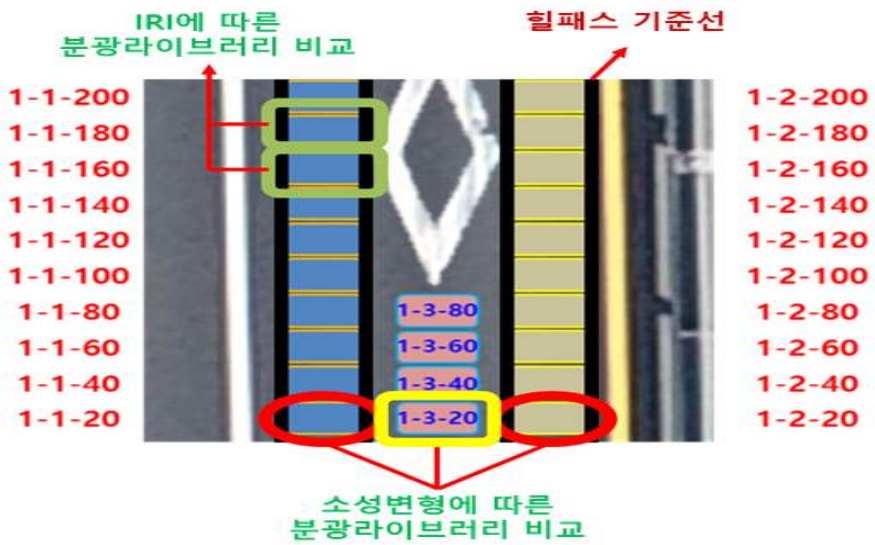
[PCI 기준값 매칭 라이브러리 구축]

- 촬영 대상지 드론 운용을 통한 초분광영상 데이터 수집 및 DB 구축
- Pes 기반 구축된 PCI(기준값)와 비교분석을 위한 초분광영상 가공(소성변형, IRI 등)
  · 기존 PCI는 10m 단위로 하나의 지수로 산출되어지고 있다. 이는 다시 말해 10m 안에 위치한 도로표면의 상태를 평균하여 하나의 수치로 나타낸다고 할 수 있다. 이는 도로 일부 파손 및 포토홀 등 극소적인 도로의 상태까지 판단하기는 한계가 있다고 판단된다. 따라서 본 연구에서는 세부적인 도로포장상태 판단을 위해 PCI를 결정하는 주요 3요소인 소성변형, IRI, 균열율을 고려하여 각 요인에 따라 생성되는 도로 분광라이브러리 특성을 파악하기 위해 기존 PCI를 기준으로 소성변형, IRI, 균열율 각각이 좋고 안좋은 도로를 선정함



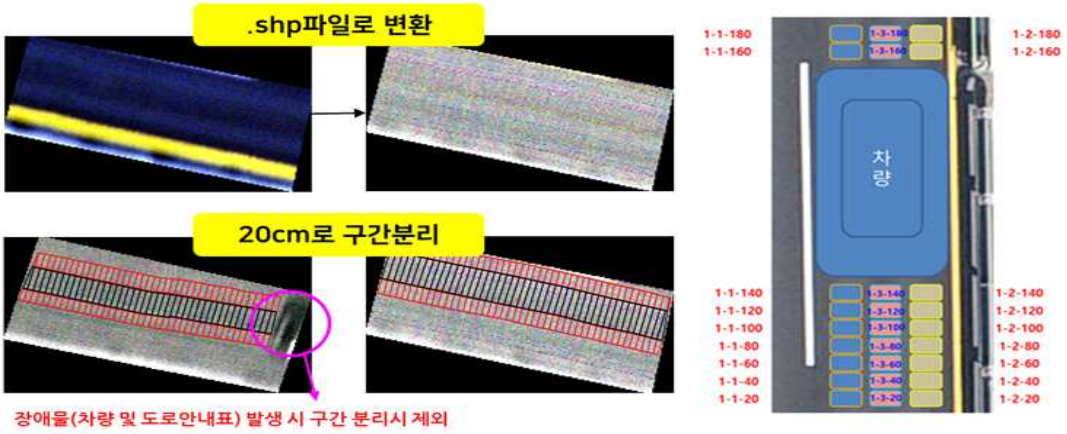
[드론 초분광 데이터 가공(10m)]

· (소성변형) 자동차 하중 및 힐패스에 의하여 도로의 횡방향으로 요철이 발생하는 현상을 말함, 소성변형에 따른 도로 반사도값을 파악하기 위해서는 차로 양쪽의 힐패스 구간의 영역과 자동차 바퀴가 지나가지 않는 도로 중앙부분의 도로 분광라이브러리를 추출하여 비교하였음
· (IRI) 차량의 진행방향으로 요철이 발생하는 현상으로 운전자의 승차감에 직접적인 영향을 주는 도로변형의 한 현상을 말함, 이러한 IRI에 따른 도로 반사도값 특성을 파악하기 위해서는 차량의 바퀴가 지나가는 힐패스 구간의 20cm 단위로 데이터를 추출하였다. 균열율의 경우 일정한 원인에 의해 발생하지 않기 때문에 PES 측정 데이터값 중 균열율이 좋은 도로와 안좋은 도로를 선정하여 데이터를 추출하였음



[소성변형, IRI를 고려한 데이터 가공]

·초분광영상 데이터 가공의 절차는 취득한 초분광영상자료(.hrd) 파일을 백터데이터(.shp) 파일로 변환하여 QGIS SW를 통해 도로를 20cm의 일정한 간격으로 대상지 내 구간분리를 수행하였음, 단 도로 위에 장애물(차량 및 도로안내표)이 존재할 경우 장애물을 제외한 순수 도로표면만 구간분리를 진행하였다. 이후 분리한 shp파일을 이용하여 envi 프로그램 내에서 각 셀의 분광영역대 ROI값을 min, max, mean 등 으로 추출하여 기존 PCI값과 비교 분석하는데 활용함



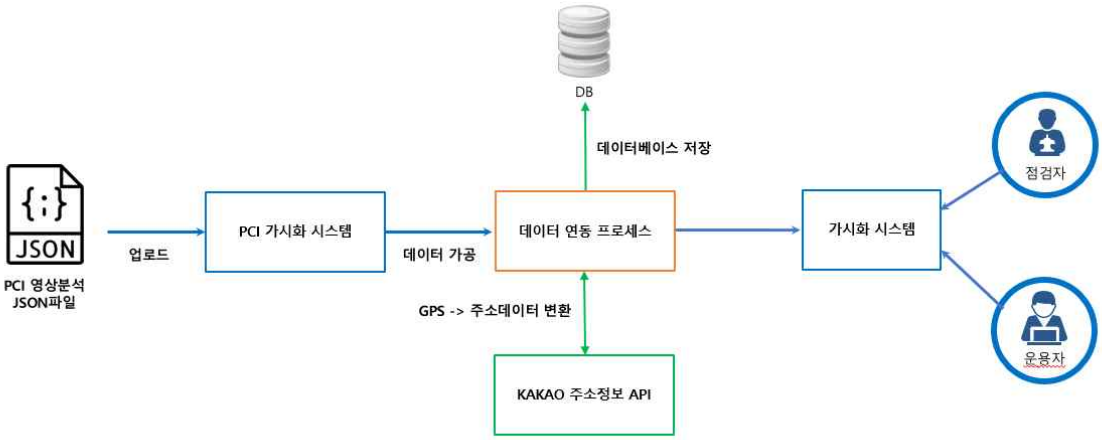
[PCI 산정 주요요소를 고려한 초분광영상 데이터 가공]

- 드론수집 초분광 데이터와 PCI 기준값 매칭 라이브러리 구축
  · 셀 단위로 가공된 영상에서 도로포장 분광값을 통해 분석한 결과, 도로포장상태가 좋은 포인트에서는 파장영역 800~900nm 사이에 높은 반사도를 나타내는 특성을 가지며, 도로포장상태가 안 좋은 지점에서는 같은 파장영역에서 상대적으로 낮은 반사도 값을 가지는 것을 알 수 있음

(6) PCI 가시화 기술 개발

○ PCI 가시화시스템

- 드론이 촬영한 초분광영상의 AI영상분석 영상분석 결과데이터를 가시화하여 운용할 수 있는 웹 서비스 시스템개발. 운용자는 PCI 영상분석 결과데이터를 실시간 업로드하여 Vworld기반 지도 레이어에 가시화 할 수 있으며, 작업자는 시스템을 통해 포장상태 분석결과를 모델별로 조회하여 포장상태 점검에 대해 조치여부를 판단함.



[PCI가시화 시스템 아키텍처]

○ PCI 분석데이터 모델 정의

- PCI 분석데이터를 시스템에 연동하기위한 데이터 모델 분석, PCI분석결과 JSON데이터에서 구간별, 모델별(VIS1, VIS1R, VIS2, VIS2R, VIS3, VIS3R,NDVI, NDVIR, PVISI1, PVISI1R, PRI, PRIR, ARI1, ARI2, REI, WBI) 시스템 데이터 구조설계 연구를 진행

PCI 분석정보 Metadata	PCI 분석 결과

<pre>"meta":⊡[
"ENVI",
"description = {HEADWALL Hyperspec III[HEADWALL Hyperspec III],RADIANCE,
[HEADWALL Hyperspec III OR],REFLECTANCE}",
"samples = 5180",
"lines = 6400",
"bands = 99",
"header offset = 0",
"file type = ENVI Standard",
"data type = 4",
"interleave = bsq",
"sensor type = Unknown",
"byte order = 0",
"wavelength units = nm",
"map info = {Geographic
Lat/Lon,1,1,127.5577671839702,37.19308702983902,3.270202789040048e-
07,2.605081648798801e-07,WGS84,units=Degrees}",
"wavelength ={",
"414.151",</pre>
	<pre>"1":⊡{
	"rect_points":"[[401, 100], [320, 166], [526, 421], [606, 355]]",
	"IRI1":⊡{
	"VIS1":0.002267281298832441,
	"VIS1R":0.004625035876392403 5,
	"VIS2":0.0030895579210189145,
	"VIS2R":0.005724722110018 1975,
	"VIS3":0.003096406873342639,
	"VIS3R":0.005527003947851238,
	"NDVI":0.001764212921886488,
	"NDVIR":0.003430269001364606,
	"PVISI1":0.0026873109881064426,
	"PVISI1R":0.0049615452269757 6,
	"PRI":0.0010355239673223561,
	"PRIR":0.002118509524036578,
	"ARI1":0.02866311314065268,
	"ARI2":0.0035644636331584384,
	"REI":0.0028760939071574266,
	"WBI":0.2617738190272639
	},
	"IRI2":⊞{…},
	"Rutting1_1-3":⊞{…},
	"Rutting1_2-3":⊞{…},
	"Rutting2_1-3":⊞{…},
	"Rutting2_2-3":⊞{…},
	"Crack":⊞{…}</pre>

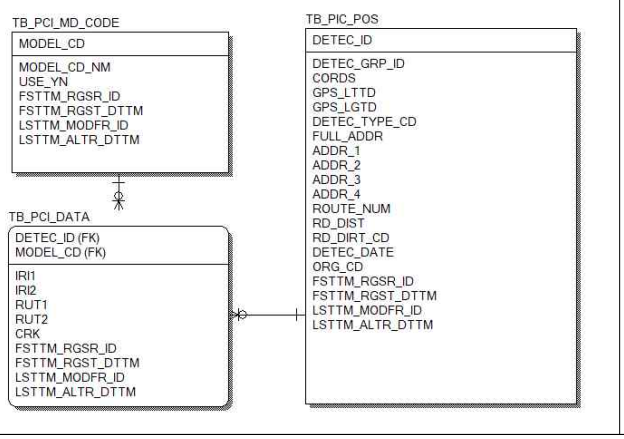
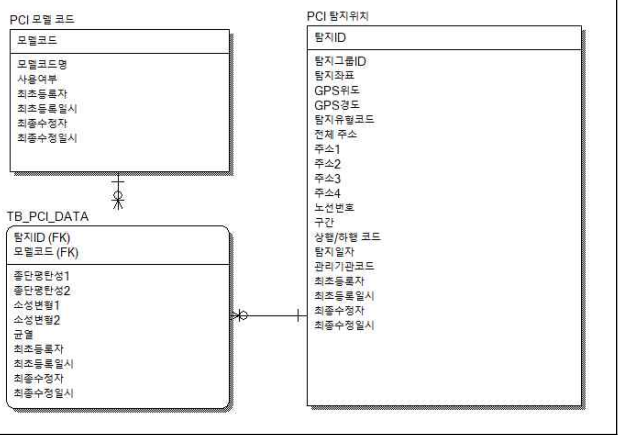
[PCI 연동 데이터 자료구조]

- Meta data의 map_info 정보와 PCI분석 결과의 reect_points를 다음 계산공식을 적용하여 구간 polygon 좌표 4건을 추출함.

경도	meta_base_longitude + (rect_pixel_x * meta_weight_longitude)
위도	meta_base_latitude + (rect_pixel_y * meta_weight_latitude)
구간 GPS	[127.55881900585233,37.19217297873709], [127.55879251720974,37.192156306214535], [127.55885922934664,37.19208935561616], [127.55888571798923,37.19210602813872]

○ PCI 분석데이터 기반 Database 설계

- PCI 분석시스템의 추가된 데이터 요소를 PCI 가시화 시스템에 연동시키기 위하여 Database 아키텍처를 추가 개선함.
- Database ER-Diagram은 Logical Data Model과 Physical DataModel을 구분하여 작성하였으며 각 Entity 항목별로 DB Table을 작성함.
- PCI가시화 시스템에 대한 Database ER-Diagram(Logical, Physical Data Model) 설계는 PCI 분석 구간정보를 기본정보로 하며, 탐지한 PCI구간의 Polygon좌표정보, 분석한 모델정보, 종단평탄성, 소성변형, 균열의 분석정보를 하위정보로 하여 각 Table에 대하여 Primary Key, Foreign key등 Relation구성을 보다 더 고도화 하여 설계하였음.

물리ERD	논리ERD
	

[PCI 가시화시스템 ER-Diagram]

TB_PCI_POS (PCI 구간 위치 정보)		
컬럼명	Type	Desc
DETEC_GRP_ID	NUMBER	탐지그룹ID
DETEC_ID	NUMBER	탐지ID
CORDS	CLOB	탐지좌표
GPS_LTTD	NUMBER(12,9)	GPS위도
GPS_LGTD	NUMBER(12,9)	GPS경도
DETEC_TYPE_CD	VARCHAR(3)	탐지유형코드
FULL_ADDR	VARCHAR(100)	전체주소
ADDR_1	VARCHAR2(20)	주소1
ADDR_2	VARCHAR2(20)	주소2
ADDR_3	VARCHAR2(20)	주소3
ADDR_4	VARCHAR2(20)	주소4
ROUTE_NUM	NUMBER(3,0)	노선번호
RD_DIST	NUMBER	구간(10m,20m...)
RD_DIRT_CD	VARCHAR2(3)	상/하행코드
DETEC_DATE	DATE	탐지일자
ORG_CD	VARCHAR(3)	관리기관코드

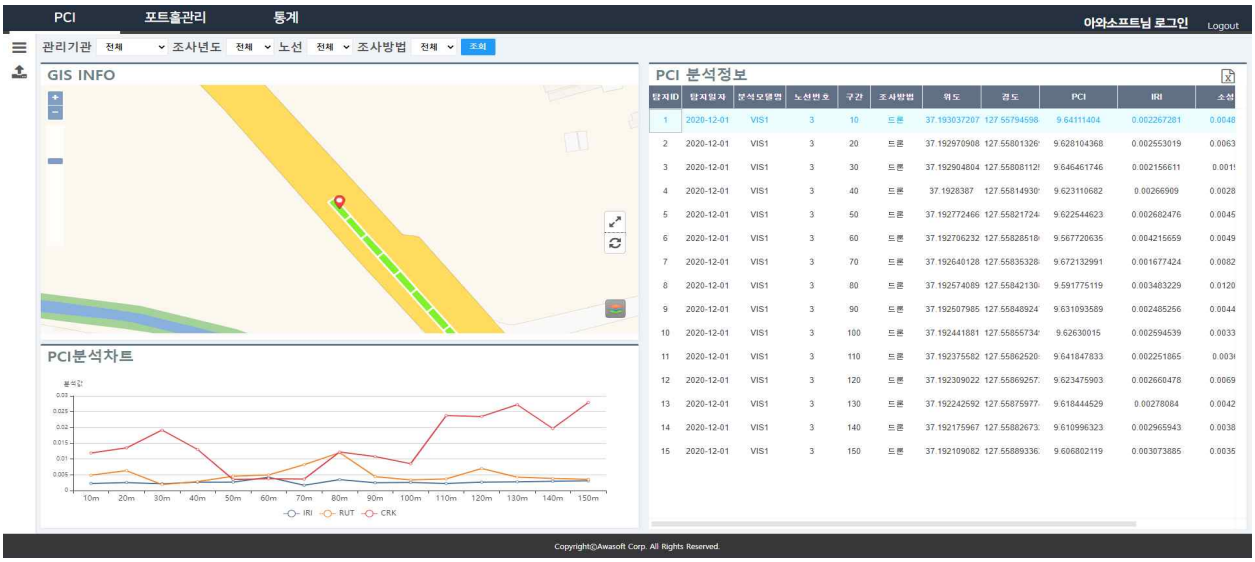
[PCI 구간 위치 정보 테이블]

TB_PCI_DATA (PCI 분석데이터)		
컬럼명	Type	Desc
DETEC_ID	NUMBER	탐지ID
MODEL_CD	VARCHAR(2)	모델코드
IRI	NUMBER	종단평탄성 분석값
RUT	NUMBER	소성변형 분석값
CRK	NUMBER	균열 분석값

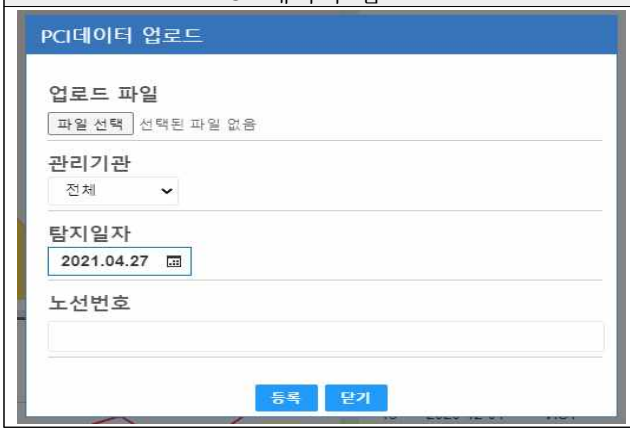
[PCI 구간 위치 정보 테이블]

○ PCI 가시화시스템 UI고도화

- 사용자 관점에서의 Interaction Design을 고려하여 레이아웃을 배치하였으며 ES6환경의 Frontend Architecture를 적용함.
- 지도 API의 Polygon 좌표형태를 표한하기 위한 Openlayer Feature Rendering 기술 연구 및 적용
- 대량의 List데이터의 브라우저 랜더링 최적화를 위하여 Toast UI Grid Library의 Wrapping module을 개발함. 대용량 Grid Data에 대한 Virtual Scroller기술을 적용함.
- PCI 분석구간의 GPS 고도, 렌즈등에 의한 오차가 있을 경우 데이터의 변동없이 지도API의 구간보정을 할 수 있는 기능을 개발함. 보정거리 오차수치와 구간 Polygon Rendering에 사용된 GPS 좌표정보를 재계산하여 Database에 반영함.



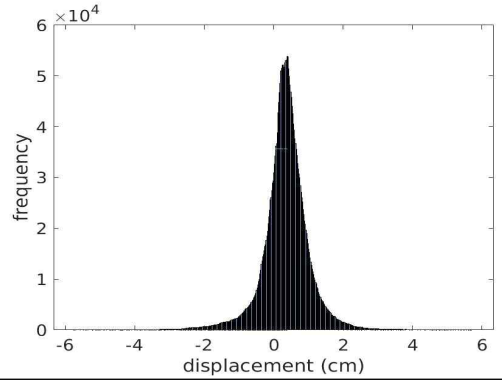
[PCI 메인UI]

PCI 데이터 업로드	PCI 데이터 조회 설정 UI
	

- PCI분석결과 데이터 모델정보를 연동하기위한 데이터 연동 프로세스를 개발함. 데이터 업로드 UI를 통해 PCI 분석데이터를 업로드하여 시스템 등록이 자동적으로 이루어짐.
- PCI 적용공식, 분석결과 임계기준, 적용 모델등의 분석결과에 대한 설정기능을 제공하는 UI개발
- 분석 결과 설정값의 수정에 대응하는 PCI분석결과값이 조회됨

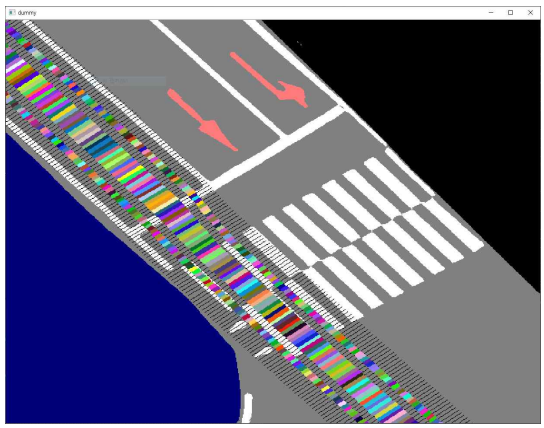
<3차년도 연구성과 요약>

□ 드론 SAR 데이터 기반 비탈면 변위 측정 기술 개발
- ㅇ 시계열 L-band RailSAR 자료 수집
- ㅇ 정밀위치자료와 고해상도 DEM을 이용한 영상레이더 정밀 보정 정사영상 자료 생성을 위한 코너 리플렉터 제작
- ㅇ Sentinel-1 위성 영상을 이용한 코너 리플렉터 실험
- ㅇ 테스트베드 운영을 통한 드론 측정 데이터와 지상 라이더 데이터 비교 분석

테스테베드(남양주 조안면)	관측 변위량 오차 히스토그램
	

□ 드론을 활용한 포트홀 탐지 기술 개발
- ㅇ 항공 영상 내 포트홀 자동탐지 결과 관리시스템 구현
- ㅇ 객체 분할 딥러닝 기반 항공 영상 이미지 내 포트홀 탐지 알고리즘 구현
- ㅇ 도로 내 포트홀 탐지를 위한 항공 영상 내 도로 영역 분할 알고리즘 구현
- ㅇ 포트홀의 위험도 측정을 위한 포트홀 탐지 영역의 실면적 측정 알고리즘 구현
- ㅇ 항공 영상 및 도로 영상 기반 딥러닝용 데이터 셋(약 730,134장) 구축
- ㅇ 딥러닝 기반 항공 영상 이미지 내 포트홀 탐지 결과 분석
  - 포트홀 검지율 : 97.2%, 포트홀 오검지율 : 9.8%

□ 포장 노후화 상태(PCI) 측정을 위한 초분광 분석 모델 고도화 방안 개발
- ㅇ PCI 고도화를 위한 균열율, 소성변형, 종단평탄성 분석 방법 개발
- ㅇ 초분광 분석 고도화를 위한 초분광 데이터 기반 도로 이미지 추출 기술 개발
- ㅇ 초분광 데이터 기반 도로 상태 평가 알고리즘 개발
- ㅇ CSV 데이터 기반 수치해석 알고리즘 개발



(a) 차선 인식 및 영역 분할 이미지

항목	조사자평결과	VIS1	VIS1R	VIS2	VIS2R	VIS3	VIS3R	NDVI	NDVIR	PVISI1	PVISI1R	PRI	PRIR	ARI1	ARI2	REI	WBI	Sum_Refec	Sum_Stdev	Sum_VIS1_SD
IRI-L	R^2	0.064	0.076	0.003	0.023	0.015	0.002	0.066	0.038	0.002	0.015	0.080	0.099	0.052	0.000	0.004	0.053	0.004	0.059	0.033
	상관계수	0.253	-0.276	-0.054	-0.151	-0.123	-0.050	0.257	0.196	-0.041	0.123	-0.284	-0.314	0.227	-0.018	0.060	0.230	-0.066	-0.244	-0.181
IRI-L (좌우고제)	R^2	0.109	0.128	0.025	0.066	0.017	0.001	0.145	0.097	0.009	0.051	0.051	0.082	0.108	0.002	0.167	0.312	0.070	0.130	0.106
	상관계수	0.330	0.358	0.158	0.257	0.130	0.034	0.380	0.311	0.051	0.226	0.225	0.286	0.328	0.046	0.408	0.558	0.265	0.361	0.325
IRI-R	R^2	0.030	0.113	0.063	0.089	0.003	0.022	0.076	0.063	0.001	0.092	0.003	0.004	0.008	0.074	0.062	0.324	0.167	0.015	0.002
	상관계수	0.173	0.336	0.251	0.298	0.052	0.149	0.275	0.252	0.039	0.303	0.053	0.064	0.092	0.272	0.250	0.569	0.409	0.124	0.041
IRI-R (좌우고제)	R^2	0.035	0.047	0.029	0.050	0.132	0.130	0.246	0.143	0.023	0.040	0.012	0.003	0.010	0.008	0.357	0.133	0.008	0.096	0.171
	상관계수	0.173	0.231	0.349	0.390	0.142	0.144	0.176	0.010	0.038	0.125	0.006	0.087	0.206	0.211	0.459	0.211	0.243	0.338	0.483
Rut1	R^2	0.008	0.007	0.018	0.016	0.003	0.003	0.024	0.061	0.035	0.071	0.014	0.016	0.038	0.024	0.192	0.087	0.000	0.006	0.000
	상관계수	0.089	0.084	0.133	0.127	0.057	0.056	0.155	0.248	0.188	0.266	0.118	0.127	0.194	0.156	0.439	0.296	0.009	0.074	0.020
Rut2	R^2	0.004	0.013	0.011	0.009	0.010	0.011	0.007	0.026	0.001	0.042	0.019	0.021	0.003	0.004	0.095	0.057	0.001	0.005	0.000
	상관계수	0.067	0.112	0.104	0.096	0.102	0.104	0.081	0.162	0.038	0.206	0.137	0.145	0.053	0.061	0.308	0.239	0.025	0.071	0.001
Crack	R^2	0.066	0.073	0.164	0.162	0.003	0.001	0.521	0.501	0.057	0.106	0.131	0.122	0.041	0.057	0.480	0.435	0.081	0.053	0.016
	상관계수	0.257	0.271	0.405	0.403	0.057	0.030	0.722	0.708	0.238	0.326	0.362	0.350	0.203	0.239	0.693	0.660	0.284	0.230	0.127
NHPCI	R^2	0.064	0.053	0.044	0.097	0.109	0.120	0.004	0.002	0.024	0.005	0.001	0.003	0.116	0.074	0.016	0.073	0.073	0.011	0.016
Crack	상관계수	0.252	0.231	0.209	0.192	0.330	0.346	0.066	0.049	0.155	0.073	0.025	0.057	0.340	0.272	0.125	0.270	0.271	0.106	0.125

(b) 데이터 분석 결과 이미지

[초분광 데이터를 이용한 PCI 산출]

3. 연구개발과제의 수행 결과 및 목표 달성 정도

1) 연구수행 결과

(1) 정량적 연구개발성과

가. 비탈면 변위측정 기술

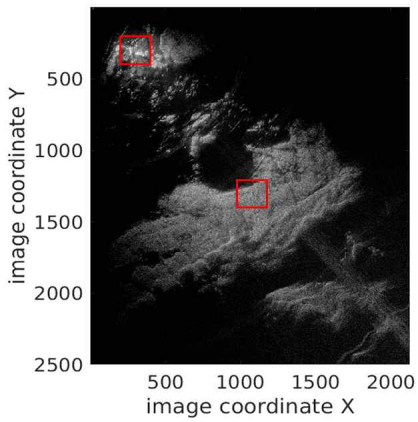
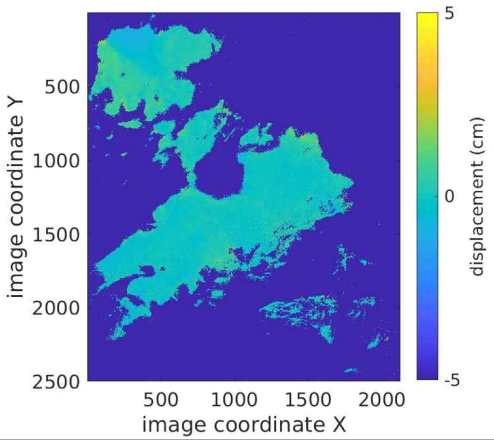
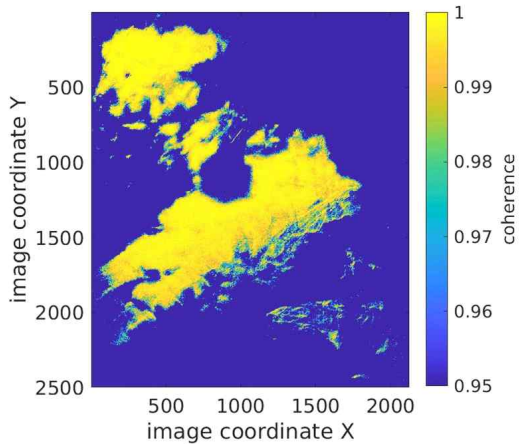
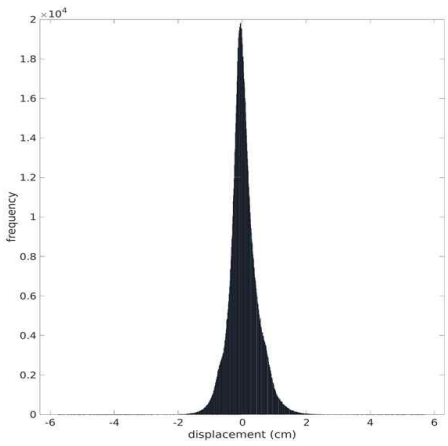
○ 평가환경
- 비탈면 변위 측정 시스템
  · 드론 촬영 허가 필요 및 심한 강우와 강설 환경은 피하는 것이 좋음
  · 드론과 영상레이더 관측기기는 지상에서 통신이 가능하여야 함
  · 해당 실험은 스위스 베른 인근 부지에서 진행
- 영상 촬영 환경
  · 헬기형 드론 등 안정적인 궤도와 비행을 할 수 있는 비행탑재체를 운용하여야 함
  · 고도제한은 없으나 해당 지역에 대해 허가가능한 고도로 비행 필요 (100~500m)
  · 기상조건, 주야 조건과 관계없음
  · 스위스 실험 평가환경은 일반적인 도로 인근 비탈 사면 환경을 가지고 있으며, 계절적 요인으로 1차 실험 평가시기(2019. 02. 13)에는 눈으로 피복되어 있었음
  · 눈이나 얼음의 경우 마이크로파 영역에서의 유전율에 영향을 줄 수 있고, 투과심도에 영향을 주어 간섭기법 적용에 다소 제한적일 수 있는 환경이 될 수 있음
  · 영상레이더 변위측정 기법에서 제한 요소가 될 수 있는 식생의 경우 비교적 잎의 밀도와 수고가 높지 않은 식생이 피복되어 있는 환경에서 평가 진행

○ 전제조건
- 비탈면이 영상레이더의 방향에 놓이도록 관측 필요
- 반복 비행시 동일한 궤도를 유지할 수 있도록 비행 필요
- 영상레이더 관측시 동일한 관측 변수 설정 필요
- 정밀 궤도 측정이 필요한 DGPS 운용이 필수적이므로 GPS 위성망을 활용할 수 있는 환경에서 평가가 필요함
- 영상레이더 기반 위상간섭기법의 경우 지표면의 급작스러운 변화는 지표변위 측정의 정밀도를 판단할 수 있는 기준인 긴밀도 유지를 불가능하게 하므로 굴착, 파괴 등으로 인한 표면상태가 변하지 않아야 함
- 마이크로파 전자기파는 철골, 펜스 등의 전자기파의 산란을 불규칙하게 할 수 있는 구조물에 쉽게 영향을 받으므로 이들의 구조를 고려한 편광 선택을 하여야 함
- 해안가, 전파 기지국 등 마이크로파의 전파 방해 또는 jamming 현상을 야기할 수 있는 곳에서의 평가환경을 고려해야 함
- 드론 비행 고도 제한 지역 여부를 판단하여야 하며, 비행 고도에 따른 마이크로파 입사각이 달라질 수 있으므로 평가 진행 전 비행 궤도 모사 실험이 필수적임

○ 평가방법
- 드론 탑재 영상레이더로부터 관측 다운링크된 데이터를 노트북 또는 외장하드에 저장하여 자료처리
- 정밀궤도 분석을 위한 INS 자료 처리 수행
- 영상레이더 원시 자료 처리 기법을 적용하여 지형고도모형 기반 single look complex 자료로의 자료처리과정 수행
- 반복 관측된 영상을 위상간섭기법 적용을 통해 비탈면 변위 탐지 수행
- 다음과 같은 평가 방법과 기준을 통해 비탈면 변위 측정 기술 탐지

- 2019. 02. 13, 2019. 03. 28 2회에 걸쳐 동일한 지역에 대한 비탈면 변위 측정 기술을 수행하였음
- 위상간섭기법의 기술 적용 특성상 비탈면 변위를 인공적, 임의적으로 생성하여 실험할 수 없으므로 변위가 없는 지역을 동일한 방법으로 측정하여 비탈면 측정 기술을 적용한 후 변위가 생성되는지 여부를 판단하였음
- 동일한 날, 유사한 시간대에 측정된 드론 영상레이더를 기반으로 비탈면 측정 기술을 적용하여 발생된 변위량은 개발된 시스템의 비행 궤도, 전자기파 산란, 관측시간 차이에 따른 비상관성 등 정확도 오차로 해석할 수 있음
- 위치정확도 평가는 기 구축된 해당 지역의 지형표고모형 자료와의 공간 상관관계 분포를 계산으로 평가를 실시하였음

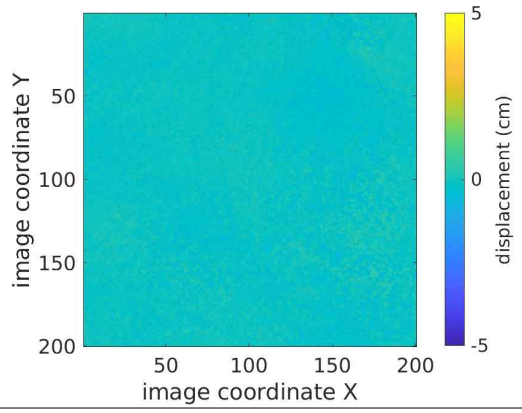
○ 비탈면 변위 분석 결과값은 드론 SAR 관측 시스템의 정확도로 판단 가능함
※ 아래 그림에서 상대적으로 밝은 부분이 관심 비탈면 지역임

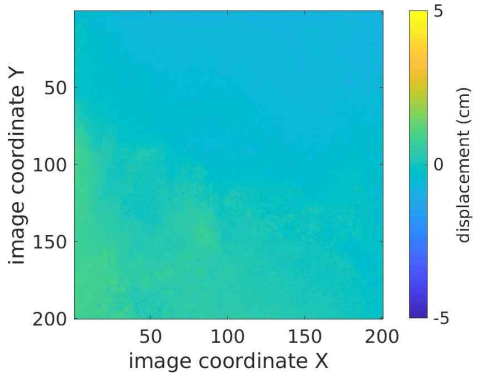
<table>
<tr><td>드론 SAR 시스템 후방산란지도</td><td>드론시스템으로 관측된 비탈면 변위 지도</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>영상레이더 자료 사이의 긴밀도</td><td>관측 변위량의 히스토그램</td></tr>
<tr><td> 
</td><td> 
</td></tr>
</table>

○ 비탈면 변위 평가 분석 결과 과제목표인 5 ㎝ 이하에 해당하는 화소는 전체 유효화소수 (1,504,633개)의 99.99 %에 해당하는 결과를 보임

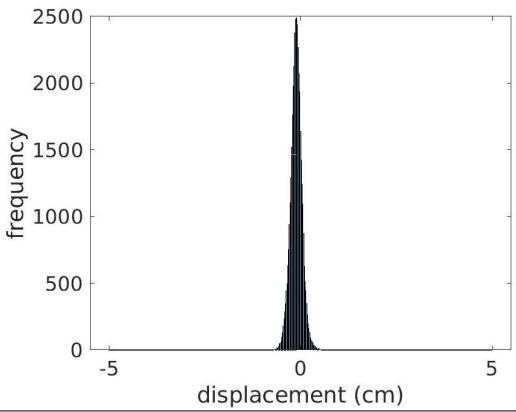
변위 정확도 기준	해당 화소 수 (개)	비율 (%)
5 ㎝ 이하	1,504,484	99.99
4 ㎝ 이하	1,504,326	99.98
3 ㎝ 이하	1,504,065	99.96
2 ㎝ 이하	1,502,439	99.85
1 ㎝ 이하	1,448,781	96.29
0.5 ㎝ 이하	1,175,566	78.13

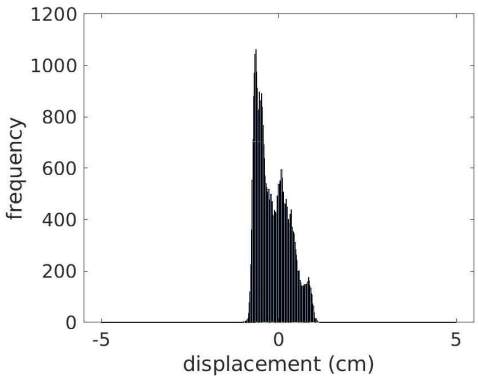
드론시스템으로 관측된 비탈면 변위 지도



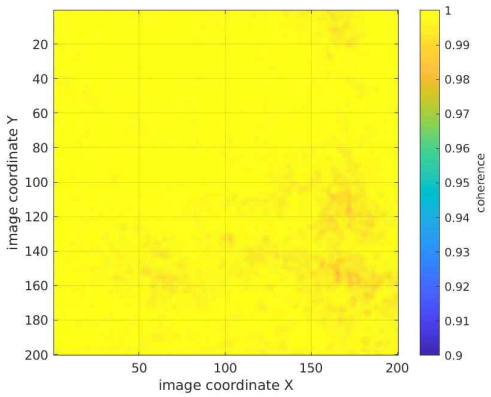


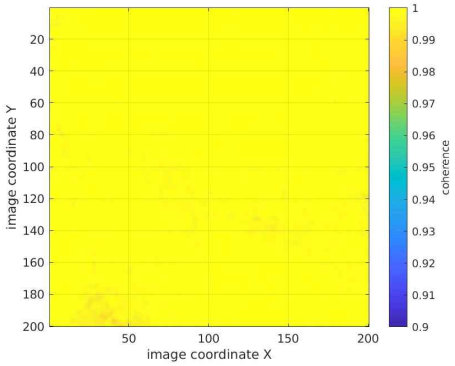
관측 변위량의 히스토그램



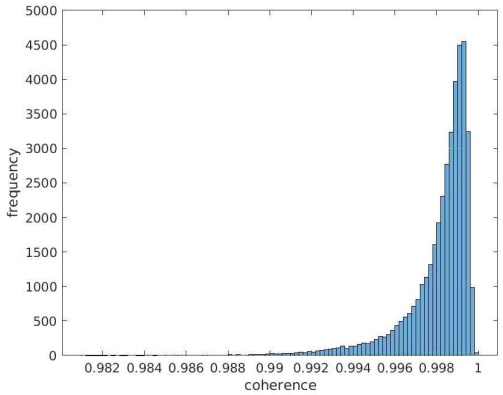


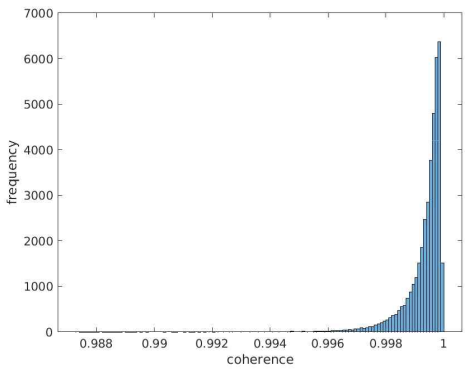
영상레이더 자료 사이의 긴밀도





관측 긴밀도의 히스토그램





○ 200 × 200 화소 영역 두 지역에 대한 변위 측정 추가 분석 수행

구분	최소변위값 (cm)	최대변위값 (cm)	평균 (cm)	표준편차 (cm)
상세 지역 1	-0.74	0.62	-0.12	0.03
상세 지역 2	-1.03	1.2	-0.16	0.08

○ L-밴드 파장을 적용하여 얻은 실험자료의 변위의 통계 계산 분석
○ 긴밀도는 0.98 이상의 매우 우수한 영상레이더 시스템이 개발되었음을 확인

나. 포트홀 탐지 기술

(1) 평가방법론 정립

○ 평가환경
- 포트홀 자동 탐지시스템
· 웹브라우저 중 Firefox, Chrome, Opera, Microsoft Edge 사용
· 포트홀 자동 탐지 서비스 웹 어플리케이션을 통해 항공 영상 업로드 및 탐지 결과 확인
· 항공 영상 업로드 시, 반드시 온라인 상태
- 영상 촬영 환경
· 고정익과 회전익 드론을 사용하여 항공 영상 촬영
· 약 30~70m 고도에서 도로 이미지 촬영
· 악천후가 아닌 조건에서 항공 영상 촬영
· 일몰 후부터 일출 전까지 촬영금지 및 태양과 지표면이 수직일 때, 촬영 권장

○ 전제조건
- 포트홀 면적 계산을 위한 고도 데이터와 CMOS (Complementary Metal Oxide Semiconductor)의 가로와 세로 길이 정보 필요
- 영상 촬영지의 좌표와 촬영된 드론의 Yaw, Pitch 등의 메타데이터 필요
- 항공 이미지 내 도로가 있도록 촬영

○ 평가방법
- 웹 페이지를 통해 약 30~70m 고도에서 촬영한 항공 영상을 포트홀 자동 탐지시스템에 업로드
- 포트홀 자동 탐지시스템 웹 페이지에서 포트홀 탐지 결과 확인
- 포트홀 탐지 결과 분석을 위해 평가방법론 정립
· 딥러닝을 이용한 항공 영상 내 포트홀 탐지 알고리즘의 성능 평가 방법론
· 정확한 포트홀 면적 계산을 위한 포트홀 탐지 영역 추출 알고리즘 성능 평가 방법론
- 평가방법론을 기준으로 탐지 결과 분석
· 포트홀의 검지율과 오검지율 계산
· 포트홀 면적 계산을 위한 포트홀 탐지 영역의 IoU(Intersection over Union) 계산

○ 딥러닝을 이용한 항공 영상 내 포트홀 탐지 알고리즘의 성능 평가방법론
- 분류 성능 평가지표
· 혼동 행렬(Confusion matrix) : 분류 모델에서 구한 분류의 예측값과 데이터의 실제 값의 발생 빈도를 나열한 행렬

분할표(Confusion Matrix)

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

· 혼동 행렬에서 파생되는 지표

성능 평가 지표

지표	계산식	의미
Precision	$\frac{TP}{TP+FP}$	True로 예측된 것 중 실제로도 True인 경우의 비율
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	전체 예측에서 옳은 예측의 비율
Recall	$\frac{TP}{TP+FN}$	실제로 True인 것들 중 예측이 True로 된 경우의 비율
Specificity	$\frac{TN}{FP+TN}$	실제로 False인 것들 중 예측이 False으로 된 경우의 비율
FP Rate	$\frac{FP}{FP+TN}$	실제로는 False인데 True로 예측된 비율
Negative Prediction Value	$\frac{TN}{TN+FN}$	False로 예측한 것 중에 실제 False의 비율
Prevalence	$\frac{FN+TP}{TP+FP+FN+TN}$	전체 중 실제 True가 나타난 비율
Detection Rate	$\frac{TP}{TP+FP+FN+TN}$	전체 중 모델이 실제 True를 예측한 비율
Detection Prevalence	$\frac{TP+FP}{TP+FP+FN+TN}$	전체 중 모델이 True로 예측한 비율
F-1 Score	$2\times\frac{1}{\frac{1}{\mathrm{Pr}ecision}+\frac{1}{Recal}}$	Precision과 Recall의 조화 평균

- 항공 영상 내 포트홀 탐지 알고리즘 성능 평가방법

· 촬영되는 항공 영상들에서 포트홀을 탐지할 때, 중요한 지표는 탐지율과 오탐지율이다. 포트홀 탐지율은 실제 포트홀 중 포트홀로 예측된 것의 비율이다. 오탐지율은 포트홀로 예측한 것 중 포트홀이 아닌 것의 비율

· 분류 성능 평가지표에서 탐지율은 Recall에 해당하며 오탐지율은 1-Precision과 같은 계산식

· 약 30~70m 고도에서 촬영한 항공 영상 100장(포트홀 총 71개)을 이용하여 알고리즘 성능 평가를 진행

· 탐지율과 오탐지율은 각 항공 영상에 대해 평가되었으며, 항공 영상에 포트홀이 없는 경우 탐지와 오탐지를 하지 않았다면, 탐지율과 오탐지율을 100%로 계산

· 각 항공 영상의 탐지율과 오탐지율의 평균으로 항공 영상 내 포트홀 탐지 알고리즘 성능을 평가

○ 정확한 포트홀 면적 계산을 위한 포트홀 탐지 영역 추출 알고리즘 성능 평가방법론

- 면적 계산 성능 평가방법

· IoU(Intersection over Union) : 두 영역의 교차영역의 넓이를 합 영역의 값으로 나눈 값을 의미

· IoU 수식 : $\frac{A\cap B}{A\cup B}$

· 중복 오차율 : 두 영역의 교차영역의 여집합에 대한 넓이를 합 영역으로 나눈 값을 의미

· 중복오차율 수식 : $\frac{(A\cap B)^C}{A\cup B}$

- 포트홀 면적 계산을 위한 포트홀 탐지 영역 추출 알고리즘 성능 평가방법

· 드론으로 촬영된 항공 영상은 높은 고도에서 촬영되어 정확한 면적 계산을 위해서는 포트홀

의 탐지 영역이 정확해야 한다.
· 포트홀의 탐지 영역의 오차값을 계산하기 위해 중복오차율을 사용한다.
· 항공 영상 100장에서 탐지된 포트홀을 이용하여 알고리즘 성능 평가를 진행

(2) 평가자료 분석

○ 딥러닝을 이용한 항공 영상 내 포트홀 탐지 알고리즘의 성능 평가결과 분석
- 항공 영상 내 포트홀 탐지 알고리즘 성능 평가결과
· 약 30~70m 고도에서 촬영한 항공 영상 100장 기준(포트홀 총 71개)
· 포트홀 검지율 : 탐지 포트홀 수 / 포트홀 수 [TP/(TP+FN)]
· 포트홀 오검지율 : 오탐지 횟수 / 포트홀 탐지 횟수 [FP/(TP+FP)]
· 포트홀 검지율 : 97.2%(세부내역 : 부록 1)
· 포트홀 오검지율 : 9.8%(세부내역 : 부록 1)
- 항공 영상 내 포트홀 탐지 알고리즘 성능 평가결과 분석
· 목표 검지율 90%보다 7.2%p 높은 포트홀 검지율을 보임
· 목표 오검지율 10%보다 0.2%p 낮은 포트홀 오검지율을 보임

○ 정확한 포트홀 면적 계산을 위한 포트홀 탐지 영역 추출 알고리즘 성능 평가결과 분석
- 포트홀 면적 계산을 위한 포트홀 탐지 영역 추출 알고리즘 성능 평가결과
· Mask R-CNN 포트홀 영역 추출 결과 : 58.7%
· Steerable filter와 DBSCAN, kd tree을 이용한 포트홀 영역 추출 결과 : 47.9%
- 포트홀 면적 계산을 위한 포트홀 탐지 영역 추출 알고리즘 성능 평가결과 분석
· 딥러닝만 사용하여 포트홀을 탐지하였을 때 보다 약 10.8%p 개선된 성능을 보임

다. PCI 측정 기술

○ 평가환경

- 드론을 이용한 도로 포장 조사 환경
  · 드론: Matrice 600 PRO (최대이륙중량 15.5kg)
  · 초분광 카메라: 미국 Headwall사 Nano-HyperSpec (파장대 400~1000nm)
  · 드론에는 짐벌, GPS/IMU 시스템 포함
- 드론 운용 소프트웨어: 범용 지상통제 소프트웨어(UgCS)
- 드론 초분광영상 전처리:
  · 초분광 카메라 회사인 Headwall사에서 제공하는 SpectralView 소프트웨어
  · 기하보정, 방사보정, 정사보정 등의 전처리
- 드론 초분광영상 데이터 변환 : QGIS(오픈소스 기반 소프트웨어)
- 드론 초분광영상을 이용한 포장상태분석
  · 연구개발한 초분광 포장상태분석 프로그램(HSPCA 1.0) 사용하여 텍스트(CSV)로 변환된 자료를 분석하여 PCI 산출

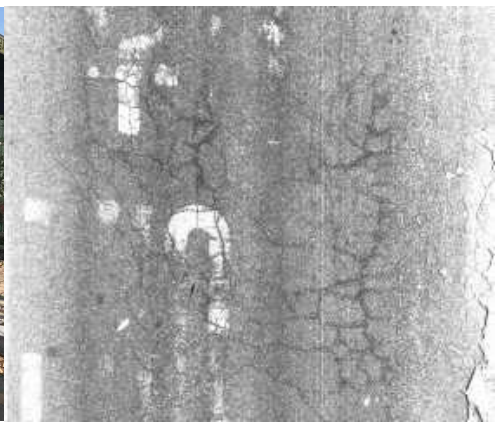


[드론 탑재형 초분광시스템 촬영 (예)]

- 드론 초분광 조사 자료를 비교하기 위한 PCI 산출
  · 도로포장 조사 차량 PES를 이용한 도로 조사 및 결과산출



[조사차량(PES)]



[균열 조사결과(예)]

  · 소성변형, 종단평탄성: 10m 별 분석한 자동 계측결과 이용
  · 균열율: 균열 조사결과를 육안으로 확인하여 화면에 표시하여 10m 별 산출
  · PCI: 소성변형, 종단평탄성, 균열율 결과를 NHPCI 산출식으로 계산

$$NHPCI = \frac{1}{(0.33 + 0.003 \times X_{CR} + 0.004 \times X_{RD} + 0.0183 \times X_{IRI})^2}$$

여기서, X_{CR} : 균열율(%)

X_{CR} : 소성변형 깊이(㎜)

X_{IRI} : 종단평탄성(m/㎞))

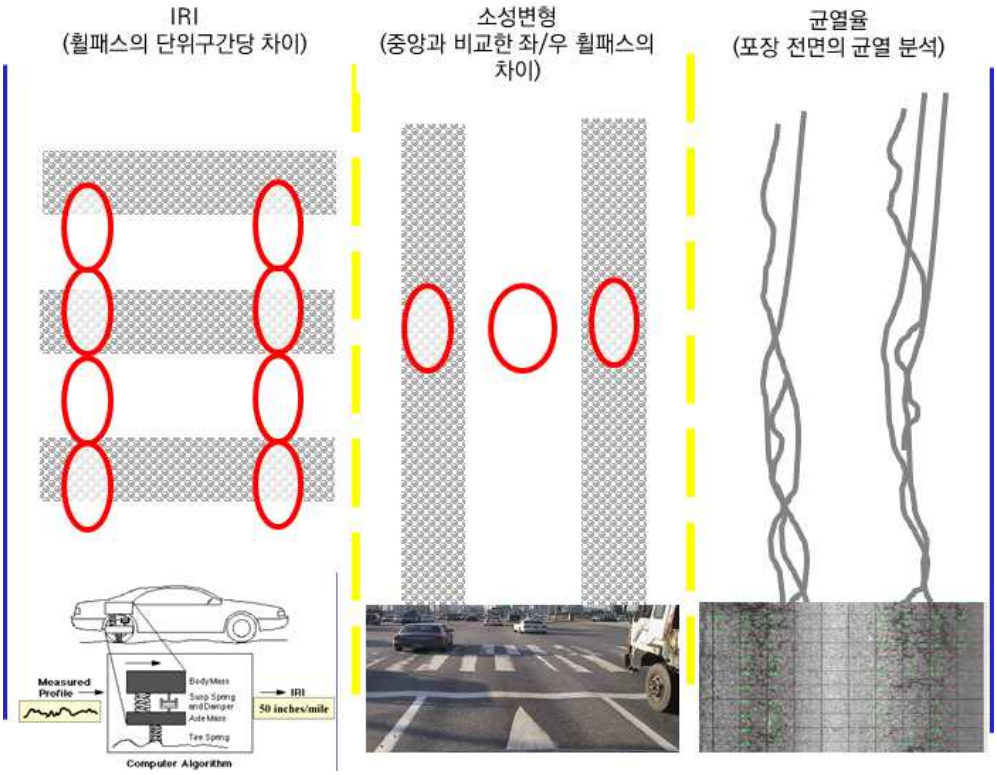
○ 전제조건

- 드론 조사결과에서 평가 제외 구간을 딥러닝 기반 프로그램(HSPCA 1.0)을 이용하여 제외
  · 도로의 차량, 흰색차선, 중앙차선, 버스전용차로 차선, 노면표지, 횡단보도 표시 등의 구간은 평가구간에서 제외하여 포장 평가의 신뢰성 높임



[도로 포장 평가 제외 구간 분류]

- 드론 조사 결과를 10m 구간으로 분할하고, 기본 데이터 산출
  · 종단평탄성은 휠패스 영역을 25cm 단위구간으로 분할하여 단위구간 이전 영역의 데이터와 비교할 수 있도록 2개의 데이터 세트 추출
  · 소성변형은 좌·우 휠패스 영역과 변형이 거의 일어나지 않는 도로의 중심부의 데이터를 비교하여 분석할 수 있도록 3개의 데이터 세트 추출
  · 균열: 도로 전체에서 발생한 균열로써 분석 영역의 전체 데이터를 비교하여 데이터 검출
- 드론 조사 결과와 차량 조사 결과를 딥러닝을 통해 분석모듈의 개발 및 HSPCA 1.0 프로그램에 탑재하여 분석



[도로포장 조사항목 및 특성]

○ 평가방법
- 조사 구간 결정 및 드론 초분광 아스팔트 포장상태 조사 매뉴얼(부록 7) 방법에 준하여 드론을 이용한 초분광 분석
- 도로포장 조사 차량(PES)를 이용하여 조사 및 NHPCI 산출
- 도로 조사구간의 결과를 10m 단위로 분석하여 드론 초분광 평가결과와 조사차량 평가결과의 상관계수 산출

○ 딥러닝을 이용한 포장 평가 방법 및 평가 결과
- 드론 초분광 조사 결과에서 10m 단위의 분석 범위 결정
- 초분광 데이터 대표파장 범위 산출
- 드론 초분광 데이터를 딥러닝으로 학습하여 제작한 프로그램을 통해 IRI, 소성변형, 균열을, IRI 산출
- 딥러닝 모델로 산출된 도로포장 평가 결과는 조사차량으로 평가된 포장평가지수(PCI)와 상관계수가 0.769이었으며, 개발 목표인 0.7 이상을 만족하였음

[조사차량 및 드론 초분광 아스팔트 포장 평가 결과]

No.	IRI (m/km)			rutting (mm)			균열 (%)			PCI		
	조사차량	초분광	편차	조사차량	초분광	편차	조사차량	초분광	편차	조사차량	초분광	편차
1	0.780	1.044	0.264	4.400	4.686	0.286	0.000	0.000	0.000	0.890	1.133	0.243
2	1.200	1.040	0.160	4.200	4.778	0.578	0.000	0.000	0.000	1.380	1.243	0.137
3	1.840	1.108	0.732	4.500	4.643	0.143	0.000	0.000	0.000	2.270	1.378	0.892
4	1.370	1.164	0.206	3.800	4.344	0.544	0.000	0.000	0.000	1.470	1.300	0.170
5	0.970	1.060	0.090	3.800	4.375	0.575	0.000	0.000	0.000	1.000	1.045	0.045
6	0.840	1.091	0.251	4.000	4.382	0.382	0.000	0.000	0.000	0.840	1.053	0.213
7	0.940	1.018	0.078	5.200	5.136	0.064	0.000	0.000	0.000	1.020	1.086	0.066
8	1.040	1.275	0.235	4.200	5.903	1.703	0.000	1.779	1.779	1.330	1.415	0.085
9	1.070	1.239	0.169	4.100	6.218	2.118	0.000	2.979	2.979	1.080	1.388	0.308
10	1.510	1.484	0.026	4.600	4.940	0.340	0.000	0.000	0.000	1.700	1.324	0.376
11	1.620	1.299	0.321	4.800	4.361	0.439	0.000	0.000	0.000	1.630	1.305	0.325

No.	IRI (m/km)			rutting (mm)			균열 (%)			PCI		
	조사 차량	초분광	편차	조사 차량	초분광	편차	조사 차량	초분광	편차	조사 차량	초분광	편차
12	1.110	1.094	0.016	4.000	4.100	0.100	0.000	0.000	0.000	1.190	1.179	0.011
13	1.440	1.088	0.352	4.300	4.702	0.402	0.000	0.000	0.000	1.990	1.265	0.725
14	1.390	1.425	0.035	4.300	4.396	0.096	0.000	0.000	0.000	1.720	1.552	0.168
15	0.930	1.054	0.124	5.700	4.920	0.780	0.000	0.000	0.000	1.080	1.344	0.264
16	1.540	1.362	0.178	4.700	4.791	0.091	0.000	0.000	0.000	1.580	1.552	0.028
17	1.100	1.201	0.101	4.200	4.603	0.403	0.000	0.000	0.000	1.320	1.434	0.114
18	1.460	1.522	0.062	4.800	4.613	0.187	0.000	0.000	0.000	1.960	2.015	0.055
19	1.020	1.011	0.009	4.400	4.411	0.011	0.000	0.000	0.000	1.300	1.308	0.008
20	1.630	1.394	0.236	4.700	4.613	0.087	0.000	0.000	0.000	1.950	1.753	0.197
21	1.590	1.576	0.014	4.300	4.552	0.252	0.000	0.000	0.000	2.000	2.006	0.006
22	2.390	1.779	0.611	4.200	4.399	0.199	0.000	0.000	0.000	2.560	2.559	0.001
23	1.430	1.301	0.129	4.500	4.416	0.084	0.000	0.000	0.000	1.510	1.337	0.173
24	1.100	1.142	0.042	4.700	4.743	0.043	0.000	0.000	0.000	1.140	1.222	0.082
25	0.820	0.800	0.020	4.300	4.335	0.035	0.000	0.000	0.000	0.890	0.897	0.007
26	0.700	0.642	0.058	5.100	5.330	0.230	0.000	0.000	0.000	0.890	0.836	0.054
27	0.810	0.781	0.029	4.000	4.088	0.088	0.000	0.000	0.000	0.940	0.926	0.014
28	0.870	0.842	0.028	5.000	4.990	0.010	0.000	0.000	0.000	1.350	1.382	0.032
29	1.210	1.490	0.280	5.100	6.127	1.027	0.000	0.000	0.000	1.420	2.110	0.690
30	1.390	1.662	0.272	5.400	5.561	0.161	0.000	0.000	0.000	1.640	1.543	0.097
31	1.320	1.362	0.042	4.000	5.158	1.158	0.000	0.000	0.000	1.590	1.932	0.342
32	1.240	1.403	0.163	4.600	4.992	0.392	0.000	0.000	0.000	1.490	1.829	0.339
33	1.560	1.263	0.297	4.300	4.852	0.552	0.000	0.000	0.000	1.570	1.591	0.021
34	1.600	1.601	0.001	4.900	4.830	0.070	0.000	0.000	0.000	2.250	2.253	0.003
35	1.010	1.041	0.031	4.500	4.690	0.190	0.000	0.000	0.000	1.120	1.391	0.271
36	0.930	1.128	0.198	5.000	4.723	0.277	0.000	0.000	0.000	1.300	1.420	0.120
37	0.720	0.935	0.215	4.700	4.726	0.026	0.000	0.000	0.000	0.720	0.863	0.143
38	0.580	0.619	0.039	4.300	4.171	0.129	0.000	0.000	0.000	0.610	0.520	0.090
39	0.790	0.969	0.179	4.900	4.586	0.314	0.000	0.000	0.000	0.850	0.917	0.067
40	1.010	1.001	0.009	5.600	5.563	0.037	0.000	0.000	0.000	1.120	1.106	0.014
41	1.120	1.122	0.002	5.400	5.387	0.013	0.000	0.000	0.000	1.260	1.260	0.000
42	0.800	0.977	0.177	4.800	4.704	0.096	0.000	0.000	0.000	0.970	1.220	0.250
43	0.710	1.052	0.342	4.800	4.684	0.116	0.000	0.000	0.000	0.790	1.299	0.509
44	0.840	0.871	0.031	5.500	5.161	0.339	0.000	0.000	0.000	0.870	0.968	0.098
45	1.160	1.216	0.056	5.300	5.017	0.283	0.000	0.000	0.000	1.400	1.369	0.031
46	1.220	1.121	0.099	5.800	5.037	0.763	0.000	0.000	0.000	1.230	1.359	0.129
47	0.660	1.024	0.364	4.400	4.588	0.188	0.000	0.000	0.000	0.700	1.219	0.519
48	0.960	1.074	0.114	5.200	4.722	0.478	0.000	0.000	0.000	1.260	1.261	0.001
49	0.970	1.162	0.192	4.900	4.416	0.484	0.000	0.000	0.000	1.040	1.221	0.181
50	1.090	1.229	0.139	4.700	4.417	0.283	0.000	0.000	0.000	1.290	1.306	0.016
51	1.330	1.249	0.081	4.700	4.682	0.018	0.000	0.000	0.000	1.430	1.379	0.051
52	1.320	1.147	0.173	5.200	5.056	0.144	0.000	0.000	0.000	1.430	1.390	0.040
53	0.780	1.234	0.454	6.100	5.144	0.956	0.000	0.000	0.000	1.000	1.420	0.420
54	1.570	1.226	0.344	6.400	5.141	1.259	0.000	0.000	0.000	1.870	1.508	0.362
55	1.960	1.319	0.641	5.600	5.593	0.007	0.000	0.000	0.000	2.110	1.601	0.509
56	1.610	1.311	0.299	7.600	6.328	1.272	0.000	0.000	0.000	1.850	1.318	0.532
상관 계수	0.760			0.568			-			0.769		

<총괄 성과점검기준표>

핵심성과 (CTE)		단위성과		기관	성과점검기준						양적 성과 건수
					연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
1	SAR 센서 기반 비탈면 변위 측정 드론시스템	1-1	비탈면 변위 측정을 위한 SAR(L 밴드) 센서 장착 드론	메타 빌드 (공동)	①	비탈면 측정 드론 임무계획 요구사항 분석	1건	분석서	분석서 작성 확인	0.9	특허 : 출원 1/1건 등록 건 논문 : 건 시제품 : 건 S/W : 건
					②	비탈면 측정 드론 임무계획 소프트웨어	1식	소프트웨어	소프트웨어 개발 확인	2.5	특허 : 출원 1/1건 등록 건 논문 : 건 시제품 : 건 S/W : 1/1건
					③	비탈면 측정 드론 관제 소프트웨어	1식	소프트웨어	소프트웨어 개발 확인	0.5	특허 : 출원 1/0건 등록 1/2건 논문 : 건 시제품 : 건 S/W : 1/1건
		1-2	드론 SAR 데이터 기반 비탈면 변위 측정 소프트웨어	부산대 (공동)	①	변위측정 정확도	15cm	측량 기준값 수집	기준값과 변위 비교분석	3.5	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
					②	변위측정 정확도	10cm	측량 기준값 수집	기준값과 변위 비교분석	3.5	특허 : 출원 건 등록 건 논문 : 0/1건 시제품 : 건 S/W : 0/1건
					③	변위측정 정확도*	5cm	측량 기준값 수집	기준값과 변위 비교분석	8.7	특허 : 출원 건 등록 건 논문 : 0/1건 시제품 : 건 S/W : 1/2건
		1-3	비탈면 변위 측정 및 가시화 시스템	부산대 (공동) 메타 빌드 (공동)	①	비탈면 변위위치 가시화 정확도	5m	측량 기준값 수집	기준값과 가시화 위치 비교분석	0.6	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
					②	비탈면 변위위치 가시화 정확도	3m	측량 기준값 수집	기준값과 가시화 위치 비교분석	1.0	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
					③	비탈면 변위위치 가시화 정확도	1m	측량 기준값 수집	기준값과 가시화 위치 비교분석	2.0	특허 : 출원 1/0건 등록 건 논문 : 건 시제품 : 건 S/W : 1/1건
		1-4	평가/검증 보고서	롯데 건설 (공동)	①	비탈면 계측 보고서	1건	보고서	보고서 작성 확인	0.6	특허 : 출원 1/1건 등록 건 논문 : 건 시제품 : 건 S/W : 건
					②	비탈면 가시화 평가/검증 계획서	1건	계획서	계획서 작성 확인	0.6	특허 : 출원 1/1건 등록 건 논문 : 건 시제품 : 건 S/W : 건
					③	비탈면 가시화 정확도 평가/검증 보고서	1건	평가 보고서	평가 보고서 작성 확인	1.4	특허 : 출원 1/2건 등록 1/2건 논문 : 건 시제품 : 건 S/W : 건

핵심성과 (CTE)		단위성과		기관	성과점검기준						양적 성과 건수
					연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
		1-5	비탈면 변위 가시화 시스템 활용 지침	롯데 건설 (공동)	①	비탈면 변위 활용지침 요구사항 정의	1건	정의서	정의서 작성 확인	0.6	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
					②	기존 비탈면 지침 조사 분석서	1건	분석서	분석서 작성 확인	0.6	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
					③	개발시스템 활용지침	1건	활용 지침	활용 지침 작성 확인	0.4	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
		1-6	비탈면 기술개발 테스트베드	건설연 (주관)	③	테스트베드	1식	파일럿 비탈면	파일럿 비탈면 구축 확인	1.0	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
2	영상 센서 기반 포트홀 사전/사후 탐지 드론시스템	2-1	포트홀 탐지를 위한 영상 센서 장착 드론	공간 정보 (공동)	①	포트홀 탐지 드론시스템 기술	1건	시제품	시제품 제작 확인	2.1	특허 : 출원 1/2건 등록 건 논문 : 건 시제품 : 0/1건 S/W : 건
					②	포트홀 탐지 포장 영상 DB 구축	100 km	포장 영상 DB	포장 영상 DB 구축 확인	2.1	특허 : 출원 1/1건 등록 건 논문 : 건 시제품 : 건 S/W : 건
					③	포트홀 탐지 드론 운용 지침	1건	운용 지침	운용 지침 작성 확인	5.1	특허 : 출원 1/1건 등록 1/2건 논문 : 건 시제품 : 건 S/W : 건
		2-2	드론 영상 데이터 기반 포트홀 측정 소프트웨어	건설연 (주관) 대구가톨릭대 (공동)	①	검지율 및 오검지율	검지 : 60% 오검지 : 30%	기준값 수집	기준값과 비교분석	3.9	특허 : 출원 1/2건 등록 건 논문 : 건 시제품 : 건 S/W : 1/1건
					②	검지율 및 오검지율	검지 : 70% 오검지 : 20%	기준값 수집	기준값과 비교분석	3.9	특허 : 출원 1/2건 등록 건 논문 : 건 시제품 : 건 S/W : 1/4건
					③	검지율 및 오검지율	검지 : 80% 오검지 : 10%	기준값 수집	기준값과 비교분석	0.45	특허 : 출원 1/2건 등록 1/2건 논문 : 0/1건 시제품 : 건 S/W : 1/5건
		2-3	포트홀 위치 가시화 시스템	공간 정보 (공동) 아와 소프트 (공동)	①	포트홀 위치 가시화 정확도	2m	DGPS 기준값 수집	기준값과 비교분석	0.6	특허 : 출원 1/0건 등록 건 논문 : 건 시제품 : 0/1건 S/W : 건
					②	포트홀 위치 가시화 정확도	1m	DGPS 기준값 수집	기준값과 비교분석	0.6	특허 : 출원 1/1건 등록 건 논문 : 건 시제품 : 건 S/W : 건

핵심성과 (CTE)		단위성과		기관	성과점검기준						양적 성과 건수
					연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
					③	포트홀 위치 가시화 정확도	0.5m	DGPS 기준값 수집	기준값과 비교분석	1.4	특허 : 출원 1/0건 등록 1/1건 논문 : 건 시제품 : 건 S/W : 1/1건
		2-4	평가/검증 보고서	건설연 (주관)	①	포트홀 탐지 평가/검증 계획서	1건	평가 계획서	계획서 작성 확인		특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
					②	포트홀 탐지 평가/검증 보고서	1건	평가 보고서	평가 보고서 작성 확인	2.3	특허 : 출원 0/1건 등록 건 논문 : 0/1건 시제품 : 건 S/W : 0/1건
					③	포트홀 가시화 정확도 평가/검증 보고서	1건	평가 보고서	평가 보고서 작성 확인	3.5	특허 : 출원 건 등록 0/1건 논문 : 건 시제품 : 건 S/W : 건
		2-5	포토홀 탐지 및 가시화 시스템 활용 지침	건설연 (주관)	①	포트홀 활용지침 요구사항 정의서	1건	정의서	정의서 작성 확인		특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
					②	기존 포트홀 지침 조사 분석서	1건	분석서	분석서 작성 확인	1.4	특허 : 출원 건 등록 건 논문 : 0/1건 시제품 : 건 S/W : 0/1건
					③	개발시스템 활용지침	1건	활용 지침	활용 지침 작성 확인	1.2	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
		2-6	포트홀 기술개발 테스트베드	건설연 (주관)	③	포트홀 탐지 실험시설	1식	실험 시설	실험 시설 구축 확인	2.0	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
3	초분광 센서 기반 포장상태지수(PCI) 측정 드론시스템	3-1	PCI 측정을 위한 초분광 센서 장착 드론	신한 항업 (공동)	①	PCI 측정 드론시스템 기술	1건	시제품	시제품 제작 확인	2.3	특허 : 출원 1/0건 등록 건 논문 : 건 시제품 : 0/1건 S/W : 건
					②	PCI 측정 포장 초분광 DB 구축	100 km	포장 초분광 DB	포장 초분광 DB 구축 확인	2.3	특허 : 출원 1/1건 등록 건 논문 : 건 시제품 : 건 S/W : 건
					③	PCI 측정 드론 운용 지침	1건	운용 지침	운용 지침 작성 확인	5.7	특허 : 출원 1/3건 등록 1/3건 논문 : 건 시제품 : 건 S/W : 건
		3-2	초분광 데이터 기반 PCI 추정 소프트웨어	건설연 (주관)	①	기준값 대비 PCI 측정 상관계수**	0.5	PCI 조사차량 이용 기준값 수집	기준값과 상관계수 비교평가	3.9	특허 : 출원 1/2건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
					②	기준값 대비 PCI 측정 상관계수	0.6	PCI 조사차량 이용 기준값 수집	기준값과 상관계수 비교평가	3.9	특허 : 출원 0/1건 등록 건 논문 : 0/1건 시제품 : 건 S/W : 0/1건

핵심성과 (CTE)	단위성과		기관	성과점검기준						양적 성과 건수
				연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
				③	기준값 대비 PCI 측정 상관계수	0.7	PCI 조사차량 이용 기준값 수집	기준값과 상관계수 비교평가	5.2	특허 : 출원 건 등록 1/2건 논문 : 0/1건 시제품 : 건 S/W : 1/2건
	3-3	PCI 가시화 시스템	신한 항업 (공동) 아와 소프트 (공동)	①	PCI 측정 위치 가시화 정확도	2m	DGPS 기준값 수집	기준값과 가시화 위치 비교분석	0.6	특허 : 출원 1/1건 등록 건 논문 : 건 시제품 : 건 S/W : 건
				②	PCI 측정 위치 가시화 정확도	1m	DGPS 기준값 수집	기준값과 가시화 위치 비교분석	0.6	특허 : 출원 1/0건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
				③	PCI 측정 위치 가시화 정확도	0.5m	DGPS 기준값 수집	기준값과 가시화 위치 비교분석	1.4	특허 : 출원 1/0건 등록 1/1건 논문 : 건 시제품 : 건 S/W : 1/1건
	3-4	평가/검증 보고서	건설연 (주관)	①	PCI 추정 평가/검증 계획서	1건	평가 계획서	계획서 작성 확인	-	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
				②	PCI 추정치 평가/검증 보고서	1건	평가 보고서	평가 보고서 작성 확인	1.7	특허 : 출원 0/1건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
				③	PCI 가시화 정확도 평가/검증 보고서	1건	평가 보고서	평가 보고서 확인	4.1	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
	3-5	PCI 측정 및 가시화 시스템 활용 지침	건설연 (주관)	①	PCI 활용지침 요구사항 정의서	1건	정의서	정의서 작성 확인	-	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
				②	기존 PCI 지침 조사 분석서	1건	분석서	분석서 작성 확인	1.4	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 건
				③	개발시스템 활용지침	1건	활용 지침	활용 지침 작성 확인	1.2	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
	3-6	PCI 측정 기술개발 테스트베드	건설연 (주관)	③	PCI 측정 실험시설	1식	실험 시설	실험 시설 구축 확인	2.0	특허 : 출원 건 등록 건 논문 : 건 시제품 : 건 S/W : 0/1건
계				-	-	-	-	-	100	특허 : 출원 23/27건 등록 8/16건 논문 : 0/7건 시제품 : 0/3건 S/W : 10/33건

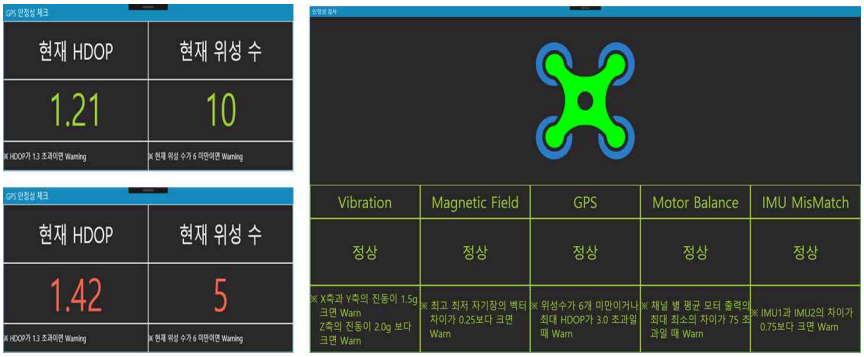
<1차년도 성과점검기준표>

핵심성과 (CTE)		단위성과		기관	성과점검기준						목표 달성도
					연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
1	SAR 센서 기반 비탈면 변위 측정 드론시스템	1-1	비탈면 변위 측정을 위한 SAR(L 밴드) 센서 장착 드론	메타빌드 (공동)	①	비탈면 측정 드론 임무계획 요구사항 분석	1건	분석서	분석서 작성 확인	0.9	달성 완료
		1-2	드론 SAR 데이터 기반 비탈면 변위 측정 소프트웨어	부산대 (공동)	①	변위측정 정확도	15cm	측량 기준값 수집	기준값과 변위 비교분석	3.5	달성 완료
		1-3	비탈면 변위 측정 및 가시화 시스템	부산대 (공동) 메타빌드 (공동)	①	비탈면 변위위치 가시화 정확도	5m	측량 기준값 수집	기준값과 가시화 위치 비교분석	0.6	달성 완료
		1-4	평가/검증 보고서	롯데건설 (공동)	①	비탈면 계측 보고서	1건	보고서	보고서 작성 확인	0.6	달성 완료
		1-5	비탈면 변위 가시화 시스템 활용 지침	롯데건설 (공동)	①	비탈면 변위 활용지침 요구사항 정의	1건	정의서	정의서 작성 확인	0.6	달성 완료
2	영상 센서 기반 포트홀 사전/사후 탐지 드론시스템	2-1	포트홀 탐지를 위한 영상 센서 장착 드론	공간정보 (공동)	①	포트홀 탐지 드론시스템 기술	1건	시제품	시제품 제작 확인	2.1	달성 완료
		2-2	드론 영상 데이터 기반 포트홀 측정 소프트웨어	건설연 (주관)	①	검지율 및 오검지율	검지 : 60% 오검지 : 30%	기준값 수집	기준값과 비교분석	3.9	달성 완료
		2-3	포트홀 위치 가시화 시스템	공간정보 (공동) 아와소프트 (공동)	①	포트홀 위치 가시화 정확도	2m	DGPS 기준값 수집	기준값과 비교분석	0.6	달성 완료
		2-4	평가/검증 보고서	건설연 (주관)	①	포트홀 탐지 평가/검증 계획서	1건	평가 계획서	계획서 작성 확인		달성 완료
		2-5	포토홀 탐지 및 가시화 시스템 활용 지침	건설연 (주관)	①	포트홀 활용지침 요구사항 정의서	1건	정의서	정의서 작성 확인		달성 완료
3	초분광 센서 기반 포장상태지	3-1	PCI 측정을 위한 초분광 센서 장착	신한항업 (공동)	①	PCI 측정 드론시스템 기술	1건	시제품	시제품 제작 확인	2.3	달성 완료

핵심성과 (CTE)		단위성과		기관	성과점검기준						목표 달성도
					연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
	수(PCI) 측정 드론시스템		드론								
		3-2	초분광 데이터 기반 PCI 추정 소프트웨어	건설연 (주관)	①	기준값 대비 PCI 측정 상관계수	0.5	PCI 조사차량 이용 기준값 수집	기준값과 상관계수 비교평가	3.9	달성 완료
		3-3	PCI 가시화 시스템	신한 항업 (공동) 아와 소프트 (공동)	①	PCI 측정 위치 가시화 정확도	2m	DGPS 기준값 수집	기준값과 가시화 위치 비교분석	0.6	달성 완료
		3-4	평가/검증 보고서	건설연 (주관)	①	PCI 추정 평가/검증 계획서	1건	평가 계획서	계획서 작성 확인	-	달성 완료
		3-5	PCI 측정 및 가시화 시스템 활용 지침	건설연 (주관)	①	PCI 활용지침 요구사항 정의서	1건	정의서	정의서 작성 확인	-	달성 완료

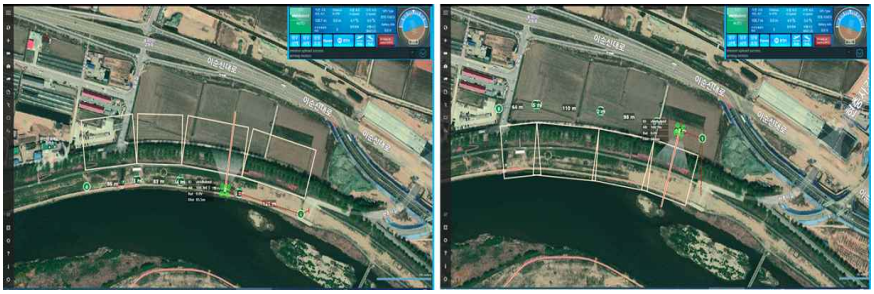
○ 비탈면 측정 드론 임무계획 요구사항 분석

- 비탈면 측정 드론의 상태를 고려한 임무비행이 가능한 비행환경 기준 필요



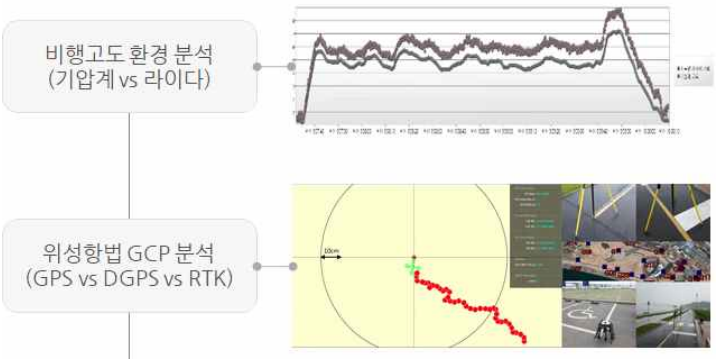
[드론 상태 자동점검 예시]

- 비탈면 방향•형태•크기•개수에 따른 (비)연속 또는 복수 임무계획 생성 기능



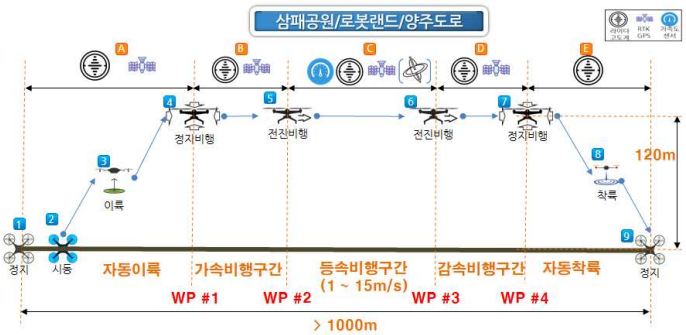
[임무장치 위치 및 각도에 따른 임무비행계획서 예시]

- 동일한 비탈면의 반복•연속 측정을 위한 임무계획 생성 기능
- 임무계획 생성•저장•수정•삭제 등 일반 관리 기능
- 드론의 임무장치 특성에 따른 임무비행 고도와 경로 설정 매칭 기능



[항법장치 성능의 차이 분석]

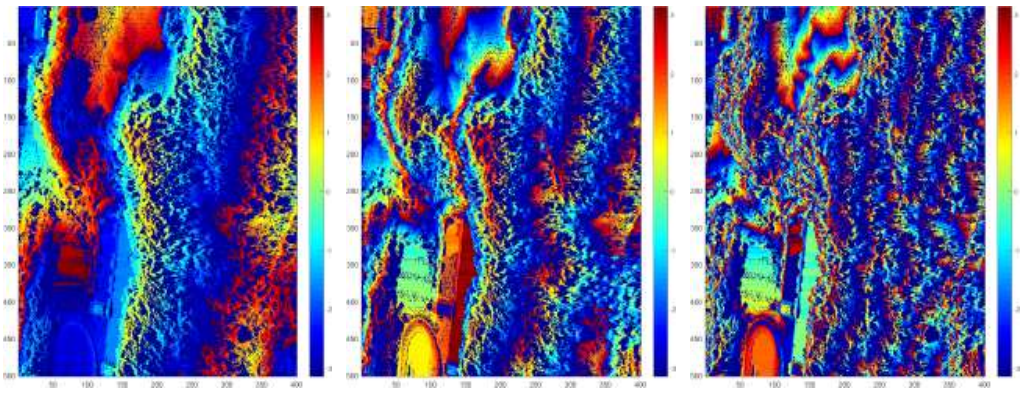
- 비탈면 측정을 위한 이륙•전진•리턴•착륙이 포함된 임무계획 생성 기능



[임무비행 패턴 예시]

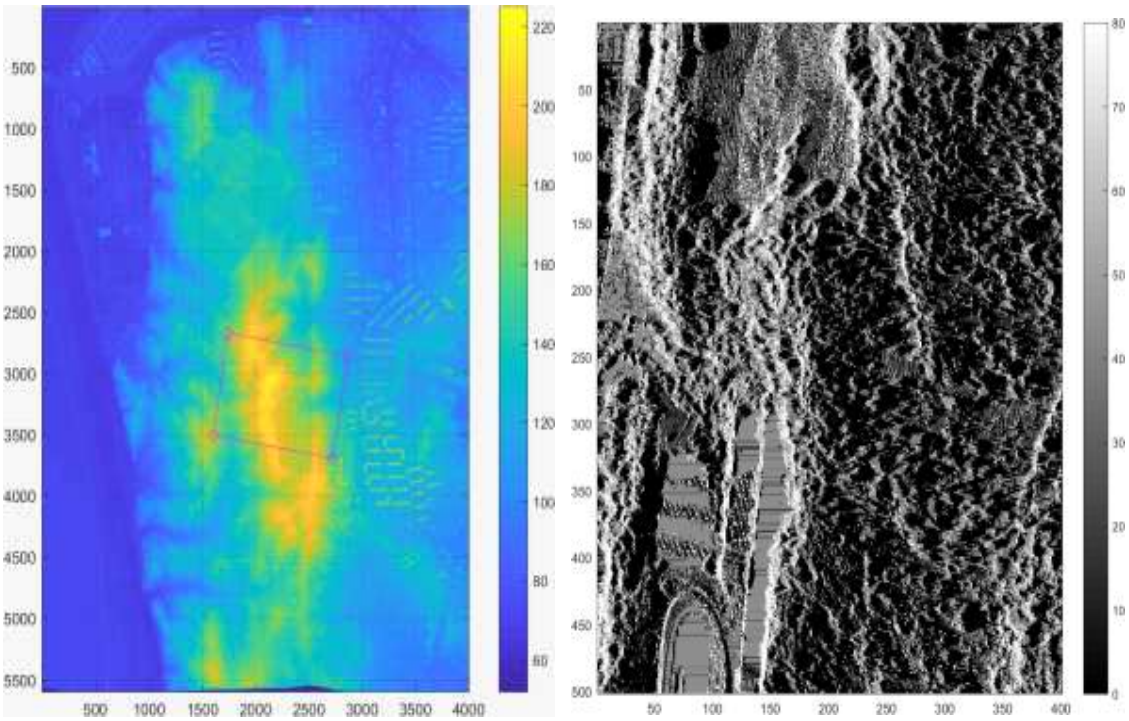
○ 비탈면 변위 및 가시화 검증계획서 및 평가분석서

- 1차년도는 비탈면 변위 및 가시화 측정을 위한 드론 기반 영상레이더 개발을 설계하는 과정이므로 활용 가능한 인공위성 기반 자료를 이용하여 성과목표에 대한 검증을 실시하였음
- 변위 측정 정확도 오차는 드론 궤도간격 (기선거리)에 따른 모사된 간섭영상 생성에 기반 하여 성과목표를 검증하였음



[기선거리별 모사된 간섭영상 (좌) 100m, (중) 200m, (우) 400m]

- 개발된 시뮬레이터를 이용한 지형 정보 위상 제거를 통해 잔여 위상 (변위측정 오차)은 최대 1 fringe (~12cm) 이내로 발생하였으며 성과목표를 달성 완료함
- 비탈면 가시화 정확도 평가는 실제 드론으로 촬영된 SAR 영상과 모사된 SAR 영상간의 정합을 통해 좌표오차를 개선하는 방식으로 검증을 실시하였음
- 보정된 영상에서 두 개의 기준점을 통해 분석한 좌표오차는 최대 3 m 이내로 성과목표 달성 완료함



[사면을 포함한 DSM (좌), 개발시뮬레이터로부터 모사된 SAR 영상 (우)]

○ 비탈면 변위 활용지침 요구사항 정의서

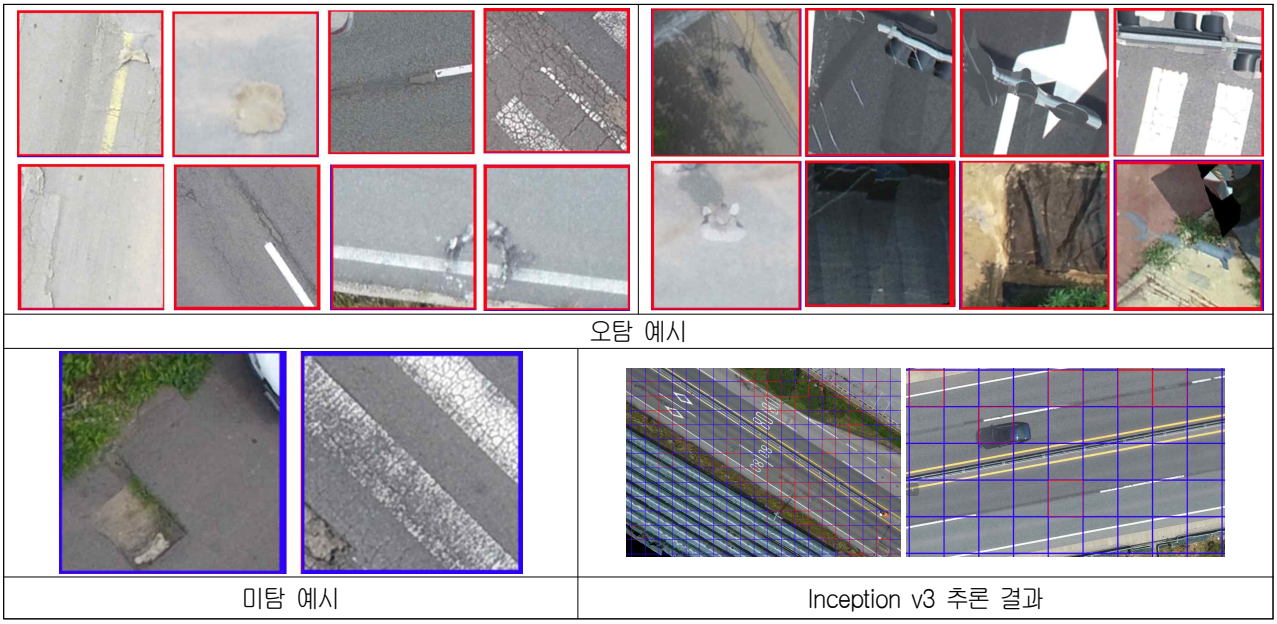
- 국내 비탈면 특성
  - 자연재해 : 집중호우와 같은 강우패턴 변화로 인해 비탈면 피해 증가
  - 인공재해 : 건설부지 부족으로 부지 개발로 인한 급경사 비탈면 증가
- 다양한 비탈면 파괴 유형을 대비해야 함
  - 비탈면 배면의 인장균열 파악 시 대규모 비탈면 붕괴 대처 가능

- 국내산지는 대부분 수목으로 이루어져 있으며, 절취된 표면도 수풀로 덮여있기 때문에 지표 변화 감지가 어려움
- 기존 비탈면 변위 측정 방법 : 비탈면에 무선망 관리 계측기 설치
  · 설치 지점만 측정 확인가능
- 비탈면 외부에서 거동 확인
  · 정기적 육안 관찰 조사 : 접근이 어려운 지점 발생, 소요 시간이 김
  · 지상 광파기 조사 : 타겟 설치 지점만 확인가능
  · 지상 3D 레이저 스캐너 조사 : 지상 관측시 비탈면 조사 음영 발생, 수풀 식생이 반영된 표면 측정
  · 실 화상 드론 조사 : 수목, 수풀 식생이 반영된 영상 측정
  · 열화상, 다분광, 초분광 활용 드론 조사 : 수목, 수풀 식생에서 반사된 영상 측정
  · 라이다 활용 드론 조사 : 수풀 식생이 반영된 표면 측정
- 비탈면 변위 측정 활용지침 요구사항 정의
  · 우리나라 3만개의 국도 비탈면 및 건설부지 시공 굴착 비탈면 모두 관리할 수 있는 신속한 측정 기술이 요구됨
  · 일부 지점이 아닌 비탈면 모든 지점의 거동 확인이 가능한 측정 기술이 필요
  · 비탈면 수풀뿐만 아니라 비탈면 배면의 수목과 수풀 아래 지표 변위를 감지가 가능한 L밴드 SAR 활용한 기술 개발 추진

○ 포트홀 탐지 드론시스템 시제품
- 기존 상용화된 드론 제품 및 기술을 바탕으로 국도 및 지방도 등 전국 도로를 대상으로 지진, 기후변화 등에 따라 해마다 증가하는 포트홀의 긴급복구를 위해 영상을 수집하는 장치이다. 시제품은 상용품 및 국내 기술로 개발한 드론을 활용하였고, 취득된 영상을 기반으로 데이터 분석 기반 포트홀 자동탐지기술을 적용하여 포트홀 위치 및 크기를 측정하고, 이를 활용하여 포트홀을 신속보수하기 위한 가시화 기술로 구성
- 드론 촬영 영상 분석 기반 포트홀 및 균열 객체 인지 · 추출 기술 개발
- 드론 촬영 영상 분석 기반 포트홀 면적 산출 기술 개발
- GIS 맵 기반 드론 측정 포트홀 데이터 가시화 기술 개발
- 드론 탑재 영상 센서 포트홀 탐지 및 가시화 결과 평가/검증 기술 개발
- 드론 측정 데이터 기반 포트홀 사전보수/긴급보수 효율화 방법론 개발

포트홀 탐지 드론 시스템 시제품	
	
	

○ 포트홀 탐지 검지율/오검지율 평가계획서 및 평가결과 분석서

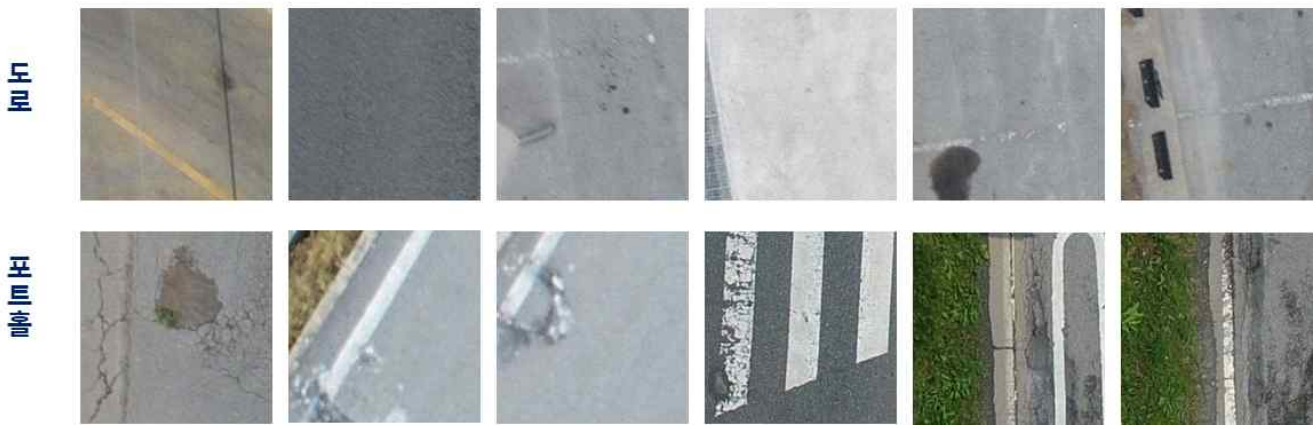


오탐 예시

미탐 예시 | Inception v3 추론 결과

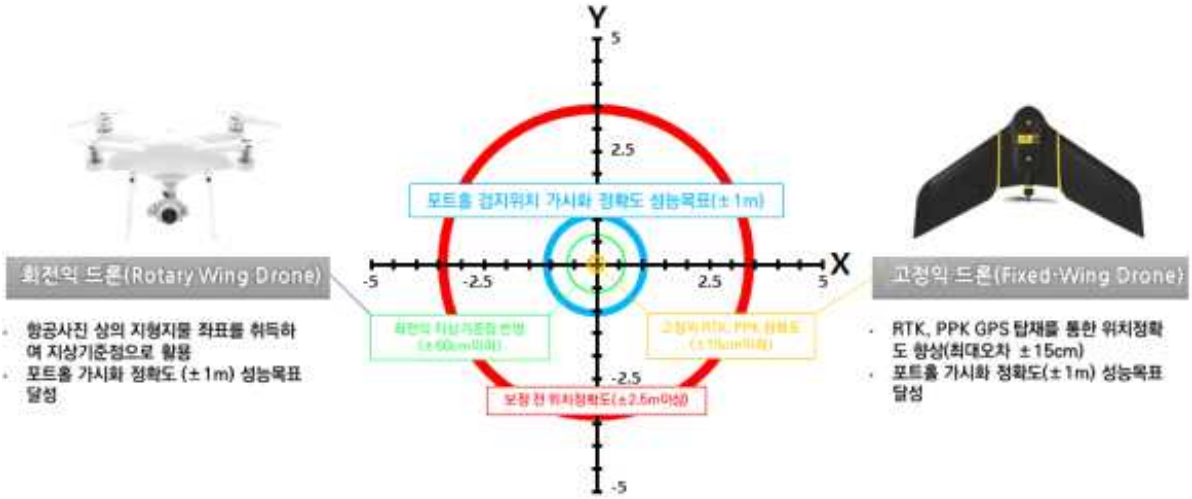
- 포트홀 탐지 검지율/오검지율 평가계획 및 결과
  - · 검지율 : 탐지 포트홀 수 / 전체 포트홀 수
  - · 오검지율 : 비 포트홀 탐지 횟수 / 포트홀 탐지 횟수
  - · 탐지 에러율 : 비 포트홀 탐지 횟수 / 전체 탐지 실행 횟수
  - · MobileNet : 검지율 0% (0/25)
  - · InceptionNet : 검지율92%(23/25) 오검지율29.9(345/1001) 탐지에러율0.05%(2344/44,343)
  - · 타일 기반 MobileNet 딥러닝 모델의 부적함으로 인해 낮은 분류 정확도
  - · 타일 기반 Inception v3 딥러닝 모델의 낮은 도로 영역 분류 정확도

- 포트홀 탐지 검지율/오검지율 평가 결과 분석
  - · 정합 이미지 내 합성된 영역 내 영상 변형 발생 → 오탐 발생 요인
  - · 원본 촬영 이미지 기반 도로 파손 영역(포트홀)탐지



도로 이미지와 포트홀 이미지 예시

○ 포트홀 위치 가시화 정확도
- 고정밀 GPS 탑재 드론, 지상기준점을 활용 포트홀 위치 정확도 향상
- 고정익드론 : RTK, PPK 기능 탑재 드론활용 위치정확도 향상(RMSE 0.1m)
- 회전익드론 : 지상기준점 사용 위치정확도 향상(RMSE 0.6m)



○ 포트홀 탐지 드론시스템 활용지침 요구사항 정의서

- 비행 전 준비사항(내업)

· 비행공역확인
· 촬영전 항공촬영 및 비행승인 인허가 획득
· 비행 및 촬영계획 수립
· 기상 및 풍속, 지자기 이상 등 사전분석

- 비행 전 준비사항(외업)

· 드론 촬영전 예비 비행으로 안전성 확보
· 드론 조종자(촬영자)의 조종자 준수사항 숙지
· 드론의 이착륙 지점의 설정과 안전대책 수립
· 드론 탑재 센서 보정(Calibration)

- 비행

· 기체 조종, 카메라 조종 등 역할을 분담하여 업무 수행
· 드론 비행은 전진, 후진, 측면, 상승, 하강 등 대상지의 특성에 맞게 조정
· 안전한 비행을위한 고려사항 숙지
· 차량이나 행인 위에서의 비행 금지
· 카메라 및 배터리 성능 수시 체크 및 최대능력치 사용금지
· 비행 중 온도, 풍속 등 기상상태 수시체크
· 조종기의 수신감도 상시 확인 및 가시권비행 수행
· 조종자와 부조종자 간의 원할한 소통 필요
· 안개 등으로 지상 목표물 식별할 수 없는 경우 비행 금지
· 기체이상 및 기상악화 등 돌발 및 비상상황 발생시 즉각 비상착륙
· 사고발생시 항공안전법에 의거하여 사고의 즉각적인 보고 조취

- 비행 완료 후

· 비행 전후 체크리스트를 작성하여 드론의 상태 및 기능의 작동여부 점검
· 기체의 외관 점검
· 모터 및 날개, 프로펠러 등 부속품 점검
· 라디오 모뎀 송수신기 점검
· 카메라, GPS, 배터리, 조종기 등 점검
· 안전점검 조사지역의 드론운영 현황 파악을 위해 드론 일지작성
· 촬영영상 확인 및 후속작업을 위한 데이터 추출
· 데이터 백업 및 메타데이터 작성

○ PCI 측정 드론시스템 시제품

- 본 연구 수행을 위해 투입된 드론 초분광시스템은 드론을 활용한 도로포장상태 평가 기술 적용을 목적으로 제작하였으며 시스템의 구성은 소형 초분광센서, IMU/GPS, 드론, SW로 구성됨



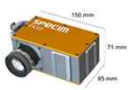
[드론초분광 시스템의 구성]

- 드론 탑재형 초분광 센서

· 도로포장상태 평가 적용을 위한 파장범위 400-1,000nm, 밴드폭 10nm 이하의 초분광센서를 조사하였으며, 도로조사 목적에 맞는 초분광센서는 뛰어난 공간 및 스펙트럼 해상도, 넓은 시야, 높은 SNR 등 고해상도로 영상을 취득이 가능한 Headwall社의 Nano-Hyperspec 센서를 선정함

· 이는 타 센서에 비해 크기가 작어 드론 탑재하여 안전적인 운영이 가능하며, 높은 신호대 잡음과 흔들림을 최소화하는 설계로 안정성 및 내구성이 뛰어난 강점이 있음

[드론 탑재용 초분광 카메라 조사]

센서	Nano-Hyper spec	Micro-CASI1920	FX-10	SOC710-GX	SHARK	OCI UAV
제조사	- headwall - (미국)	- ITRES - (캐나다)	- Specim - (핀란드)	- Surface Optics (미국)	- Corning - (미국)	- Bay Spec - (미국)
크기(mm) /무게(kg)	- 76×76×120/ - 0.52	- 102×190×178/ - 1.50	- 150×71×85/ - 1.40	- 103×200×200/ - 1.25	- 137×88×71/ - 0.680	- 80×80×30/ - 0.272
밴드수	- 270	- 288	- 220	- 120	- 180	- 100
밴드폭(nm)	- 2.2	- 2.1	- -	- 4.2	- 3.3	- 5
파장대(nm)	- 400~1,000	- 400~1,000	- 400~1,000	- 400~1,000	- 400~1,000	- 600~1,000
센서사진						

- 초분광 센서 탑재 드론

· 초분광센서의 탑재 드론은 DJI社의 Matrice 600 PRO기이며, 최대 이륙중량은 15.5kg로 소형 초분광센서를 탑재하고도 안정적인 운영과 성과도출이 가능함

[Matrice 600 PRO 드론 스펙]

장비	항목	세부 사양
 Matrice 600 PRO	- 크 기	- 1668×1518×727 mm - (프로펠러, 프레임 암, GPS 마운트)
	- 무 게	- 9.5 kg
	- 최대이륙중량	- 15.5 kg
	- 최대속도	- 65km/h
	- 최대 실용 상승 한도 - (해발 고도)	- 2500m

- GPS / IMU 시스템

· GPS(Global Positioning System)는 인공위성으로부터 수신된 전파를 해석하여 위치와 속도, 시간을 알 수 있는 범 세계위치결정 시스템이며, IMU(Inertial Measurement Unit)는 중력의 형상, 대상의 기울기 및 회전력을 감지하는 센서들로 구성되어있는 관성측정장치로서 GPS/IMU 시스템은 초분광센서에 장착되어 탑재 비행체의 위치, 시간, 속도, 방향 등의 종합적인 자세정보를 제공함



[GPS Antenna 및 IMU 센서]

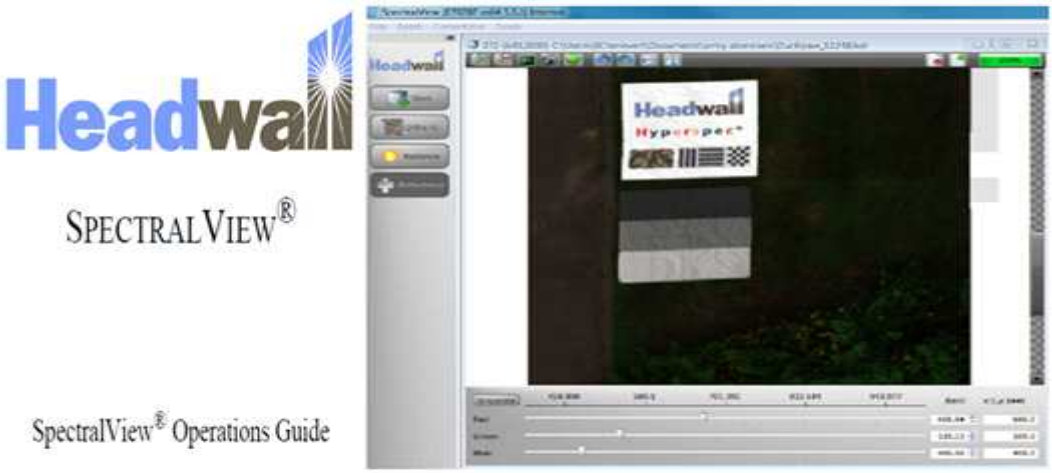
- 드론 초분광시스템 운용 및 처리를 위한 SW

· (드론 자동운용 SW) 도로포장상태 평가를 위한 객관적인 도로 분광라이브러리 구축을 위해서는 안정적인 성과도출이 필수적으로 요구됨에 따라 안정적인 성과도출에 많이 활용되어지고 있는 'Ugcs'드론 운용 SW를 활용하여 드론 영상촬영 활용함



[드론 운용 SW(Ugcs-Universal Ground Control Software)]

· (초분광영상 처리 SW) 기술개발된 드론 초분광시스템의 데이터는 제작사의 독자적인 파일형식인 큐브 형식으로 취득되어짐에 따라 최종성과물로 구축하기 위해서는 기하보정, 방사보정, 정사보정 과정이 요구되며, 이러한 전처리 과정은 제작사에서 제공하는 SpectralView SW를 통해 초분광영상을 구축함



[초분광영상 처리 SW(SpectralView)]

· (초분광영상 가공 SW) QGIS는 오픈소스 기반의 GIS 소프트웨어로 Widndow, Linux, Unix 등에서 실행되며 다양한 공간정보 포맷(vector, raster, database)을 지원한다. 또한 데이터 보기, 생성, 편지, 관리, 내보내기 등의 기능을 지원하여 영상데이터의 시각화, 데이터 분석에 활용할 수 있으며, SW를 통해 기존 PES장비로부터 산출되는 PCI(10m 단위)와 드론 초분광시스템 토대로 산출된 PCI 비교 분석을 위해 초분광영상 데이터 가공에 활용함



[초분과영상 가공을 위한 SW(QGIS)]

○ PCI 추정 평가/검증 계획서 및 평가분석서

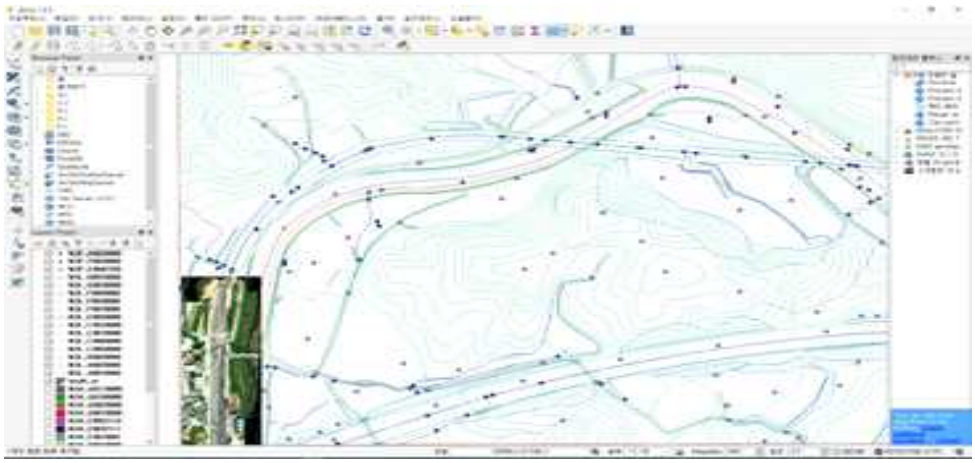
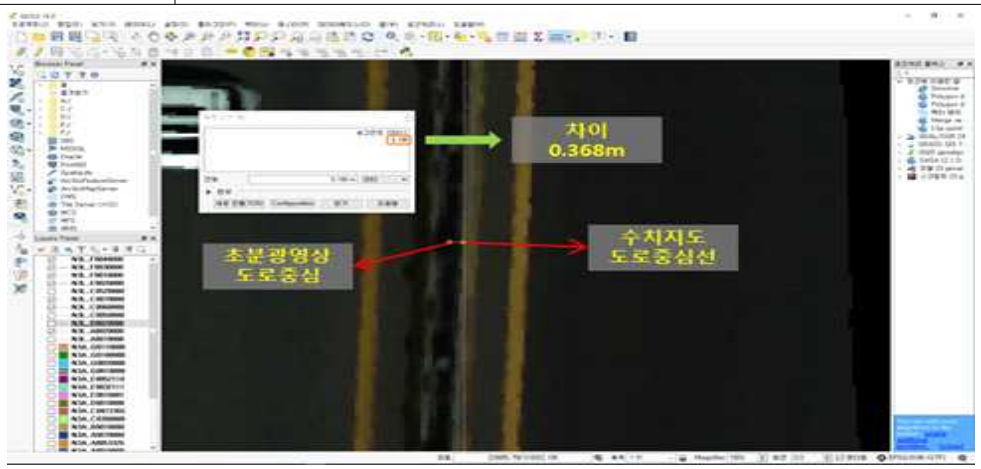
- 조사 구간 결정 및 드론 초분광 아스팔트 포장상태 조사 매뉴얼(부록 7) 방법에 준하여 드론을 이용한 초분광 분석
- 도로포장 조사 차량(PES)를 이용하여 조사 및 NHPCI 산출
- 도로 조사구간의 결과를 10m 단위로 분석하여 드론 초분광 평가결과와 조사차량 평가결과의 상관계수 산출
- 딥러닝 모델로 산출된 도로포장 평가 결과는 조사차량으로 평가된 포장평가지수(PCI)와 상관계수가 0.769이었으며, 개발 목표인 0.7 이상을 만족하였음

[조사차량 및 드론 초분광 아스팔트 포장 평가 결과]

No.	IRI (m/km)			rutting (mm)			균열 (%)			PCI		
	조사차량	초분광	편차	조사차량	초분광	편차	조사차량	초분광	편차	조사차량	초분광	편차
1	0.780	1.044	0.264	4.400	4.686	0.286	0.000	0.000	0.000	0.890	1.133	0.243
2	1.200	1.040	0.160	4.200	4.778	0.578	0.000	0.000	0.000	1.380	1.243	0.137
3	1.840	1.108	0.732	4.500	4.643	0.143	0.000	0.000	0.000	2.270	1.378	0.892
4	1.370	1.164	0.206	3.800	4.344	0.544	0.000	0.000	0.000	1.470	1.300	0.170
5	0.970	1.060	0.090	3.800	4.375	0.575	0.000	0.000	0.000	1.000	1.045	0.045
6	0.840	1.091	0.251	4.000	4.382	0.382	0.000	0.000	0.000	0.840	1.053	0.213
7	0.940	1.018	0.078	5.200	5.136	0.064	0.000	0.000	0.000	1.020	1.086	0.066
8	1.040	1.275	0.235	4.200	5.903	1.703	0.000	1.779	1.779	1.330	1.415	0.085
9	1.070	1.239	0.169	4.100	6.218	2.118	0.000	2.979	2.979	1.080	1.388	0.308
10	1.510	1.484	0.026	4.600	4.940	0.340	0.000	0.000	0.000	1.700	1.324	0.376
11	1.620	1.299	0.321	4.800	4.361	0.439	0.000	0.000	0.000	1.630	1.305	0.325
12	1.110	1.094	0.016	4.000	4.100	0.100	0.000	0.000	0.000	1.190	1.179	0.011
13	1.440	1.088	0.352	4.300	4.702	0.402	0.000	0.000	0.000	1.990	1.265	0.725
14	1.390	1.425	0.035	4.300	4.396	0.096	0.000	0.000	0.000	1.720	1.552	0.168
15	0.930	1.054	0.124	5.700	4.920	0.780	0.000	0.000	0.000	1.080	1.344	0.264
16	1.540	1.362	0.178	4.700	4.791	0.091	0.000	0.000	0.000	1.580	1.552	0.028
17	1.100	1.201	0.101	4.200	4.603	0.403	0.000	0.000	0.000	1.320	1.434	0.114
18	1.460	1.522	0.062	4.800	4.613	0.187	0.000	0.000	0.000	1.960	2.015	0.055
19	1.020	1.011	0.009	4.400	4.411	0.011	0.000	0.000	0.000	1.300	1.308	0.008
20	1.630	1.394	0.236	4.700	4.613	0.087	0.000	0.000	0.000	1.950	1.753	0.197
21	1.590	1.576	0.014	4.300	4.552	0.252	0.000	0.000	0.000	2.000	2.006	0.006
22	2.390	1.779	0.611	4.200	4.399	0.199	0.000	0.000	0.000	2.560	2.559	0.001
23	1.430	1.301	0.129	4.500	4.416	0.084	0.000	0.000	0.000	1.510	1.337	0.173
24	1.100	1.142	0.042	4.700	4.743	0.043	0.000	0.000	0.000	1.140	1.222	0.082
25	0.820	0.800	0.020	4.300	4.335	0.035	0.000	0.000	0.000	0.890	0.897	0.007
26	0.700	0.642	0.058	5.100	5.330	0.230	0.000	0.000	0.000	0.890	0.836	0.054
27	0.810	0.781	0.029	4.000	4.088	0.088	0.000	0.000	0.000	0.940	0.926	0.014
28	0.870	0.842	0.028	5.000	4.990	0.010	0.000	0.000	0.000	1.350	1.382	0.032
29	1.210	1.490	0.280	5.100	6.127	1.027	0.000	0.000	0.000	1.420	2.110	0.690
30	1.390	1.662	0.272	5.400	5.561	0.161	0.000	0.000	0.000	1.640	1.543	0.097
31	1.320	1.362	0.042	4.000	5.158	1.158	0.000	0.000	0.000	1.590	1.932	0.342
32	1.240	1.403	0.163	4.600	4.992	0.392	0.000	0.000	0.000	1.490	1.829	0.339
33	1.560	1.263	0.297	4.300	4.852	0.552	0.000	0.000	0.000	1.570	1.591	0.021
34	1.600	1.601	0.001	4.900	4.830	0.070	0.000	0.000	0.000	2.250	2.253	0.003
35	1.010	1.041	0.031	4.500	4.690	0.190	0.000	0.000	0.000	1.120	1.391	0.271
36	0.930	1.128	0.198	5.000	4.723	0.277	0.000	0.000	0.000	1.300	1.420	0.120
37	0.720	0.935	0.215	4.700	4.726	0.026	0.000	0.000	0.000	0.720	0.863	0.143
38	0.580	0.619	0.039	4.300	4.171	0.129	0.000	0.000	0.000	0.610	0.520	0.090
39	0.790	0.969	0.179	4.900	4.586	0.314	0.000	0.000	0.000	0.850	0.917	0.067
40	1.010	1.001	0.009	5.600	5.563	0.037	0.000	0.000	0.000	1.120	1.106	0.014
41	1.120	1.122	0.002	5.400	5.387	0.013	0.000	0.000	0.000	1.260	1.260	0.000
42	0.800	0.977	0.177	4.800	4.704	0.096	0.000	0.000	0.000	0.970	1.220	0.250
43	0.710	1.052	0.342	4.800	4.684	0.116	0.000	0.000	0.000	0.790	1.299	0.509
44	0.840	0.871	0.031	5.500	5.161	0.339	0.000	0.000	0.000	0.870	0.968	0.098
45	1.160	1.216	0.056	5.300	5.017	0.283	0.000	0.000	0.000	1.400	1.369	0.031
46	1.220	1.121	0.099	5.800	5.037	0.763	0.000	0.000	0.000	1.230	1.359	0.129
47	0.660	1.024	0.364	4.400	4.588	0.188	0.000	0.000	0.000	0.700	1.219	0.519
48	0.960	1.074	0.114	5.200	4.722	0.478	0.000	0.000	0.000	1.260	1.261	0.001
49	0.970	1.162	0.192	4.900	4.416	0.484	0.000	0.000	0.000	1.040	1.221	0.181
50	1.090	1.229	0.139	4.700	4.417	0.283	0.000	0.000	0.000	1.290	1.306	0.016
51	1.330	1.249	0.081	4.700	4.682	0.018	0.000	0.000	0.000	1.430	1.379	0.051
52	1.320	1.147	0.173	5.200	5.056	0.144	0.000	0.000	0.000	1.430	1.390	0.040
53	0.780	1.234	0.454	6.100	5.144	0.956	0.000	0.000	0.000	1.000	1.420	0.420
54	1.570	1.226	0.344	6.400	5.141	1.259	0.000	0.000	0.000	1.870	1.508	0.362
55	1.960	1.319	0.641	5.600	5.593	0.007	0.000	0.000	0.000	2.110	1.601	0.509
56	1.610	1.311	0.299	7.600	6.328	1.272	0.000	0.000	0.000	1.850	1.318	0.532
상관계수	0.760			0.568			-			0.769		

○ PCI 측정 위치 가시화 정확도 분석서

- 드론초분광시스템 기반으로 취득한 초분광영상의 PCI 측정을 위한 위치정확도 검증을 위해 국토정보플랫폼에서 제공되어지는 국가기본도(1:5,000)를 취득하였으며, 이를 통해 초분광영상의 위치 정확도는 평가함
- 평가는 수치지도의 도로중심선(N3L_A0020000), 인도(도로-인도 경계) 경계선(N3L_A0033320) 레이어를 통해 위치 정확도 검증을 수행함
- 이를 통해 1차년도 구축된 초분광영상의 위치정확도는 약 0.368m의 수준으로 본 과제 PCI 측정을 위한 위치정확도 최종목표치인 0.5m 이내로 목표를 달성하였음

<table>
<tr><td colspan="2">Overlap 수치지도-초분광영상</td></tr>
<tr><td>소재지</td><td>경기도 용인 양지면 추계리 53-2</td></tr>
<tr><td colspan="2">  </td></tr>
<tr><td>위치정확도 검증</td><td></td></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><td colspan="2">도로중심선(N3L_A0020000) 기준으로 초분광영상의 위치정확도는 0.368m 수준임</td></tr>
</table>

○ PCI 활용지침 요구사항 정의서

- 드론 초분광 아스팔트 포장상태 조사 시스템은 초분광센서, 드론, IMU/GPS, 데이터 처리 SW로 구성
- 초분광 카메라는 미국 Headwall사의 Nano-HyperSpec이 표준이며, 드론에 탑재가 가능하여야 하고, 400~1,000 nm 파장대를 촬영할 수 있어야 함.
- 드론은 초분광센서를 탑재하고 안정적인 운영이 가능하여야 하며, 최대 이륙중량이 15.5kg인 DJI사의 Matrice 600 PRO를 표준
- 드론에는 짐벌을 설치하여 초분광센서가 안정적으로 촬영할 수 있어야 하며, GPS/IMU 시스템를 탑재하여야 함
- 드론 초분광시스템 운용 소프트웨어는 UgCS를 표준으로 함
- 드론 초분광영상 기하보정, 방사보정, 정사보정 등의 전처리를 하여야 하며, 일반적으로 제작

사에서 제공하는 Headwall사의 SpectralView 소프트웨어를 표준
- 드론 초분광영상 데이터 변환 소프트웨어는 영상데이터의 시각화, 데이터 분석을 위해 오픈소스 기반의 QGIS를 표준
- 드론 초분광영상을 이용한 포장상태분석 소프트웨어인 초분광 포장상태분석 프로그램(HSPCA 1.0)을 개발하여 텍스트(CSV)로 변환 및 분석
- HSPCA (Hyper-spectral Pavement Condition Analysis)는 딥러닝을 이용한 도로인식과 차량, 차선 등의 데이터 제거, 포장상태 분석 등을 수행
- 드론 포장상태분석 가시화 시스템은 분석 결과를 기반으로 PCI 데이터 산출 및 시각화하는 서비스를 제공
- 운용자는 초분광 영상 분석 자료를 실시간 업로드하여 Vworld 기반 지도 레이어에 위치정보를 가시화하고, 10m 구간별 포장상태 분석자료 조회 수행

<2차년도 성과점검기준표>

핵심성과 (CTE)		단위성과		기관	성과점검기준						목표 달성도
					연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
1	SAR 센서 기반 비탈면 변위 측정 드론시스템	1-1	비탈면 변위 측정을 위한 SAR(L 밴드) 센서 장착 드론	메타빌드 (공동)	②	비탈면 측정 드론 임무계획 소프트웨어	1식	소프트웨어	소프트웨어 개발 확인	2.5	달성 완료
		1-2	드론 SAR 데이터 기반 비탈면 변위 측정 소프트웨어	부산대 (공동)	②	변위측정 정확도	10cm	측량 기준값 수집	기준값과 변위 비교분석	3.5	달성 완료
		1-3	비탈면 변위 측정 및 가시화 시스템	부산대 (공동) 메타빌드 (공동)	②	비탈면 변위위치 가시화 정확도	3m	측량 기준값 수집	기준값과 가시화 위치 비교분석	1.0	달성 완료
		1-4	평가/검증 보고서	롯데건설 (공동)	②	비탈면 가시화 평가/검증 계획서	1건	계획서	계획서 작성 확인	0.6	달성 완료
		1-5	비탈면 변위 가시화 시스템 활용 지침	롯데건설 (공동)	②	기존 비탈면 지침 조사 분석서	1건	분석서	분석서 작성 확인	0.6	달성 완료
2	영상 센서 기반 포트홀 사전/사후 탐지 드론시스템	2-1	포트홀 탐지를 위한 영상 센서 장착 드론	공간정보 (공동)	②	포트홀 탐지 포장 영상 DB 구축	100km	포장 영상 DB	포장 영상 DB 구축 확인	2.1	달성 완료
		2-2	드론 영상 데이터 기반 포트홀 측정 소프트웨어	건설연 (주관)	②	검지율 및 오검지율	검지 : 70% 오검지 : 20%	기준값 수집	기준값과 비교분석	3.9	달성 완료
		2-3	포트홀 위치 가시화 시스템	공간정보 (공동) 아와소프트 (공동)	②	포트홀 위치 가시화 정확도	1m	DGPS 기준값 수집	기준값과 비교분석	0.6	달성 완료
		2-4	평가/검증 보고서	건설연 (주관)	②	포트홀 탐지 평가/검증 보고서	1건	평가 보고서	평가 보고서 작성 확인	2.3	달성 완료
		2-5	포토홀 탐지 및 가시화 시스템 활용 지침	건설연 (주관)	②	기존 포트홀 지침 조사 분석서	1건	분석서	분석서 작성 확인	1.4	달성 완료

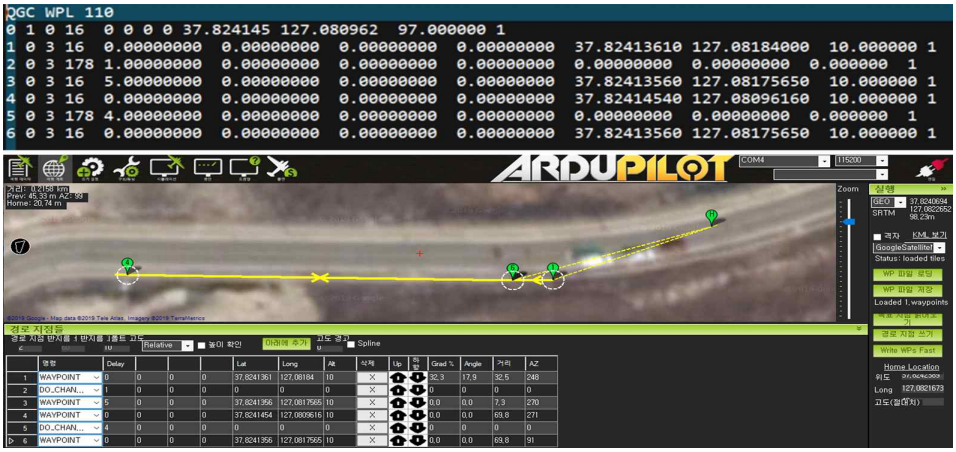
핵심성과 (CTE)		단위성과		기관	성과점검기준						목표 달성도
					연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
3	초분광 센서 기반 포장상태지수(PCI) 측정 드론시스템	3-1	PCI 측정을 위한 초분광 센서 장착 드론	신한항업 (공동)	②	PCI 측정 포장 초분광 DB 구축	100 km	포장 초분광 DB	포장 초분광 DB 구축 확인	2.3	달성 완료
		3-2	초분광 데이터 기반 PCI 추정 소프트웨어	건설연 (주관)	②	기준값 대비 PCI 측정 상관계수	0.6	PCI 조사차량 이용 기준값 수집	기준값과 상관계수 비교평가	3.9	달성 완료
		3-3	PCI 가시화 시스템	신한항업 (공동) 아와소프트 (공동)	②	PCI 측정 위치 가시화 정확도	1m	DGPS 기준값 수집	기준값과 가시화 위치 비교분석	0.6	달성 완료
		3-4	평가/검증 보고서	건설연 (주관)	②	PCI 추정치 평가/검증 보고서	1건	평가 보고서	평가 보고서 작성 확인	1.7	달성 완료 (1차년도 완료)
		3-5	PCI 측정 및 가시화 시스템 활용 지침	건설연 (주관)	②	기존 PCI 지침 조사 분석서	1건	분석서	분석서 작성 확인	1.4	달성 완료

○ 비탈면 측정 드론 임무계획 소프트웨어

- 비탈면 변위 측정을 위한 드론 제어 명령이 포함된 임무계획

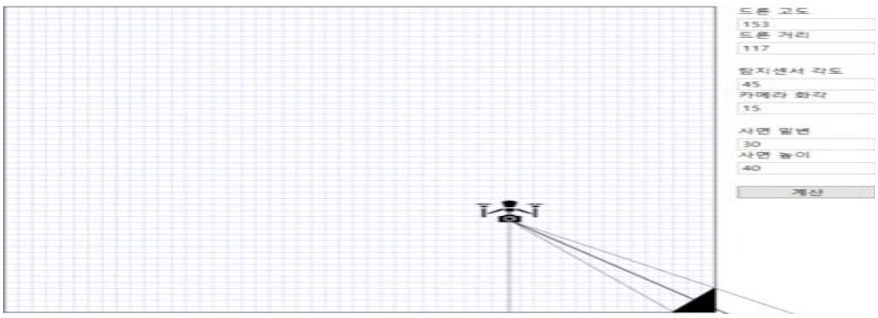
① TAKEOFF, LAND, RTH, GOTO, Mission UPLOAD, Mission START

- 시험용 드론 기반 3D 임무 비행포인트 시험비행 비행계획서



[임무비행포인트 비행계획 예시]

- 고정밀 항법센서 및 고도센서 기반 비탈면 DEM 생성 및 비행계획서에 반영
- 3D 임무비행포인트 산출 검증 시뮬레이터



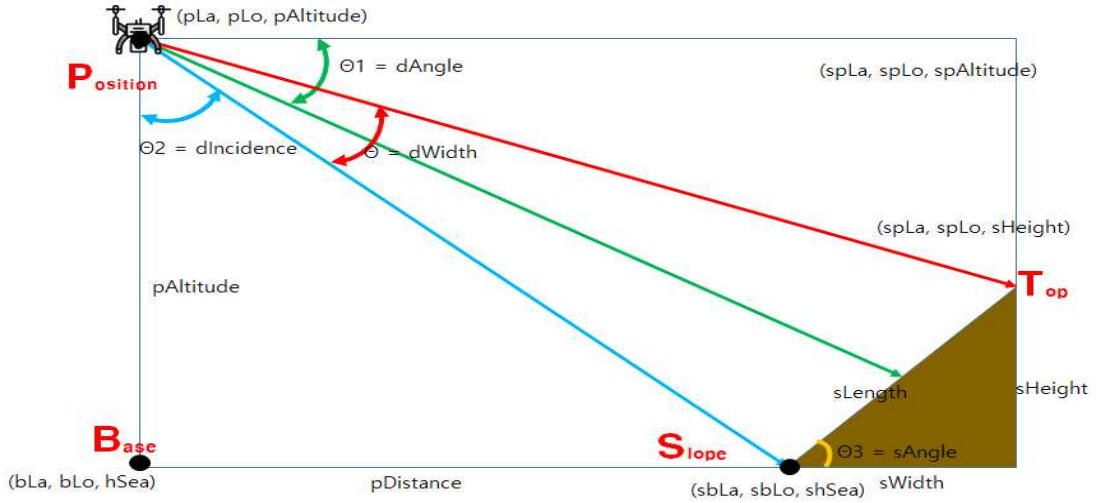
[시뮬레이션 실행 화면]

- 3D 임무비행포인트 산출 및 임부비행데이터 분석 로직



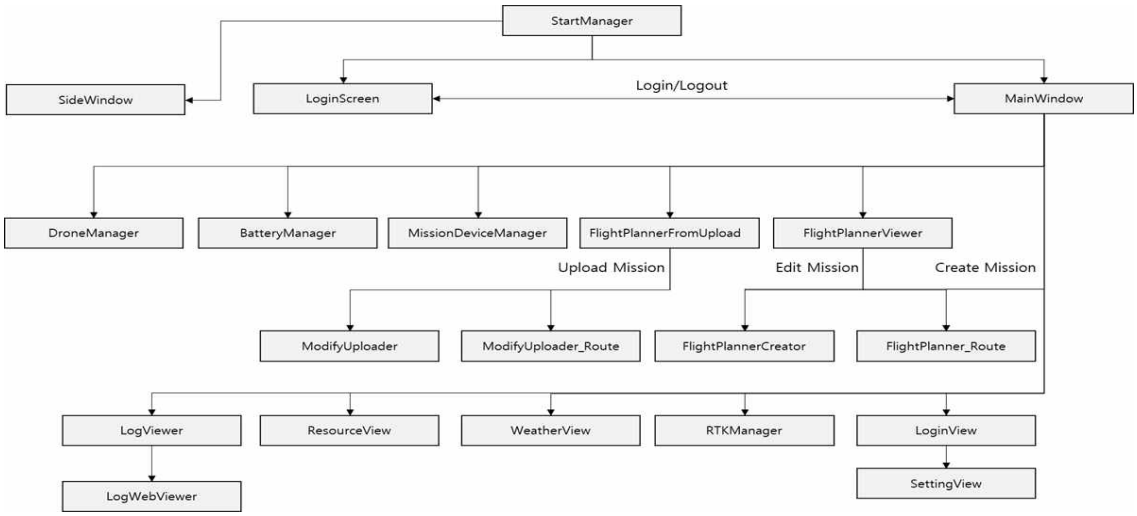
[분석 로직 실행화면]

- 비탈면 측정을 위한 3D 임무비행포인트 생성



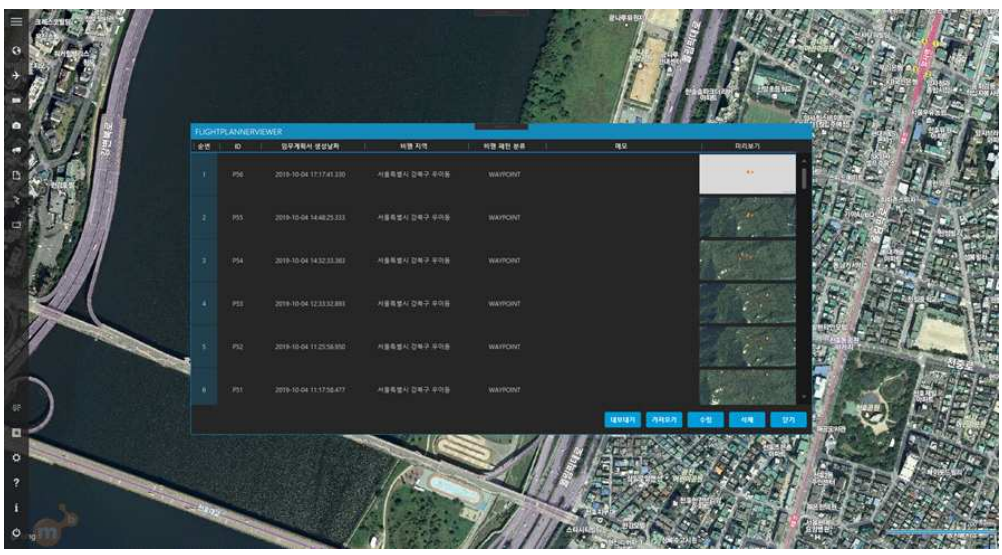
[3D 임무비행포인트 생성 설계]

- 자동운항 기반 임무비행계획 UI 구조



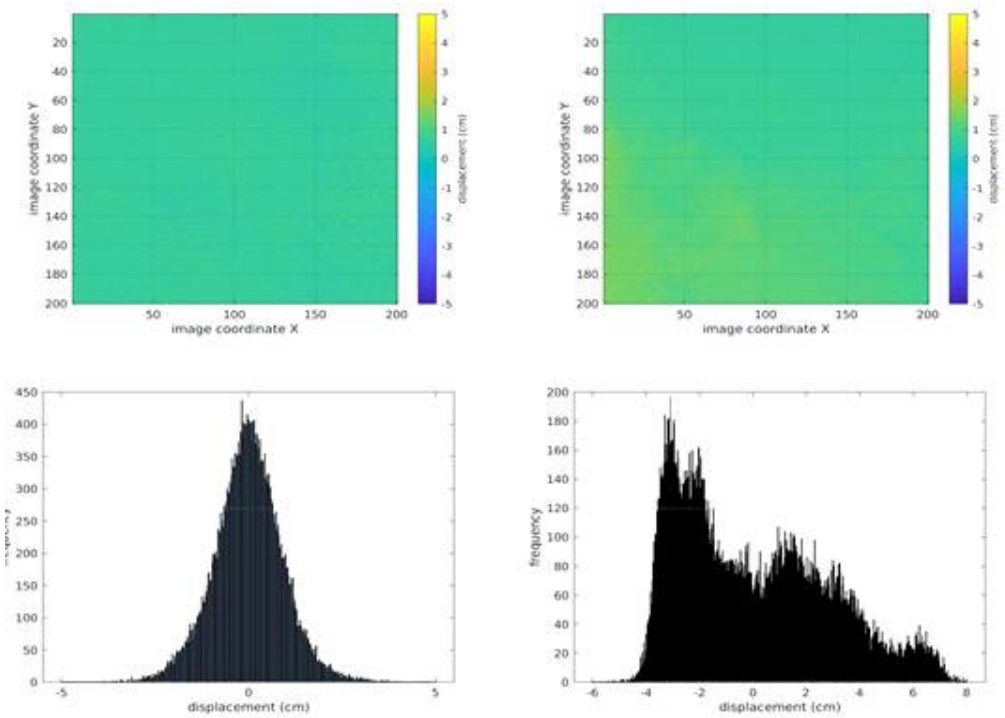
[자동운항 기반 임무비행계획 UI 구조도]

- 임무비행계획서 관리 기능



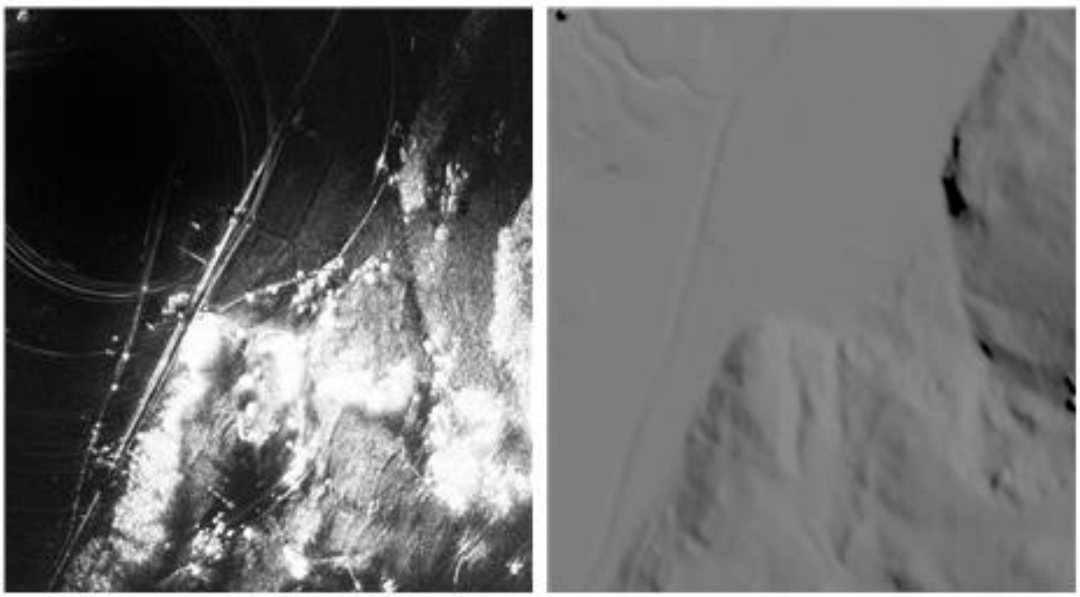
[작성된 비탈면 비행임무계획서의 조회 및 수정 기능]

○ 비탈면 변위 및 가시화 검증계획서 및 평가분석서
- 2차년도는 스위스 현지 실험 관측을 통해 성과목표에 대한 검증을 실시하였음
- 변위 측정 정확도 분석 (평균 0.7-1.0 cm, 편차 0.2-0.5 cm)



[두 지역 변위 측정치 분포도(상), 변위분포 히스토그램 (하)]

- L-밴드 파장인 23 cm을 적용하여 얻은 시험 변위의 평균과 표준편차 등 통계 계산 분석
- 최종 개발목표치인 5 cm 변위 측정 정확도 요구 성능을 만족하는 결과를 조기에 달성함
- 변위위치 정확도 분석 (평균: 0.975 (range: 1.1-1.4 m, azimuth: 0.4-1.0 m))
- 향상된 공간해상도를 가진 DEM을 이용하여 정사영상을 제작함으로써 변위위치 정확도 향상 기술 개발 진행하였음
- 생성된 정사 영상은 약 1m 내외로 개발 목표치를 만족하며 성과목표 달성 완료함



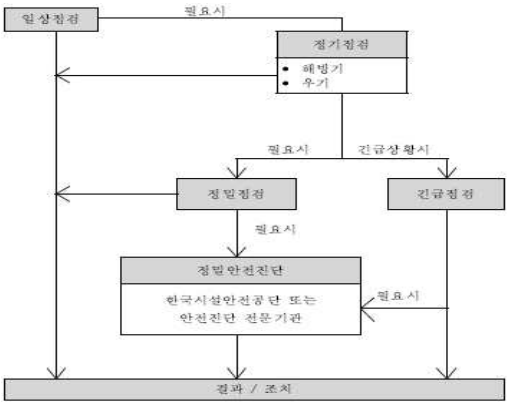
[관측된 영상레이더와 지형표고모형과의 위치정확도 분석]

○ 기존 비탈면 지침 조사 분석서

- 절토사면 유지관리 매뉴얼 보완 · 개정 연구(2013.12), 한국시설안전공단
  · 지표면 설치된 계측기기로 지표 변위량 계측으로 설치 위치, 수량 등 계측관리 한계가 있음
  · 전문가 육안조사 시 접근 불가 위치, 소요 시간, 전문 인력 등 점검 한계 있음
  · 급경사 또는 대절토 비탈면의 경우 식생, 경사에 의해 비탈면 지표면, 비탈면 상단 측량이 불가함

4.4 절토사면의 계측관리

: 지표 변위량은 지표면 신축계, 지표면 경사계, 광파측량기 등



· 제5장 절토사면의 점검 및 진단 : 정기점검, 정밀점검, 긴급점검, 정밀안전진단으로 나뉨

: 정기점검 - 반기에 1회 이상 실시, 세심한 육안검사 수준의 점검으로서 간단한 점검장비(카메라, 필리도구, 줄자, 망치 등)

: 정밀점검, 긴급점검, 정밀안전진단 : 근접조사를 위한 접근장비 사용 가능, 측량기 사용, 외관조사

5.3 현장조사 및 시험법

: 절토사면 측량 : 외관 조사를 실시한 결과, 상태 평가 또는 안정성 평가를 위해 정밀한 사면형성이 필요한 경우 사면에 대한 측량을 실시

: 광파거리계에 의한 측량기를 사용하며 측정간격은 20~100m 범위에서 사면의 형상이나 손상정도에 따라 위치를 선정하여 측량

: 측점별 횡단 측량은 보통 20m마다 1개소씩 측량하며, 단면 변화구간 및 육안조사 단면의 이상부위는 보다 정확한 측량을 실시

- 건설공사 비탈면 설계기준(2011.12), 국토해양부
  · 지상 계측, 계측기기에 의한 계측 한계에 대한 언급이 없지만 계측목적에 부합하는 계측기 선정이 중요함을 언급함

- 철도설계지침 및 편람 : KR C-02050 계측 (2017.3), 한국철도시설공단
  · 지표면에 설치된 계측기기로 계측, 사진 촬영 측정 등 설치 위치, 수량 등 계측 관리 한계가 있으며, 식생에 의한 육안, 지표 계측 관리 한계도 있음

- 비탈면 시공 또는 유지관리시 기존 방법으로는 육안조사, 지표면에 설치된 계측기기 지점만 점검, 계측 관리되는 한계점을 가지고 있음. 본 연구 기술로는 대절토 비탈면, 급경사지, 식생이 많은 곳, 계측기가 설치되지 않은 지점 등에서 발생되는 지표면 변위를 감지하여 비탈면 붕괴 조짐을 사전에 알 수 있음

○ 포트홀 탐지 포장 영상 DB 구축 보고서
- 고정밀 RTK GPS가 탑재된 고정익드론 eBee Plus 활용
- 신속촬영이 가능한 회전익드론 Phantom4 Pro 활용
- 1차년도 연구성과인 포트홀 탐지를 위한 최적의 드론 비행방법 반영하여 해상도, 고도, 중복도, 경로 설정
- 해상도 2cm/px, 비행고도 73m, 중복도 70%, 비행속도 10~12m/s 경로 2Line

고정익 드론

SenseFly eBee Plus



구분	내용
날개길이	110cm
무게	약 1.1kg
최대비행시간	59분
수신범위	3km
자동착륙방식	선형착륙, 선회착륙 (정확도 5m)
경사이미지	0° - 50°
GSD	최소 1.5cm
GNSS	RTK & PPK Ready
카메라	20MP(5472x3648)
해상도(100m)	2.6cm/px

회전익 드론

DJI Phantom4 Pro



구분	내용
대각선길이	35cm
무게	약 1.3kg
최대비행시간	30분
수신범위	1km
자동착륙방식	수직착륙 (정확도 2m)
경사이미지	-90° ~ +30°
GSD	최소 1mm이하
GNSS	DGPS
카메라	20MP(5472x3648)
해상도(100m)	2.7cm/px

- 포트홀 AI학습용 DB구축
  · 총연장 105km, 포트홀 영상 61개 확보



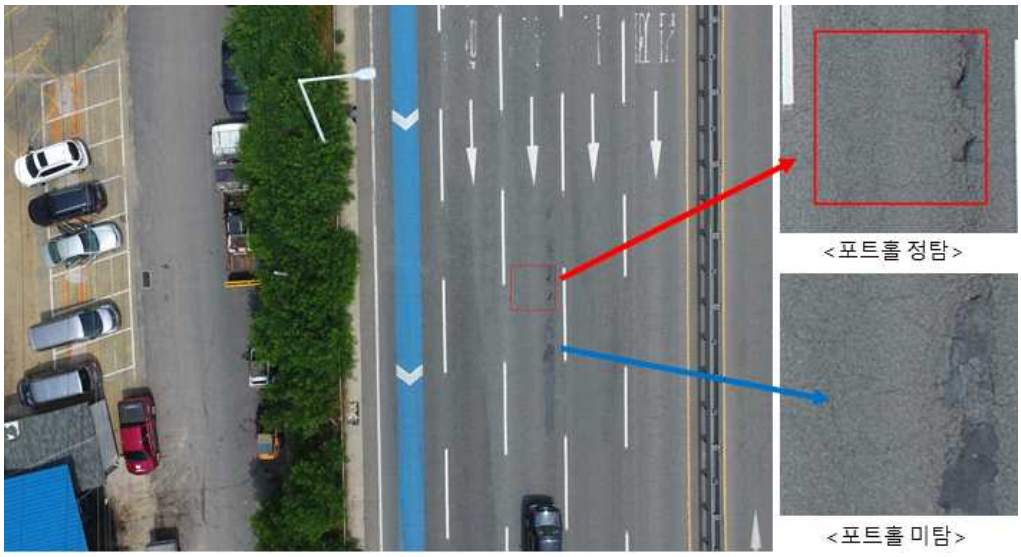
- 전남, 전북, 광주, 충북, 대전 등 주요국도에 대한 지상조사를 실시하여 포트홀 위치 파악
- 총 비행횟수 27번, 총 사진매수 2,523매, 포트홀 발생지 61곳에 대한 영상데이터 취득

○ 포트홀 탐지 검지율/오검지율 평가계획서 및 평가결과 분석서

- 포트홀 탐지 검지율/오검지율 평가계획 및 평가 결과
  · 포트홀 검지율 : 탐지 포트홀 수 / 포트홀 수 [TP/(TP+FN)]
  · 포트홀 오검지율 : 오탐지 횟수 / 포트홀 탐지 횟수 [FP/(TP+FP)]
  · 포트홀 검지율 : 78.78%
  · 포트홀 오검지율 : 18.30%



[항공 이미지 내 포트홀 정탐 및 오탐 샘플]



[항공 이미지 내 포트홀 정탐 샘플]

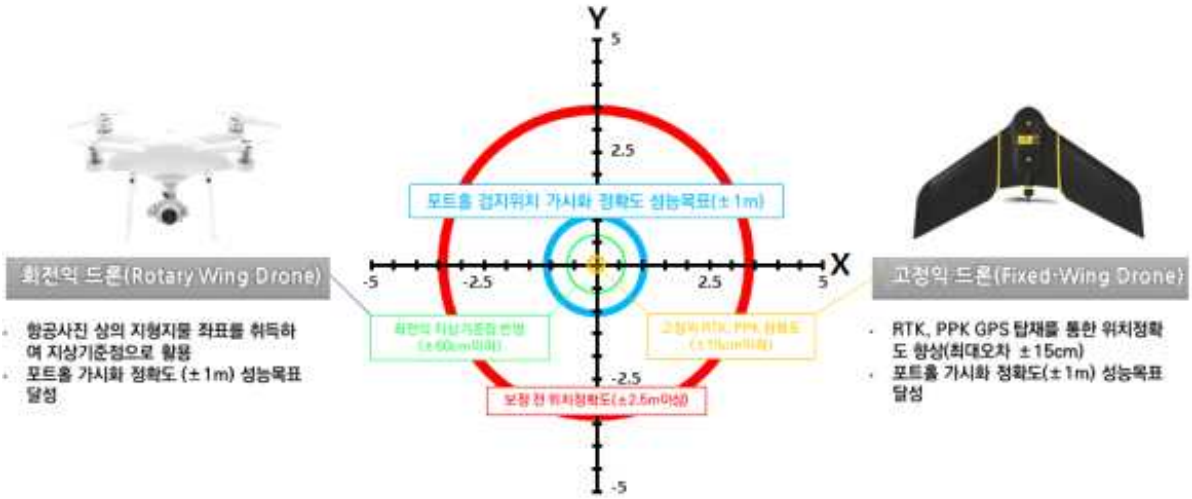
- 포트홀 탐지 검지율/오검지율 평가 결과 분석
  · 1차년도 정탐율 92% 및 오탐율 29.9%
  · 1차년도 대비 정탐율 20% 및 오탐율 11% 감소됨.
  · 포트홀 특징 강조를 위한 데이터 셋 검증 및 수정을 통한 오탐율 감소 확인
  · 1번, 2번 오탐 : (미세)크랙을 포트홀로 오탐
  · 3번 오탐 : 노후화된 도로 노면 표시를 포트홀로 오탐

· 4번 오탐 : 도로 소성변형 및 표면 오염을 포트홀로 오탐
· 5번 오탐 : 도로 노면 표시상 물체를 포트홀로 오탐



[2차년도 포트홀 오탐 사례]

○ 포트홀 위치 가시화 정확도
- 1차년도 가시화 정확도 평가 결과 RMSE 0.1m로 2차년도 목표치 달성
- 고정밀 GPS 탑재 드론, 지상기준점을 활용 포트홀 위치 정확도 향상
- 고정익드론 : RTK, PPK 기능 탑재 드론활용 위치정확도 향상(RMSE 0.1m)
- 회전익드론 : 지상기준점 사용 위치정확도 향상(RMSE 0.6m)



○ 기존 드론활용지침 조사 분석서
- 기존 드론활용지침을 분석하여 포트홀 조사방법 시나리오 작성
  1. 상황 : 수원시 00면 ~ 안성시 00동 00국도 포트홀 발생
  2. 현장에 대한 대략적인 상황을 드론운용팀에게 전파(현장 위성지도 일반국도 도면을 확인한 후 스마트기기에 저장 및 인쇄하여 준비)
  3. 취합된 내용을 토대로 현장점검팀과 함께 드론 운용가능지점을 결정
  4. 현장보수팀과 드론운용팀(장비운용팀) 현장으로 출동
  5. 현장도착 후 상급자 또는 현장책임자의 판단에 따라서 현장 상황을 종합적으로 분석 후 안전한 이 착륙장소 결정
  6. 세부적인 재해상황 판단을 위해 <드론을 이용한 포트홀 점검순서>에 의거 촬영 시작
  7. 포트홀의 전경이 사진 1장에 나오도록 드론의 고도 및 거리를 이격한 후 촬영 진행
  8. AI알고리즘 학습을위한 해상도 1cm이하로 촬영하여 포트홀 영상 수집
  9. 즉각적인 위치 및 크기를 파악하기위해 진행 방향으로 드론 운용
  10. 비행의 효율성과 빠른 작업을위해 운용팀과 보조팀으로 나누어 이륙장과 착륙장으로 배분
  11. 촬영이 완료되면 AI학습용 서버에 업로드
  12. 포트홀 상세 분석을 위해 매핑을 해야 하는 경우, <매핑을 위한 드론 촬영 순서>에 따라 촬영
  13. 1차 촬영이 끝난 후 작업팀 전체가 모여서 촬영된 데이터를 분석하고 추가촬영 및

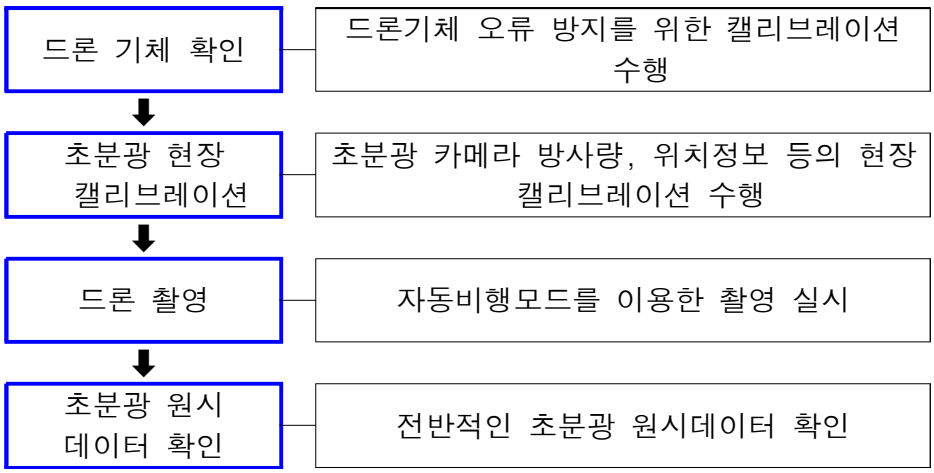
국토관리사무소에 실시간으로 상황을 보고하고 한국건설기술연구원에 중요 사진을 취합하여 전송

○ PCI 측정 포장 초분광 DB 구축

- 드론 탑재형 초분광 시스템 촬영

· 드론 탑재형 초분광시스템을 통해 대상지의 정확한 분광라이브러리를 구축을 위해 촬영공정에 따라 촬영이 진행 함

[드론 초분광시스템 촬영공정]



· 드론 기체 확인은 드론 운영 시 조정자의 과실이 아닌 드론기체의 오작동으로 인해 발생하는 사고를 방지하고 높은 품질의 성과를 취득하기 위해 현장에서 수행하는 캘리브레이션 단계임

· 특히 드론기체가 처음 운영하는 대상지역에서는 현장에서 드론 짐벌, IMU, 콤파스 캘리브레이션을 수행 후 촬영을 진행함

· 초분광 현장 캘리브레이션은 센서 특성상 다른 RGB, 열화상 센서와 달리 현장 캘리브레이션을 실시하여 초분광센서의 영상처리 과정에서 방사보정과 기하보정을 위한 필수 과정으로 영상 품질과 직결되는 문제이므로 철저히 실시해야함

[초분광센서 현장 캘리브레이션 항목]

항목	점검사항	비고
하이퍼 전용 GPS 캘리브레이션	전용 SW(Spectral III)를 이용한 GPS 연결상태 확인	Spectral III 이용
하이퍼 전용 IMU 캘리브레이션	수평축, 수직축 캘리브레이션 실시	Spectral III 이용
Black 밸런스 점검	카메라 렌즈 덮개를 막고 Black 밸런스 점검	Spectral III 이용
White 밸런스 점검	카메라 렌즈를 풀고, 반사도 95% 순백지를 이용하여 점검 실시	Spectral III 이용

· 드론 촬영 단계에서는 양질의 초분광영상 취득을 위해 비행고도, 촬영시간, GPS 수신 등 다양한 요소를 파악하여 최적의 비행방안을 도출해야 함

· 비행고도는 약 70m일 때 20m의 촬영 폭으로 영상을 취득할 수 있으며 촬영에 활용되는 Nano-Hyperspec 센서의 분광범위는 400nm ~ 1,000nm이며 라인스캔방식으로 촬영을 진행함

· 초분광센서로 취득되는 분광데이터는 반사 및 방사와 같은 빛에너지를 취득하게 되는데, 이러한 과정에서 태양고도각에 의한 경면반사와 같이 대기 및 분광패턴에 영향을 미치는 요소

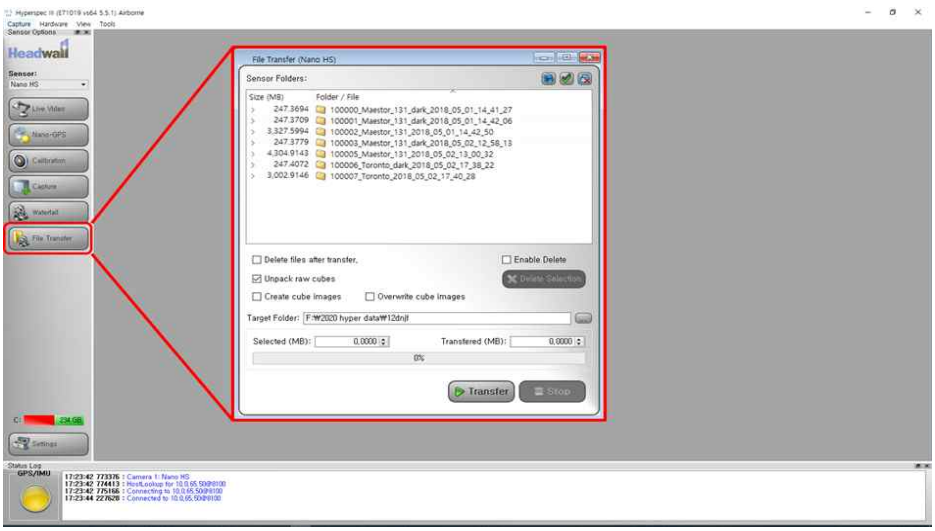
를 최소화하기 위하여 촬영 시 태양고도각이 65°이하가 되는 오전 9시 ~ 오전 11시, 오후 2시 ~ 오후 4시에 촬영을 수행함

· GPS 관측위성은 수신지역의 환경 및 시간의 변화에 따라 유동적으로 변화하여 GPSS/IMU데티어 품질에 많은 영향을 미치기 때문에 초분광 촬영을 단일코스로 계획하여 드론기체가 안정적으로 유지하여 양호한 위성 신호를 수신하며, 위성 고도 각은 15°이상, 위성개수는 7개 이상이 되도록 함

[드론 탑재형 초분광시스템 촬영 현장사진]



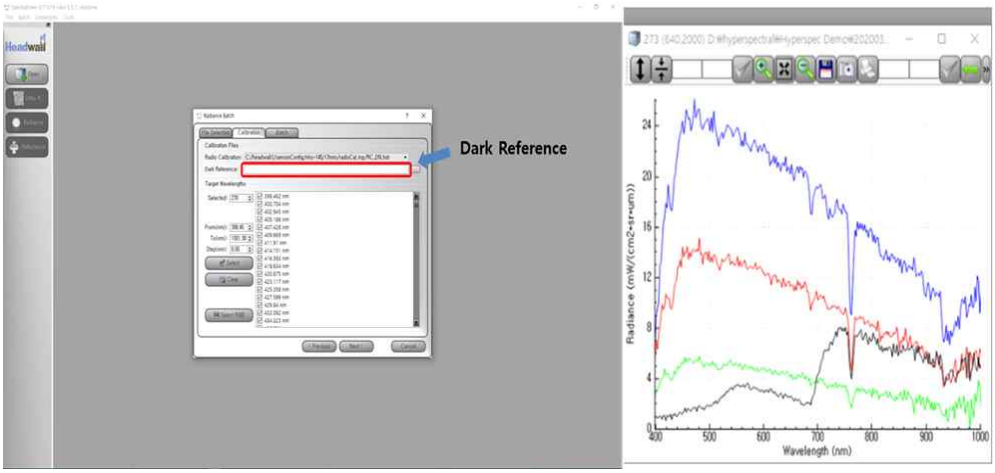
· 초분광 원시데이터 확인은 촬영 완료 후 HyperspecIII SW 활용하여 센서 자체스토리지에 있는 영상을 유선네트워크 연결을 통해 영상처리가 가능한 컴퓨터로 데이터를 옮긴 후 영상의 품질을 육안으로 검토하여 촬영계획과 동일하게 데이터가 취득되었는지를 확인하는 단계임



- 초분광 영상자료는 기존의 다중분광영상자료와 달리 수많은 분광밴드를 가지고 있기 때문에 보정을 거치지 않은 RAW 데이터의 경우 의미가 없는 Digital Number로 저장되어 있어, 이러한 무의미한 데이터를 분석에 활용하기 위해서 오차를 보정하는 방사보정, 기하보정, 대기보정 등 전처리 과정을 통해 의미있는 수치로 바꿔주어야 함

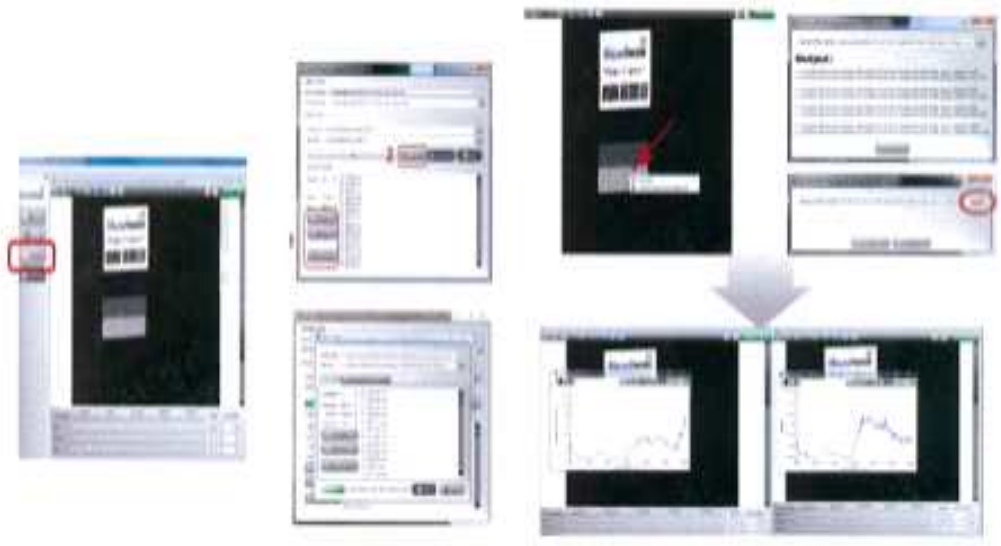


· (방사보정) 태양의 고도각, 탐측기의 응답 특성, 대기의 상태, 지형적 용인에 의한 영상의 밝기값 변화로 나타나는 왜곡을 보정하기 위해 실시하는 과정으로 Headwall社 소프웨어 Spectral View를 통해 초분광 원시데이터가 복사된 에너지의 양인 Radiance 값을 가지게 됨
· 방사보정은 현장에서 센서 캘리브레이션 수행 시 현장에서 White Reference 및 Dark Reference를 수행하게 되는데 이러한 현장의 환경이 적용된 데이터를 통해 방사 보정을 수행하였음



[방사보정을 통한 Radiance값 취득]

· (대기보정) 대기보정은 영상 내에 고정된 반사율을 가지는 물체 TRAF를 현장에 설치하고, 반사판이 영상에서 가지는 반사율을 기준으로 전체 영상을 정규화하여 반사율을 얻은 후 수행함
· 대기보정 반사율 식은 $\frac{(Raw\,data - Dark\,Reference)}{(White\,Reference - Dark\,Reference)}$ 를 이용하여 적용하였음



[초분광 영상 대기보정 과정]

· (기하보정) 초분광 데이터를 취득 할 때 지구와 드론 기체 및 센서의 특징으로 인한 기기적 한계로 인해 초분광데이터 영상의 공간적 왜곡이 발생하게 되며 이를 기하오차라고 한다. 이런 기하오차를 보정하기 위해서 위성 영상의 픽셀좌표와 지상의 지리좌표를 대응하여 초분광 데이터영상을 적걸한 실제 위치에 위치 할 수 있도록 프로세스를 진행해줘야 하는데 이를 기하보정이라고 함

· 기하보정 시 기복변위 및 기복으로 인한 반사광 변화를 고려해야 하므로 반드시 DEM을 이용해야하며 DEM의 정밀도가 높을수록 초분광영상의 품질에 좋은 영향을 미친다. 따라서 본 연구에서는 5m 해상도를 가지는 DEM을 적용하여 기하보정을 수행하였다. 또한 Headwall社의 Nano-Spectral 제품의 기하보정 시 GPSS/IMU Offsets 대상지역에 맞는 매개변수를 입력하여 영상 품질을 향상 시킬 수 있음



[초분광 영상 기하보정]

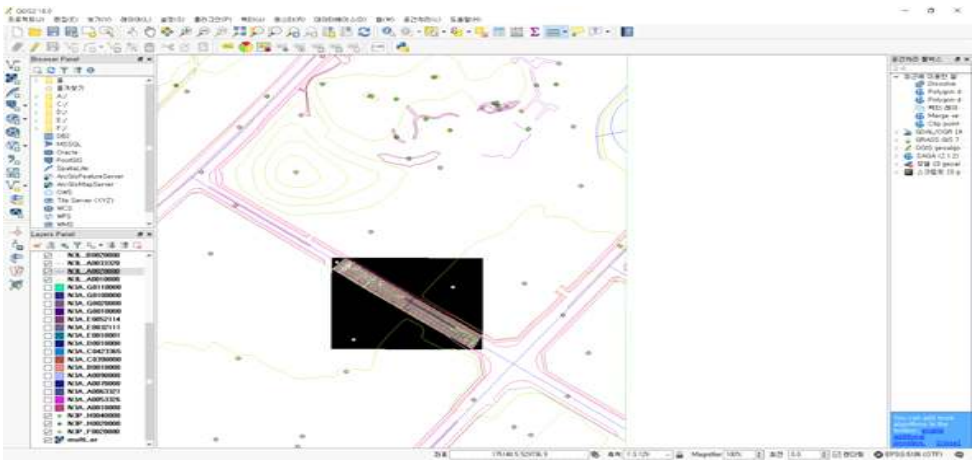
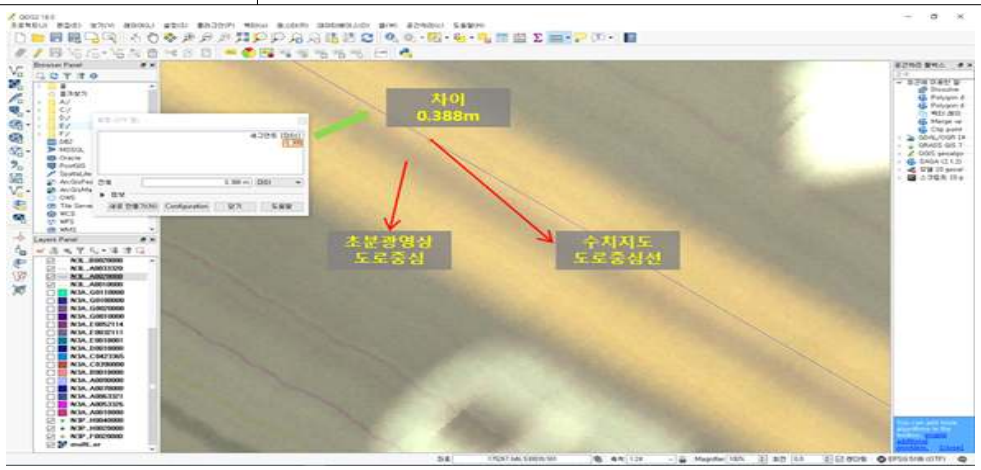
[드론시스템을 활용한 도로포장 초분광데이터 구축 현황]

No	대상지역	기존 PES DB현황	드론 초분광DB	촬영 횟수	비고
1	경기 용인 양지면 추계리 53-2	20.0 km	0.4 km	2회	PES DB 중 최적 구간 선정
2	경기 여주 하동 3-2	20.0 km	1.0 km	2회	상동
3	경기 여주 가남읍 심석리 100-1	20.0 km	0.5 km	2회	상동
4	경기 고양 일산서구 대화동 882	0.2 km	0.2 km	2회	상동
5	경기 시흥 정왕동	0.6 km	0.6 km	2회	상동
6	경기 안성 일죽면 월정리 719	10.0 km	0.4 km	2회	상동
7	경기 연천 국도 37호선 자장사거리	10.0 km	-	2회	비행승인 불가
8	인천 중구 신흥동3가 80	0.7 km	0.7 km	2회	PES DB 중 최적 구간 선정
9	인천 동구 송현동 146	0.5 km	0.5 km	2회	상동
10	경기 안성 삼죽면 진촌리 307-3(상행)	20.0 km	0.5 km	2회	상동
11	경기 안성 삼죽면 진촌리 307-3(하행)		0.5 km	2회	상동
12	경기 안성 죽산면 두현리 109-1		0.5 km	2회	상동
13	경기 평택 포승읍 도곡리 170-76	-	1.5 km	2회	상동
14	경기 화성 장안면 노진리 1416	-	3.3 km	1회	상동
합계		100.0 km	10.6 km	27회	도로 1차선 기준

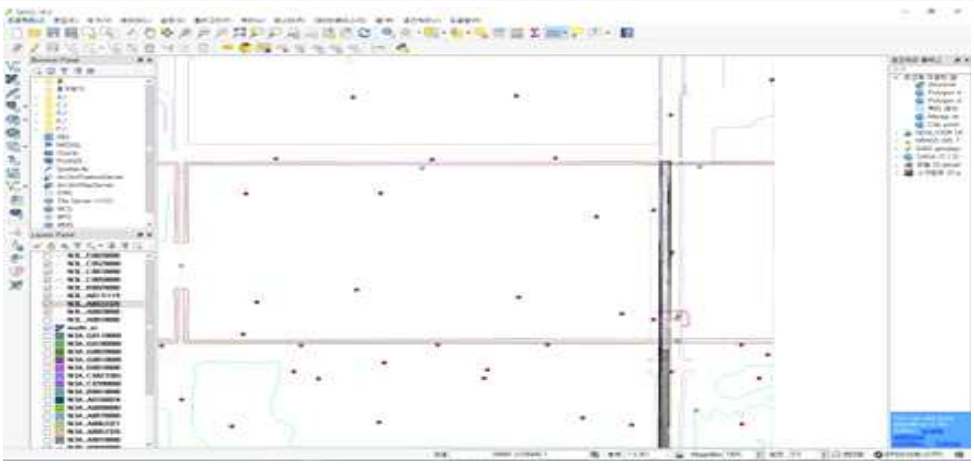
○ PCI 측정 위치 가시화 정확도 분석서

- 드론초분광시스템 기반으로 취득한 초분광영상의 PCI 측정을 위한 위치정확도 검증을 위해 국토정보플랫폼에서 제공되어지는 국가기본도(1:5,000)를 취득하였으며, 이를 통해 초분광영상의 위치 정확도는 평가함
- 평가는 수치지도의 도로중심선(N3L_A0020000), 인도(도로-인도 경계) 경계선(N3L_A0033320) 레이어를 통해 위치 정확도 검증을 수행함
- 이를 통해 2차년도 구축된 초분광영상의 위치정확도는 **약 0.359m**의 수준으로 본 과제 PCI 측정을 위한 위치정확도 최종목표치인 0.5m 이내로 목표를 달성하였음

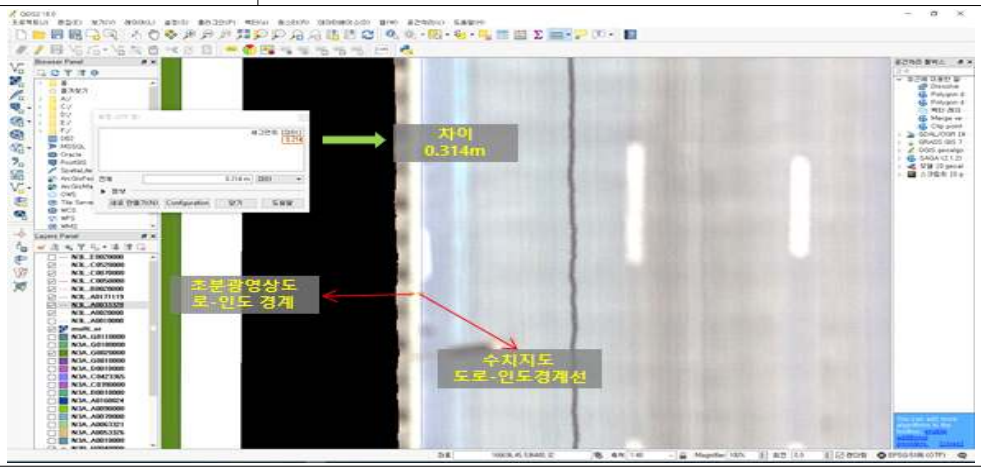
[경기도 시흥시 정왕동 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 시흥시 정왕동
	
위치정확도 검증	
	
도로중심선 기준 초분광영상의 위치정확도는 0.388m 수준임	

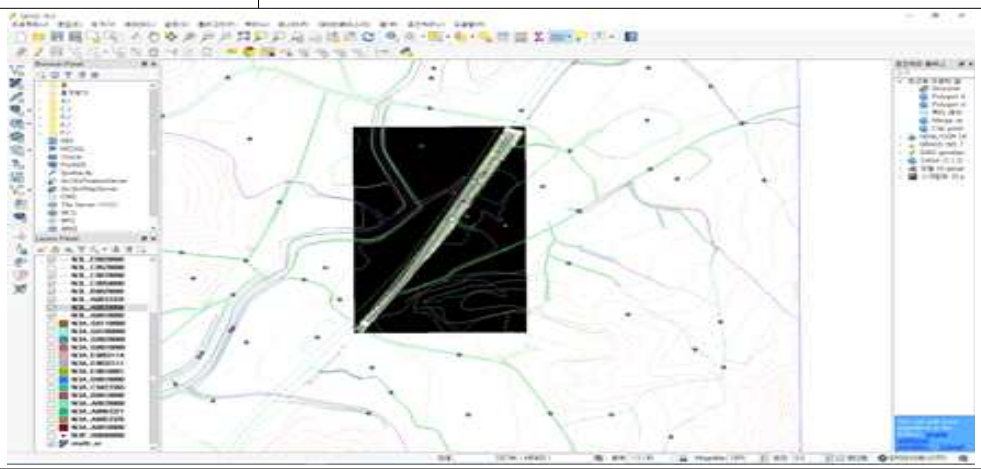
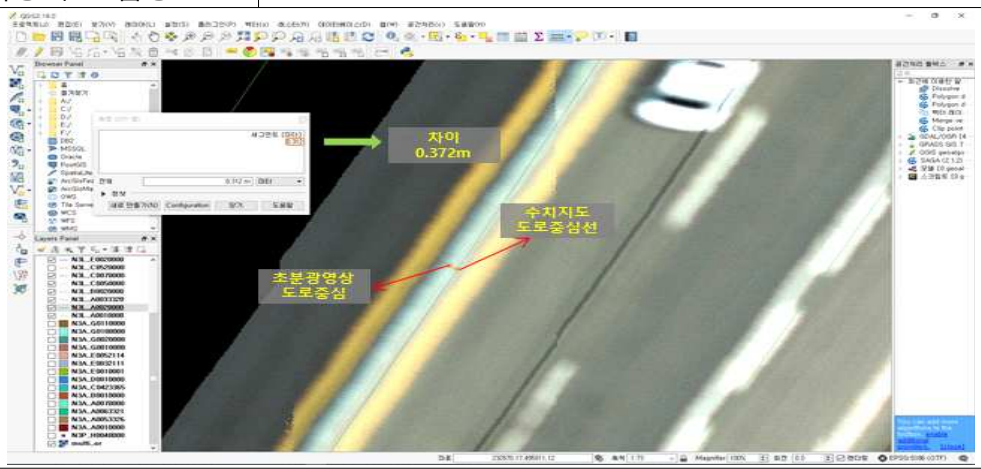
[인천광역시 중구 신흥동3가 80 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	인천광역시 중구 신흥동3가 80
	

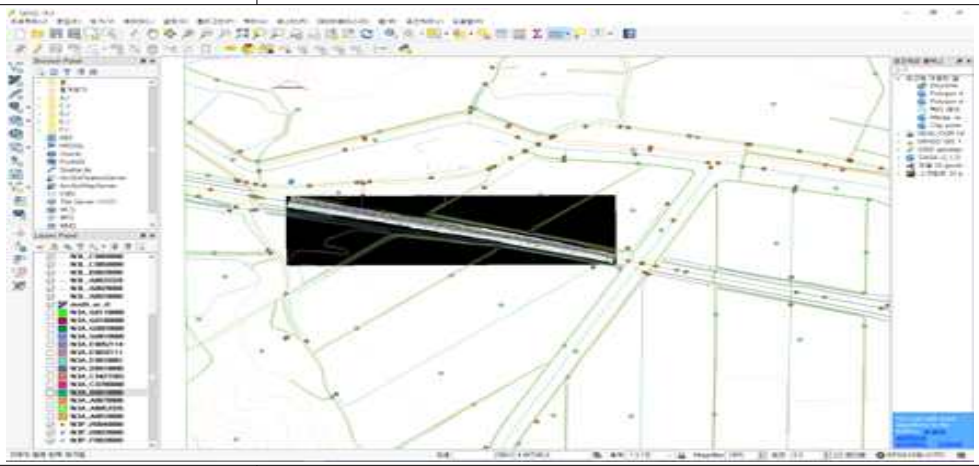
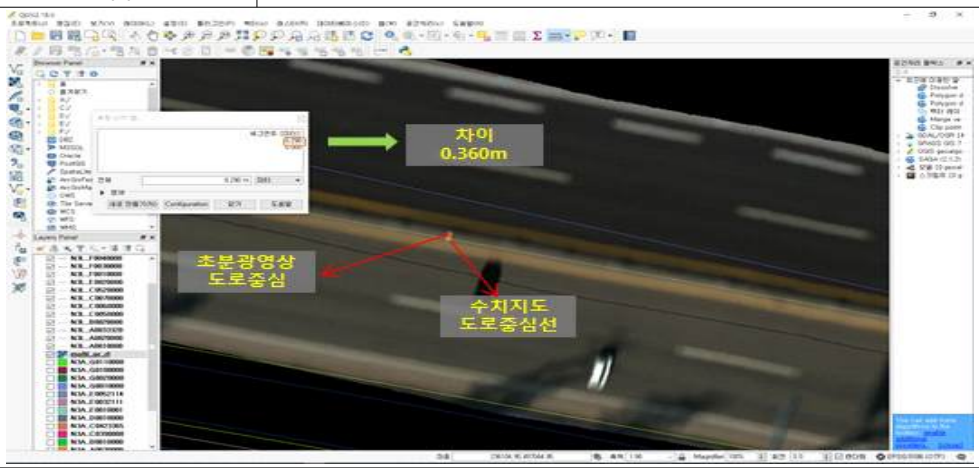
[인천광역시 중구 신흥동3가 80 위치 정확도 검증]

위치정확도 검증	
	
도로-인도 경계선(N3L_A0033320) 기준 초분광영상의 위치정확도는 0.314m 수준임	

[경기도 안성시 삼죽면 진촌리 307-3 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기 안성 삼죽면 진촌리 307-3
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준 초분광영상의 위치정확도는 0.372m 수준임	

[경기도 안성시 죽산면 두현리 109-1 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기 안성 죽산면 두현리 109-1
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준 초분광영상의 위치정확도는 0.360m 수준임	

○ 기존 PCI 지침 조사 분석서

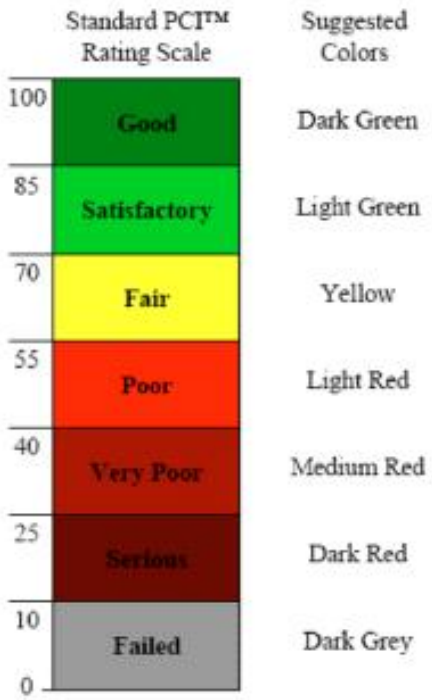
- 포장상태평가지수란, 도로이용자가 포장상태를 주관적인 의견에 대하여 균열, 패칭, 소성변형, 종단평탄성 등과 관련시킨 수학적 모델을 뜻하며, 기존의 주관적인 포장상태를 벗어나 객관적이고 정량적인 수치를 뜻한다(권수안외, 2000). 도로의 포장상태를 평가하기 위한 주요인자는 아래 그림과 같은 균열률, 소성변형, 종단평탄성 등이 있음



- Highway Pavement Condition Index) 일반국도의 포장상태를 정량적으로 파악하기 위해 한국건설기술연구원에서는 2007년 도로포장관리시스템 연구과제의 일환으로써 포장평가지수인 NHPCI를 개발
- NHPCI는 국도관리사무소의 관리자, 산학연 포장 전문가로 구성된 패널(Panel)과 상세조사 구간의 현장자료(균열률, 소성변형, 종단평탄성 등) 간의 통계 분석을 통하여 다음과 같

은 모델식으로 개발됨(국토해양부, 2010)

- 국외에서 현재 사용 중인 대표적인 포장평가지수로는 미국 주도로교통공무원협회(AASHTO)의 PSI(Present Serviceability Index), 일본 건설성의 MCI (Maintenance Control Index), 미국 공항포장 등의 상태를 평가하기위해서 사용하는 PCI(Pavement Condition Index)가 있음(권수안 외, 2000)



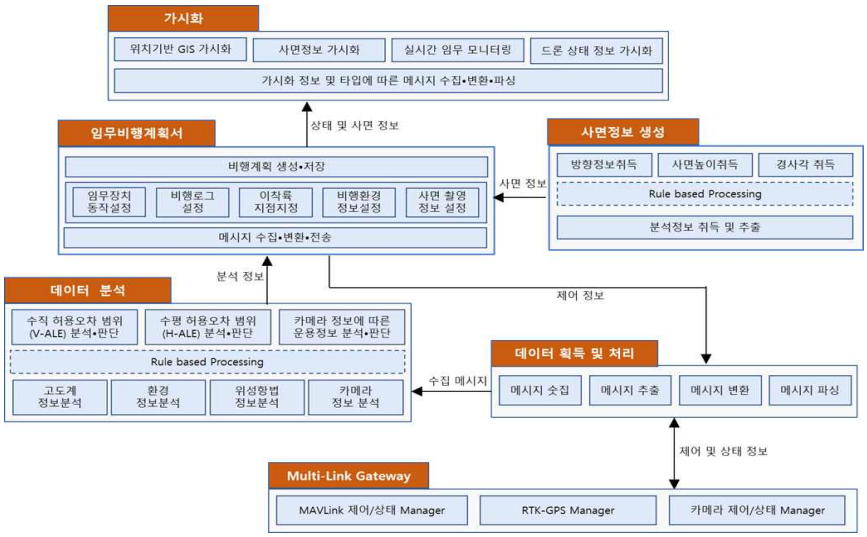
<3차년도 성과점검기준표>

핵심성과 (CTE)		단위성과		기관	성과점검기준						목표 달성도
					연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
1	SAR 센서 기반 비탈면 변위 측정 드론시스템	1-1	비탈면 변위 측정을 위한 SAR(L 밴드) 센서 장착 드론	메타빌드 (공동)	③	비탈면 측정 드론 관제 소프트웨어	1식	소프트웨어	소프트웨어 개발 확인	0.5	달성 완료
		1-2	드론 SAR 데이터 기반 비탈면 변위 측정 소프트웨어	부산대 (공동)	③	변위측정 정확도	5cm	측량 기준값 수집	기준값과 변위 비교분석	8.7	달성 완료
		1-3	비탈면 변위 측정 및 가시화 시스템	부산대 (공동) 메타빌드 (공동)	③	비탈면 변위위치 가시화 정확도	1m	측량 기준값 수집	기준값과 가시화 위치 비교분석	2.0	달성 완료
		1-4	평가/검증 보고서	롯데건설 (공동)	③	비탈면 가시화 정확도 평가/검증 보고서	1건	평가 보고서	평가 보고서 작성 확인	1.4	달성 완료
		1-5	비탈면 변위 가시화 시스템 활용 지침	롯데건설 (공동)	③	개발시스템 활용지침	1건	활용 지침	활용 지침 작성 확인	0.4	달성 완료
		1-6	비탈면 기술개발 테스트베드	건설연 (주관)	③	테스트베드	1식	파일럿 비탈면	파일럿 비탈면 구축 확인	1.0	달성 완료
2	영상 센서 기반 포트홀 사전/사후 탐지 드론시스템	2-1	포트홀 탐지를 위한 영상 센서 장착 드론	공간정보 (공동)	③	포트홀 탐지 드론 운용 지침	1건	운용 지침	운용 지침 작성 확인	5.1	달성 완료
		2-2	드론 영상 데이터 기반 포트홀 측정 소프트웨어	건설연 (주관)	③	검지율 및 오검지율	검지 : 80% 오검지 : 10%	기준값 수집	기준값과 비교분석	0.45	달성 완료
		2-3	포트홀 위치 가시화 시스템	공간정보 (공동) 아와소프트 (공동)	③	포트홀 위치 가시화 정확도	0.5 m	DGPS 기준값 수집	기준값과 비교분석	1.4	달성 완료
		2-4	평가/검증 보고서	건설연 (주관)	③	포트홀 가시화 정확도 평가/검증 보고서	1건	평가 보고서	평가 보고서 작성 확인	3.5	달성 완료

핵심성과 (CTE)		단위성과		기관	성과점검기준						목표 달성도
					연차	질적 성과지표	목표치	측정 방법	검증 방법	가중치	
		2-5	포토홀 탐지 및 가시화 시스템 활용 지침	건설연 (주관)	③	개발시스템 활용지침	1건	활용 지침	활용 지침 작성 확인	1.2	달성 완료
		2-6	포트홀 기술개발 테스트베드	건설연 (주관)	③	포트홀 탐지 실험시설	1식	실험 시설	실험 시설 구축 확인	2.0	달성 완료
3	초분광 센서 기반 포장상태지수 (PCI) 측정 드론시스템	3-1	PCI 측정을 위한 초분광 센서 장착 드론	신한 항업 (공동)	③	PCI 측정 드론 운용 지침	1건	운용 지침	운용 지침 작성 확인	5.7	달성 완료
		3-2	초분광 데이터 기반 PCI 추정 소프트웨어	건설연 (주관)	③	기준값 대비 PCI 측정 상관계수	0.7	PCI 조사차량 이용 기준값 수집	기준값과 상관계수 비교평가	5.2	달성 완료
		3-3	PCI 가시화 시스템	신한 항업 (공동) 아와 소프트 (공동)	③	PCI 측정 위치 가시화 정확도	0.5 m	DGPS 기준값 수집	기준값과 가시화 위치 비교분석	1.4	달성 완료
		3-4	평가/검증 보고서	건설연 (주관)	③	PCI 가시화 정확도 평가/검증 보고서	1건	평가 보고서	평가 보고서 확인	4.1	달성 완료
		3-5	PCI 측정 및 가시화 시스템 활용 지침	건설연 (주관)	③	개발시스템 활용지침	1건	활용 지침	활용 지침 작성 확인	1.2	달성 완료 (부록 7)
		3-6	PCI 측정 기술개발 테스트베드	건설연 (주관)	③	PCI 측정 실험시설	1식	실험 시설	실험 시설 구축 확인	2.0	달성 완료

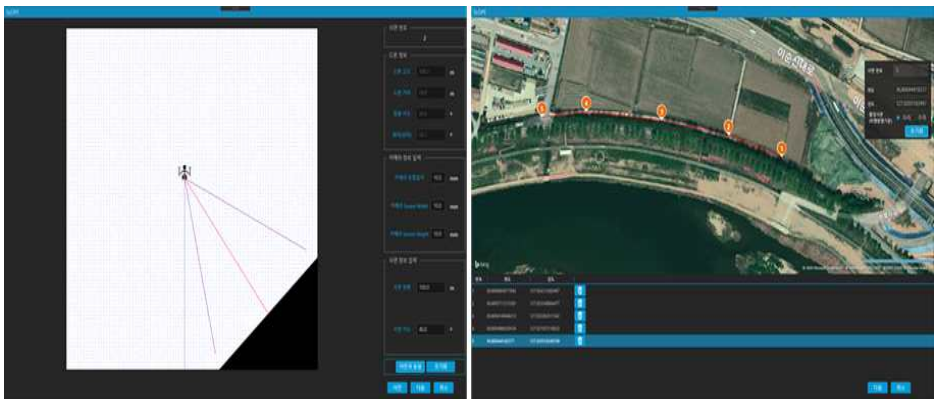
○ 비탈면 측정 드론 관제 소프트웨어

– 다중 비탈면 측정을 위한 임무비행을 위한 기능별 구조 정의



[임무비행계획서 기능구조도]

– 다중 비탈면 측정 비행 및 비행경로의 좌측, 우측 선택 기능



[(좌)드론 운영정보 입력, (우)측정 지점에서 측정센서 방향 설정]

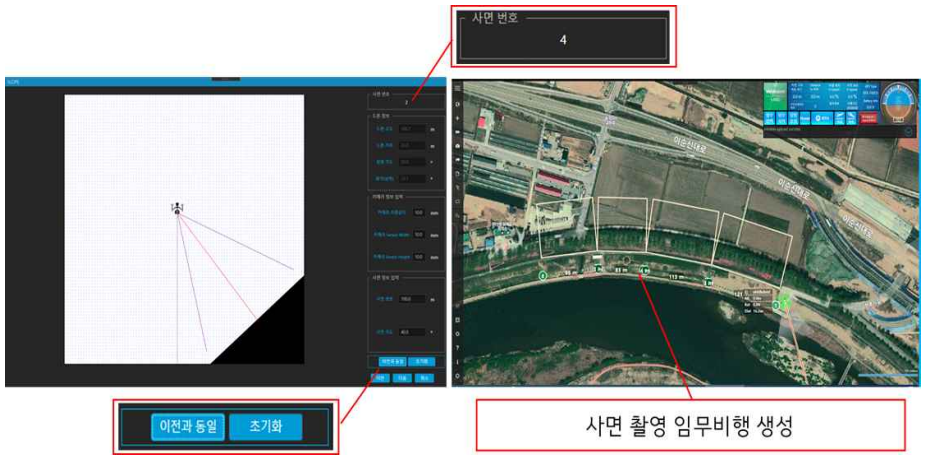
– 다중 사면 촬영 기능

① 곡면형태의 비탈면에 대해 다수의 직선임무비행계획서로 분할 생성 기능

② GIS상에 비탈면 표출 및 임무비행계획서를 분할 생성 기능

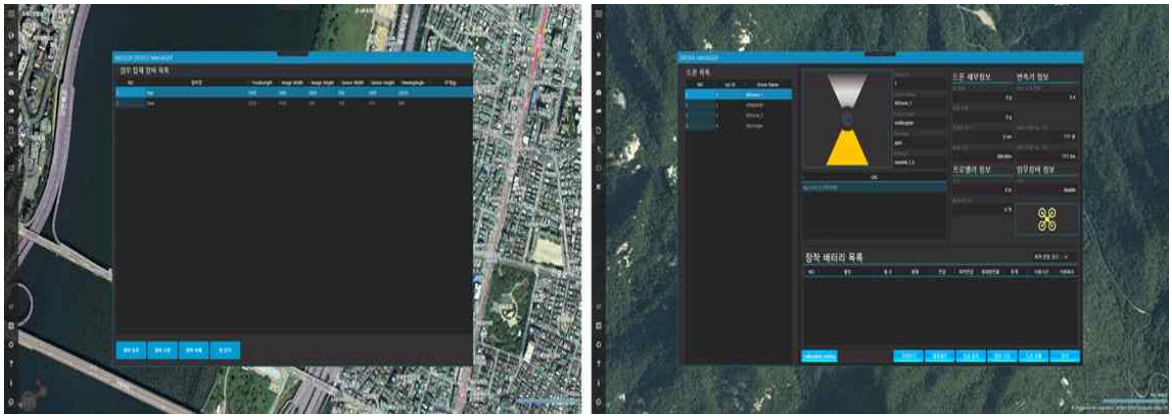
③ 비탈면의 촬영을 위한 드론의 운영고도 및 카메라 각도 복수 생성 기능

④ 복수의 비탈면 특정 영역에 대한 개별 정보 수정 및 생성 기능

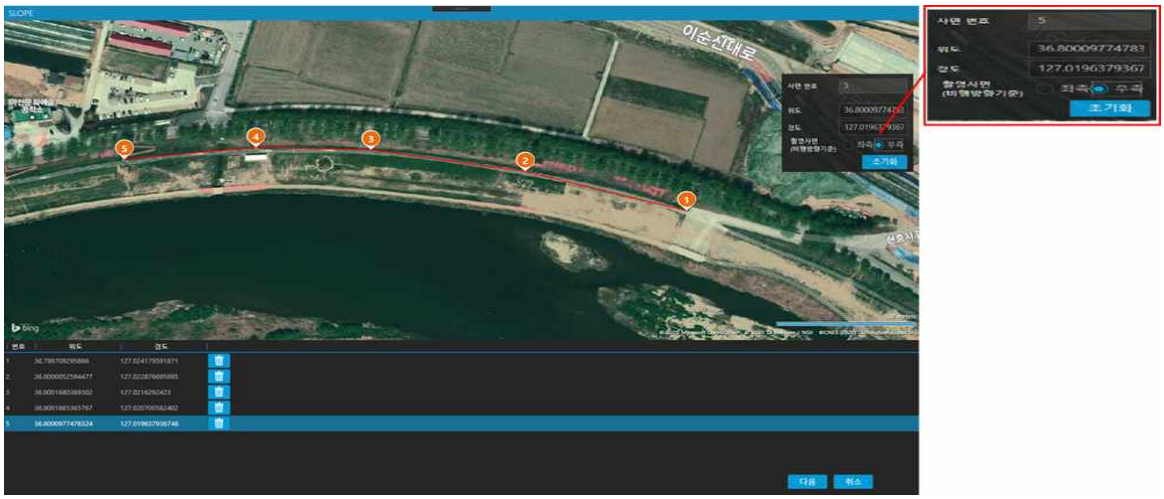


[드론 임무고도 및 카메라 각도 입력 및 복수의 비행계획서 생성]

- 비탈면의 위치에 따른 드론 카메라 촬영 방향 제어
  ① 비탈면의 위치에 따라 드론의 카메라 방향 설정 기능
  ② 마주 보는 비탈면 양쪽 촬영을 위한 카메라 방향제어 기능
  ③ 드론의 카메라 인터페이스에 따른 제어 메시지 전송 기능

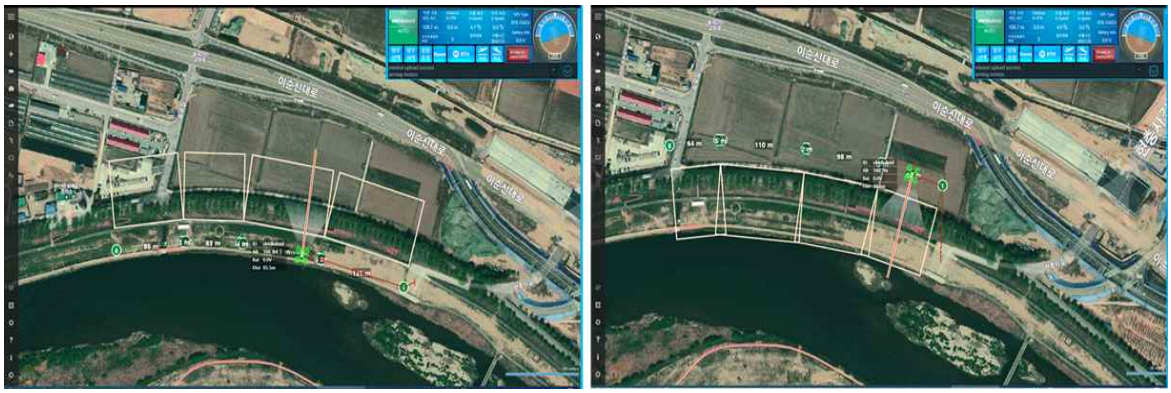


[(좌)임무장치 정보 등록, (우) 드론 등록]



[비탈면의 측정방향에 따른 임무장치 방향 설정 기능]

- 드론의 카메라 방향과 각도 표시 기능

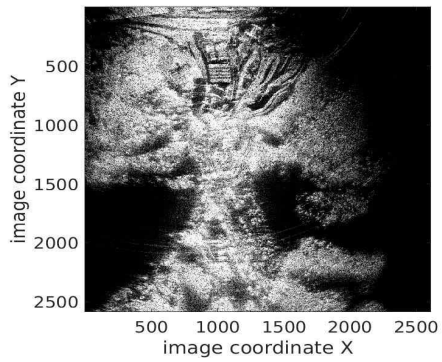


[임무비행 및 임무장치 각도 표시]

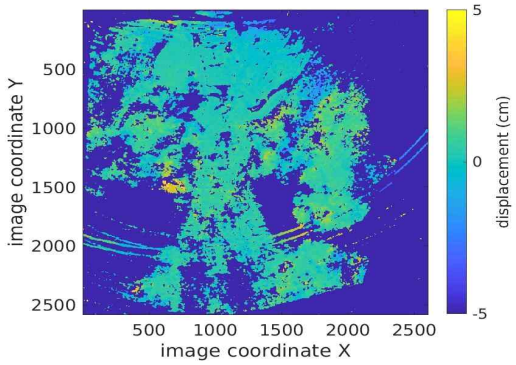
○ 비탈면 변위 및 가시화 검증계획서 및 평가분석서

- 3차년도 검증은 2021년 5월 31일 경기도 남양주 사이트에서 실시한 두 장의 드론 영상레이더 관측 시간은 약 7분 차이로 해당 시간 동안 비탈면 변위는 없다고 가정
- 비탈면 변위 분석 결과값은 드론 SAR 관측 시스템의 정확도로 판단 가능함

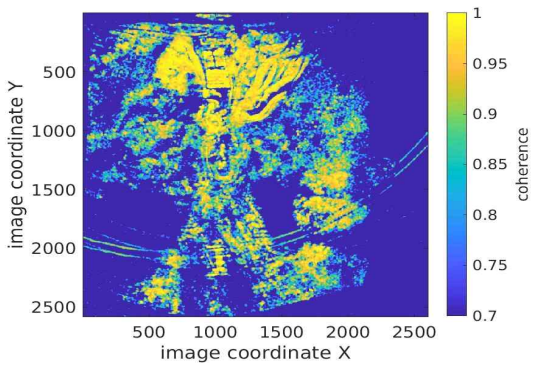
드론 SAR 시스템 후방산란지도



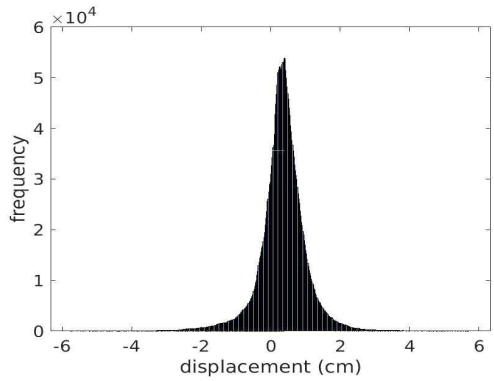
드론시스템으로 관측된 비탈면 변위 지도



영상레이더 자료 사이의 긴밀도



관측 변위량의 히스토그램



- 비탈면 변위 평가 분석 결과 과제목표인 5 ㎝ 이하에 해당하는 화소는 전체 유효 화소수 (2,888,826개)의 99.95 %에 해당하는 결과를 보임

변위 정확도 기준	해당 화소 수 (개)	비율 (%)
5 ㎝ 이하	2,887,478	99.95
4 ㎝ 이하	2,885,892	99.90
3 ㎝ 이하	2,876,332	99.57
2 ㎝ 이하	2,827,286	97.87
1 ㎝ 이하	2,462,789	85.25
0.5 ㎝ 이하	1,597,292	55.29

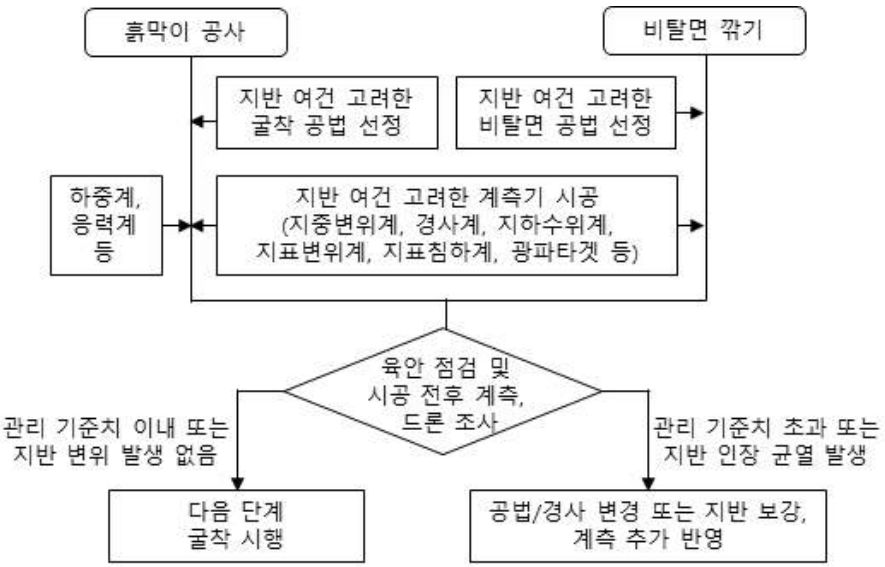
- 변위위치 정확도 분석는 약 1m 내외로 개발 목표치를 달성 완료하였음

○ 드론활용 비탈면 조사지침

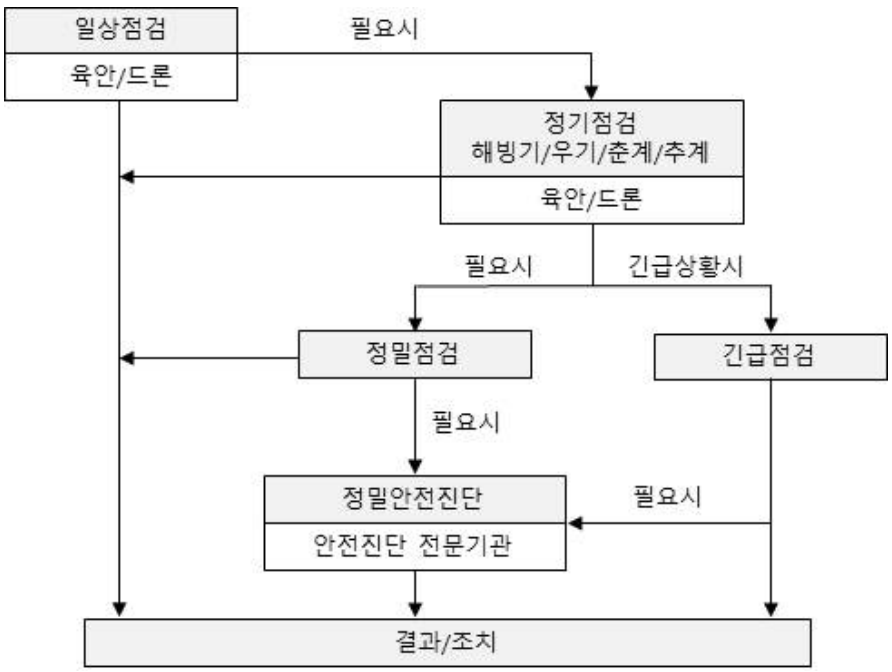
· 비탈면 관리를 위한 방법으로는 통상 육안점검, 지상 계측기 설치를 통한 계측관리가 대표적이며, 육안점검으로는 일상점검, 정기점검으로 나뉘며, 위험도가 큰 비탈면의 경우 지표면에 계측기를 설치하여 실시간 또는 주기적 계측관리로 구분됨

· 대깎기 비탈면, 급경사지, 식생이 많은 곳, 계측기가 설치되지 않은 지점 등에서는 비탈면 시공 또는 유지관리시 육안조사, 지표면에 설치된 계측기기 지점만 점검, 계측 관리되는 한 계점을 가지고 있음

· 대깎기 비탈면, 급경사지, 식생이 많은 곳, 계측기가 설치되지 않은 지점 등에서는 비탈면 시공 또는 유지관리시 육안조사, 지표면에 설치된 계측기기 지점만 점검, 계측 관리만으로는 확인하지 못한 곳이 발생하므로, 적합한 센서(RGB카메라, 열화상카메라, 초분광카메라, 라이다, 레이더 등)를 장착한 드론을 활용하여 조사 점검



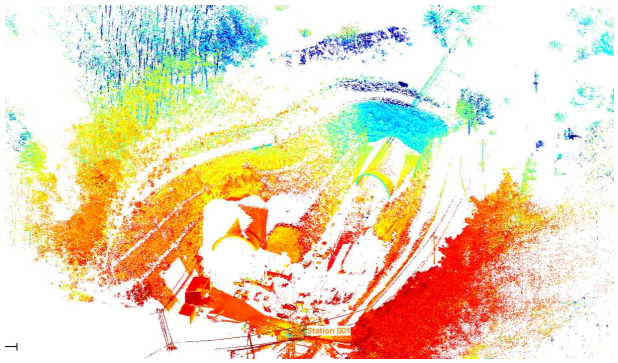
[공사중 관리 점검 업무 흐름도]



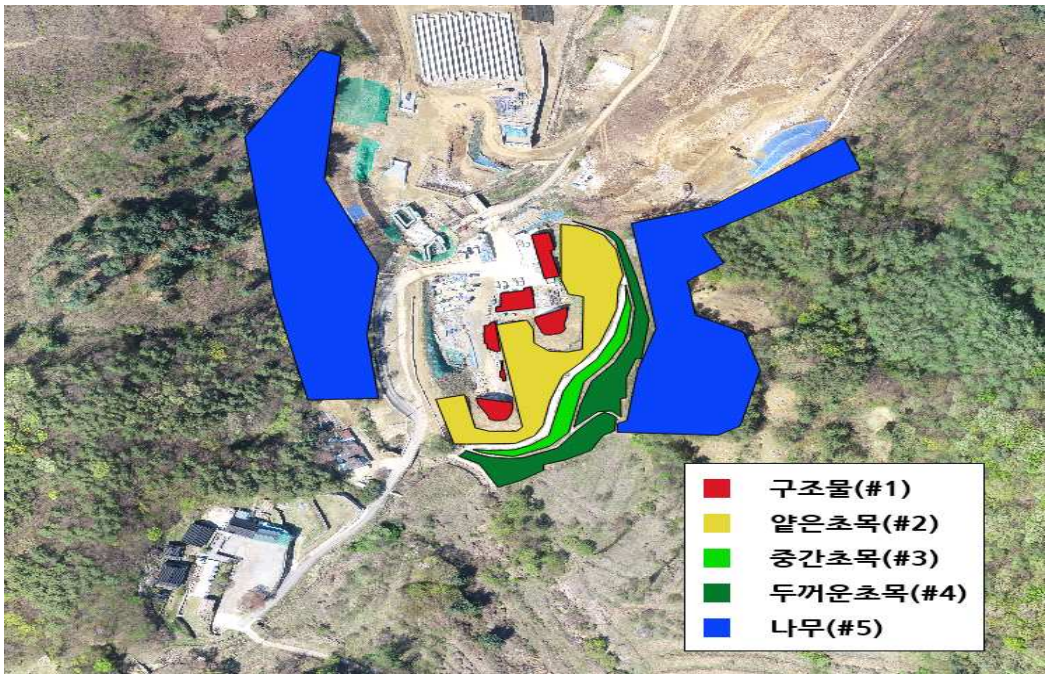
[공사후 관리 점검 업무 흐름도]

○ 비탈면 변위측정 드론 테스트베드

- 비탈면 현장 테스트베드에서 지상 장비 이용 측정 기준값과 L-SAR 센서 기반 드론 변위 데이터 비교 분석
  - · 지상 레이저 스캐너로 식생부/미식생부 혼재 구간 절대 기준값 측정
  - · 미식생부 주변 식생부 L-SAR 센서 기반 드론 변위 데이터과 지상레이저 스캐너와 비교분석

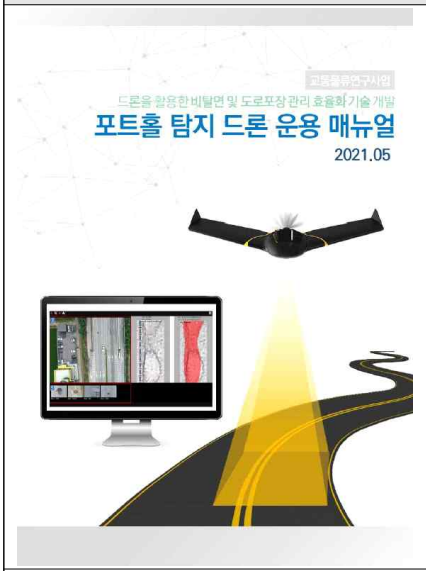
<table>
<tr><td>  </td><td>  </td></tr>
<tr><td>깎기 완료 영역</td><td>레이저 스캔 자료</td></tr>
<tr><td>  </td><td>  </td></tr>
<tr><td>지상 계측점 위치에서 바라본 현장</td><td>식생(두께 20~40㎝)과 미식생부 혼재영역</td></tr>
</table>

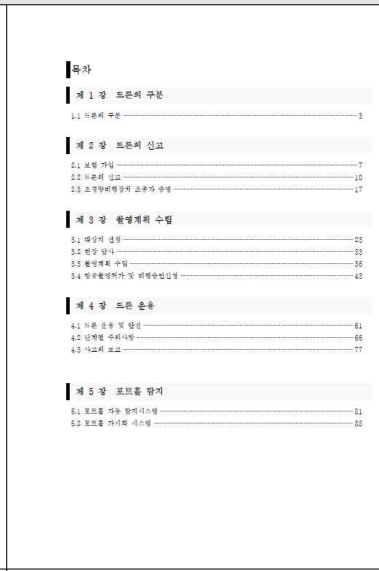
[테스트베드 현장 (화도-양평 o공구)]



○ 포트홀 탐지 드론 운용 지침
- 드론의 분류, 장치신고, 보험가입과 같은 드론운용 사전 준비사항 수록
- 비행전 대상지선정, 공역확인, 촬영허가 및 비행승인신청 등 사전 준비항목 작성
- 드론 운용 및 안전, 단계별 주의사항, 사고 발생 시 조취사항 작성

포트홀 탐지 드론 운용 매뉴얼




목차

제 1 장 드론의 구분
1.1 드론의 구분 ····· 3

제 2 장 드론의 신고
2.1 보험 가입 ····· 7
2.2 드론의 신고 ····· 10
2.3 초경량비행장치 조종자 증명 ····· 17

제 3 장 촬영계획 수립
3.1 대상지 선정 ····· 23
3.2 현장 답사 ····· 33
3.3 촬영계획 수립 ····· 38
3.4 항공촬영허가 및 비행승인신청 ····· 43

제 4 장 드론 운용
4.1 드론 운용 및 탑재 ····· 61
4.2 단계별 주의사항 ····· 66
4.3 사고시 보고 ····· 77

제 5 장 포트홀 탐지
5.1 포트홀 자동 탐지시스템 ····· 81
5.2 포트홀 가시화 시스템 ····· 83


제1장 드론의 구분

1장 드론의 구분

1.1 드론의 구분

○ 항공안전법(이하 "법") 제2조 제3호에서 "초경량비행장치"항공기와 경량항공기 외 공기의 반작용으로 뜰 수 있는 장치로서 자체중량, 화물 수 등 국토교통부령으로 정하는 기준에 해당하는 무인비행장치, 행글라이더, 패러글라이더, 기구류 및 무인비행장치 등을 말한다.

○ 국토교통부령으로 정하는 "초경량비행장치의 기준"은 항공안전법 시행규칙(이하 "규칙") 5조에 규정하고 있다. 여기서 드론은 초경량비행장치 중 무인비행장치(사람이 탑승하지 아니하는 것)로 무인동력비행장치(연료의 중량을 제외한 자체중량이 150kg 이하인 무인비행기, 무인멀티콥터 또는 무인헬리콥터)와 무인비행선(연료의 중량을 제외한 자체중량이 180kg 이하이고 길이가 20미터 이하인 무인비행선)에 해당된다.

<그림 1-1> 초경량비행장치의 분류

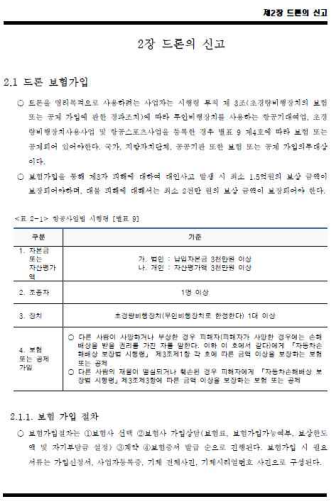

포트홀 탐지 드론 운용 매뉴얼

○ 무인비행장치는 무인동력비행장치, 무인비행선으로 분류되며, 무인동력비행장치의 세부류로 무인비행기, 무인멀티콥터, 무인헬리콥터가 있다. 일반적으로 드론은 무인동력비행장치에 해당된다.

○ 본 매뉴얼에서는 포트홀 탐지를 위해 주로 사용되는 무인비행기와 무인멀티콥터를 위주로 설명하였다.

<표 1-1> 무인비행장치 분류

구분	무인동력비행장치			무인비행선
	무인비행기	무인헬리콥터	무인멀티콥터	
사진				
특성	사람이 타지 않고 무선통신 장비를 이용하여 조종하거나, 내장된 프로그램에 의해 자동으로 비행하는 비행체로, 주 조저으로 일반 비행기와 거의 같고, 항공촬영 등에 활용하고 있음 비행속도가 빠르며 장거리 비행 가능 활주로 또는 발사대를 이용하여 이륙	사람이 타지 않고 무선통신장비를 이용하여 조종하거나, 내장된 프로그램에 의해 자동으로 비행하는 비행체로 주조저으로 일반 헬리콥터와 거의 같고, 항공촬영 등에 활용하고 있음 저가의 비행 가능 고정도 대비 속도 비행거리 등이 짧음	사람이 타지 않고 무선통신장비를 이용하여 조종하거나, 내장된 프로그램에 의해 자동으로 비행하는 비행체로 4개 이상의 전동모터로 구동되며 조종자의 없이 모터의 속도로 기체를 제어하는 것이 특징임 저가의 비행 가능 고정도 대비 속도 비행거리 등이 짧음 항공촬영 등으로 활용 주로 소형 드론에 주로, 일러레터 최진적 발전이	가스기구와 같은 기구 비행체에 소스로 추진 장치로 엔진을 장착하여 추진할 수 있는 추진장치를 부착하여 만들어 가볍게 만든 비행체이며 가스를 이용하여 비행하는 것으로 기구 류가, 기술 등 비행선에 가스를 충전하여 양력을 만들며, 비행속도는 느리지만 체공시간이 긴 장점이 있음


제2장 드론의 신고

2장 드론의 신고

2.1 드론 보험가입

○ 드론을 영리목적으로 사용하려는 사업자는 시행령 부록 제 3조(초경량비행장치의 보험 또는 공제 가입에 관한 경과조치)에 따라 무인비행장치를 사용하는 항공기대여업, 초경량비행장치사용사업 및 항공스포츠사업을 등록한 경우 별표 9 제4호에 따라 보험 또는 공제회에 가입하여야 한다. 국가, 지방자치단체, 공공기관 또한 보험 또는 공제 가입의무대상이다.

○ 보험가입을 통해 제3자 피해에 대하여 대인사고 발생 시 최소 1.5억원의 보상 금액이 보장되어야하며, 대물 피해에 대해서는 최소 2천만 원의 보상 금액이 보장되어야 한다.

<표 2-1> 항공사업법 시행령 [별표 9]

구분	기준
1. 자본금 또는 자산평가액	가. 법인 : 납입자본금 3천만원 이상 나. 개인 : 자산평가액 3천만원 이상
2. 조종자	1명 이상
3. 장치	초경량비행장치(무인비행장치로 한정한다) 1대 이상
4. 보험 또는 공제 가입	○ 다른 사람이 사망하거나 부상한 경우 피해자(피해자가 사망한 경우에는 손해배상을 받을 권리를 가진 자를 말한다. 이하 이 호에서 같다)에게 「자동차손해배상 보장법 시행령」 제3조제1항에 따른 금액 이상을 보장하는 보험 또는 공제 ○ 다른 사람의 재물이 멸실되거나 훼손된 경우 피해자에게 「자동차손해배상 보장법 시행령」 제3조제1항에 따른 금액 이상을 보장하는 보험 또는 공제

2.1.1. 보험 가입 절차

○ 보험가입절차는 ①보험사 선택 ②보험사 가입상담(보험료, 보험가입가능여부, 보상한도액 및 자기부담금 설정) ③계약 ④보험증서 발급 순으로 진행한다. 보험가입 시 필요서류는 가입신청서, 사업자등록증, 기체 견적서류, 기체사진첨부 사진으로 구성된다.


포트홀 탐지 드론 운용 매뉴얼

<표 2-3> 보험가입확인

등/허/사진첨부여부 확인	보험가입증서 확인

2.2 드론 신고

○ 초경량비행장치 신고 목적은 체계적인 장치 신고 관리로 안전 관련 위험행위 예방 및 국민 인명 재산 보호와 기체별 고유번호 관리를 통한 향후 추인기 교부 관리, 드론 택시 택배 사용화에 기여하기 위함이다.

○ 드론 신고는 사업용(영리)과 비사업용(비영리)에 따라 장치신고 및 사업등록의 필요성이 구분되고 영리목적은 항공사업법 제2조에 따른 항공기대여업, 초경량비행장치사용사업, 항공레저스포츠사업이 사용하는 초경량 비행장치를 말한다.

○ 드론 비행 및 항공촬영허가를 신청하기 위해서는 비사업용의 경우 12kg이상, 사업용의 경우 장치신고가 필수로 되어있어야 한다.


제3장 촬영계획수립

3장 촬영계획수립

○ 포트홀 탐지를 위한 촬영은 주로 무인비행기 형식인 고정익 드론과 무인멀티콥터 형식인 회전익 드론을 활용한다. 촬영계획은 <그림 3-1>과 같이 대상지선정, 현장답사, 촬영계획수립, 촬영 및 비행승인신청, 드론촬영 순으로 이루어진다.

<그림 3-1> 촬영계획수립 프로세스

3.1 대상지 선정

○ 포트홀 탐지를 위한 대상지 선정은 매우 중요한 업무로 대상지 선정 시 활용성, 시급성, 안정성, 기능성 등을 고려해야 한다. 또한 비행경로와 비행고점, 그리고 드론 장비에 있어 대상지 선정은 매우 중요한 업무이며 활용성, 시급성, 안정성, 기능성 등을 고려해 추천순위를 선정하고, 노선길이, 비행고점, 이착륙장소, 기상 등을 고려하고 선정하여야 한다.


제3장 촬영계획수립

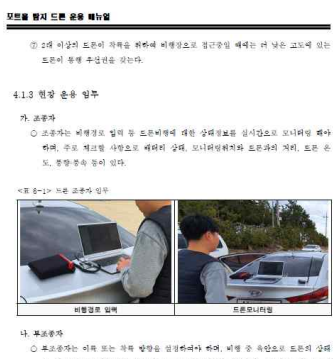
<그림 3-3> 국토교통노드링크 다운로드

○ 국가교통정보통합센터 홈페이지의 상단메뉴 표준노드링크 탭에서 노드링크 개요설을 클릭한다. 노드링크 개요설의 가장 최신날짜 '[2021-02-15]NODELINKDATA.zip (2021-02-15, 122.84MB)의 파일을 다운받는다. 다운받은 압축파일을 해제하면 SHP가 생성되며, SHP파일은 QGIS 소프트웨어를 통해 확인할 수 있다. 또한 kml 포맷으로 변환하여 구글어스와 같은 프로그램에서도 사용할 수 있다.

<표 3-1> 전국도로 노드링크

전국도로 노드,링크	노드,링크 확대
	

○ 노드는 링크를 구분하는 점(링크, 시작점, 링크, 종료점)으로 표준노드/링크 구축 운영지침에 따라 도로교차점, 도로 시종점, 교통통제점, 도로부고 변화점, 행정구역 변화점, 도로 운영 변화점, 교통 건설일점, 그 외에 ITS사업 수혜가 필요에 따라 정하는 지점이다. 노드에 대한 속이를 설명은 아래의 표와 같다.


포트홀 탐지 드론 운용 매뉴얼

② 2대 이상의 드론이 작목을 위하여 비행장으로 접근중일 때에는 더 낮은 고도에 있는 드론이 통행 우선권을 갖는다.

4.1.3 현장 운용 업무

가. 조종자

○ 조종자는 비행경로 입력 등 드론비행에 대한 상세정보를 실시간으로 모니터링 하여야 하며, 주로 체크할 사항으로 배터리 상태, 모니터링화면과 드론과의 거리, 드론 온도, 통합통속 등이 있다.

<표 6-1> 드론 조종자 업무

나. 부조종자

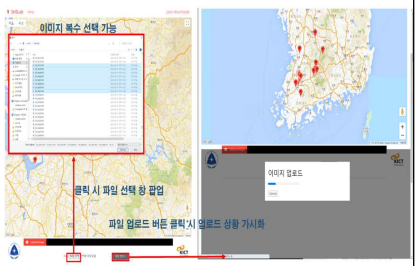
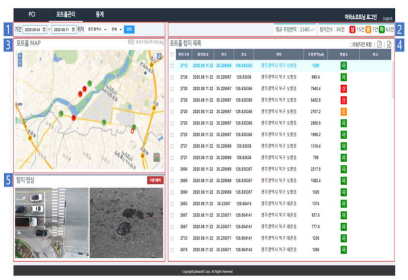
○ 부조종자는 이륙 또는 착륙 방향을 설정하여야 하며, 비행 중 특이점으로 드론의 상태 및 위치 등을 주의하여야 한다. 드론 비행 중 이상한 움직임이 포착되면 즉시 조종자에게 보고하여야 하며, 비상시 비상대처방안에 따라 비상착륙을 실시하여야 한다. 드론의 추락은 곧 인명사고로 발생할 수 있음을 인지하여야 한다.

○ 포트홀 탐지 검지율/오검지율 평가계획서 및 평가결과 분석서

- 항공 영상 내 포트홀 탐지 알고리즘 성능 평가 계획
· 촬영되는 항공 영상들에서 포트홀을 탐지할 때, 중요한 지표는 탐지율과 오탐지율이다. 포트홀 탐지율은 실제 포트홀 중 포트홀로 예측된 것의 비율이다. 오탐지율은 포트홀로 예측한 것 중 포트홀이 아닌 것의 비율
· 분류 성능 평가지표에서 탐지율은 Recall에 해당하며 오탐지율은 1-Precision과 같은 계산식
· 약 30~70m 고도에서 촬영한 항공 영상 100장(포트홀 총 71개)을 이용하여 알고리즘 성능 평가를 진행
· 탐지율과 오탐지율은 각 항공 영상에 대해 평가되었으며, 항공 영상에 포트홀이 없는 경우 탐지와 오탐지를 하지 않았다면, 탐지율과 오탐지율을 100%로 계산
· 각 항공 영상의 탐지율과 오탐지율의 평균으로 항공 영상 내 포트홀 탐지 알고리즘 성능을 평가
- 항공 영상 내 포트홀 탐지 알고리즘 성능 평가결과
· 약 30~70m 고도에서 촬영한 항공 영상 100장 기준(포트홀 총 71개)
· 포트홀 검지율 : 탐지 포트홀 수 / 포트홀 수 [TP/(TP+FN)]
· 포트홀 오검지율 : 오탐지 횟수 / 포트홀 탐지 횟수 [FP/(TP+FP)]
· 포트홀 검지율 : 97.2%(세부내역 : 부록 1)
· 포트홀 오검지율 : 9.8%(세부내역 : 부록 1)
- 항공 영상 내 포트홀 탐지 알고리즘 성능 평가결과 분석
· 목표 검지율 90%보다 7.2%p 높은 포트홀 검지율을 보임
· 목표 오검지율 10%보다 0.2%p 낮은 포트홀 오검지율을 보임

○ 포트홀 위치 가시화 정확도

- 포트홀 위치 가시화 정확도 0.5m 이내 달성을 위한 현장실증
- 일시 : 2021년 5월 21일 10:00~14:00
- 장소 : 연천 SOC 실증센터
- 방법 : 1. GPS를 활용한 네트워크 RTK방식으로 포트홀 위치 취득
  2. 드론을 활용한 포트홀 영상 수집
  3. AI학습용 서버에 수집된 영상자료 업로드
  4. AI자동판독 및 필터, 위치 및 크기 도출
  5. 가시화시스템에서 포트홀 위치 및 크기 판독
  6. 수집된 GPS데이터와 영상에서 추출된 위치 비교

1. 포트홀 위치 취득	2. 포트홀 드론촬영	3. 영상 업로드
		
4. AI자동 판독	**5. 가시화시스템 표출**	**6. 데이터 비교**
 업로드된 이미지의 자동탐지 결과		

구분	관측값(GRS80 중부)		가시화 값(GRS80 중부)		오차	
	X	Y	X	Y	X	Y
포트홀1	206979.034	613051.536	206979.125	613051.594	-0.091	-0.058
포트홀2	207009.451	613055.225	207009.372	613055.296	0.079	-0.072
포트홀3	207029.648	613056.130	207029.674	613056.052	-0.026	0.078

○ 포트홀 탐지 드론 활용지침

- 포트홀 자동 탐지시스템 및 가시화 시스템 운용법 작성

포트홀 탐지 가시화 시스템 운용 매뉴얼

5장 포트홀 탐지

5.1 포트홀 자동 탐지시스템

○ 포트홀 자동 탐지시스템은 항공 이미지를 업로드하면 항공 영상 내 포트홀을 탐지하여 포트홀의 위치 및 면적, 가로, 세로, 크기한 포트홀 이미지 등을 반환한다. <그림 5-1>은 항공 영상 업로드 및 위치 가시화를 위한 포트홀 자동 탐지시스템의 메인 페이지다.



<그림 5-1> 포트홀 자동 탐지시스템 메인 페이지

5.1.1 이미지 업로드

○ 포트홀 자동탐지 시스템은 웹 어플리케이션으로 구현이 되어있으며 사용자가 주로 사용하는 Firefox, Chrome, Opera, Microsoft Edge에서 가동성 테스트를 진행하였다. 결과는 아래의 <표 5-1>과 같으며 4개의 웹브라우저 모두 이미지 업로드 및 포트홀 자동탐지 결과 확인이 가능하다.

○ 촬영한 항공 이미지 업로드 방법은 그림 <그림 5-2>와 같으며, 먼저 파일선택 버튼을 클릭하여 파일선택 창을 확인한 뒤추고 업로드를 원하는 이미지를 선택한다. 그 리고 선택된 이미지를 자동탐지시스템에 업로드 하기 위해 파일 업로드 버튼을 클릭

한다. 이후 <그림 5-2>의 추측에 이미지 업로드 진행 상황을 볼 수 있는 창을 확인 할 수 있다. 이미지 업로드가 끝날 때까지는 해당 페이지에서 대기하여야 한다.

○ 이미지 업로드 시, DJI 의원칙 드론, SensorFly 고정익 드론의 영상은 포트홀의 면적 측정을 위해 가장으로 픽셀 당 단위면적을 구할 수 있지만, 이미지 촬영 시 아래하의 CMOS 스펙이 다르거나 비하게이터의 키-값 구성이 다른 이미지가 들어올 시 업데이트가 필요하다.

○ 이미지 업로드가 끝나고 처리 현황을 보기 위해서는 좌측 상단의 가시화 버튼에서 처리목록 버튼을 클릭하여 <그림 5-3>과 같이 현재 처리 대기 중인 이미지들을 확인 할 수 있다.

<표 5-1> 웹 어플리케이션 가동성 테스트

구분	Opera	Chrome	Firefox	Microsoft Edge
이미지 업로드	O	O	O	O
탐지결과 확인	O	O	O	O



<그림 5-2> 이미지 업로드 선정



<그림 5-4> 포트홀 자동탐지 결과 확인



<그림 5-5> 최근 업로드된 이미지의 포트홀 자동탐지 결과



<그림 5-6> 메인 페이지 지도 내 포트홀 자동탐지 결과

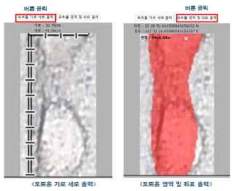


<그림 5-7> 포트홀 시각화 페이지


<그림 5-8> 항공 영상 확인



<그림 5-9> 항공 영상 내 포트홀 위치 가시화 기능



<그림 5-10> 포트홀 자동탐지 결과 및 정보 시각화 기능

5.2.3 메타데이터 정보

○ 포트홀 탐지결과 데이터는 JSON(JavaScript Object Notation) 포맷으로 구성이 되어 있다.

○ 포트홀 탐지결과 데이터는 원본 영상의 이름과 class, 이미지 촬영 경도, 위도, 경도, 고도, 포트홀의 면적, 포트홀의 좌표, 포트홀의 가로, 세로 등의 데이터가 저장되어 있다.

○ <그림 5-11>은 JSON 포맷으로 구성된 포트홀 탐지결과 데이터의 예시이다.


<그림 5-11> 포트홀 탐지결과 데이터의 예시

5.2 포트홀 가시화 시스템

○ 드론의 촬영 영상의 포트홀 탐지 영상분석 결과데이터를 가시화하여 운용할 수 있는 웹 서비스 시스템이다. 운용자는 포트홀 영상분석 결과데이터를 실시간 업로드하여 Vworld기반 지도 레이어에 가시화 할 수 있으며, 포트홀 영상분석 데이터의 상세적인 조회기능을 제공한다. 포트홀 영상분석 전용 툴을 통해 포트홀의 탐지위치 확인과 원본 촬영영상 다운로드 등의 운용관리성 UI를 제공한다.

○ 취합도 기준은 726개의 차량 데이터를 조사하여 DB화 하였으며 DB데이터를 바탕으로 대형, 중형, 소형 차량의 타이어 면적에 따라 취합도 일계적는 다각 삼중화로 표현된다.

○ 포트홀 영상분석 결과와 탐지위치에 대한 각종 리포트 PDF출력 기능을 제공한다.

○ 포트홀 영상분석 결과에 대한 메모 출력 기능을 제공한다.

5.2.1 메타데이 연동

○ 포트홀 데이터 업로드 시 시스템 서버는 외부네트워크가 오픈되어 있어야 한다. (KAKAO API 연동 GPS 주소정보 변환, 영상이미지 수집을 위한 분석서버와의 외부 통신 요구)

○ 포트홀 분석 웹서비스(http://203.250.35.243:5938)에 접속하여 [메뉴 "json download" 버튼을 클릭하면 시스템에 업로드 가능한 포트홀 영상분석결과가 다운로드 된다.


<그림 5-12> 포트홀 분석 데이터 다운로드

○ 포트홀 업로드 데이터 형식은 다음과 같다.

<표 5-1> 데이터 업로드 형식 (JSON)

키	값(샘플예시)	설명
name	IX-12-82312_0105_0319.JPG	드론 촬영이미지 파일명
type	pothole	탐지유형
image_url	http://203.250.35.243:5938/media/IX-12-82312_0105_0319/pothole/286_366_2670_2729.jpg	탐지 이미지 URL
latitude	37.015123742777774	드론 촬영위치 위도
longitude	127.31366238472221	드론 촬영위치 경도
altitude	139.7861	고도
angle	251.28594	정북방향 차이각도
area	6059.839098111819	탐지된 포트홀 면적
axis_x	2670	탐지 된 Tile의 시작 x좌표
axis_y	286	탐지 된 Tile의 시작 y좌표
axis_w	2729	탐지 된 Tile의 종료 x좌표
axis_h	366	탐지 된 Tile의 종료 y좌표
pothole_width	121.94353047835	포트홀 너비
pothole_height	169.48084393047 2	포트홀 높이
pothole_lat	37 0 54.653815817855 75 N	포트홀 위도
pothole_long	127 18 49.064825 96514736 E	포트홀 경도

5.2.2 포트홀 가시화 시스템 매뉴얼

○ 로그인

가. 등록된 사용자에 대해서만 시스템 접속이 가능하다.

나. 로그인 성공 시 포트홀 대시보드 페이지로 이동한다.


<그림 5-13> 포트홀 가시화 시스템 로그인

○ 포트홀 대시보드


<그림 5-14> 포트홀 대시보드

① 탐지기간 및 취득 검색조건을 설정할 수 있다. "설정" 버튼은 시스템 관리 권한 측정에 대한 노출되며 포트홀레이터 업로드, 오탐지 일괄처리 등 운용관리 기능을 제공한다.

④ ⑤번의 선택된 포트홀의 확대된 영상을 출력한다.

⑤ ⑥번 드론 촬영 위치 내 탐지된 포트홀들의 면적과 함께 출력한다. 탐지 된 포트홀 클릭 시 ⑤번영상에 하고업데이트 ⑥번 영상이 갱신된다.


<그림 5-15> 포트홀 통계 화면

① 포트홀 탐지현황 통계자료의 조회기간을 설정한다.

② 일별 시스템 내 포트홀 탐지결과에 대한 차트를 출력한다.

③ ②번 차트 데이터를 기준으로 취합도 분포도를 출력한다.

④ ②번 차트를 클릭 시 해당 날짜의 취득별 포트홀 탐지결과를 출력한다.

⑤ ③번 차트 데이터를 기준으로 취합도 분포도를 출력한다.

○ 포트홀 탐지 드론 테스트베드

- 고정밀 GPS(RTK/PPK)탑재 드론을 통한 포트홀 영상수집

No	촬영일	노선번호	지역	투입드론	해상도	사진매수	노선길이(km)
1	02월 03일	38번 국도	안성	Phantom4 Pro ebee X	1.0cm/px 1.5cm/px	73	1
2	02월 04일	38번 국도	음성, 이천	Phantom4 Pro	1.0cm/px	64	1
3	02월 05일	42번국도	원주, 여주, 이천	ebee X	1.5cm/px	713	11
4	02월 06일	38번국도	이천, 안성	ebee X	1.5cm/px	2,982	42.3
5	05월 11일	42번국도	여주, 이천	eBee Plus	1.8cm/px	297	16.5
6	05월 20일	38번 국도	평택	ebee X	1.5cm/px	512	7.4
7	06월 09일	하남진곡산단로	광주, 담양, 장성	ebee X	1.5cm/px	505	6.8
8	07월 22일	26번국도	군산	ebee X	1.5cm/px	1,274	16.2
9	08월 04일	29번국도	화순	ebee X	1.5cm/px	867	8.11
10	08월 11일	빛고을대로, 13번, 15번 국도	광주, 담양, 장성	Phantom4 Pro	0.53~1.87/px	93	1
합계						7,385	111.39
평균						738	11.14

- 서울지방국토관리청 산하 수원국토관리 사무소 관리국도 40㎞ 실증



- 일 시 : 2020.2.6
- 장 소 : 국도 38호선(안성터미널~감곡 IC) 약 40km
- 내 용 : 도로 포트홀 탐지기술 현장조사, 자료처리, 결과협의 등 기관간 기술개발 현장실험

담당자 요구사항

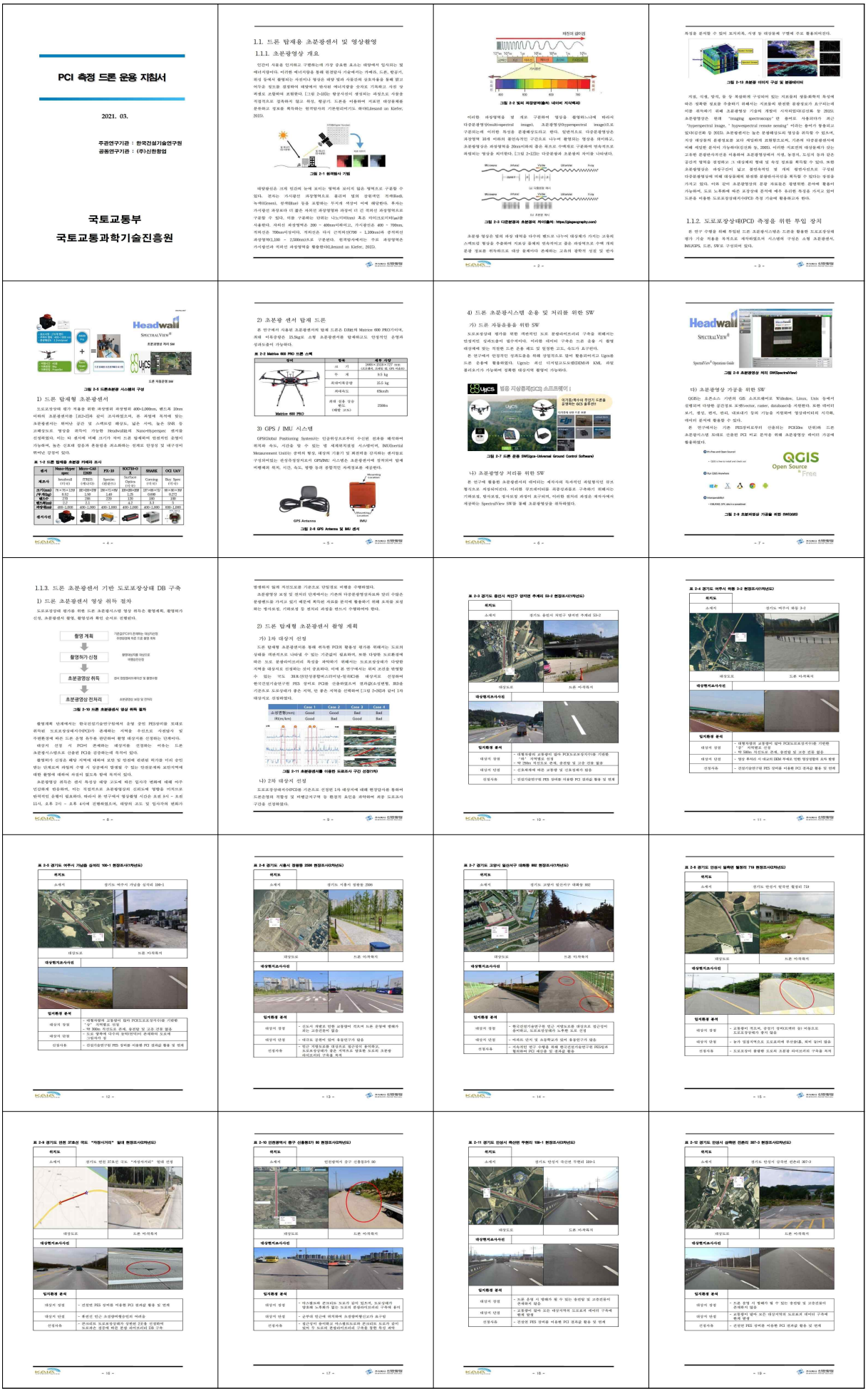
- 영상에서 육안으로 포트홀 식별이 어려움
  (회전익 드론 활용 고도별 영상을 취득하여 식별 가능한 해상도 재선정)
- 조사 및 분석시간 단축 필요
  (상습 포트홀 발생구간 사전답사를 통한 신속 DB구축체계 마련)







○ PCI 측정 드론 운용 지침



PCI 측정 드론 운용 지침서

2021. 03.

주관연구기관 : 한국건설기술연구원
공동연구기관 : (주)신한항업

국토교통부
국토교통과학기술진흥원

1.1. 드론 탑재용 초분광센서 및 영상촬영

1.1.1. 초분광영상 개요

그림 2-1 원격탐사 기법

그림 2-2 빛의 파장영역(출처: 네이버 지식백과)

그림 2-3 다른분광과 초분광의 차이(출처: https://jiqcgeography.com)

1.1.2. 도로포장상태(PCI) 측정을 위한 투입 장치

그림 2-4 초분광영상 처리 SW(SpectralView)

1) 드론 탑재형 초분광센서

그림 2-5 드론초분광 시스템의 구성

2) 초분광 센서 탑재 드론

표 2-2 Matrice 600 PRO 드론 스펙

3) GPS / IMU 시스템

그림 2-6 GPS Antenna 및 IMU 센서

4) 드론 초분광시스템 운용 및 처리를 위한 SW

가) 드론 자동운용을 위한 SW

그림 2-7 드론 운용 SW(Qgcs-Universal Ground Control Software)

나) 초분광영상 처리를 위한 SW

그림 2-8 초분광영상 기준을 위한 SW(QGIS)

다) 초분광영상 가공을 위한 SW

1.1.3. 드론 초분광센서 기반 도로포장상태 DB 구축

1) 드론 초분광센서 영상 취득 절차

그림 2-10 드론 초분광센서 영상 취득 절차

2) 드론 탑재형 초분광센서 촬영 계획

가) 1차 대상지 선정

그림 2-11 초분광센서를 이용한 도로조사 구간 선정(1차)

나) 2차 대상지 선정

표 2-3 경기도 한산시 원평동 산96-1 현장조사(1차선도)

표 2-4 경기도 여주시 가남읍 상석리 100-1 현장조사(1차선도)

표 2-5 경기도 여주시 흥천면 상동리 산96-1 현장조사(1차선도)

표 2-6 경기도 여주시 흥천면 상동리 산96-1 현장조사(2차선도)

표 2-7 경기도 고양시 일산서구 대화동 882 현장조사(1차선도)

표 2-8 경기도 고양시 일산서구 대화동 882 현장조사(2차선도)

표 2-9 경기도 안성시 원곡면 월정리 719 현장조사(1차선도)

표 2-10 경기도 안성시 원곡면 월정리 719 현장조사(2차선도)

표 2-11 경기도 안성시 죽산면 두현리 산96-1 현장조사(1차선도)

표 2-12 경기도 안성시 죽산면 두현리 산96-1 현장조사(2차선도)

표 2-13 경기도 안성시 원곡면 월정리 산96-1 현장조사(1차선도)

표 2-14 경기도 안성시 원곡면 월정리 산96-1 현장조사(2차선도)

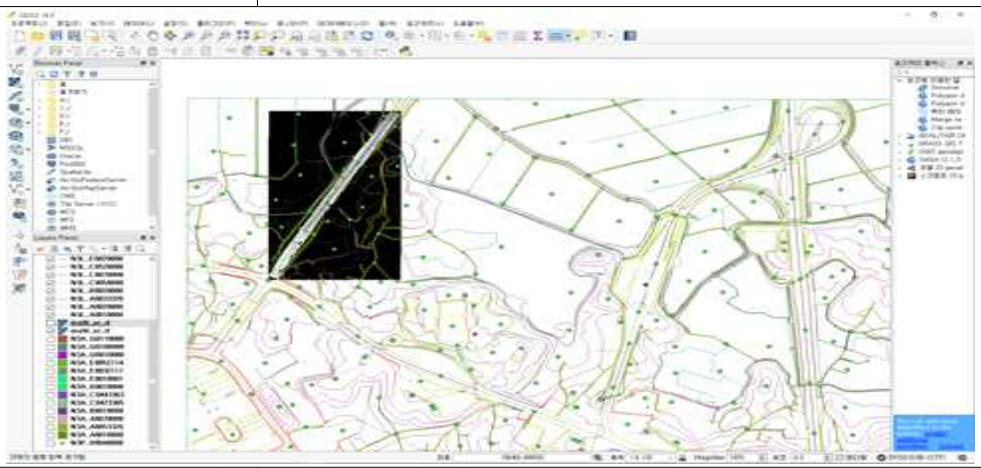
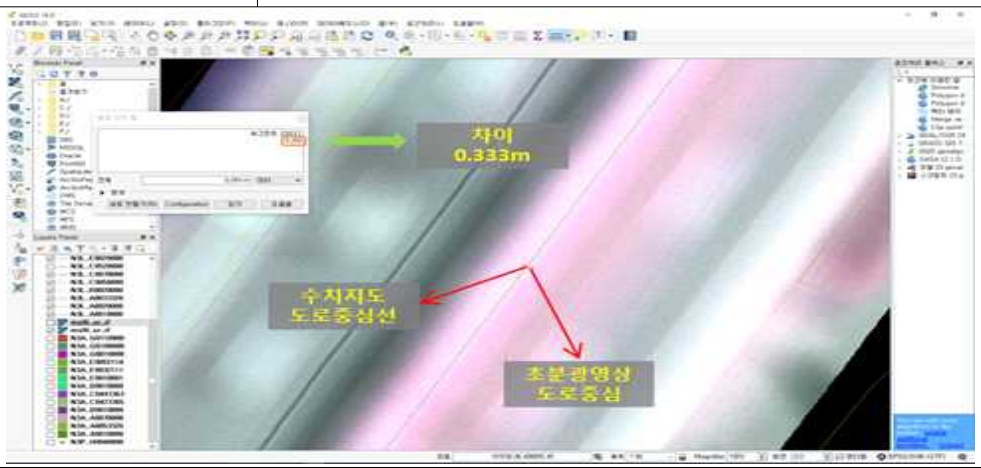
표 2-15 경기도 안성시 원곡면 월정리 산96-1 현장조사(1차선도)

표 2-16 경기도 안성시 원곡면 월정리 산96-1 현장조사(2차선도)

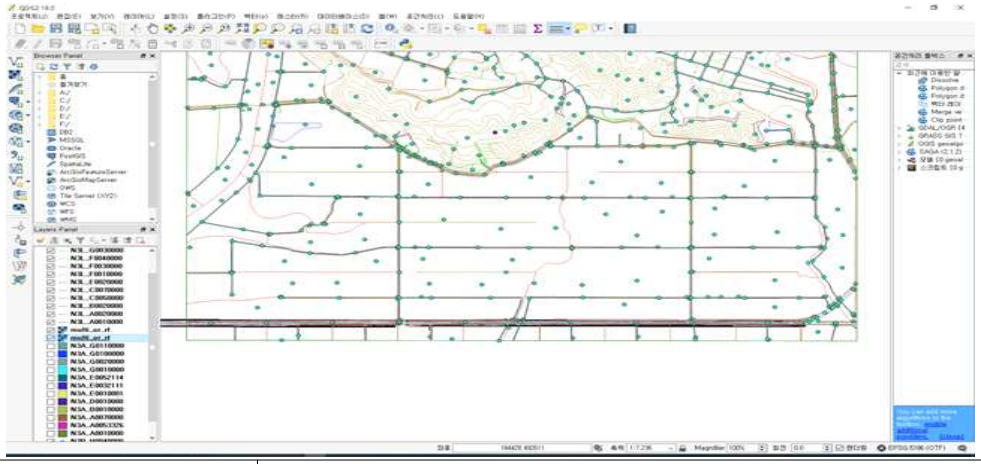
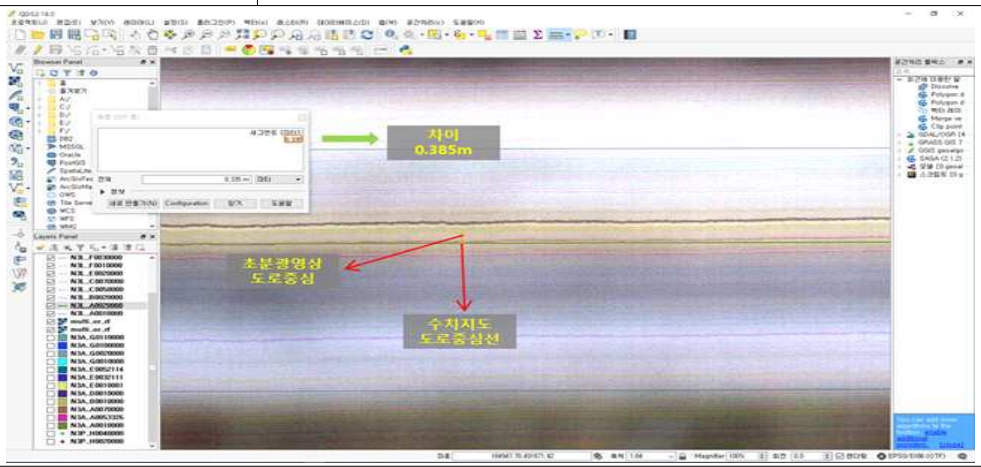
○ PCI 측정 위치 가시화 정확도 분석서

- 드론초분광시스템 기반으로 취득한 초분광영상의 PCI 측정을 위한 위치정확도 검증을 위해 국토정보플랫폼에서 제공되어지는 국가기본도(1:5,000)를 취득하였으며, 이를 통해 초분광영상의 위치 정확도는 평가함
- 평가는 수치지도의 도로중심선(N3L_A0020000), 인도(도로-인도 경계) 경계선(N3L_A0033320) 레이어를 통해 위치 정확도 검증을 수행함
- 이를 통해 3차년도 구축된 초분광영상의 위치정확도는 **약 0.359m**의 수준으로 본 과제 PCI 측정을 위한 위치정확도 최종목표치인 0.5m 이내로 목표를 달성하였음

[경기도 평택시 포승읍 도곡리 170-76 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 평택시 포승읍 도곡리 170-76
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준 초분광영상의 위치정확도는 **0.333m 수준**임	

[경기도 화성시 장안면 노신리 1416 위치 정확도 검증]

Overlap 수치지도-초분광영상	
소재지	경기도 화성시 장안면 노신리 1416
	
위치정확도 검증	
	
도로중심선(N3L_A0020000) 기준 초분광영상의 위치정확도는 **0.385m 수준**임	

○ PCI 측정 드론 테스트베드

- 100km이상 PCI 측정 포장 초분광 DB 구축을 위해서는 세부적으로 초분광데이터 구축, 초분광데이터 가공으로 나뉨
- (초분광데이터 구축) 도로포장상태 평가를 위한 드론 초분광시스템 영상 취득을 말하며, 절차는 촬영계획, 촬영허가 신청, 초분광센서 촬영, 촬영성과 확인 순서로 진행함

· 촬영계획 단계에서는 한국건설기술연구원에서 운영 중인 PES장비를 토대로 취득된 도로포장상태지수(PCI)가 존재하는 지역을 우선으로 사전답사 및 주변 환경에 따른 드론 운영 유무를 판단하여 촬영 대상지를 선정하는 단계임, 대상지 선정 시 PCI이 존재하는 대상지를 선정하는 이유는 드론 초분광시스템으로 산출된 PCI을 검증하는데 목적이 있음

· 촬영허가 신청 단계에서는 해당 지역에 대하여 보안 및 안전에 관련된 허가를 미리 승인 받는 단계로써 과업의 수행 시 상공에서 발생될 수 있는 안전문제와 보안지역에 대한 촬영에 대하여 차질이 없도록 함에 목적이 있음

· 초분광영상 취득은 센서 특성상 태양 고도에 따른 입사각 변화에 대해 아주 민감하게 반응하며, 이는 직접적으로 초분광영상의 신뢰도에 영향을 미치므로 탄력적인 운행이 필요함에 따라 영상촬영 시간은 오전 9시 ~ 오전 11시, 오후 2시 ~ 오후 4시에 진행하였으며, 태양의 조도 및 입사각의 변화가 발생하지 않게 직선도로를 기준으로 단일경로 비행을 수행함

· 초분광영상 보정 및 전처리 단계에서는 기존의 다중분광영상자료와 달리 수많은 분광밴드를 가지고 있기 때문에 획득된 자료를 분석에 활용하기 위해 오차를 보정하는 방사보정, 기하보정 등 전처리 과정을 반드시 수행되어야 함



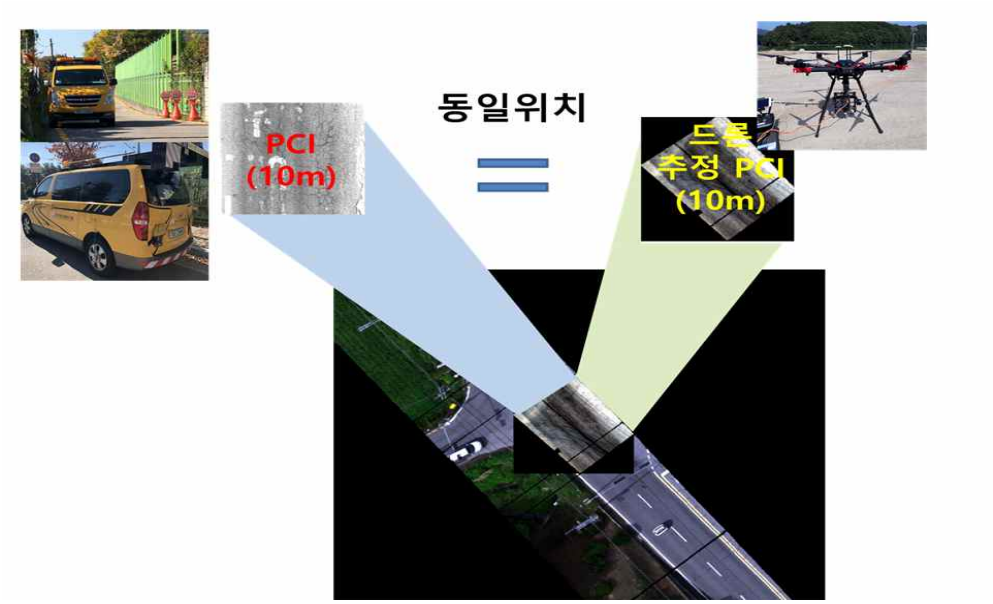
[드론 초분광센서 영상 취득 절차]

· (1차대상지 선정) 드론 탑재형 초분광센서를 통해 취득한 PCI의 활용성 평가를 위해서는 도로의 상태를 객관적으로 나타낼 수 있는 기준값이 필요하며, 또한 다양한 도로환경에 따른 도로 분광라이브러리 특성을 파악하기 위해서는 도로포장상태가 다양한 지역을 대상지로 선정하는 것이 중요함
· 이에 위 조건을 반영할 수 있는 국도 38호선(안성종합버스터미널~일죽IC)를 대상지로 선정하여 한국건설기술연구원 PES 장비로 PCI를 산출하였으며 결과값(소성변형, IRI)을 기준으로 도로상태가 좋은 지역, 안 좋은 지역을 선택하여 1차 대상지로 선정함
· (2차대상지 선정) 도로포장상태지수(PCI)를 기준으로 선정된 1차 대상지에 대해 현장답사를 통하여 드론운영의 적합성 및 비행금지구역 등 환경적 요인을 파악하여 최종 도로조사 구간을 선정함

[도로조사를 위한 초분광 시스템 촬영 대상지]

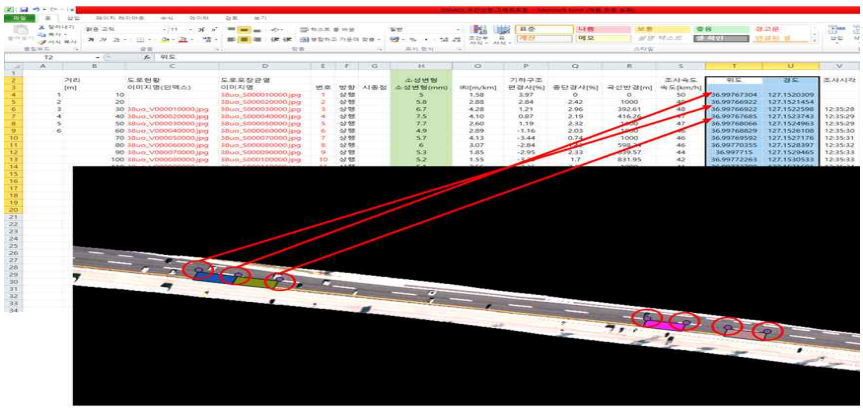
구분	대상지역	비고
1	경기 용인 양지면 추계리 53-2	
2	경기 여주 하동 3-2	
3	경기 여주 가남읍 심석리 100-1	
4	경기 고양 일산서구 대화동 882	
5	경기 시흥 정왕동	
6	경기 안성 일죽면 월정리 719	
7	경기 연천 37호선 국도 자장사거리 일대	휴전선 인근 지역에 따른 비행승인 불가
8	인천 중구 신흥동3가 80	
9	인천 동구 송현동 146	
10	경기 안성 삼죽면 진촌리 307-3(상행)	
11	경기 안성 삼죽면 진촌리 307-3(하행)	
12	경기 안성 죽산면 두현리 109-1	
13	경기 평택 포승읍 도곡리 170-76	
14	경기 화성 장안면 노진리 1416	

- (초분광데이터 가공)
· 기존 도로포장상태를 정량적으로 나타내는 지수인 PCI는 도로의 10m 단위로 수치값이 주어지고 있으며, 이를 바탕으로 도로포장 유지보수 여부를 판단하고 있음
· 이에 개발된 드론 초분광시스템으로 추정된 PCI의 신뢰성을 판단하기 위해서는 기존의 PCI와 동일한 위치에 도로 분광라이브러리를 제공되어야 함



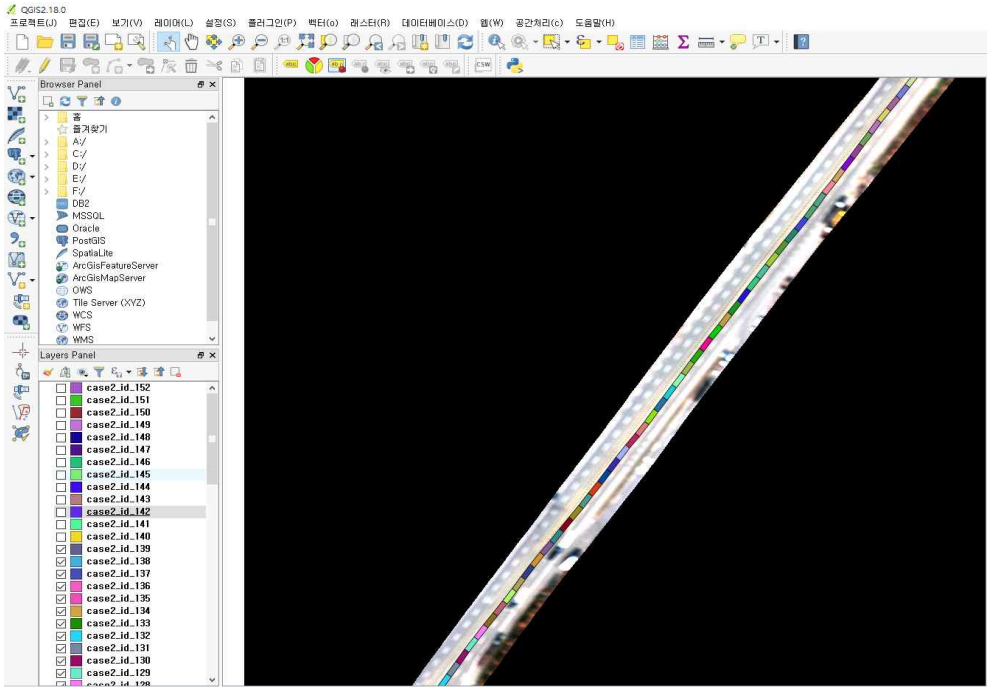
[기존 PCI vs. 드론 기반 추정된 PCI 비교검증]

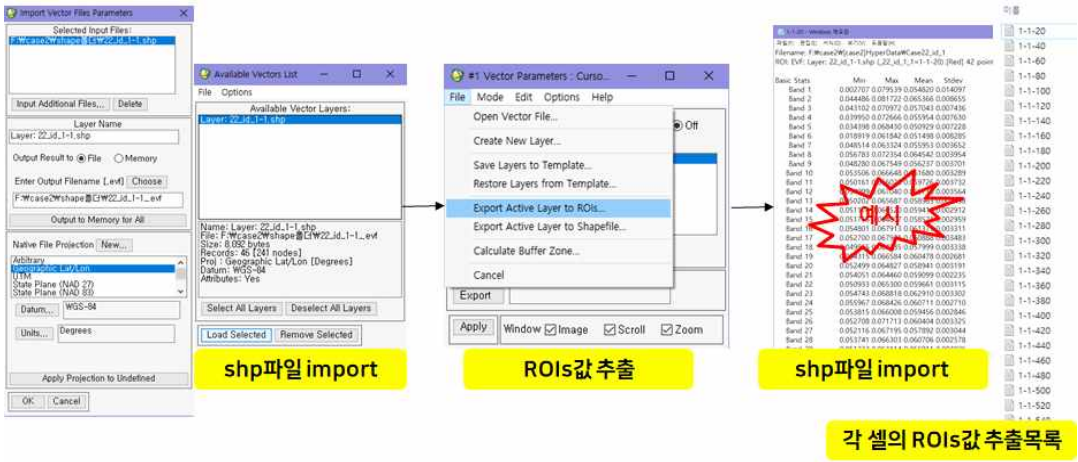
· 도로조사구간에서 기존 PCI의 위치를 알기 위해서는 PES 장비가 도로 데이터 취득 시 차량에 장착 된 GPS를 통해 위치(10m씩)를 아래와 같이 취득하게 됨, 이러한 위치를 기준으로 초분광영상에서 동일한 위치의 도로 분광데이터를 추출함



[PES에 장착된 GPS를 통한 기존 PCI 위치 파악]

· 초분광영상 데이터 가공의 절차는 취득한 초분광영상자료(.hrd) 파일을 백터데이터(.shp) 파일로 변환하여 QGIS SW를 통해 조사된 도로를 10m의 일정한 간격으로 대상지 내 구간분리를 수행한 후 분리한 shp파일을 이용하여 envi 프로그램 내에서 각 셀의 분광영역대 ROI값을 min, max, mean 등 으로 추출하여 PCI 추정에 활용하였음





[초분광영상 데이터 가공의 절차]

- 초분광 데이터 구축 약 60km, 데이터 가공 약60km로 최종목표인 100km PCI 측정 포장 초분광 DB 구축 목표를 달성하였음

[드론시스템을 활용한 도로포장 초분광데이터 구축 현황]

구분	대상지역(도로차선 개수)	기존 PES DB현황	초분광 DB구축	초분광 데이터	비고
1	경기 용인 양지면 추계리 53-2(5차로)	20.0 km	4km	4km	PES DB 중 최적 구간 선정
2	경기 여주 하동 3-2(4차로)	20.0 km	8km	8km	상동
3	경기 여주 가남읍 심석리 100-1(4차로)	20.0 km	4km	4km	상동
4	경기 고양 일산서구 대화동 882(2차로)	0.2 km	0.8km	0.8km	상동
5	경기 시흥 정왕동(4차선)	0.6 km	4.8km	4.8km	상동
6	경기 안성 일죽면 월정리 719 (2차로)	10.0 km	1.6km	1.6km	상동
7	경기 연천 37호선 국도 자장사거리 일대	10.0 km	-	-	비행승인 불가
8	인천 중구 신흥동3가 80(6차로)	0.7 km	8.4km	8.4km	PES DB 중 최적 구간 선정
9	인천 동구 송현동 146(2차로)	0.5 km	2km	2km	상동
10	경기 안성 삼죽면 진촌리 307-3(상행2차로)	20.0 km	2km	2km	상동
11	경기 안성 삼죽면 진촌리 307-3(하행2차로)		2km	2km	상동
12	경기 안성 죽산면 두현리 109-1(4차선)		4km	4km	상동
13	경기 평택 포승읍 도곡리 170-76(4차선)	-	12km	12km	상동
14	경기 화성 장안면 노진리 1416(2차선)	-	6.6km	6.6km	상동
합계		100.0 km	60.2km	60.2km	PCI 측정 초분광 DB 총 120.4km

(2) 세부 정량적 연구개발성과

[과학적 성과]

□ 논문(국내외 전문 학술지) 게재

번호	논문명	학술지명	주저자명	호	국명	발행기관	SCI여부	게재일
1	Advanced sensor technologies in geospatial sciences and engineering	Journal of Sensors	H.S. Jung	3095901	England	Hindawi	SCI급	19.09.04
2	Nonlinear Modeling of Subsidence From a Decade of InSAR Time Series	Geophysical Research Letters	S.W. Park	e2020GL090814	미국	American Geophysical Union	SCI급	20.12.29
3	Development Of A Pothole Detection Result Management System In Road Aerial Images	JP Journal of Heat and Mass Transfer	한윤성	P.77 - 81	인도	Pushpa Publishing House	SCOPUS	20.06.
4	Instantaneous Incident Detection System Based on Analysis of Acoustic Signal from Crash and Skid in Tunnel	The Open Transportation Journal	장진환	2018, Vol 12	아랍에미리트	Bentham Open	비SCI	18.12.31
5	Real-Time Traffic Information Provision Using Individual Probe and Five-Minute Aggregated Data	J. Korea Inst. Intell. Transp. Syst.	장진환	2019. Vol.18	한국	한국ITS학회	KCI	19.02.22
6	Acoustic Signal-Based Tunnel Incident Detection System	J. Korea Inst. Intell. Transp. Syst.	장진환	2019. Vol.18	한국	한국ITS학회	KCI	19.10.02
7	Data Processing Methods For Real-Time Travel-Time Information: Use Of Dedicted Short-Range Communications Probes On Suburban Arterial	The Open Transportation Journal	장진환	2020, Vol 14	아랍에미리트	Bentham Open	비SCI	20.01.13

□ 국내 및 국제 학술회의 발표

번호	회의 명칭	발표자	발표 일시	장소	국명
1	International Conference on Consumer Electronics 2019	서동만	19.01.11	Las Vegas Convention Center	USA
2	KCC 2019	정연욱	19.06.26	제주국제 컨벤션센터	대한민국
3	2019 Internation Conference on Advanced Computing, Communication, and Information Sciences	한윤성	19.12.13	세종대학교	대한민국
4	2019 IEEE Internation Conference on Consumer Electronics	이성원	19.01.13	Las Vegas Convention Center	USA
5	KCC 2020 (항공 영상 내 물체의 정확한 면적 측정을 위한 외곽선 검출 방법)	한윤성	20.07.04	온라인	대한민국
6	KCC 2020 (딥러닝 서비스 시스템에서의 적정 프로세스 수 예측방법)	한윤성	20.07.04	온라인	대한민국
7	KCC 2020 (항공 영상 내 오브젝트 면적 계산 기법)	김수종	20.07.04	온라인	대한민국
8	KCC 2020 (항공 영상 내 오브젝트의 정확한 GPS좌표 예측 방법 및 정북 표현)	김수종	20.07.04	온라인	대한민국
9	KCC 2020 (R-CNN기반 포트홀 탐지 웹 서비스 시스템 구현)	전민건	20.07.04	온라인	대한민국
10	KCC 2020 (항공영상 기반 포트홀 위치 가시화 웹 서비스 구현 웹 어플리케이션)	김재훈	20.07.04	온라인	대한민국
11	KCC 2020 (항공 이미지에서의 Mask R-CNN 기반 포트홀 검출 연구)	전민건	20.07.04	온라인	대한민국
12	KSC 2020	한윤성	20.12.23	온라인	대한민국
13	12th ARC of IAEG Asian Regional Conference of IAEG	홍상훈(부산대)	19.09.26.	제주부영호텔	대한민국
14	12th ARC of IAEG Asian Regional Conference of IAEG	김상와(세종대)	19.09.26.	제주부영호텔	대한민국
15	EGU	홍상훈(부산대)	21.04.26.	온라인	유럽
16	EGU	박서우(부산대)	21.04.28.	온라인	유럽
17	EUSAR	홍상훈(부산대)	21.03.31.	온라인	독일
18	ISRS	주정헌(부산대)	21.05.27.	온라인	대한민국
19	ISRS	이제윤(부산대)	21.05.27.	온라인	대한민국
20	2018 대한토목학회 정기학술대회	이석진	2018.10.18	경주 화백컨벤션센터	대한민국
21	2021 한국지반공학회 정기학술대회	이석진	2021.03.18	서울 aT센터	대한민국
22	2021 대한지질공학회 정기학술대회	김봉찬	2021.04.08	제주 캠퍼트리 호텔	대한민국
23	2018 공동추계 학술대회(지리정보학회+지형공간정보학회+공간정보학회)	함건우(신한)	18.11.02.	제주대학교 아라컨벤션홀	대한민국
24	2019한국BIM학회정기학술대회	배경호(신한)	19.06.13	건설회관(서울 논현동)	대한민국
25	2019 KAGIS&ICSANE 추계학술대회	함건우(신한)	19.11.01	제주대학교 아라컨벤션홀	대한민국
26	한국BIM학회 정기학술대회	배경호(신한)	20.07.03	건설회관(서울 논현동)	대한민국
27	2020 한국측량학회 정기학술발표회	김동한(신한)	20.07.03	온라인 개최(Zoom)	대한민국
28	TRB(Transportation Research Board) 98th Annual Meeting	장진환(건설연)	20.01.13	미국, 워싱톤	대한민국

[기술적 성과]

□ 지식재산권(특허, 실용신안, 의장, 디자인, 상표, 규격, 신품종, 프로그램)

번호	지식재산권 등 명칭 (건별 각각 기재)	국명	출원				등록				기여율
			출원인	출원일	출원번호	등록번호	등록인	등록일	등록번호	등록번호	
1	도로 사면 점검을 위한 드론 운용 시스템 및 방법	대한민국	메타빌드	18.12.10	10-2018-0158369	214-86-27387	메타빌드	20.07.30	10-2141667	214-86-27387	100%
2	드론을 이용한 도로 사면 점검을 위한 드론 비행경로 생성 시스템 및 방법	대한민국	메타빌드	19.12.23	10-2019-0173066	214-86-27387	메타빌드	21.02.15	10-2217627	214-86-27387	100%
3	지표면 경사 감지장치 및 감지시스템	대한민국	롯데건설	18.10.16.	10-2018-0123181	114-81-16377	롯데건설	20.02.12	10-2078929	114-81-16377	100%
4	험지 설치용 경사 감지장치	대한민국	롯데건설	19.11.15	10-2019-0146504	114-81-16377	롯데건설	20.03.30	10-2235814	114-81-16377	100%
5	무인비행기를 이용한 지표면 변위 감지 시스템	대한민국	롯데건설	20.08.13	10-2020-0101540	114-81-16377					100%
6	무인비행기를 이용한 비탈면 모니터링 방법	대한민국	롯데건설	20.09.04	10-2020-0112811	114-81-16377					100%
7	간이 측정용 스마트 폴 드론	대한민국	공간정보	18.11.20	10-2018-0143178	412-81-30180	공간정보	20.06.10	10-2123367	412-81-30180	100%
8	3차원 안전정보 측정을 위한 드론 탐지시스템	대한민국	공간정보	18.11.20	10-2018-0143180	412-81-30180	공간정보	20.06.10	10-2123360	412-81-30180	100%
9	무인비행체와 5G 통신시스템의 연계용 라이브맵 생성방법 및 시스템	대한민국	공간정보	19.09.20	10-2019-0115951	412-81-30180					100%
10	댐유역 및 수자원시설물의 상시 자동점검을 위한 인공지능 드론 무인스테이션 시스템	대한민국	공간정보	20.10.30	10-2020-0143598	412-81-30180					100%
11	도로 이미지에 기반한 포트홀 탐지 시스템 및 방법	대한민국	대구가톨릭대	19.11.25	10-2019-0151996	515-82-06711					100%
12	딥러닝 기반 항공 영상 내 포트홀 탐지 결과 시각화 장치 및 그 방법	대한민국	대구가톨릭대	21.06.09	10-2021-0074779	515-82-06711					100%
13	무인비행체에 탑재된 합성구경레이더 센서를 활용한 비탈면 변위 측정 시스템 및 그 방법	대한민국	건설연	18.06.11	10-2018-0066628	229-820-01135	건설연	21.01.18	10-2206161	229-820-01135	100%
14	무인비행체에 탑재된 비전센서를 활용한 포트홀 탐지 시스템 및 그 방법	대한민국	건설연	18.07.12	10-2018-0080961	229-820-01135					100%
15	시설물 점검을 위한 드론 관제 시스템 및 그 방법	대한민국	건설연	19.10.01	10-2019-0121717	229-820-01135	건설연	20.12.11	10-2192686	229-820-01135	100%
16	분광 데이터와 노면온도센서 데이터를 이용한 노면상태 분류 시스템 및 그 방법	대한민국	건설연	20.10.20	10-2020-0135800	229-820-01135					100%
17	CCTV 연동 포트홀 실시간 모니터링 시스템	대한민국	아와소프트	19.11.27	10-2019-0154267	105-86-57078	아와소프트	21.05.10	10-2252369	105-86-57078	100%
18	스마트폰과 클라우드 서버를 이용한 도로교통 위험정보 수집제공 시스템 및 그 방법	대한민국	건설연	19.11.17	10-2019-0141749	229-820-01135	건설연	21.02.09	10-2116272	229-820-01135	100%
19	드론장착 초분광 카메라용 보정기구	대한민국	신한항업	19.02.01	10-2019-0013779	107-81-74705	신한항업	20.03.02	10-2085628	107-81-74705	100%
20	드론 SAR SLC 영상 제작 방법	대한민국	신한항업	20.11.27	10-2020-0162863	107-81-74705	신한항업	20.06.22	10-2269752	107-81-74705	100%
21	변위 관측을 위한 L-밴드 SAR 센서 탑재 드론	대한민국	신한항업	20.11.27	10-2020-0162850	107-81-74705	신한항업	21.06.22	10-2269751	107-81-74705	100%
22	드론 장착 초분광센서 기반 도로포장상태지수 생성 방법	대한민국	신한항업	20.11.27	10-2020-0162859	107-81-74705					100%
23	무인비행체에 탑재된 초분광 센서를 활용한 포장상태지수 측정 시스템 및 그 방법	대한민국	건설연	18.07.20	10-2018-0084478	229-820-01135	건설연	20.05.29	10-2119242	229-820-01135	100%
24	적외선 카메라와 적외선 조명을 이용한 버스전용차로 위반 차량 단속 시스템 및 그 방법	대한민국	건설연	18.09.27	10-2018-0114671	229-820-01135					100%
25	교통 데이터를 이용한 실시간 도로변 대기오염 추정 시스템 및 그 방법	대한민국	건설연	19.10.01	10-2019-0121711	229-820-01135	건설연	21.02.24	10-2222381	229-820-01135	100%
26	웹 GIS기반 지능형 도로 위험도 관리 시스템 및 모바일 정보 알림 서비스	대한민국	아와소프트	18.11.21	10-2018-0144311	105-86-57078	아와소프트	20.10.22	10-2171295	105-86-57078	100%
27	중력가속도 센서와 휠스피드 센서를 이용한 노면 미끄럼정보 제공 시스템 및 그 방법	대한민국	건설연	19.12.11	10-2019-0164465	229-820-01135					100%

□ 저작권(소프트웨어, 서적 등)

번호	저작권명	창작일	저작자명	등록일	등록 번호*	저작권자명	기여율
1	비탈면 측정을 위한 임무계획 소프트웨어	19.11.04	메타빌드	19.12.02	C-2019-036791	메타빌드	100%
2	자동운항 기반 비탈면 비행임무계획 소프트웨어	20.06.05	메타빌드	20.06.24	C-2020-020751	메타빌드	100%
3	영상 레이더 위상 모사 프로그램	18.10.30	홍상훈,김상완	18.12.00	C-2018-037072	부산대학교	100%
4	영상레이더 위상 영상 제작 프로그램	19.06.30	박서우,홍상훈	19.10.31	C-2019-030941	부산대학교	100%
5	비선형 시계열 변위 분석 프로그램	20.06.02	박서우,홍상훈	20.10.28	C-2020-037956	부산대학교	100%
6	쌍곡선법 및 호시노법을 이용한 비선형 침하 추정 프로그램	20.02.03	박서우,홍상훈	20.11.13	C-2020-041604	부산대학교	100%
7	영상레이더 진폭 영상 제작 프로그램	19.04.30	박서우,홍상훈	19.10.31	C-2019-030940	부산대학교	100%
8	레벤버그-마쿼트 최적화 기법을 통한 시계열 변위 분석 프로그램	21.02.01	박서우,홍상훈	21.03.10	C-2021-011846	부산대학교	100%
9	딥러닝 활용한 항공영상 내 포트홀 탐지(KK PotDet v1.2)	18.11.20	박근형,장진환	18.11.26	C-2018-034909	건설기술연구원	100%
10	항공 영상 기반 도로내 포트홀 탐지를 위한 Accuracy 및 Recall Precision 판단 프로그램	19.08.30	서동만	19.11.11	C-2019-032806	대구가톨릭대	100%
11	항공 영상 기반의 딥러닝 결과 출력 프로그램	19.04.08	서동만	19.11.11	C-2019-032807	대구가톨릭대	100%
12	항공 영상 기반 도로내 포트홀 탐지를 위한 딥러닝 학습 프로그램	19.04.08	서동만	19.11.11	C-2019-032808	대구가톨릭대	100%
13	딥러닝 활용한 항공영상 내 포트홀 탐지(KK PotDet v2.1)	19.09.18	장진환,박근형	19.11.01	C-2019-031198	건설기술연구원	100%
14	항공 영상기반 오브젝트 외곽선 추출 프로그램	20.10.15	서동만	21.01.04	C-2021-000248	대구가톨릭대	100%
15	항공 영상 내 포트홀 병렬 추론 프로그램	20.04.24	서동만	21.01.04	C-2021-000247	대구가톨릭대	100%
16	항공 영상 기반 상대 위치 계산 프로그램	20.07.23	서동만	21.01.04	C-2021-000246	대구가톨릭대	100%
17	이미지 헤더 정보를 이용한 면적 계산 프로그램	20.06.12	서동만	21.01.04	C-2021-000245	대구가톨릭대	100%
18	도로 영역 추출 모델 학습 프로그램	20.11.20	서동만	21.01.04	C-2021-000244	대구가톨릭대	100%
19	항공영상에서 도로포트홀 탐지(DronePotDet3.1)	20.05.18	박근형,장진환	20.08.31	C-2020-027239	건설기술연구원	100%
20	항공이미지 포트홀특징도출SW(ImgPothole.v1.1	18.11.06	박근형,장진환	18.12.13	C-2018-037944	건설기술연구원	100%
21	항공이미지 포트홀특징도출SW(ImgPothole.v1.3)	19.10.04	박근형	19.11.01	C-2019-031200	건설기술연구원	100%
22	계측자료 검증 매핑(KK 2D v2.2)	19.10.15	박근형	19.11.01	C-2019-031201	건설기술연구원	100%
23	포트홀특징 이미지도출(PotholeImgEx3.1)	20.07.13	박근형,장진환	20.08.31	C-2020-027238	건설기술연구원	100%
24	초분광 결과 분석 프로그램(AROD.HSA.v3)	18.10.31	정규동	18.11.26	C-2018-034908	건설기술연구원	100%
25	초분광 결과 분석 프로그램(AROD.HSA.v3.2)	19.09.27	정규동	19.11.01	C-2019-031197	건설기술연구원	100%
26	도로PIC초분광정보분석(RoadPciAnal3.1)	20.05.27	박근형,장진환	20.08.31	C-2020-027240	건설기술연구원	100%
27	도로포장상태정보취합(PaveStatManager2.2)	20.07.30	박근형,박상규	20.08.31	C-2020-029145	건설기술연구원	100%
28	포장 포트홀 및 PCI 가시화 프로그램	19.11.15	아와소프트	19.11.25	C-2019-035247	아와소프트	100%
29	도로포장PCI등급관리(PavePCILeveler2.1)	20.08.20	박근형,장진환	20.08.31	C-2020-029143	건설기술연구원	100%
30	포장상태지수 수집프로그램(Pave.IAP.v2.3)	19.10.21	박근형,장진환	19.10.21	C-2019-031199	건설기술연구원	100%
31	도로포장상태분석결과 등급관리(PaveLeveler3.1)	20.08.07	박근형,장진환	20.08.31	C-2020-027241	건설기술연구원	100%
32	도로포장계측자료 검증매핑(RoadDataMap3.1)	20.06.10	박근형,장진환	20.08.31	C-2020-029142	건설기술연구원	100%
33	3차원로컬좌표변환(3DCoodiChanger3.4)	20.08.07	박근형,박상규	20.08.31	C-2020-029144	건설기술연구원	100%

[경제적 성과]

□ 시제품 제작

번호	시제품명	출시/제작일	제작 업체명	설치 장소	이용 분야	사업화 소요 기간	인증기관 (해당 시)	인증일 (해당 시)
1	포트홀 탐지 드론시스템	2018.06.12	공간정보	공간정보	포트홀 촬영	2년		
2	초분광센서 장착 드론 시스템 개발	2018.06.01	신한항업	신한항업	드론 초분광 기반 PCI 측정	종료 후 2년		
3	L-Band SAR 장착 드론 시스템		신한항업	신한항업	비탈면 지반 변위관측	종료 후 2년		

□ 기술 실시(이전)

번호	기술 이전 유형	기술 실시 계약명	기술 실시 대상 기관	기술 실시 발생일	기술료 (해당 연도 발생액)	누적 징수 현황
1	통상	딥러닝 기반 항공 영상 내 포트홀 탐지 기술	㈜공간정보	2021.07.14	4,000,000	4,000,000

□ 사업화 계획 및 무역 수지 개선 효과

성과			비탈면 변위측정 드론시스템		
사업화 계획	사업화 소요기간(년)		2년		
	소요예산(천원)		150,000		
	예상 매출규모(천원)		현재까지	3년 후	5년 후
	시장 점유율	단위(%)	현재까지	3년 후	5년 후
		국내			
		국외			
	향후 관련기술, 제품을 응용한 타 모델, 제품 개발계획				
무역 수지 개선 효과(천원)	수입대체(내수)		현재	3년 후	5년 후
	수출				

성과			포트홀 탐지 드론시스템		
사업화 계획	사업화 소요기간(년)		2년		
	소요예산(천원)		150,000		
	예상 매출규모(천원)		현재까지	3년 후	5년 후
			0	100,000	300,000
	시장 점유율	단위(%)	현재까지	3년 후	5년 후
		국내	0%	1%	3%
		국외	0%	0%	0.01%
	향후 관련기술, 제품을 응용한 타 모델, 제품 개발계획		시설물 안전점검 자동 탐지 드론시스템		
무역 수지 개선 효과(천원)	수입대체(내수)		현재	3년 후	5년 후
			0	100,000	300,000
	수출		0	0	100,000

성과			PCI 측정 드론시스템		
사업화 계획	사업화 소요기간(년)		2년		
	소요예산(천원)		150,000		
	예상 매출규모(천원)		현재까지	3년 후	5년 후
	시장 점유율	단위(%)	현재까지	3년 후	5년 후
		국내			
		국외			
	향후 관련기술, 제품을 응용한 타 모델, 제품 개발계획				
무역 수지 개선 효과(천원)	수입대체(내수)		현재	3년 후	5년 후
	수출				

[사회적 성과]

□ 홍보 실적

번호	홍보 유형	매체명	제목	홍보일
1	전시회	DCU 소프트웨어 대전	드론 포트홀 프로젝트	2019.11.28
2	전시회	DCU 소프트웨어 대전	항공 영상 내 물체 면적 계산	2020.10.06
3	영상매체	유튜브	포트홀 탐지 시스템 소개	2020.12.24
4	기술기사	대한토목학회지 KSCE Magazine 2020년 1월호	건설 분야 비탈면 및 시공관리 효율화를 위한 드론 적용 기술 개발	2020.01.01
5	뉴스기사	건설경제 외	롯데, 드론으로 '안전정밀시공' 잡는다	2020.02.13
6	뉴스기사	서울경제 외	드론으로 지표면 확인, 안전정밀 '두토끼' 잡아	2020.09.01

□ 포상 및 수상 실적

번호	종류	포상명	포상 내용	포상 대상	포상일	포상 기관
1	대회입상	KCC 2020 우수발표논문상	우수발표	한윤성, 전민건, 전수빈, 서동만	2020.8.10.	한국정보과학회
2	대회입상	KSC 2020 우수논문상	우수논문	한윤성, 김수종, 전수빈, 서동만	2020.12.22.	한국정보과학회

2) 목표 달성 수준

성과 목표	자체 평가
드론 SAR 데이터 기반 비탈면 변위 측정 및 가시화 (부산대)	○ 목표 – (정성) L-밴드 드론 SAR 시스템을 통한 비탈면 변위 측정 시스템 구현 및 분석기술 개발 – (정량) 논문 1건, SW 2건 – (성능) 변위측정 정확도: 5cm, 변위위치 가시화 정확도: 1m ○ 수행 내용 – (정성) 시계열 SAR 자료 수집/처리분석 기술 및 정밀 변위 분석 및 가시화 기술 개발 완료 • SBAS 시계열 위상간섭기법 기반 비탈면 변위 측정 및 분석 정확도 검증 완료 • Corner Reflector(CR) 제작을 통한 위치정확도 평가 확인 • 구글 어스 기반 시각화 자료 생성, 변위 색상 분포 범례 도시 구현 완료 – (정량) 논문 1건, SW 3건 – (성능) 변위측정 정확도: 평균 0.12–0.16cm/편차 0.03–0.08cm, 비탈면 변위위치 가시화 정확도: 0.4–1.0m (azimuth 방향, 드론 비행 방향) / 1.1–1.4m(range 방향, 레이더 전파 방향) ○ 평가 결과 – 정성, 정량, 성능목표 달성 완료
비탈면 측정 드론 임무계획 소프트웨어 (메타빌드)	○ 목표 – (정성) 자동운항 기반 임무비행계획 소프트웨어 고도화 ○ 수행 내용 – (정성) 가상비행 기반 자동운항 임무비행계획 소프트웨어 기능적·성능적 테스트 수행 • 곡면 형태의 사면을 위한 임무비행계획서 분할 생성 및 임무비행, 촬영기능 테스트 • 다양한 형태의 사면 촬영을 위한 드론의 비행고도 및 카메라 각도 복수 생성 및 임무비행 테스트 • 사면의 위치에 따른 드론의 임무카메라 촬영 방향 제어 테스트 – (정량) 특허등록 1건 ○ 평가 결과 – 정성, 정량적 목표 달성 완료
비탈면 측정기술 평가 및 비탈면 지침 조사 (롯데건설)	○ 목표 – (정성) 비탈면 가시화 정확도 평가/검증 및 비탈면 변위가시화 시스템 활용 지침 – (정량) 특허등록 1건, 특허출원 1건 ○ 수행 내용 – (정성) 비탈면 현장 테스트베드를 운영하고 지상 장비 이용 측정 기준값과 L-SAR 센서 기반 드론 변위 데이터 비교분석 • 비탈면 변위 측정 절대 기준값과 L-SAR 센서 기반 식생부 아래 실제 지표값과 비교 • L-SAR 센서 기반 드론 비탈면 변위 가시화 개발시스템 활용 지침

	작성함 - (정량) 특허등록 1건, 특허출원 2건 ○ 평가 - 정성, 정량목표 달성 완료
포트홀 탐지 포장 영상 DB 구축 (공간정보)	○ 목표 - (정성) 포트홀 DB구축 200km, 포트홀 탐지 비행방법 도출, 접도구역 고려 운항기법 도출, 포트홀탐지 운용 매뉴얼 제작 - (정량) 시제품 1건, 특허출원 3건, 특허등록 1건 - (성능) 포트홀 위치 가시화 정확도 0.5m 이내 ○ 수행 내용 - (정성) 포트홀 탐지 포장 영상 DB구축 216km 달성, 해상도 1cm이하 비행시 포트홀 판독 확인 검증, 8차로 미만 1라인, 8차로 초과 2라인 경로비행방법 도출, 포트홀 탐지 시스템 운용 매뉴얼 제작 완료 - (정량) 시제품 1건, 특허출원 4건, 특허등록 2건 완료 - (성능) RTK/PPK 장착 드론 활용으로 가시화 정확도 0.1m 달성 ○ 평가 - 정성, 정량목표 달성 완료
드론 영상 데이터 기반 포트홀 측정 소프트웨어 (대구가톨릭)	○ 목표 - (정성) 드론 영상 데이터 기반 포트홀 측정 소프트웨어 개발 - (정량) SCI 2건, 특허 출원 1건, 소프트웨어 등록 1건 - (성능) 검지율 80% 이상, 오검지율 10% 이하 ○ 수행 내용 - (정성) 드론 영상 데이터 기반 포트홀 탐지 웹 어플리케이션 개발, 딥러닝 기반 항공 영상 이미지 내 포트홀 탐지 알고리즘 구현, 면적 측정 알고리즘 및 마스크 내 포트홀 영역 추출 알고리즘 개발, 항공 영상 내 포트홀 좌표 계산 알고리즘 개발 - (정량) SCI 1건 심사예정, KCI 1건 게재예정, SCOPUS 1건, 특허 1건 출원 진행중, 소프트웨어 등록 5건 수행 - (성능) 검지율 97.2%, 오검지율 9.8% ○ 평가 결과 - 정성, 정량목표 달성 완료
포트홀 위치 가시화 시스템 (아와소프트)	○ 목표 - (정성) MMS 기반 포트홀 정보 가시화 - (정량) 특허등록 1건 - (성능) 검지된 포트홀의 위치 정확도 0.5m 내에 맵 표출 가시화 ○ 수행 내용 - (정성) 현장 기반 영상 수집 및 딥러닝을 통해 검지된 포트홀 정보의 GIS 및 MMS 기반의 맵에 가시화 및 발생된 포트홀의 상세한 정보 제공 프로세스 실증 - (정량) 특허등록 1건 - (성능) 위치 오차 0.6~0.9m - 정성, 정량목표 달성 완료

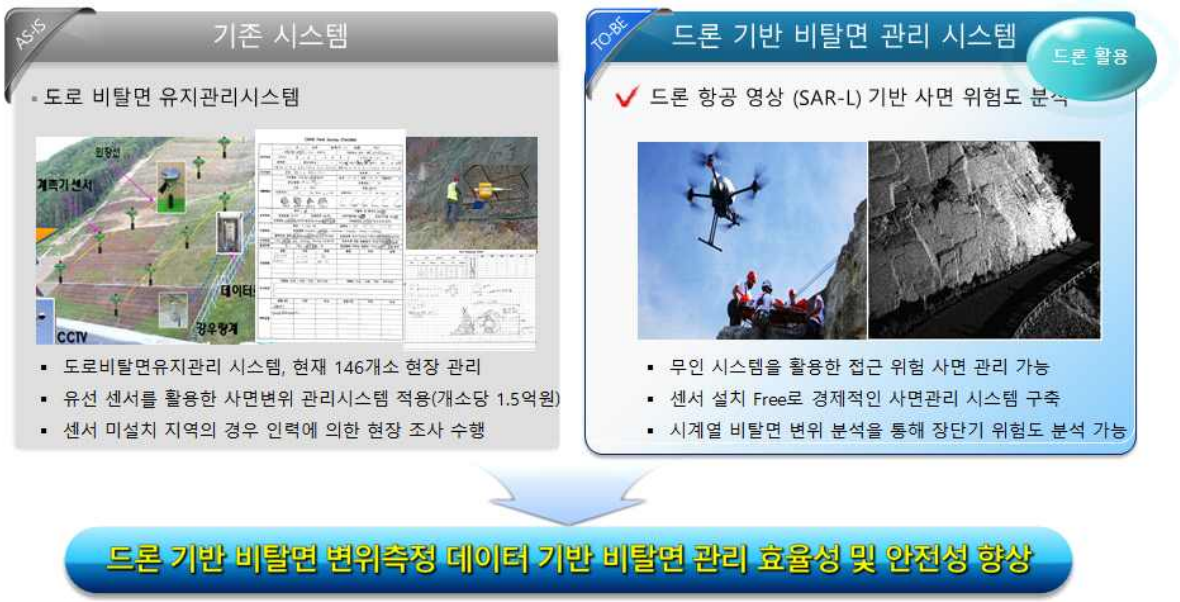
초분광 데이터 기반 PCI 추정 소프트웨어 (건설연)	○ 목표 – (정성) 드론 수집 초분광 센서 데이터 분석 기반 PCI 추정 기술 개발 및 포장상태(PCI) 측정 및 가시화 시스템 활용 지침 개발을 위해 기존 PCI 조사 방법 분석 및 평가 – (정량) 특허출원 1건 – (성능) 초분광 데이터와 PCI 기준값의 상관계수 0.6 이상 ○ 수행 내용 – (정성) 도로포장 평가를 위한 초분광 분석 모델 개선 및 해외 및 국내 포장상태 지수의 개발현황과 국내 도로 포장상태(PCI) 조사 방법 및 활용 현황 분석 • 초분광 계측자료의 자동 분석을 위한 초분광 분석 엑셀 VBA 프로그램 개발 • 아스팔트 포장 조사차량을 이용한 PCI와 개발된 초분광 모델을 이용한 지수값의 상관계수 평가 • 도로 포장 평가 제외 구간 자동 추출을 위한 라이브러리 구축 – (정량) 특허출원 4건, 논문 2건 – (성능) PCI 추정 모델을 적절히 개발하였으며, 이를 드론으로부터 수집된 초분광 데이터와 도로 조사장비로 측정한 PCI로 적합하게 검증하였으며, 상관계수가 목표기준인 0.6보다 높은 0.83 달성함 ○ 평가 – 정성, 정량목표 달성 완료
PCI 측정 포장 초분광 DB 구축 (신한항업)	○ 목표 – (정성) PCI 측정 드론 운용 지침서 작성 – (정량) 특허등록 1건, 특허출원 1건 – (성능) PCI 위치 정확도 0.5m 이내 가시화 ○ 수행 내용 – (정성) PCI 측정 드론 운용 지침서 작성 완료 – (정량) 특허등록 1건, 특허출원 3건 – (성능) PCI 가시화 위치 오차 0.361m 달성 ○ 평가 – 정성, 정량목표 달성 완료
PCI 가시화 시스템 (아와소프트)	○ 목표 – (정성) MMS 기반 포장상태(PCI) 정보 가시화 – (정량) 소프트웨어 등록 1건 ○ 수행 내용 – (정성) MMS 기반의 포장상태(PCI) 가시화를 위한 도로정밀지도를 기본 맵에 매칭작업 수행 및 분석 단위 10m에 대한 포장상태지수(PCI) 산출 및 정보 가시화 – (정량) 소프트웨어 등록 1건 – 정성, 정량목표 달성 완료

4. 연구개발성과의 관련 분야에 대한 기여 정도

가. 비탈면 변위측정 기술

○ L-band SAR 드론 시스템 시험 운영 및 정밀 변위 해석 기술 개발
- 기존 인공위성, 항공기의 경우 원하는 지역을 정확히 또는 자주 관측하는데 비용이나 기술적인 한계가 존재하였으나, 정밀궤도를 기준으로 운행이 가능한 드론의 경우 해당 지역에 대한 반복 관측 운항이 가능하여 위성간섭기법 적용을 통한 비탈면 변위 측정이 가능함을 확인함
- 드론을 이용한 위상간섭기법은 세계 최초로 구현된 기술로서 비탈면 변위측정 뿐만 아니라 미세 지표변위 측정을 요구하는 다양한 국토관리 분야에 활용 가능할 것으로 판단됨
- 또한 개발된 드론 영상레이더 시스템의 기술적 접근 방식은 우리나라에서 처음으로 도입되는 방식으로서 RF 공학, 원격 탐사학, 측지학, 항공우주공학 등 다양한 분야에서 참고자료로 활용될 수 있는 최신 고급 기술에 해당하여 관련 분야 파급력이 상당하다고 사료됨
- 시험 비행 중 획득된 다중시기 SAR 영상 간 상관관계를 나타내는 긴밀도 평가를 완료하였고 변위 및 위치 정확도 분석을 수행함. 평균 0.7-1.0 ㎝, 편차 0.2-0.5 ㎝의 변위 측정 정확도를 분석해내었고 최종 개발 목표치인 5 ㎝ 변위 측정 정확도 요구 성능을 만족하는 결과를 달성하였으며 긴밀도 0.98 이상을 상회하는 높은 값을 보여 우수한 영상레이더 획득 시험임을 입증하여 해당 분야에 기여함

○ 드론 촬영 비탈면 SAR 영상 기반 비탈면 변위 측정 기술 개발
- L-band SAR 데이터의 처리 및 시계열 분석을 통한 정밀 변위 해석을 통한 자료처리 기술을 축적함으로써 드론 시스템뿐만 아니라 영상레이더를 활용한 지표변위 분석 연구 분야에 있어 널리 활용할 수 있는 기초 연구 토대를 마련하였음
- 드론 영상레이더 관측 시스템을 개발하는 과정에서 고정형 지상레이더, Rail형 지상레이더, 차량용 영상레이더 시스템의 시험 평가와 기술을 평가함으로써 관련 기술 개발을 위한 발판을 마련하였다고 판단됨
- 정밀 위치 자료와 정밀 보정 정사영상 자료 생성을 위한 정밀 지상 기준점 비교를 위해 코너리플렉터를 제작하고 실험을 진행함으로써 관련 기반 기술을 획득하였음. 국내에서 해당 기기를 제작하는 경우가 많지 않아 제작 기술과 설치 운용에 대한 노하우가 적은 점을 고려할 때 관련 기기의 제작 시험 운영 경험을 통해 향후 관련 기술 개발이나 항공우주 검·보정 기술 분야에 크게 기여할 수 있을 것으로 판단됨



나. 포트홀 탐지 기술

○ 사업계획서 대비 변경 내용 및 사유

- 본 연구는 드론 영상 데이터 기반 포트홀 측정 소프트웨어, 포트홀 위치 가시화 시스템 및 활용 지침, 포트홀 탐지 및 가시화 결과 평가/검증 보고서 등을 연구하였다. 딥러닝 기반의 포트홀 탐지에서는 분류(Classification)를 위한 Inception v3 모델 망 학습을 수행하였다. 하지만 CNN(Convolution Neural Network) 기반의 Inception v3 모델 망의 포트홀 분류 성능의 한계가 있는 것으로 판단하였고, 각 객체 인스턴스 별로 분할 탐지가 가능한 Instance Segmentation 방식의 모델 망으로 연구 주제를 변경하였다.

○ 드론 영상 데이터 기반 포트홀 측정 소프트웨어

- CNN 기반의 Inception v3 모델 망 기반의 포트홀 탐지 및 커스템 데이터 셋 학습, 탐지 방법론 연구를 진행함. Inception v3 모델 망 기반의 포트홀 탐지 연구는 224p X 224p 크기의 타일 추론 방식과 포트홀 이외의 객체(그림자, 도로 영역, 비도로 영역, 맨홀 뚜껑, 차선) 탐지 방식의 새로운 방법론으로 해당 분야에 기여함
- 포트홀 탐지를 위한 커스텀 데이터셋 학습과 타일 추론 방식은 컴퓨터 비전 알고리즘(오츠 이진화, 슈퍼픽셀 + 컨벡스헐, 슈퍼픽셀 + Wavelet Energy Field, 세일리언시 맵, 배경제거(MOG) 등)을 이용한 탐지 방법론보다 정확한 분류 결과를 도출하여 해당 분야에 기여함
- 이미지 분류 방식인 Inception v3 모델 망의 분류 한계로 인해 인스턴스 별 분할 탐지가 가능한 Mask R-CNN 모델 망을 연구함. 인스턴스 분할 방식의 포트홀 탐지 연구는 이미지 추론 속도 개선을 위한 멀티프로세싱과 데이터셋 불균형 해결을 위한 데이터증강 기법, 오탐지율 개선을 위한 방법론으로 해당 분야에 기여함
- 포트홀 면적 계산 알고리즘 개발 연구는 이미지 픽셀 당 단위 면적을 구하기 위해 드론 영상 센서의 이미지 센서(CCD, CMOS)를 기반으로 지표면과의 상대 고도 값과 렌즈 사이의 거리비를 이용하며, 인스턴스 분할 방식의 포트홀 탐지 결과값을 기반을 단위 면적을 구할 수 있어 해당 분야에 기여함

○ 포트홀 위치 가시화 시스템 및 활용 지침

- 포트홀 위치 가시화 시스템은 구글 지도 API와 인스턴스 별 분할 탐지결과를 기반으로 시스템에 업로드된 이미지의 GPS 좌표와 이미지 센서 데이터 등을 이용해 포트홀이 존재하는 위치와 면적을 가시화해주는 연구를 진행함. 포트홀 위치 가시화 시스템의 매뉴얼(사용 설명서)과 API를 제공함으로써 해당 분야에 기여함

○ 포트홀 탐지 및 가시화 결과 평가/검증 보고서

- 인스턴스 별 분할 탐지를 이용한 포트홀 탐지와 포트홀 위치 가시화 시스템은 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), IoU(Intersection of Union) 지표와 검지율(80% 이상), 오검지율(10% 이하), 탐지 에러율, 포트홀 위치 가시화 정확도(1m 이내)에 대한 보고서 작성을 통해 해당 분야에 기여함

다. 포장 노후화 상태 측정기술

○ 드론 수집 초분광 데이터 처리 및 분석 기술 개발

- 초분광 센서 및 분석 모델을 조사하고, 분광계를 이용하여 아스팔트 바인더 노후화 상태, 포장 표면 텍스쳐, 아스팔트 포장 노화 상태 변화 등에 실험 및 평가하여 대표 파장대를 선정하고 초분광 데이터를 이용하여 아스팔트 포장 상태를 분석하는 기술을 개발하였음.
- 드론으로 측정한 대용량의 초분광 계측자료의 자동분석을 위해 초분광 분석 엑셀 VBA 프로그램을 개발하여 유의확률을 분석하였음. 또한 이를 고도화 하여 인공지능 모델을 이용하여 도로 영역의 자동화 이미지 추출 및 분할 기술을 개발하였고, 도로차선, 노면표지, 차량 등을 자동으로 처리함에 따라 분석 결과의 신뢰성을 높였음.
- 드론 초분광 데이터를 이용하여 대표 파장대의 선정과 분석 방법은 기존의 대용량 초분광 데이터로 인한 문제점을 개선할 수 있는 방법으로 도로 포장 분석은 물론 초분광 측정 분야에 폭넓게 사용될 수 있을 것으로 판단됨.

○ 초분광 분석 데이터와 PCI 기준값 매칭 알고리즘 개발 및 PCI정보 추정 기술 개발

- 도로 포장 평가시 기존에 사용하는 조사차량(PES)을 이용하여 테스트 구간을 조사하여 10m 단위로 구분된 PCI를 산출하였으며, 드론 기반 초분광 데이터를 분석하여 얻어진 10m 단위의 초분광 지수와 상관관계를 분석한 결과 최종적으로 0.9를 달성하여 목표 기준인 0.7을 초과 달성하여 해당 분야에 기여함.
- 드론 초분광을 이용한 아스팔트 포장의 평가 방법은 기존에 초분광 데이터를 이용한 대략적인 노후화 여부의 확인 등에서 크게 개선된 방법으로, 향후 다양한 포장에 적용하기 위한 실용화 연구와 관련 기술 분야에 크게 기여할 수 있을 것으로 판단됨



라. 개발기술 활용에 따른 경제성 분석

○ 개발기술 경제성 분석 및 세부근거 자료

- 개발기술은 현재 운영 중인 비탈면 상시계측시스템(1.5억원/개소) 및 차량식 포장조사장비(PES)를 대체·보완하는 기술로 현재 시스템 대비 경제성 분석 결과 2.25로 분석되어 기술개발에 따른 경제성은 충분히 확보되는 것으로 나타남. 단, 포토홀 측정기술의 경우 현재 비용 등의 문제로 운영되는 비교대상 시스템이 없어 경제성 분석에서 제외하였는데, 포트홀 탐지기술에 대한 편익도 반영할 경우 경제성은 더욱 높아질 것으로 판단됨

☑ 경제성 분석표

단위 : 백만원

구분		2018	2019	2020	2,021	2,022	2,023	2024	2025	2026	2027	2028	2029	2030	합계	B/C
총비용		1,000	1,500	2,000	200	206	212	219	225	232	239	246	253	261	6,793	-
총편익		0	0	0	2,277	2,345	2,416	2,488	2,563	2,640	2,719	2,800	2,884	2,971	26,103	-
현재가치	비용	1,000	1,408	1,763	166	160	155	150	145	140	135	131	127	123	5,603	2.25 (포트홀 미반영)
	편익	0	0	0	0	0	1,763	1,705	1,649	1,595	1,543	1,492	1,443	1,395	12,585	
연차		0	1	2	3	4	5	6	7	8	9	10	11	12	-	-

비용 산출 근거

- 연구개발비용 : 1차년도(10억원), 2차년도(15억원), 3차년도(20억원)
- 드론 활용 비탈면 조사 비용 = 조사비(인건비, 출장비 등)
- 드론 활용 포장조사 비용 = 현행 조사비의 ¼ (국도 4차로 기준)

- 40개소(1개월/드론1식) 연간 조사비 = 15백만원*12개월 ≒ 200백만원
  ※40개소 근거 : 기존 드론 활용 비탈면 조사사례 근거(국토교통부, 2016)
- 연구기간 3년 + 내용연수(드론+센서) 10년 적용
- 사회적 할인율 6.5%(KDI) 및 경제성장률 3%(기제부) 적용

편익 산출 근거

- 드론 활용 비탈면 조사 편익 = 상시계측시스템 구입비
- 드론 활용 포장조사 편익 = 기존 방식 대비 드론 활용 조사비용 절감분

- 비탈면 상시계측시스템(40식) 구입비 = 150백만원/식 * 40 = 6,000백만원
- 기존방식 대비 드론활용 포장조사 비용 절감분 = 드론 활용 포장조사비용은 현행 국도 포장 조사비(연간 1,200백만원)의 ¼ 계상(근거 : 국도 4차로 기준으로 현행 조사차량 조사는 차로별로 조사하지만, 드론은 1회 비행으로 조사 가능)
- 기술기여도(사업화시 R&D가 기여하는 비율) = 33%(근거 : 국제공인기구 LES)

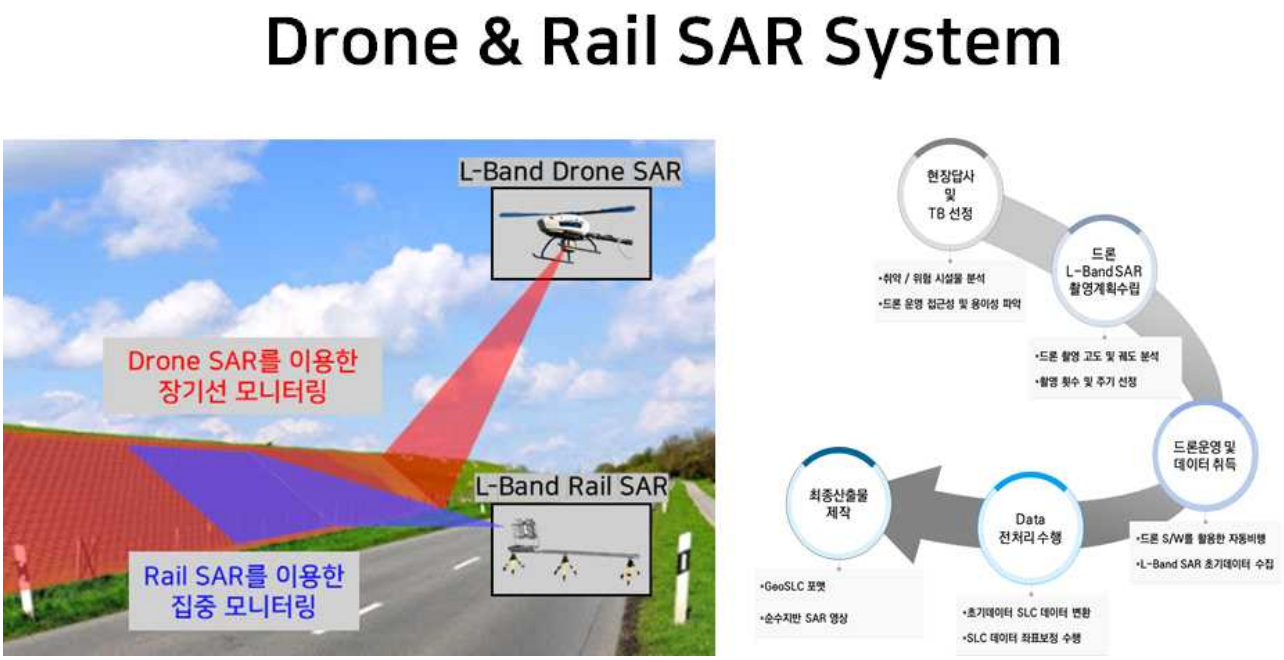
- 도로포장 조사의 경우에는 조사차량을 이용하여 국도의 경우 3년마다 조사를 수행하는데, 조사방식은 모든 차로에 걸쳐 조사차량을 운행을 하고, 조사차량으로부터 수집된 데이터를 인력식으로 분석하는 방법으로 수행되지만, 본 개발기술 적용 시 교통상황의 영향을 받지 않으면서 전체 차로를 한꺼번에 조사할 수 있어 비용 및 시간 측면에서 유리함
- 현행 조사는 재포장이 이루어져 포장상태가 양호한 지역에도 정해진 일정에 따라 조사를 수행하는 방식이기 때문에 불필요한 조사기간이 과다 소요되는 단점이 있지만, 본 개발 기술 적용 시 드론을 활용한 1차 조사(광역/단기간) 후 정밀조사가 필요한 구간(약 20%)만 조사차량을 이용하여 정밀조사를 수행하기 때문에 포장조사의 효율성을 극대화 할 수 있음
- 비탈면 변위 측정기술은 식생의 영향을 받지 않고 비탈면의 정밀변위를 측정할 수 있는 L 밴드(1~2 Ghz) SAR 데이터 처리기술을 세계 최초로 개발하는 것으로 기술개발 성공시 국내와 유사한 비탈면 특성을 나타내는 아시아, 남미 등지에 수출 가능성이 높은 기술임
- SAR-L 센서 데이터 기반 비탈면 변위 측정을 위해서는 레이더 신호의 간섭계 측정이 필수적이고 이를 위해서는 비탈면 특성을 고려한 고도의 안정적인 드론 자동운항 기술 개발이 필요
- 드론 영상 센서 기반 포트홀 측정기술은 포트홀 객체 인지를 위한 고도의 영상처리 기술과 기계학습 기술 개발이 필요하기 때문에 본 연구는 SOC 인프라 계측 기술 개발을 위한 영상처리 기술의 기술적 진보 달성이 가능함
- 포장상태지수(PCI) 측정을 위해 본 연구는 드론 장착 초분광 센서 데이터 처리기술을 개발할 계획인데, 초분광 센서를 활용한 도로포장 노후화 상태 측정은 아직까지 국내에서 개발된 사례가 없기 때문에 본 연구 개발 기술 현장 적용시 국내 포장 상태 측정기술의 다각화 및 고도화가 예상됨

5. 연구개발성과의 관리 및 활용 계획

가. 비탈면 변위측정 기술

○ 기술 개요

- 기후변화에 따라 폭우 및 폭설 등의 증가로 도심 및 도로 비탈면 시설의 노후화 및 붕괴로 위험이 증대되고 있으나 현재 운영하고 있는 비탈면 상시계측시스템은 지점별 설치(1.5억원/개소)로 구축비용이 과다하게 소요되며 다양한 인자(동물, 기상)로 인한 오경보 발생빈도가 높아 효용성이 낮은 실정임
- L-Band SAR는 cm 수준의 정밀 변위 모니터링이 가능한 최신 센서기술로 약 23cm 길이의 파장을 사용하여 투과성을 가져 초목, 관목으로 덮힌 비탈면의 순수지반 데이터 획득이 가능함
- L-Band SAR를 이용하여 다양한 3차원 공간정보 구축 및 시설물 모니터링이 가능하고 탑재체에 따라 드론, Rail, 차량에서 운행되며 유연성 있는 데이터 활용과 고정밀 데이터 구축이 가능함



○ 연구성과 확산 방안

- 1단계 : 연구성과 참여기업 대상으로 기술 확산 및 사업화 기반 마련
- 2단계 : L-Band SAR 기반의 지반 변위관측 체계화를 통한 기술 적용의 제도적 근거 마련
- 3단계 : 국토교통부, 지자체, 한국수자원공사, 한국도로공사 등 발주처를 대상으로 기술 홍보 및 시범사업 실시
- 4단계 : 국내 산업체(도로 관리 및 사면 모니터링관련 기업)을 대상으로 기술 홍보 추진
- 5단계 : 추가연구개발을 통한 기술 고도화

○ 사업화 계획
- 사업화 형태 및 상용화 계획

성과물	지반변위관측
형태/규모	o 상용화 형태 : 모니터링 시스템 구축을 통한 현장적용 및 DB 구축 o 수요처 : 롯데건설, 도로공사, 수자원공사, 지자체 o DB구축 및 변위관측 예상 단가 : 20백만원 o 개발 투입인력 및 기간 : 5명 / 12개월
상용화 능력 및 자원보유	o 주요 수요처와의 협력관계 구축 o 국내건설현장 및 지자체 실증적용 및 활용 방안 제시 o 개발성과품에 대한 현장검증 및 공인 성과 인증 실시 o 공공기관 발주사업과 연계한 사업화 전략 수립
상용화 계획 및 일정	o 주요 수요처의 요구 및 필요 항목 파악 o 다양한 현장 여건 및 환경에 맞춘 프로세스 구축 o 국내 상용화 일정 - 현장검증 및 공인 성과 인증 달성 - 장비전시회 및 홍보 활동 실시 - 주요 수요기관에 대한 시범사업 추진 - 작업 매뉴얼 및 표준 지침서 제공으로 작업 표준화 o 국외 상용화 일정 - ODA 사업 등에서 시범사업 추진 및 홍보 - 장비전시회 및 홍보 활동 실시 - 영문 매뉴얼 및 표준 지침서 작성

- 명확한 수요처 및 요구사항에 기반한 연구추진
· 산사면 및 도로 비탈면 모니터링 기관의 요구와 필요사항에 대한 명확한 분석 및 연구결과 도출
· 한국건설기술연구원 등 기존 업무 대체 방안 제시 및 사업화

- 비탈면 상시계측시스템 대체를 통한 비탈면 조사 업무 방안 제시
· 기존 고가인 비탈면 상시계측장비(1.5억원/개소)는 계측범위 및 정량적 분석에 한계가 있으며, 이러한 이유로 광범위한 도로 비탈면 등에 대한 설치 및 모니터링이 극히 미비한 실정임
· 이에 한국건설기술연구원의 비탈면 관리체계를 보완하고 대체할 기술 방안으로 연구를 추진하며 이에 대한 사업화 추진

성과물	지반변위관측
활용분야	비탈면 모니터링, 도로 시설물 모니터링 등
예상 수요처	한국건설기술연구원, 롯데건설, 수자원공사, 지방자치단체
수요처 요구사항	고가의 비탈면 계측장비 대체로 계측범위 보완 , 운영비 절감 및 모니터링 대상 확대
예상 매출액	**200억원**
사업화 전략	용역 형태 추진

○ 드론활용 비탈면 변위측정 기술 현장 활용 가능 수준
- 본 연구개발을 통해 개발된 드론 기반 영상레이더 비탈면 변위측정 기술은 하드웨어나 소프트웨어적으로 완성도가 매우 높은 기술로서 현장 활용 가능 수준이 매우 높음
- Technology Readiness Level (TRL) 기준 핵심요소기술의 기술적 성숙도 지표로 평가하는 경우 TRL7의 실용화단계인 실제 환경에서의 시제품 데모가 이미 과제를 통해 확인이 되었으며, 상용제품 시험 평가 및 신뢰성 검증 또한 확인되었음
- 사업화단계인 TRL 9는 해당 제품을 우리나라에서 개발할 수는 없는 환경으로 제품 양산의 개념으로 판단할 수는 없으며, 해당 기술을 현장 활용 측면에서 평가할 때 충분히 사업화 가능성이 있다고 판단됨
- 하지만 해당 시스템의 운용 능력은 충분한 비행 훈련, 영상레이더의 비탈면 변위측정 기술 원리 등의 종합적인 이해와 준비가 필요한 사항이 요구됨
- 또한 비탈면 변위측정 기술의 시계열 분석을 위해서는 획득된 자료의 숙련된 자료처리 기술과 해석 경험이 요구됨

○ 비탈면 파괴 유형별 SAR 센서 활용 변위측정 범위 및 실용성 검토
- 본 연구개발을 통해 개발된 드론 기반 영상레이더 비탈면 변위측정 기술은 성과목표로 제시된 5 cm를 충분히 상회하는 높은 정밀도로 측정 가능함을 보임
- 영상레이더 위상간섭기법은 지표 피복이 급격하게 변화하게 되면 해당 영상 사이의 긴밀도가 저하되거나 비상관성(de-correlation)이 급격하게 증가하여 변위측정이 불가능하므로 관측 주기 설정에 있어 주의가 요구됨
- 기 완공된 비탈면의 주기적인 관측으로 인한 미세 지표 변위에 매우 효과적인 기술이며, 굴착 등으로 인한 파괴, 낙석 등으로 인한 비탈면 변위 측정 적용은 다소 제한적임
- 드론을 이용한 영상레이더의 관측 범위는 드론 고도에 따라 달라질 수 있으며, 일반적으로 전파의 범위는 수 km 내외에서 운용 가능할 수 있음
- 다만 전파의 거리가 먼 경우, 신호의 감쇠현상이 심해지고, 주변 환경으로부터 돌아오는 잡음신호가 많아 신호 대 잡음비를 떨어뜨리게 되므로 가급적 목표 대상체에서의 과도한 원거리 측정은 다소 제한적인 성능을 보일 수 있음
- 비탈면 파괴 측정을 위한 영상레이더의 위치 정확도는 대상 지역에서 기 구축된 수치표고모형(digital elevation model: DEM) 또는 수치피복모형(ditigal surface model: DSM)의 공간해상도와 정확도에 의존함
- 비탈면 파괴 유형별 SAR 센서 활용 변위측정 오차는 대상 지역의 특성에 따라 수치표고모형의 갱신여부, 안개, 구름 등으론 인한 습도 등 기상상태, 유전율이 높은 구조물의 여부, 전파방해 여부 등 다양한 원인에 영향이 발생할 수 있으며, 이의 저감을 위해 관측일시, 전자기파 편파 선택 등 주의 깊은 관측 설계가 요구될 수 있음

나. 포트홀 탐지 기술

○ 기술 개요

- 해마다 아스팔트 포장의 포트홀 발생 건수와 민원은 증가하고 있는데 반해, 포트홀 조사에 대한 방법은 전무한 실정이며 민원에 의존하고 있음
- PPK/RTK 장착 드론은 사진의 정확도가 RMSE PCI 지수의 산출 특성을 고려한0.5m 이내로 포트홀의 위치를 정확하게 조사할 수 있으며, 많은 경험 및 실증을 토대로 제작한 매뉴얼을 통하여 간단한 교육으로 누구든지 쉽게 포트홀 데이터를 수집할 수 있음
- 수집된 데이터는 웹기반 AI분석 시스템에 업로드를 하여 자동으로 포트홀 의심 지역에 대한 위치 및 크기, 위험도를 분석하고, 인위적인 필터링을 거쳐 최종 가시화시스템에 전송하여 보수 작업계획을 세우는데 활용할 수 있음
- 또한 추가적인 연구개발로 AI분석시스템을 개선하여 균열, 누수, 박리, 철근노출 등을 학습하여 시설물 안전점검에 활용할 수 있음



○ 연구성과 확산방안

- 1단계 : 연구성과 참여기업 대상으로 기술 실시계약 및 기술이전을 통한 사업화 기반 마련
- 2단계 : 드론활용 포트홀 탐지기술의 조사방법 체계화를 통한 기술 적용의 제도적 근거 마련
- 3단계 : 국토교통부, 지방청, 지자체, 한국도로공사 등 발주처를 대상으로 기술 홍보
- 4단계 : 국내 산업체(포트홀 유지보수관련 기업 등)을 대상으로 기술 홍보 및 기술이전 추진
- 5단계 : 추가연구개발을 통한 기술 고도화 및 시설물 점검 분야로 확대

○ 사업화 계획
- 국내 드론기술의 발전과 관련 법·제도의 완화 및 정부의 드론산업 활성화 정책으로 성장과 더불어 치열한 경쟁이 예상되고 있으며, 드론관련 기술서비스 시장의 선점이 필요함
- 비즈니스 모델은 드론, AI, IOT, 지리정보시스템(GIS) 등의 4차산업 기술의 융합을 통하여 점검분야의 디지털데이터로 구축하여 관리용 정보서비스를 제공하는 것임
- 이를 통해 각종 안전사고 및 민원발생 건수를 최소화 하고, 신속 보수 체계마련 등을 목표로 하고 있으며, 드론영상과 AI기술 융합데이터를 이용한 신개념 유지관리 서비스를 제공함으로써 신시장을 개척한다는 사업모델을 설정함
- 드론 하드웨어 기술의 급속한 발전과 AI기술을 접목한 융합데이터를 활용한 기술은 각종 산업분야의 저변확대와 기술 발달에 따른 서비스가 가능해졌고, 이러한 스마트 유지관리 시스템은 포트홀조사의 혁신적인 변화를 주도할 수 있음
- 사업화의 형태는 드론과 시스템의 판매, 용역서비스가 있으며, 공공기관 및 유지보수관련 산업체, 정부출연기관, 등 폭넓은 수요처가 존재함

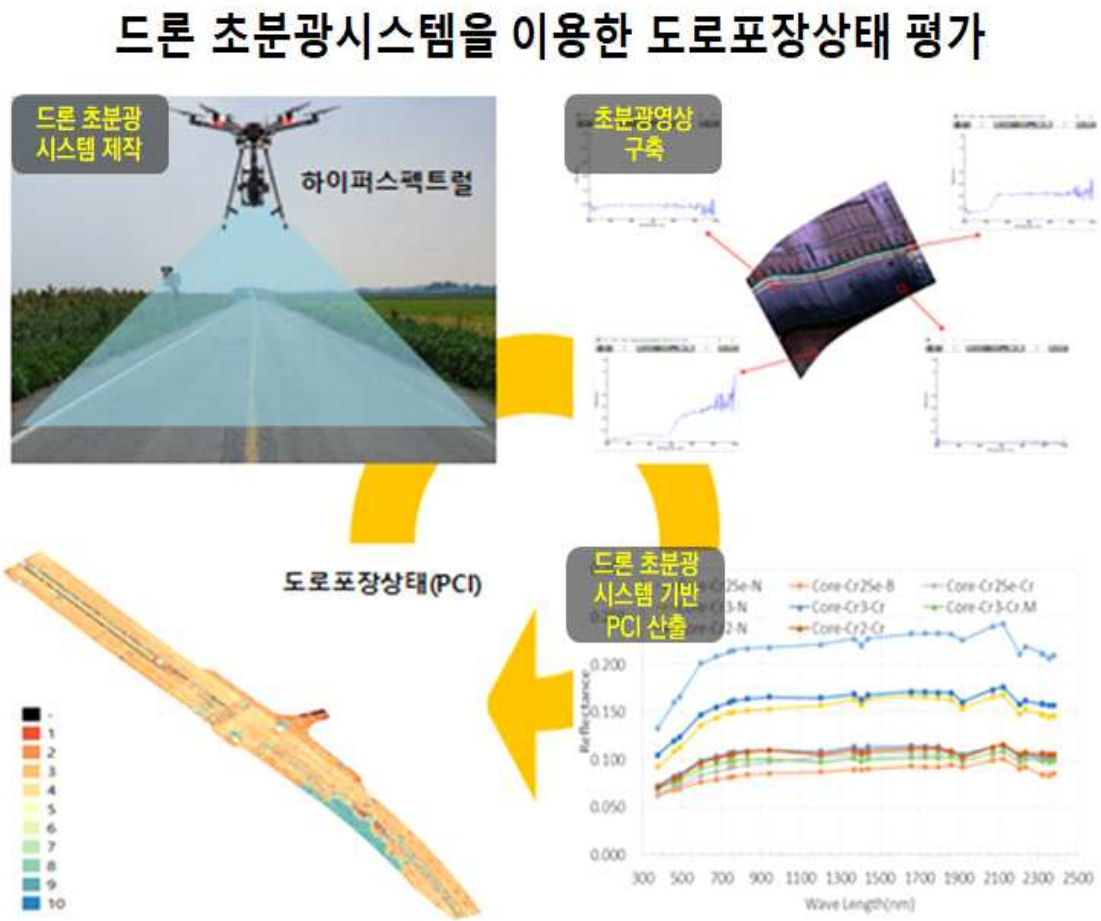
구 분	포트홀 탐지 시스템	
활용분야	포트홀 현장 조사, 시설물 점검, 시계열 모니터링, 등	
판매	드론판매 : 일반형(RTK/PPK), 사용자 맞춤형 시스템 판매 : 라이센스 이용권	
서비스	용역 : 촬영, 촬영+분석	
가격	판매	용역 서비스
	드론(회전익) : 1,200 만원 드론(고정익) : 4,500 만원 시스템 : 500만원(1년)	촬영 : 5㎢당 75만원(인건비) 촬영+분석 : 5㎢당 150만원(인건비)

※상기 가격정책은 드론의 종류, 시스템, 용역서비스의 형태에 따라 달라질 수 있음

○ 드론 활용 포트홀 탐지기술 현장 활용 가능 수준
- 본 연구를 통해 약 30~70m 고도에서 촬영한 항공 영상 기준 포트홀 검지율이 97.2%이고 오검지율이 9.8%인 포트홀 탐지 기술 개발
- 또한, 포트홀 자동 탐지 웹어플리케이션 개발과 지도 위 포트홀 위치시각화, 포트홀 면적 측정 기능, 좌표 측정 기능, 정북 방향 표시 등을 개발하여 포트홀 보수자의 편의성 증가
- 기름 자국, 그림자, 맨홀 등의 포트홀과 유사한 특징의 물체를 오검지할 수 있어 포트홀 점검의 보조적인 역할로써 현장에 활용 가능
- 차량을 이용한 포트홀 탐지에 비하여 비교적 넓은 영역에 분포한 포트홀을 빠르게 탐지 가능
- 현장 적용 후, 지속적인 포트홀 데이터 수집을 통해 포트홀 검지/오검지율의 점진적 성능 개선 가능

다. 포장 노후화 상태 측정기술

○ 기술개요
- 도로 관리를 위해 다양한 기술이 개발되고 적용되고 있음. 특히 차량 안전과 직결되는 도로포장에 관한 탐지 및 조사는 국민의 안전 확보를 위해 가장 우선적으로 개발되어야 함
- 현재 도로포장상태와 품질조사를 위해 한국건설기술연구원, 도로공사, 지자체 등은 많은 예상과 인력을 투입하여 조사하고 있음. 하지만 PES 장비는 대당 5억원 이상으로 고가의 장비로 도로 접촉방식의 자료처리의 시급성이 떨어지는 문제를 가지고 있음
- 드론 초분광시스템은 분광반사 특성을 이용하여 도로상태 및 도로포장 노후화를 모니터링이 가능한 최신 센서 기술임
- 드론 초분광시스템은 2파장폭이 약 10㎚ 또는 400~1,000㎚인 수십~수백 개의 연속적인 밴드로 취득한 영상을 분석하며 도로포장상태에 따른 표준 분광라이브러리 구축이 가능함
- 저비용의 드론 초분광시스템을 운영함으로써 효율적인 도로포장 품지 조사 체계로 활용할 수 있음



○ 연구성과 확산방안
- 1단계 : 연구성과 참여기업 대상으로 기술 확산 및 사업화 기반 마련
- 2단계 : 드론 기반 도로포장상태 조사방법 체계화를 통한 기술 적용의 제도적 근거 마련
- 3단계 : 국토교통부, 지방청, 지자체, 한국도로공사 등 발주처를 대상으로 기술 홍보 및 시범사업 실시
- 4단계 : 국내 산업체(도로 관리 및 엔지니어링 기업 등)을 대상으로 기술 홍보 및 기술이전 추진
- 5단계 : 추가연구개발을 통한 기술 고도화

○ 사업화 계획

- 사업화 형태 및 상용화 계획

성과물	도로포장상태 (PCI)
형태/규모	o 상용화 형태 : 실용화 패키징을 통한 현장적용 o 수요처 : 한국건설기술연구원, 도로공사, 지자체 o DB구축 및 변위관측 예상 단가 : 8백만원
상용화 능력 및 자원보유	o 주요 수요처와의 협력관계 구축 o 국토교통부 및 지자체 실증적용 및 활용 방안 제시 o 개발성과품에 대한 현장검증 및 공인 성과 인증 실시 o 공공기관 발주사업과 연계한 사업화 전략 수립
상용화 계획 및 일정	o 주요 수요처의 요구 및 필요 항목 파악 o 다양한 현장 여건 및 환경에 필요한 기술 이전 실시 o 국내 상용화 일정 – 현장검증 및 공인 성과 인증 달성 – 장비전시회 및 홍보 활동 실시 – 주요 수요기관에 대한 시범사업 추진 – 작업 매뉴얼 및 표준 지침서 제공으로 작업 표준화 o 국외 상용화 일정 – 당사 해외사업에서 기술 이전 실시 – ODA 사업 등에서 시범사업 추진 및 홍보 – 장비전시회 및 홍보 활동 실시 – 영문 매뉴얼 및 표준 지침서 작성

- 명확한 수요처 및 요구사항에 기반한 연구추진

· 도로포장품질 조사 기관의 요구와 필요사항에 대한 명확한 분석 및 연구결과 도출

· 한국건설기술연구원 등 기존 업무 대체 방안 제시 및 사업화

- PES 장비 대체를 통한 도로포장 품질 조사 업무 방안 제시

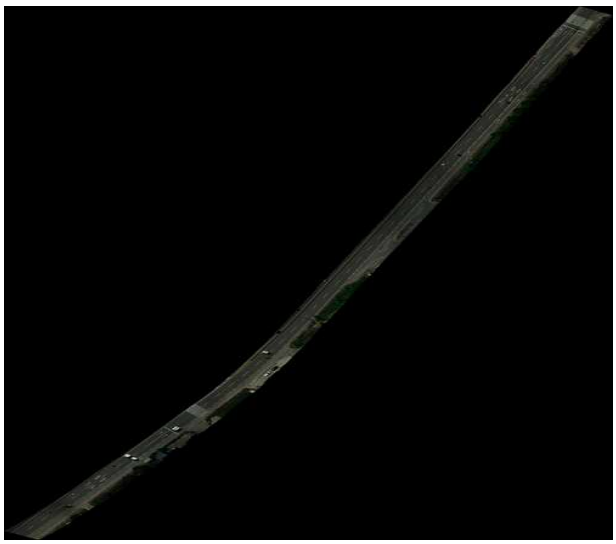
· 기존 고가의 PES 장비는 촬영빈도 및 처리 속도에 한계가 있으며, 이러한 이유로 지방도 등에 대한 도로포장품질 조사가 극히 미비한 실정임

· 이에 한국건설기술연구원의 PES 운영체계를 보완하고 대체할 기술 방안으로 연구를 추진하며 이에 대한 사업화 추진

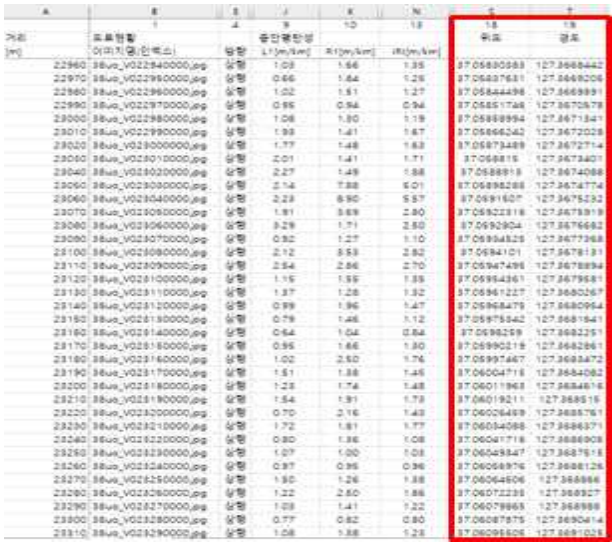
성과물	도로포장상태(PCI)
활용분야	도로포장상태 모니터링, 도로 시설물 점검 등
예상 수요처	한국건설기술연구원, 도로공사, 지방자치단체
수요처 요구사항	고가의 PES 장비 촬영빈도 및 운영비 절감 및 도로 조사 대상 확대
예상 매출액	180억원
사업화 전략	용역 형태 추진

○ 드론활용 PCI 측정기술 현장 활용 가능 수준

- 안성 지역 550M 구간에 대한 포장 평가를 진행함
- 드론을 이용한 초분광 조사 결과를 이용하여 단위 영역을 분할하고, 10m 단위로 나누어진 포장 평가 결과 도출
- 드론을 이용한 초분광 조사 결과와 전용 차량을 이용한 차량조사 결과를 비교 분석함
- 딥러닝 모델로 산출된 도로포장 평가 결과는 조사차량으로 평가된 포장평가지수(PCI)와 상관계수가 0.769이었으며, 개발 목표인 0.7 이상을 만족하였음



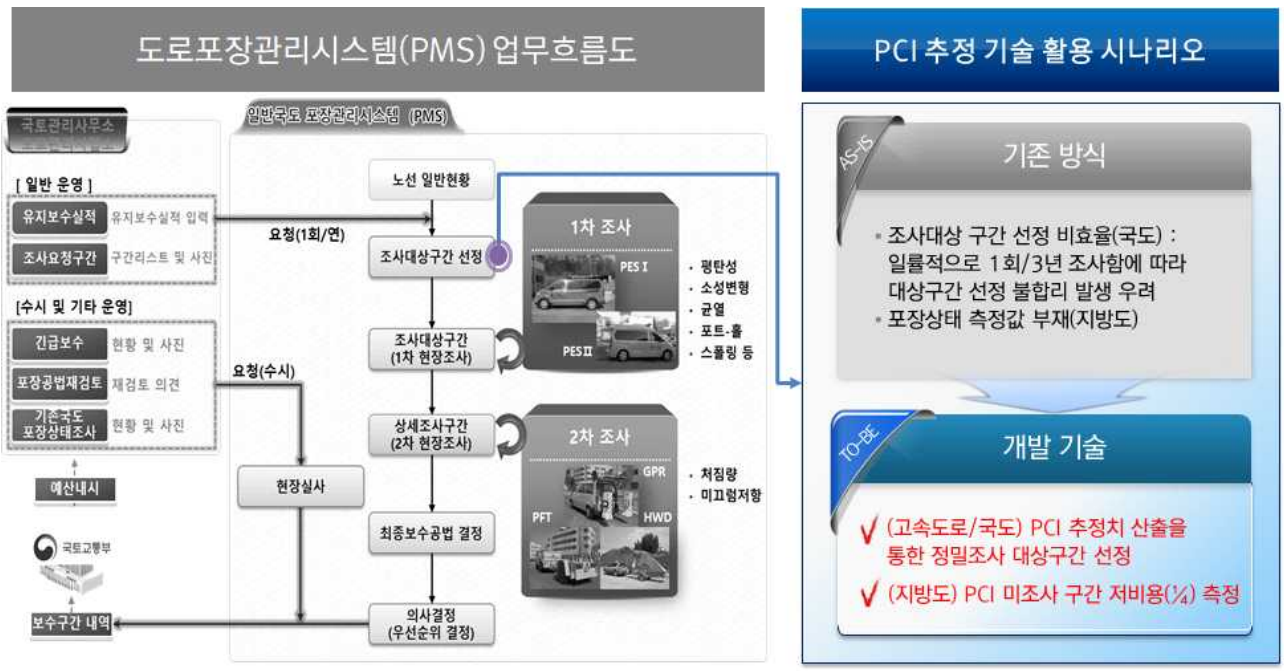
(a) 초분광 조사 시각화 이미지



(b) 조사차량 결과 및 위성 좌표

[포장조사 구간(안성)의 초분광 이미지와 조사차량 결과]

- 개발기술의 현장 활용 수준은 다음과 같음
  - 일반국도 : 현재 일반국도는 포장상태 관계없이 일률적으로 3년마다 고가의 조사차량을 이용하여 장시간 조사하고 있는데, 본 개발기술(저비용 신속) 활용 시, 포장상태가 양호한 지역에 대해서는 정밀조사를 생략하는 등의 방법으로 기존 조사방식의 효율성을 높일 수 있음
  - 지방도 아하 도로 : 현재 도로의 80% 이상을 차지하고 있는 지방도, 시군도 등에는 비용, 장비, 인력 등 문제로 PCI 조사가 이루어지지 않고 있는데, 본 개발기술 활용 시, 적은 비용으로 조사가 가능하여 포장 관리 예산투자 효율성을 높일 수 있음



라. 기술개발 성과 효과 극대화를 위한 향후 연구내용

○ 본 연구는 인력식으로 운용하는 드론을 이용하여 포트홀을 탐지하는 기술을 개발하였으나, 이는 우천 시 신속한 포트홀 조사가 어려운 단점이 있어, 이에 대한 해결방안으로 향후 국토교통부에서 추진 예정인 원격비행 시스템 등을 이용한 포트홀 조사기술로 확장·실용화 연구가 필요함

- 국토부는 국토교통 7대 혁신기술 중 하나인 드론을 활용, 도로관리 업무의 전문성 및 관리자 안전성 제고 추진하기 위해 “드론 도입 및 운영방안(국토부 차관 방침, 2019.8.19.)”을 마련함
- 위 방침의 3단계(2024년 이후) 추진계획으로 원격비행, 첨단기술 융·복합 기술을 이용한 도로관리방을 제시하고 있음

< “드론도입 및 운영방안(국토교통부 도로국, 2019.8.19.)” 일부내용 발췌>

□ (원격비행) 드론의 안정적 · 효율적 비행을 위해 원격으로 제어하여 실시간으로 영상을 전송하고 모니터링하는 시스템 구축방안 검토

ㅇ 주요 도로시설물 간 이격 거리, 점검 시간 등을 감안하여, 안정적인 드론 비행, 충전 및 보관 등이 가능한 ‘드론 스테이션 ‘ 구축 추진

ㅇ 촬영된 영상은 드론관리시스템으로 자동으로 전송되어, 영상 처리 · 분석 및 위험도 분석 데이터 등을 국토관리사무소에 전송

□ (첨단기술융합) IoT, AI 기반의 실시간 촬영영상 전송 및 모니터링 시스템 구축 등을 통해 스마트 도로관리 실현

ㅇ 각종 센서, 자동감지 · 분석기술 등을 활용, 도로시설물의 성능을 자동으로 평가하고 위험성을 조기에 감지 · 평가할 수 있는 체계 마련





- 본 연구성과는 국토부 차관 방침에 의거하여 향후 구축 예정인 원격비행 시스템과 결합함으로써, 우천 등 악천후에 따른 신속한 도로조사 기술로 확대 발전시키는 실용화 기술개발이 필요함

○ 변위 측정을 위한 측정주기(당초 연구목표에는 없었으나, 최종평가 사전점검 시 평가위원 의견사항으로 제시) 설정방안 연구 필요

- 본 연구에서 비탈면 변위측정을 위한 드론시스템 개발을 위해 센서 및 드론의 외자도입이 필요했는데, 코로나로 인한 외자도입이 지연되어 비탈면 변위측정 드론시스템 테스트베드 운영기간(1개월, 당초 6개월 이상 목표)이 제한적으로 이루어짐
- 이에 따라 비탈면 변위측정 효율성 확보를 위한 측정주기에 대한 연구는 물리적 시간부족으로 제한적으로 이루어짐
- 따라서 향후 본 드론 활용 비탈면 변위측정 기술을 상용화 기술개발을 추진하여 변위측정 효율성 극대화를 위한 운용방안, 측정주기 등에 대한 연구수행이 필요함

참고 문헌

AG, G.R.S., 2001 Gamma portable radar interferometer ii (gpri-ii). Guemligen: GAMMA Remote Sensing AG.

Bamler, R., 2000 Principles of synthetic aperture radar. Surveys in Geophysics. 21(2): pp. 147-157.

Bekaert, D.P., A.L. Handwerger, P. Agram, and D.B. Kirschbaum, 2020 Insar-based detection method for mapping and monitoring slow-moving landslides in remote regions with steep and mountainous terrain: An application to nepal. Remote Sensing of Environment. 249: pp. 111983.

Bekaert, D.P., A.L. Handwerger, P. Agram, and D.B. Kirschbaum, 2020 Insar-based detection method for mapping and monitoring slow-moving landslides in remote regions with steep and mountainous terrain: An application to nepal. Remote Sensing of Environment. 249: pp. 111983.

Berardino, P., G. Fornaro, R. Lanari, and E. Sansosti, 2002 A new algorithm for surface deformation monitoring based on small baseline differential sar interferograms. IEEE Transactions on Geoscience and Remote Sensing. 40(11): pp. 2375-2383.

Bischoff, C.A., A. Ferretti, F. Novali, A. Uttini, C. Giannico, and F. Meloni, 2020 Nationwide deformation monitoring with squeesar® using sentinel-1 data. Proceedings of the International Association of Hydrological Sciences. 382: pp. 31-37.

Calò, F., F. Ardizzone, R. Castaldo, P. Lollino, P. Tizzani, F. Guzzetti, R. Lanari, and M. Manunta. Landslide analysis through the multi-sensor sbas-dinsar approach: The case study of assisi, central italy. in 2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS. 2013. IEEE.

Colesanti, C., A. Ferretti, C. Prati, and F. Rocca, 2003 Monitoring landslides and tectonic motions with the permanent scatterers technique. Engineering Geology. 68(1): pp. 3-14.

Costantini, M., 1998 A novel phase unwrapping method based on network programming. IEEE Transactions on Geoscience and Remote Sensing. 36(3): pp. 813-821.

Dell'Amore, L., 2019, Assessment of image quality of waveform-encoded synthetic aperture radar using real data, University of Trento.

Ferretti, A., C. Prati, and F. Rocca, 2000 Nonlinear subsidence rate estimation using permanent scatterers in differential sar interferometry. IEEE Transactions on Geoscience and Remote Sensing. 38(5): pp. 2202-2212.

Ferretti, A., C. Prati, and F. Rocca, 2001 Permanent scatterers in sar interferometry. IEEE Transactions on Geoscience and Remote Sensing. 39(1): pp. 8-20.

Gabriel, A.K., R.M. Goldstein, and H.A. Zebker, 1989 Mapping small elevation changes over large areas: Differential radar interferometry. Journal of Geophysical Research: Solid Earth. 94(B7): pp. 9183-9191.

Garthwaite, M., S. Nancarrow, A. Hislop, M. Thankappan, J. Dawson, and S. Lawrie, 2015 Design of radar corner reflectors for the australian geophysical observing system. Geoscience Australia. 3: pp. 490.

Ghiglia, D.C. and L.A. Romero, 1996 Minimum lp-norm two-dimensional phase unwrapping. Journal of the Optical Society of America A. 13(10): pp. 1999-2013.

Goldstein, R.M. and C.L. Werner, 1998 Radar interferogram filtering for geophysical applications. Geophysical Research Letters. 25(21): pp. 4035-4038.

Hanssen, R.F., 2001, Radar interferometry: Data interpretation and error analysis. Vol. 2. New York City, NY. USA.: Kluwer Academic Publishers.

Herrera, G., F. Gutiérrez, J. García-Davalillo, J. Guerrero, D. Notti, J. Galve, J.

Fernández-Merodo, and G. Cooksley, 2013 Multi-sensor advanced dinsar monitoring of very slow landslides: The tena valley case study (central spanish pyrenees). Remote Sensing of Environment. 128: pp. 31-43.
Herrera, G., F. Gutiérrez, J.C. García-Davalillo, J. Guerrero, D. Notti, J.P. Galve, J.A. Fernández-Merodo, and G. Cooksley, 2013 Multi-sensor advanced dinsar monitoring of very slow landslides: The tena valley case study (central spanish pyrenees). Remote Sensing of Environment. 128: pp. 31-43.
Hong, S.-H. and J.-S. Won. Ers-envisat cross-interferometry for coastal dem construction. in Fringe 2005 Workshop. 2006.
Hong, S. and S. Wdowinski, 2014 Multitemporal multitrack monitoring of wetland water levels in the florida everglades using alos palsar data with interferometric processing. IEEE Geoscience and Remote Sensing Letters. 11(8): pp. 1355-1359.
Jakowski, N., A. Jungstand, K. Schlegel, H. Kohl, and K. Rinnert, 1992 The ionospheric response to perturbation electric fields during the onset phase of geomagnetic storms. Canadian Journal of Physics. 70(7): pp. 575-581.
Jolivet, R., R. Grandin, C. Lasserre, M.-P. Doin, and G. Peltzer, 2011 Systematic insar tropospheric phase delay corrections from global meteorological reanalysis data. Geophysical Research Letters. 38(17).
Kerle, N., L.L. Janssen, and G.C. Huurneman, 2004 Principles of remote sensing. ITC, Educational textbook series. 2: pp. 250.
Lauknes, T.R., 2011 Rockslide mapping in norway by means of interferometric sar time series analysis.
Liu, G., H. Jia, Y. Nie, T. Li, R. Zhang, B. Yu, and Z. Li, 2014 Detecting subsidence in coastal areas by ultrashort-baseline tcpinsar on the time series of high-resolution terrasar-x images. IEEE Transactions on Geoscience and Remote Sensing. 52(4): pp. 1911-1923.
Massonnet, D., M. Rossi, C. Carmona, F. Adragna, G. Peltzer, K. Feigl, and T. Rabaute, 1993 The displacement field of the landers earthquake mapped by radar interferometry. Nature. 364(6433): pp. 138-142.
Massonnet, D. and K.L. Feigl, 1998 Radar interferometry and its application to changes in the earth's surface. Reviews of Geophysics. 36(4): pp. 441-500.
Meyer, F.J., 2011 Performance requirements for ionospheric correction of low-frequency sar data. IEEE Transactions on Geoscience and Remote Sensing. 49(10): pp. 3694-3702.
Moreira, A., P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K.P. Papathanassiou, 2013 A tutorial on synthetic aperture radar. IEEE Geoscience and Remote Sensing Magazine. 1(1): pp. 6-43.
Noferini, L., M. Pieraccini, G. Luzi, D. Mecatti, G. Macaluso, and C. Atzeni. Ground-based radar interferometry for terrain mapping. in 2006 IEEE International Symposium on Geoscience and Remote Sensing. 2006. IEEE.
Osmanoğlu, B., F. Sunar, S. Wdowinski, and E. Cabral-Cano, 2016 Time series analysis of insar data: Methods and trends. ISPRS Journal of Photogrammetry and Remote Sensing. 115: pp. 90-102.
Park, S.-W. and S.-H. Hong, 2021 Nonlinear modeling of subsidence from a decade of insar time series. Geophysical Research Letters. 48(3): pp. e2020GL090970.
Roering, J.J., L.L. Stimely, B.H. Mackey, and D.A. Schmidt, 2009 Using dinsar, airborne lidar, and archival air photos to quantify landsliding and sediment transport. Geophysical Research Letters. 36(19).
Tanase, M.A., M. Santoro, U. Wegmüller, J. de la Riva, and F. Pérez-Cabello, 2010 Properties of x-, c- and l-band repeat-pass interferometric sar coherence in mediterranean pine forests affected by fires. Remote Sensing of Environment. 114(10): pp. 2182-2194.
Thiele, A., E. Cadario, K. Schulz, U. Thoennessen, and U. Soergel, 2007 Insar phase

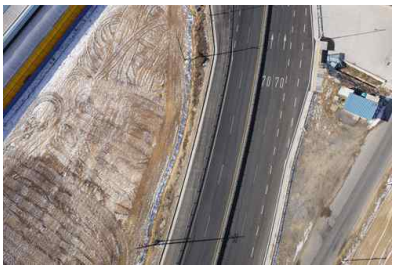
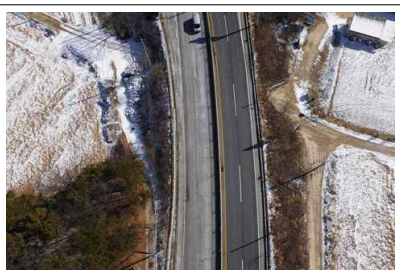
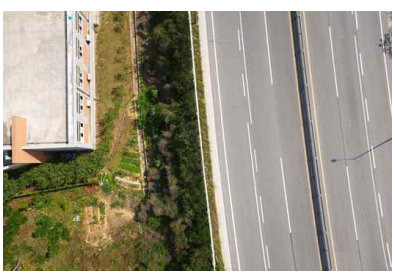
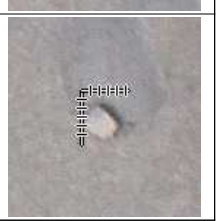
profiles at building locations. Proceeding of ISPRS photogrammetric image analysis. 36(part 3): pp. W49A.
Tribolet, J., 1977 A new phase unwrapping algorithm. IEEE Transactions on Acoustics, Speech, and Signal Processing. 25(2): pp. 170-177.
Wang, C., Cai, J., Li, Z., Mao, X., Feng, G., & Wang, Q. (2018). Kinematic parameter inversion of the Slumgullion landslide using the time series offset tracking method with UAVSAR data. Journal of Geophysical Research: Solid Earth, 123, 8110- 8124. https://doi.org/10.1029/2018JB015701
Wegmuller, U., C. Werner, and T. Strozzi. Sar interferometric and differential interferometric processing chain. in IGARSS '98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings. (Cat. No.98CH36174). 1998.
Yamazaki, F. and W. Liu. Remote sensing technologies for post-earthquake damage assessment: A case study on the 2016 kumamoto earthquake. in 6th Asia Conference on Earthquake Engineering. 2016.
Yazici, B.V. and E. Tunc Gormus, 2020 Investigating persistent scatterer insar (psinsar) technique efficiency for landslides mapping: A case study in artvin dam area, in turkey. Geocarto International. pp. 1-19.
Yueping Yin, W., 2010 L. Integration of gps with insar to monitoring of the jiaju landslide in sichuan. China. 7: pp. 359-365.
Zebker, H.A. and R.M. Goldstein, 1986 Topographic mapping from interferometric synthetic aperture radar observations. Journal of Geophysical Research: Solid Earth. 91(B5): pp. 4993-4999.
Zebker, H.A., P.A. Rosen, R.M. Goldstein, A. Gabriel, and C.L. Werner, 1994 On the derivation of coseismic displacement fields using differential radar interferometry: The landers earthquake. 99(B10): pp. 19617-19634.
Zebker, H.A., C.L. Werner, P.A. Rosen, and S. Hensley, 1994 Accuracy of topographic maps derived from ers-1 interferometric radar. IEEE Transactions on Geoscience and Remote Sensing. 32(4): pp. 823-836.
Zebker, H.A. and Y. Lu, 1998 Phase unwrapping algorithms for radar interferometry: Residue-cut, least-squares, and synthesis algorithms. Journal of the Optical Society of America A. 15(3): pp. 586-598.
홍상훈, 2006 영상레이더 위상간섭 기법을 이용한 연안 수치 표고 모형 제작 연구, 연세대학교 박사학위논문.
OpenCV 3.2.0, (n.d.) How to use background subtraction methods, Retrieved April 26, 2021 from https://docs.opencv.org/3.1.0/d1/dc5/tutorial_background_subtraction.html
OpenCV 3.2.0, (n.d.) Convex Hull, Retrieved April 26, 2021 from https://docs.opencv.org/3.0-beta/doc/tutorials/imgproc/shapedescriptors/hull/hull.html
OpenCV 3.2.0, (n.d.) More morphology transformations, Retrieved April 26, 2021 from https://docs.opencv.org/2.4/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html
Google Brain team, (n.d.) Building Intelligent Systems with Large Scale Deep Learning, Retrieved April 26, 2021, from https://drive.google.com/file/d/0B8z5oUpB2DysZWF1RTFuX1NEZUk/view
Google Tensorflow github, (n.d.) TensorFlow github, Retrieved April 26, 2021 https://github.com/tensorflow/docs/blob/master/site/en/r1/guide/extend/architecture.md
Google Tensorboard, (n.d.) Tensorboard, Retrieved April 26, 2021, from http://edwardlib.org/tutorials/tensorboard
Google TensorFlow Serving, (n.d.) TensorFlow Serving, Retrieved April 26, 2021, from http://daddynkidsmakers.blogspot.com/2017/05/windows-10-tensorflow.html

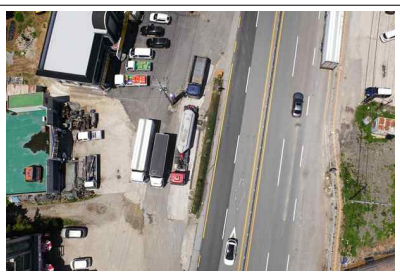
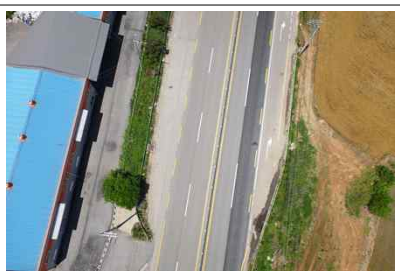
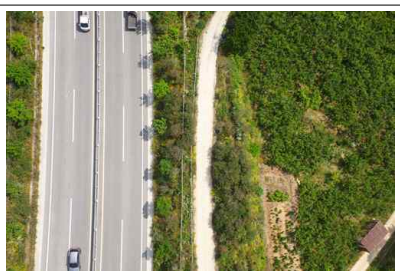
hskang9, (n.d.) Retrieved April 26, 2021, from https://hskang9.github.io/various_cnn/2017/11/24/inception/
Sai Kumar Basaveswara, (n.d.) Microsoft Residual Neural Net, Retrieved April 26, 2021, from https://towardsdatascience.com/cnn-architectures-a-deep-dive-a99441d18049
Aparna, Yukti Bhatia, Rachna Rai, Varun Gupta, Naveen Aggarwal, Aparna Akula, (2019). Convolutional neural networks based potholes detection using thermal imaging, Journal of King Saud University - Computer and Information Sciences.
Pacha, Alexander, Jan Hajič, and Jorge Calvo-Zaragoza, (2018). A baseline for general music object detection with deep learning, Applied Sciences 8.9
paperwithcode, (n.d) paper with code, Retrieved April 26, 2021 from https://paperswithcode.com/sota/object-detection-on-coco
Ren, S., He, K., Girshick, R., & Sun, J, (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497.
Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018). Single-shot refinement neural network for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4203-4212).
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 764-773).
He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).
COCO dataset, (n.d.) COCO dataset Retrieved April 27, 2021, from https://cocodataset.org/#explore
Bhatia, Y., Rai, R., Gupta, V., Aggarwal, N., & Akula, A. (2019). Convolutional neural networks based potholes detection using thermal imaging. Journal of King Saud University-Computer and Information Sciences.
Jo, Y., & Ryu, S. (2015). Pothole detection system using a black-box camera. Sensors, 15(11), 29316-29331.
Baek, J. W., & Chung, K. (2020). Pothole Classification Model Using Edge Detection in Road Image. Applied Sciences, 10(19), 6662.
Becker, Y. V. F., Siqueira, H. L., Matsubara, E. T., Gonçalves, W. N., & Marcato, J. M. (2019). Asphalt pothole detection in uav images using convolutional neural networks. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium (pp. 56-58). IEEE.
Koch, C., & Brilakis, I. (2011). Pothole detection in asphalt pavement images. Advanced Engineering Informatics, 25(3), 507-515.
이민혜, 남광우, & 이창우. (2019). Mask R-CNN 을 이용한 항공 영상에서의 도로 균열 검출. 한국산업정보학회논문지, 24(3), 23-29.
Vigneshwar, K., & Kumar, B. H. (2016, December). Detection and counting of pothole using image processing techniques. In 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1-4). IEEE.
Sulistyaningrum, D. R., Oranova, D., Pramestya, R. H., Mukhlash, I., Setiyono, B., & Ahyudanari, E. (2020, February). Pavement Distress Classification Using Deep Learning Method Based on Digital Image. In 2nd International Symposium on Transportation Studies in Developing Countries (ISTSDC 2019) (pp. 143-146). Atlantis Press.

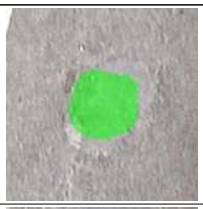
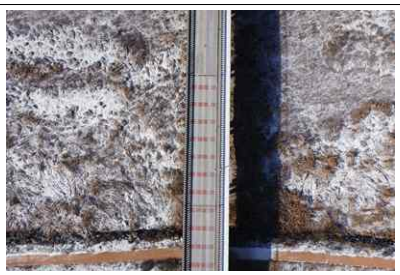
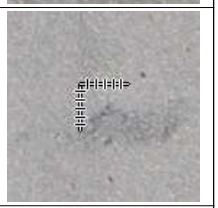
부록 1. 포트홀 탐지기술 평가 데이터

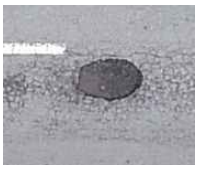
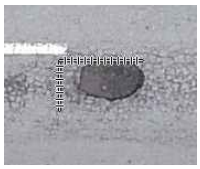
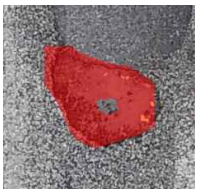
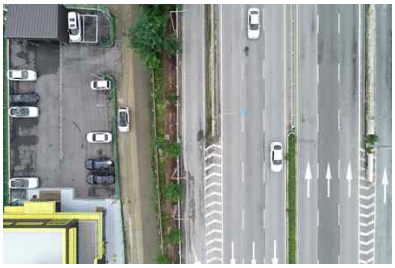
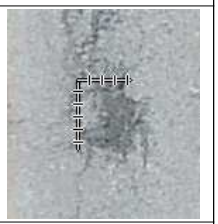
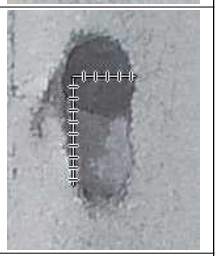
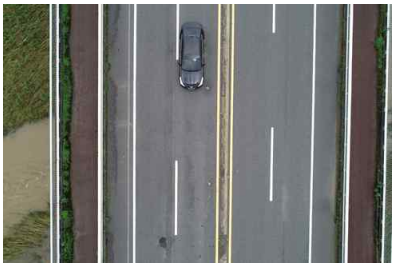
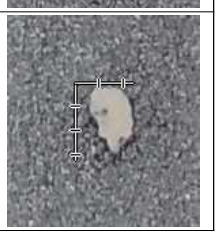
○ 포트홀 탐지기술 평가 원본 데이터 및 탐지 결과데이터

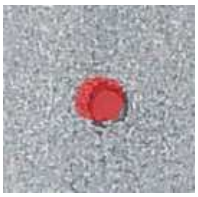
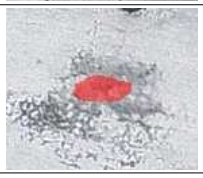
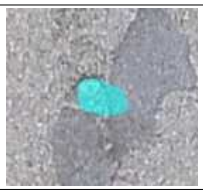
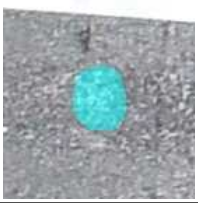
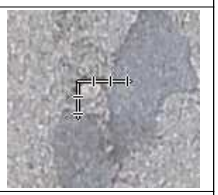
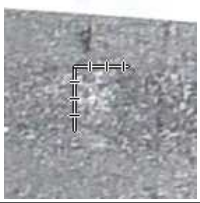
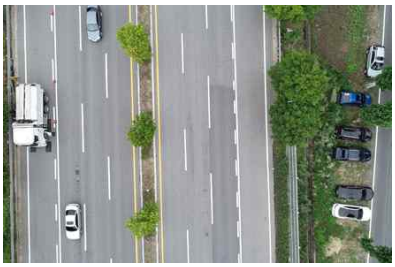
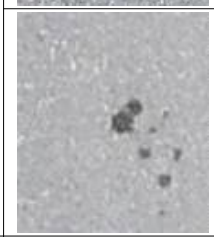
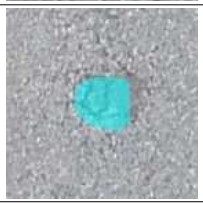
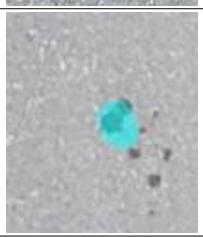
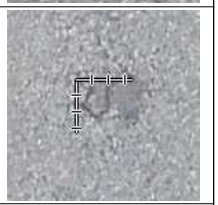
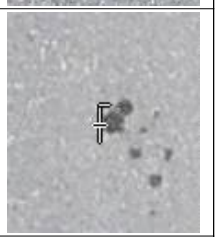
- 약 30~70m 고도에서 촬영한 항공 영상 100장 기준(포트홀 총 71개)

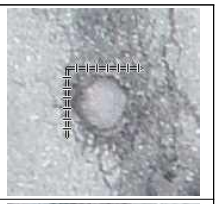
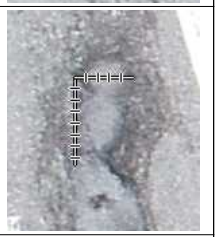
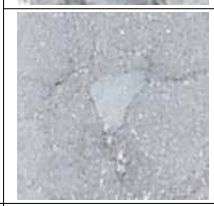
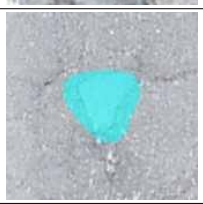
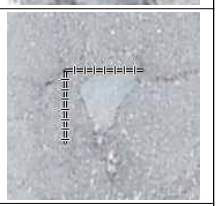
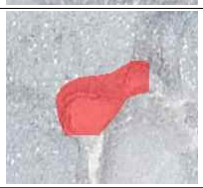
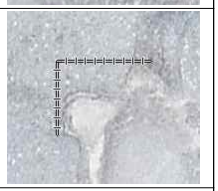
번호	원본 항공 영상	포트홀 수	탐지 결과		
			포트홀	포트홀 영역표시	포트홀 가로세로 표시
1		1			
					
2		1			
3		1			
4		1			
5		1			
					

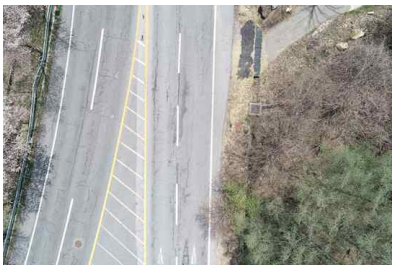
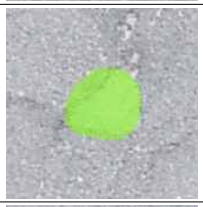
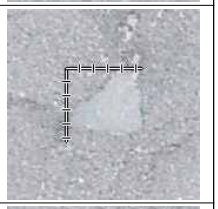
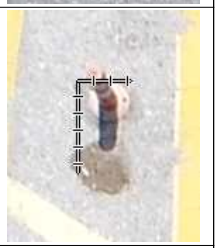
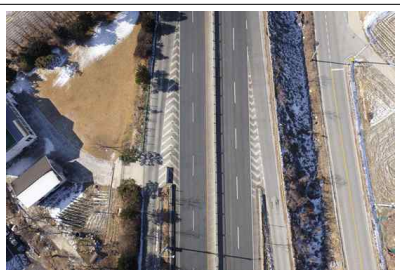
6		2	 	 	 
7		1			
8		1			
9		1			
10		1			
11		1			

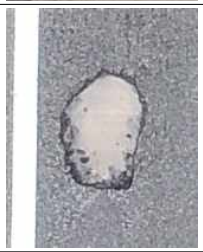
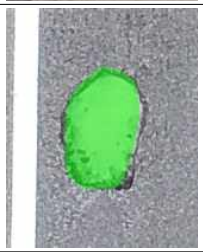
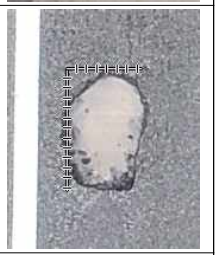
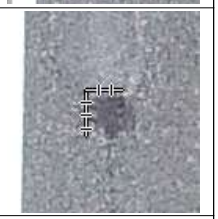
12		2			
					
					
13		1			
14		1			
					
15		1			
16		1			

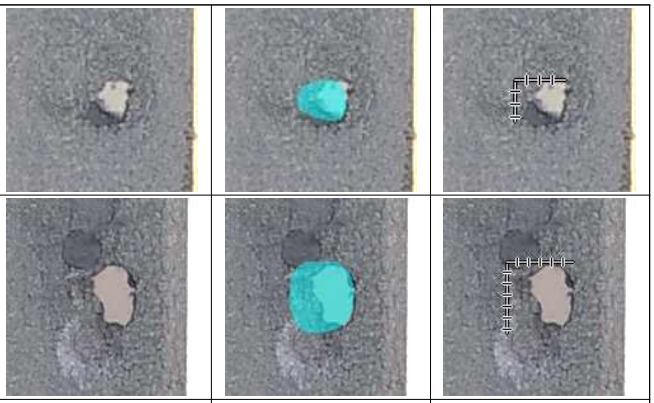
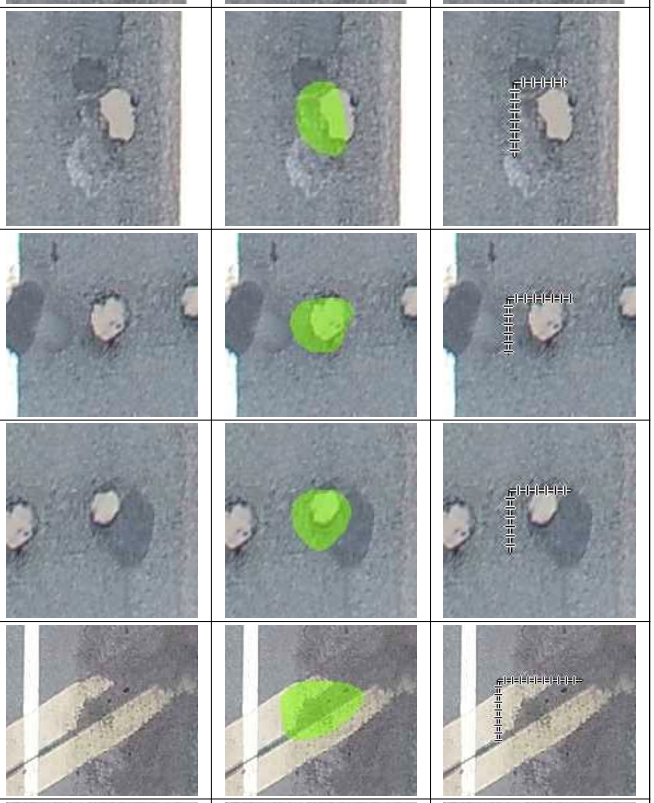
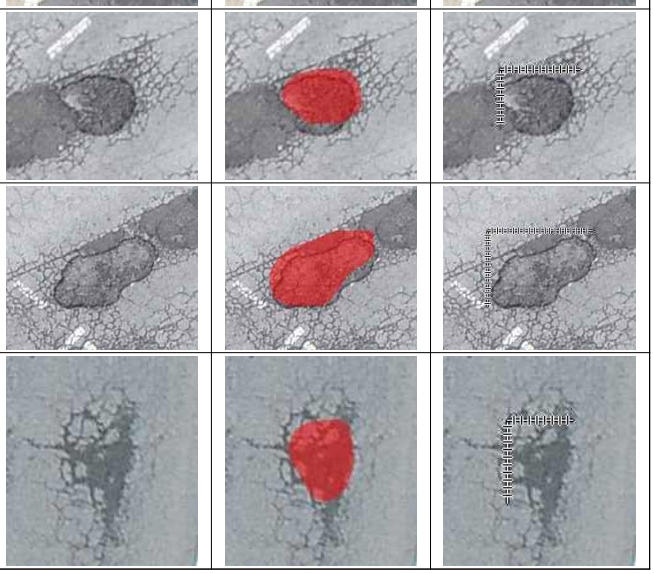
17		1			
					
18		1			
19		1			
20		1			
					
21		2			
					

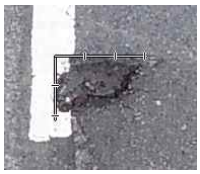
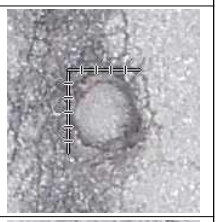
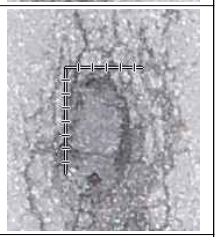
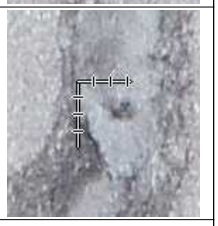
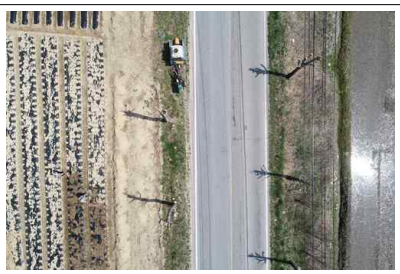
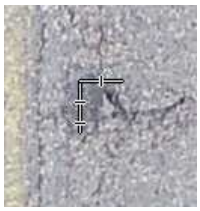
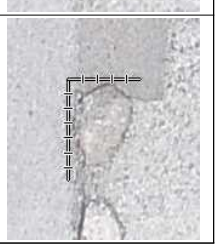
22		1			
23		2	 	 	 
24		1			
25		1	 	 	 
26		1	 	 	 
27		1			

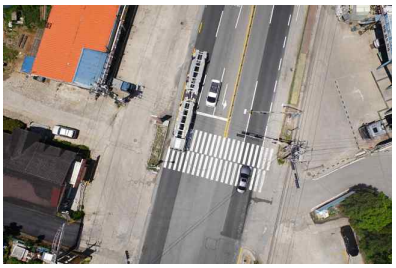
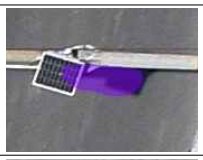
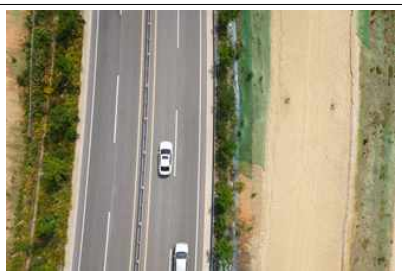
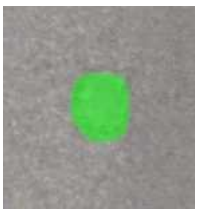
28		3			
					
					
					
29		2			
					
30		2			
					
31		1			

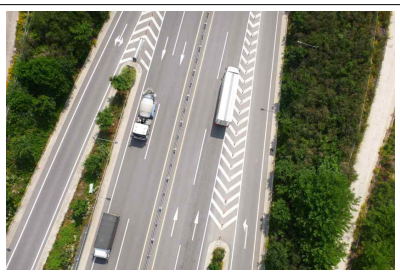
32		2			
					
					
					
33		1			
34		1			
					
35		1			

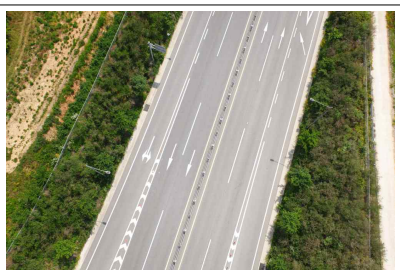
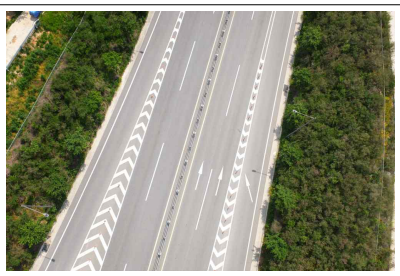
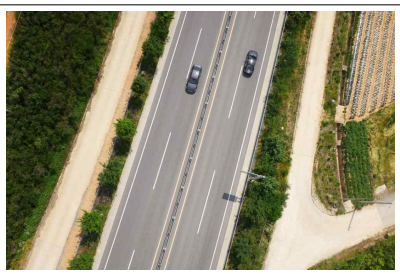
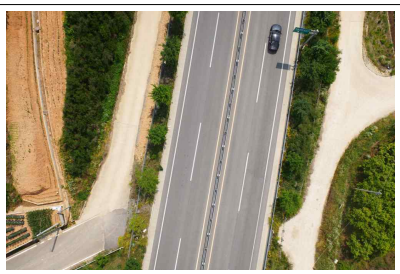
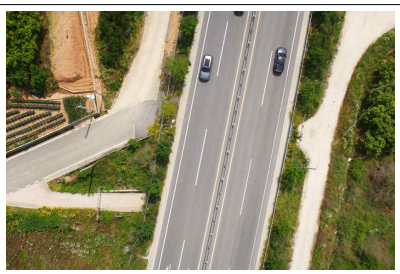
36		1			
37		2			
					
38		2			
					
					
39		2			
					

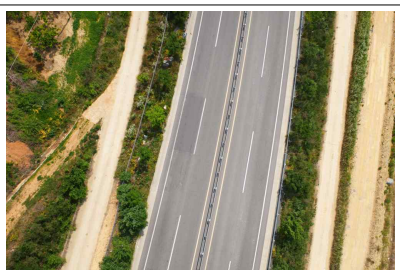
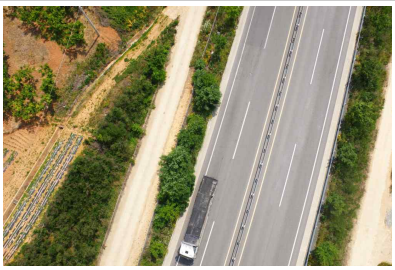
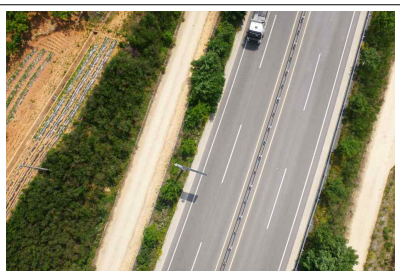
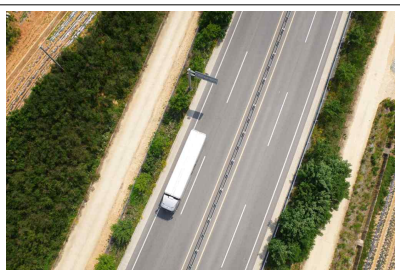
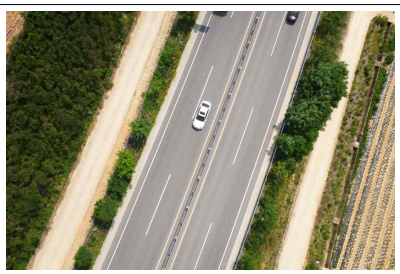
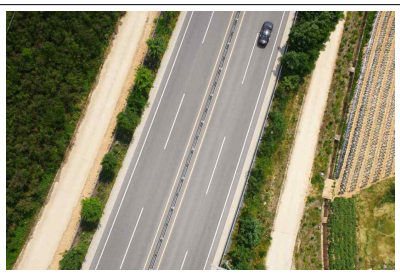
40		2			
41		3			
42		2			

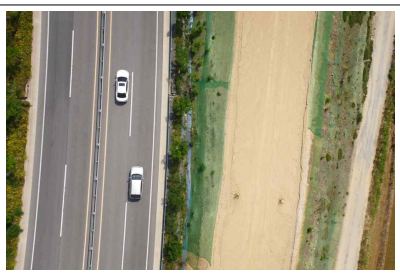
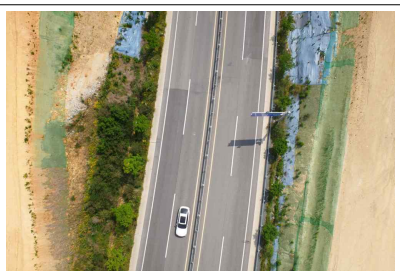
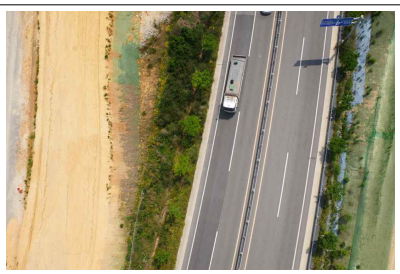
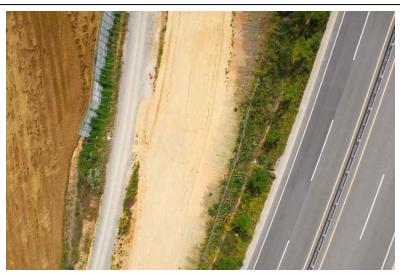
43		2			
44		4			
					
					
					
45		1			
46		2			
					

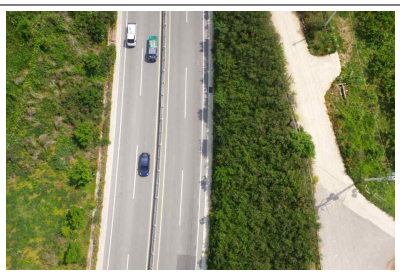
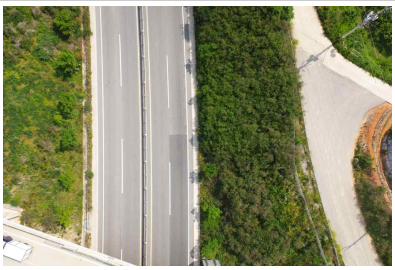
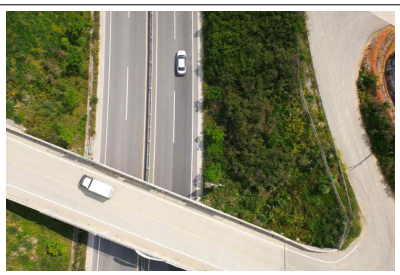
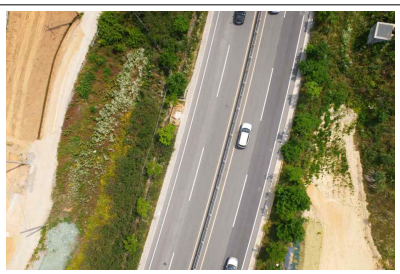
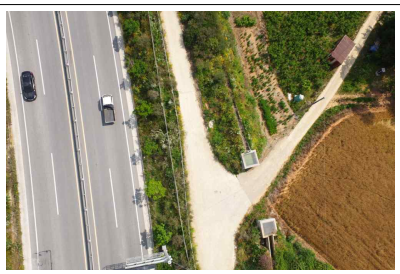
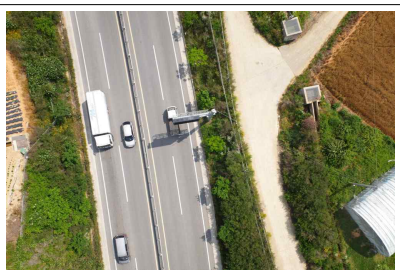
47		1			
48		1	 	 	 
49		1	없음		
50		0			
51		0			
52		0	없음		

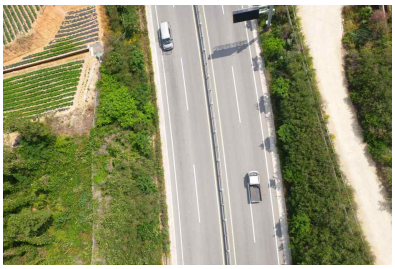
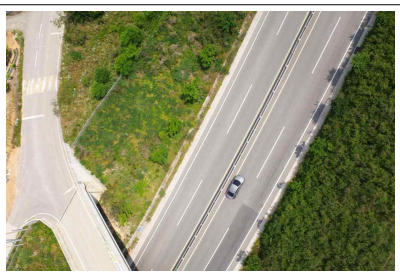
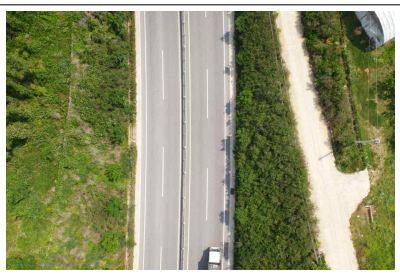
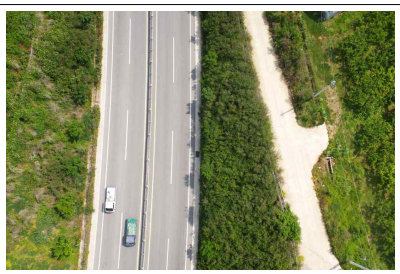
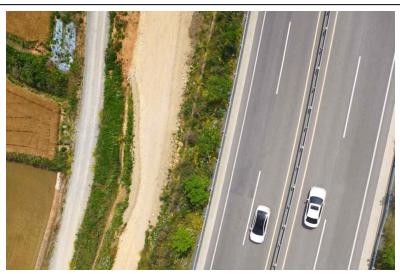
53		0	없음
54		0	없음
55		0	없음
56		0	없음
57		0	없음
58		0	없음
59		0	없음

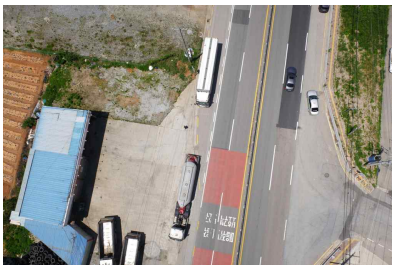
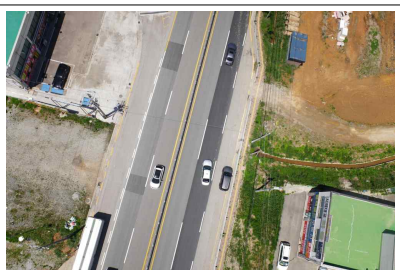
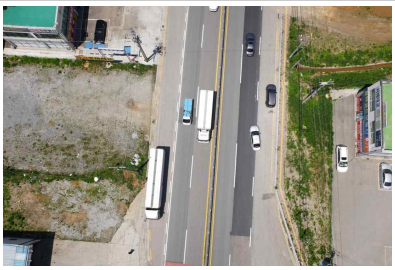
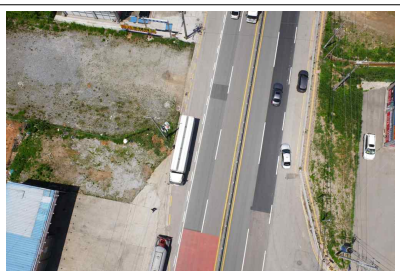
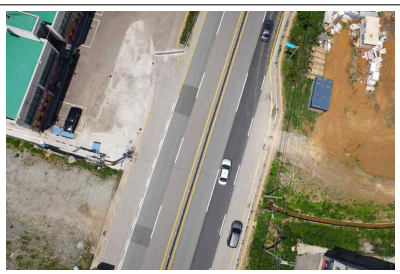
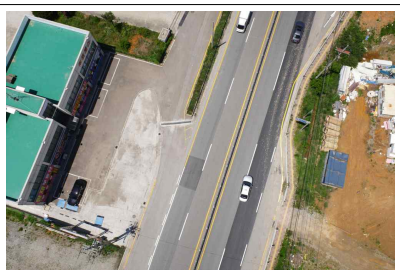
60		0	없음
61		0	없음
62		0	없음
63		0	없음
64		0	없음
65		0	없음
66		0	없음

67		0	없음
68		0	없음
69		0	없음
70		0	없음
71		0	없음
72		0	없음
73		0	없음

74		0	없음
75		0	없음
76		0	없음
77		0	없음
78		0	없음
79		0	없음
80		0	없음

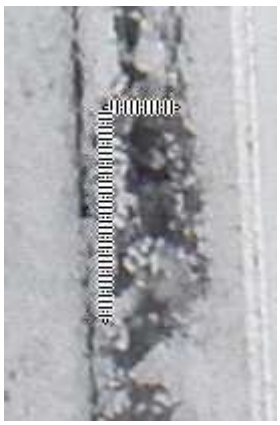
81		0	없음
82		0	없음
83		0	없음
84		0	없음
85		0	없음
86		0	없음
87		0	없음

88		0	없음
89		0	없음
90		0	없음
91		0	없음
92		0	없음
93		0	없음
94		0	없음

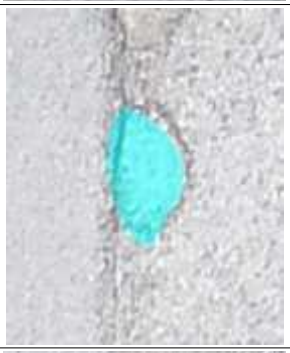
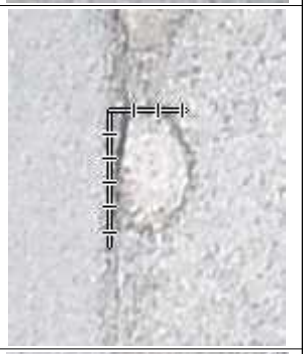
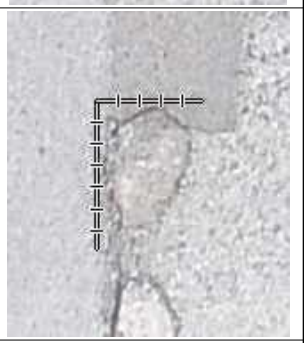
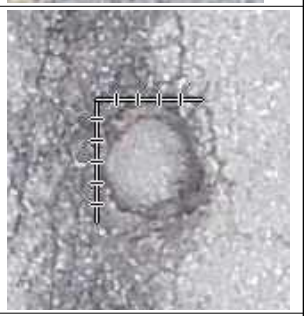
95		0	없음
96		0	없음
97		0	없음
98		0	없음
99		0	없음
100		0	없음

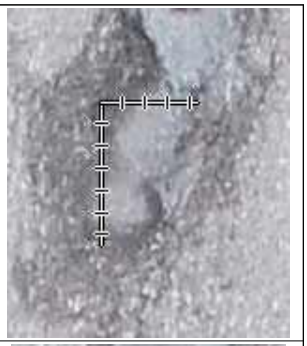
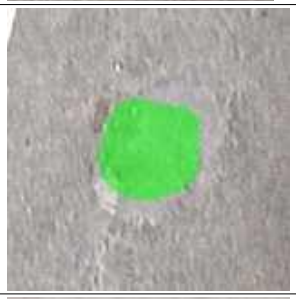
○ 포트홀 정탐지 및 오탐지 결과 정리

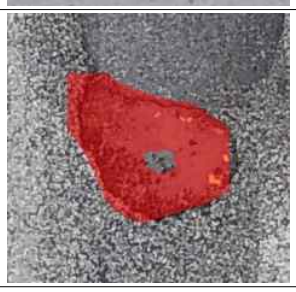
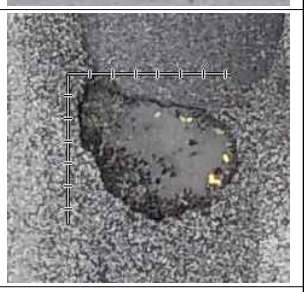
- 100장 이미지 내 71개의 포트홀 탐지 결과(정탐 65건, 미탐 6건)

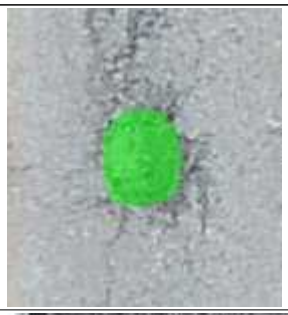
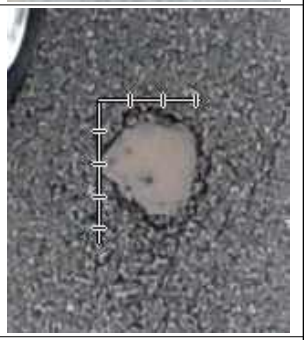
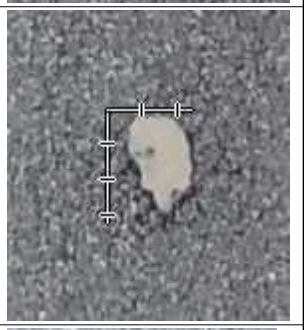
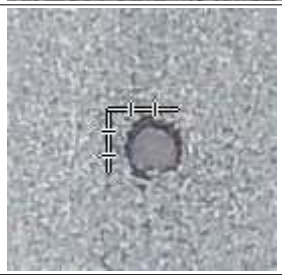
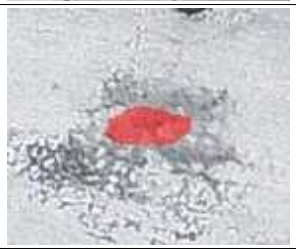
포트홀 번호	탐지 결과	정탐지 이미지		
		포트홀	포트홀 영역표시	포트홀 가로세로 표시
1	정탐지			
2	정탐지			
3	정탐지			
4	정탐지			
5	정탐지			

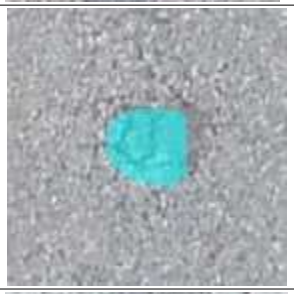
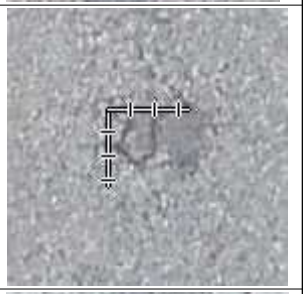
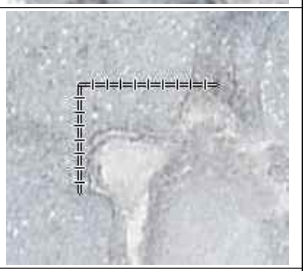
6	정탐지			
7	정탐지			
8	정탐지			
9	정탐지			
10	정탐지			
11	정탐지			
12	정탐지			

				
13	정탐지			
14	정탐지			
15	정탐지			
16	정탐지			
17	정탐지			
18	정탐지			

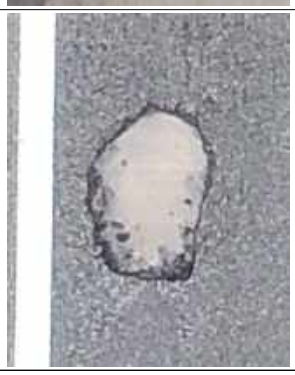
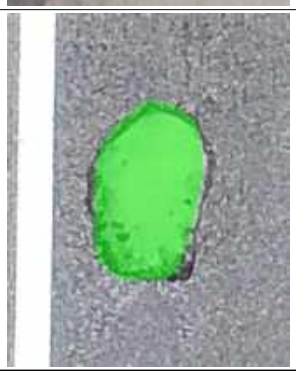
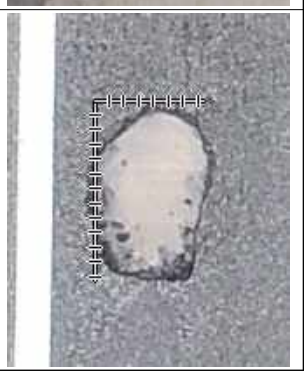
				
19	정탐지			
20	정탐지			
21	정탐지			
22	정탐지			
23	정탐지			
24	정탐지			

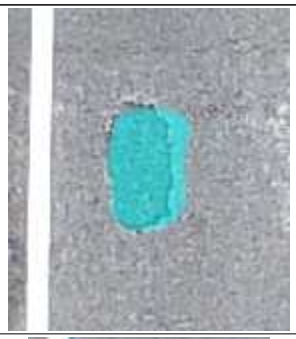
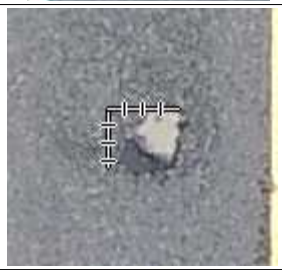
				
25	정탐지			
26	정탐지			
27	정탐지			
28	정탐지			
29	정탐지			
30	정탐지			

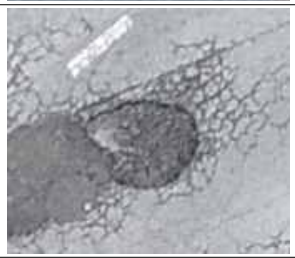
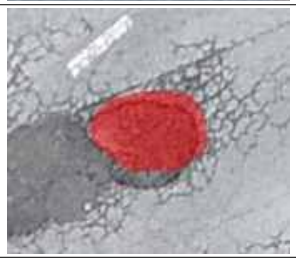
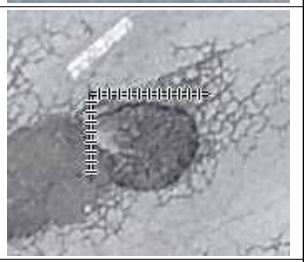
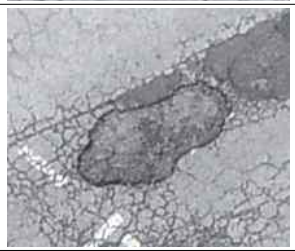
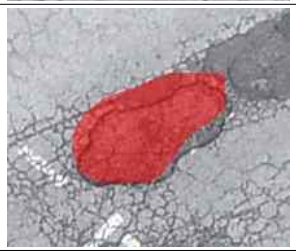
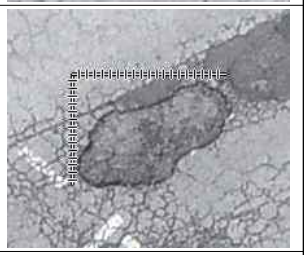
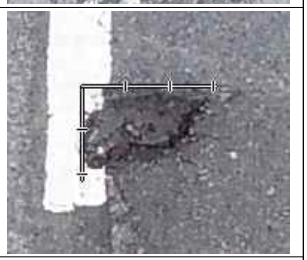
				
31	정탐지			
32	정탐지			
33	정탐지			
34	정탐지			
35	정탐지			
36	정탐지			

37	정탐지			
38	정탐지			
39	정탐지			
40	정탐지			
41	정탐지			
42	정탐지			
43	정탐지			

44	정탐지			
45	정탐지			
46	정탐지			
47	정탐지			
48	정탐지			
49	정탐지			

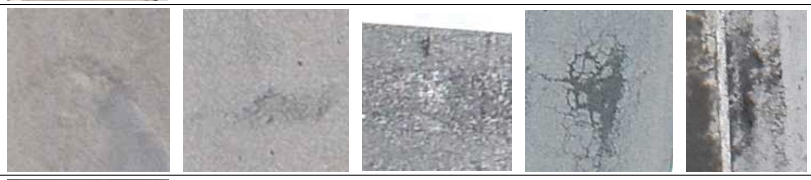
50	정탐지			
51	정탐지			
52	정탐지			
53	정탐지			
54	정탐지			
55	정탐지			

56	정탐지			
57	정탐지			
58	정탐지			
59	정탐지			
60	정탐지			
61	정탐지			

62	정탐지			
63	정탐지			
64	정탐지			
65	정탐지			
66	미탐지		없음	없음
67	미탐지		없음	없음
68	미탐지		없음	없음
69	미탐지		없음	없음

70	미탐지		없음	없음
71	미탐지		없음	없음

- 포트홀과 유사한 특징을 보이는 도로 위 패치, 라바콘, 기름자국, 등이 오탐지 사례

오탐지 결과 분석	오탐지 이미지
맨홀	
도로 포장 손상	
도로 내 구조물	
패치	
기름 자국, 물 자국	
라바콘	
도로 노면 표식	

부록 2. 비탈면 변위측정 소프트웨어 소스코드

영상레이더 진폭 영상 제작 프로그램

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Program for making amplitude from SAR Netcdf file
%Sang-Hoon Hong and Seo-Woo Park
%Before running code, you need the text file
%include the Netcdf file name
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc; clear

%Open the text file and read title of the file

file=fopen('nc_list.txt','r+');
nc_title = textscan(file,'%s');
inp_cell = nc_title{:,1};
inp_mat = cell2mat(inp_cell(:,1));

%Loop for making amplitude

for m = 1:length(inp_cell)


%file_1 is the name of Netcdf file

file_1=inp_mat(m,:);

%See the total information in the Netcdf file

diary (horzcat('info_',file_1, '.txt'))
ncdisp(file_1);
diary off

%Extract the information about real and imaginary part

Real=ncread(file_1,'SigmaImageSingleLookRealPart');
Imag=ncread(file_1,'SigmaImageSingleLookImaginaryPart');

%Wavelength is included in figure

CentralFreq = ncread(file_1, 'CentralFreq');
Wavelength=(physconst('LightSpeed')/CentralFreq)*100; %m to cm

%Calculate the amplitude using the absolute value

Amplitude = abs(Real+Imag);

%Replace the '9999' value to '0' in Amplitude %%%

for k=1:size(Amplitude,1)
for l = 1:size(Amplitude,2)
if Amplitude(k,l) == 9999
Amplitude(k,l) = 0;
end
end
end

%Export Amplitude value to binary for ENVI
name_ENVI = horzcat('Amplitude_',file_1, '.bin');
f1 = fopen(name_ENVI,'w');
fwrite(f1, Amplitude,'double');
fclose(f1);

%Change the value from double to uint8 to see the Amplitude more clearly

name = horzcat('Amplitude_',file_1, '.tif');
imwrite(Amplitude,name,'tif');
im=imread(name);

%Output result display through figure window

figure, imshow(im,[0 1]),...
title(['Amplitude_ ',num2str(m),' || row : ',num2str(size(Amplitude,1)),...
'   col : ', num2str(size(Amplitude,2)),' || {₩lambda} : ',...
num2str(Wavelength),' cm']),...

%delete the tif file (sub-creation image)
delete(name)

end
```

```
fprintf('₩n Amplitude image making is done. ₩n₩n');
```

- 영상레이더 위상 영상 제작 프로그램

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%Program for making interferogram between two SAR Netcdf files
%Sang-Hoon Hong and Seo-Woo Park
%Before running code, you need the text file
%include the Netcdf file name
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

clc; clear

%Open the file and read the name from the text file

file=fopen('nc_list.txt','r+');
nc_title = textscan(file,'%s');
inp_cell = nc_title{:,1};
inp_mat = cell2mat(inp_cell(:,1));

%Loop for making interferogram

for m = 1:length(inp_cell)

if m+1<=length(inp_cell)

%file_1 is the information of master file

file_1=inp_mat(m,:);

%See the total information in the Netcdf file

diary (horzcat('info_',file_1, '.txt'))
ncdisp(file_1);
diary off

%Extract the information about real and imaginary part of the master image

Real_1=ncread(file_1,'SigmaImageSingleLookRealPart');
imag_1=ncread(file_1,'SigmaImageSingleLookImaginaryPart');

%file_2 is the information of slave file

file_2=inp_mat(m+1,:);

%See the total information in the Netcdf file

diary (horzcat('info_',file_2, '.txt'))
ncdisp(file_2);
diary off

%Extract the information about real and imaginary part of the slave image

Real_2=ncread(file_2,'SigmaImageSingleLookRealPart');
imag_2=ncread(file_2,'SigmaImageSingleLookImaginaryPart');

%Calculate the phase from the each SLC image

A1 = Real_1 + 1i*imag_1;
A2 = Real_2 + 1i*imag_2;

%Wavelength is included in figure

CentralFreq = ncread(file_1, 'CentralFreq');
Wavelength=(physconst('LightSpeed')/CentralFreq)*100; %m to cm

%Calculate the pixel size of the master image

%Extract the information about Latitude and Longitude

LatImage1=ncread(file_1,'LatImage');
LonImage1=ncread(file_1,'LonImage');

%Extract the information about the corner value of the image

Latitude_1_11 = ncread(file_1,'Latitude11'); % Latitude of image corner (1, 1)
Longitude_1_11 = ncread(file_1,'Longitude11'); % Longitude of image corner (1, 1)
Latitude_1_12 = ncread(file_1,'Latitude12'); % Latitude of image corner (1, x)
Longitude_1_21 = ncread(file_1,'Longitude21'); % Latitude of image corner (y, 1)

%Calculate the pixel size from deg to sec of master image (unit: sec/pixel)
```

```
azimuth_m_1=(abs(Latitude_1_11-Latitude_1_12)/size(LatImage1,1));
range_m_1=(abs(Longitude_1_11-Longitude_1_21)/size(LatImage1,2));


%Calculate the pixel size of the slave image

%Extract the information about Latitude and Longitude

LatImage2=ncread(file_2,'LatImage');
LonImage2=ncread(file_2,'LonImage');

%Extract the information about the corner value of the image

Latitude_2_11 = ncread(file_2,'Latitude11'); % Latitude of image corner (1, 1)
Longitude_2_11 = ncread(file_2,'Longitude11'); % Longitude of image corner (1, 1)
Latitude_2_12 = ncread(file_2,'Latitude12'); % Latitude of image corner (1, x)
Longitude_2_21 = ncread(file_2,'Longitude21'); % Latitude of image corner (y, 1)


%Calculate the pixel size from deg to sec of slave image (unit: sec/pixel)

azimuth_m_2=(abs(Latitude_2_11-Latitude_2_12)/size(LatImage2,1));
range_m_2=(abs(Longitude_2_11-Longitude_2_21)/size(LatImage2,2));

%Claculate the offset

azimuth_pix_diff = ((Latitude_2_11-Latitude_1_11)/azimuth_m_1); %Row(Latitude,Vertical)
range_pix_diff = ((Longitude_2_11-Longitude_1_11)/range_m_1); %Column(Longitude, Parallel)

%Co-registrate the slave image to master image

A2_translation = [range_pix_diff,0]; %change 0 to row_pix_diff for normal case
A2_slave = imtranslate(A2,A2_translation);

%Resample the slave image to master image

%Azimuth size resampling

if size(A1,1) > size(A2,1)
A2_slave(:,size(A1,1)+1:size(A2,1)) = NaN;
elseif size(A1,1) < size(A2,1)
A2_slave(:,size(A1,1)+1:size(A2,1)) = [];
else
end

%Range size resampling

if size(A1,2) > size(A2,2)
A2_slave(:,size(A1,2)+1:size(A2,2)) = NaN;
elseif size(A1,2) < size(A2,2)
A2_slave(:,size(A1,2)+1:size(A2,2)) = [];
else

end

%Calculate the phase interferogram (master * conjugate_slave)

a2 = A2_slave';      A2_con = a2.';
phase1 = A1.*A2_con;
Phase1 = angle(phase1);

%Replace the '0' value to 'NaN' in inteferogram

for k=1:size(Phase1,1)
for l = 1:size(Phase1,2)
if Phase1(k,l) == 0;
Phase1(k,l) = NaN;
end
end
end

%Export phase value to binary for ENVI

name = horzcat('Phase_',file_1,'_',file_2,'.bin');
f1 = fopen(name,'w');
fwrite(f1, Phase1','double');
fclose(f1);

%Export row and column size

rcsize = horzcat(num2str(m),'. row : ',num2str(size(Phase1,1)),' col : ',num2str(size(Phase1,2)));
f2 = fopen('size.txt','a');
fwrite(f2,rcsize);
```

```
fprintf(f2,'₩n');
fclose(f2);

%Output result display through figure window

figure, imshow(Phase1,[-pi pi]), colormap jet,...
title(['Phase_ ',num2str(m),' || row : ',num2str(size(Phase1,1)),...
'   col : ', num2str(size(Phase1,2)),' || {₩lambda} : ',num2str(Wavelength),' cm']),...
colorbar('Ticks', [-pi 0 pi], 'TickLabels',{'-₩pi', '0   Phase', '₩pi'},'Fontsize',12), caxis([-pi pi])

end
end

fprintf('₩n Inteferogram making is done. ₩n₩n');
```

- L-밴드 SAR 처리 프로그램

```
function GSL_run_all(BaseDIR, GPRI_ref_north, GPRI_ref_east, GPRI_ref_alt, GPRI_geoid, GPRI_scan_heading,
first_az_ind, last_az_ind, elec_delay)

addpath(fullfile(pwd,'mfunctions_DEMO_GSL'));

% BASEDIR = fullfile(pwd, BaseDIR);
RAW_DIR = fullfile(BASEDIR, 'raw');
RAW_cplx_DIR = fullfile(BASEDIR, 'raw_cplx');
MOCO_DIR = fullfile(BASEDIR, 'raw_cplx');
SLC_DIR = fullfile(BASEDIR, 'slc');
DEM_DIR = fullfile(BASEDIR, 'DEM');
MLI_DIR = fullfile(BASEDIR, 'mli');
INT_DIR = fullfile(BASEDIR, 'int');
GEO_DIR = fullfile(BASEDIR, 'geo');

RAWFILE = fullfile(BASEDIR, 'raw', '*.raw');
RAWPARFILE = fullfile(BASEDIR, 'raw', '*.raw_par');

% Variables containing the names of the *.raw_par and *.raw radar data files are set up.
% In a addition, directory names are initialized.

%% Generate prefix, pattern, par_pattern name %%
% Only the data tested using the same parameteres should be put into
% BASEDIR
raw_struc = dir(RAWFILE);
RAWFILEname = raw_struc(1).name;
[RAWFILEsplit, matches] = strsplit(RAWFILEname, {'_', '.'});
RAWFILEstr = string(RAWFILEsplit);

prefix = char(strcat(RAWFILEstr(1), '_', RAWFILEstr(2), '_', RAWFILEstr(3), '_', RAWFILEstr(4), '_'));
pattern = char(strcat('_', RAWFILEstr(7), '_', RAWFILEstr(8), '_', RAWFILEstr(9)));

raw_par_struc = dir(RAWPARFILE);
RAWPARFILEname = raw_par_struc(1).name;
[RAWPARFILEsplit, matches2] = strsplit(RAWPARFILEname, {'_', '.'});
RAWPARFILEstr = string(RAWPARFILEsplit);
par_pattern = char(strcat('_', RAWPARFILEstr(7)));

%% preprocessing

% polarization
if strlength(RAWPARFILEstr(7)) == 1
polarization = char(strcat(RAWPARFILEstr(7), RAWPARFILEstr(7)));
else polarization = char(RAWPARFILEstr(7));
end

raw_struc = dir(fullfile(BASEDIR,'raw',[prefix '*' pattern '.raw']));
for i=1:length(raw_struc)
inRawFilename{i} = raw_struc(i).name;
end

raw_par_struc = dir(fullfile(BASEDIR,'raw',[prefix '*' par_pattern '.raw_par']));
for i=1:length(raw_par_struc)
inRawParFilename{i} = raw_par_struc(i).name;
end


% Set the 3dB beamwidth in elevation and azimuth of the radar antenna
% Rectangular L-band patch antennas
rg_3dB_bw = 20;
az_3dB_bw = 40;

% Quadratic L-band patch antennas
%rg_3dB_bw = 40;
%az_3dB_bw = 40;
```

```
% Set the azimuth oversampling for angularly sampled grid.
az_ovr_samp = 0.1

% Location and heading of pointing direction of radar
%strucLocation.GPRI_ref_north = GPRI_ref_north;    % deg
%strucLocation.GPRI_ref_east  = GPRI_ref_east  ;% deg
%strucLocation.GPRI_ref_alt   = GPRI_ref_alt    ;    % m
%strucLocation.GPRI_geoid          = GPRI_geoid     ; % m
%strucLocation.GPRI_scan_heading = GPRI_scan_heading; % deg

% Loop over the number of data sets
for i=1:length(inRawParFilename)

% Set output file names
outRawCpxFilename = regexprep(inRawFilename{i},'.raw','.raw_cplx')
outPROCparFilename = regexprep(inRawFilename{i},'.raw','.PROC_par')
outSARparFilename = regexprep(inRawFilename{i},'.raw','.SAR_par')
outPosFile = regexprep(inRawFilename{i},'.raw','.pos');
outVelFile = regexprep(inRawFilename{i},'.raw','.vel');
outPolarRecGridFilename = regexprep(inRawFilename{i},'.raw','_pol_rec_grid.XYZ');
outSLCparFilename = regexprep(inRawFilename{i},'.raw','.slc.par')

% Extract GAMMA L-band SAR data incl. Hilbert transform and range windowing
% The output *.raw_cplx, the *.PROC_par, and the *.SAR_par file are
% required as an input by the TDBP SAR processor
extractGammaSARL2SARData(fullfile(RAW_DIR,inRawFilename{i}), ...
fullfile(RAW_DIR, inRawParFilename{i}), ...
fullfile(RAW_cplx_DIR, outRawCpxFilename), ...
fullfile(RAW_cplx_DIR, outPROCparFilename), ...
fullfile(RAW_cplx_DIR, outSARparFilename), ...
rg_3dB_bw, ...
az_3dB_bw, ...
polarization);

% Create a position and velocity file for each radar echo acquired on the
% rail. Pos and vel are required as an input by the TDBP SAR processor
[railParam, pos, vel] = create_pos_vel_for_rail(fullfile(RAW_DIR,inRawParFilename{i}),fullfile(RAW_cplx_DIR,
outPosFile),fullfile(raw_cplx_dir, outVelFile));

% Create a polar reconstruction grid required as an input by the TDBP SAR processor
strucPolGrid                                                                                    =
create_polar_rec_grid_for_gsl_rail(fullfile(RAW_DIR,inRawParFilename{i}),az_3dB_bw,fullfile(RAW_cplx_DIR,outPol
arRecGridFilename),az_ovr_samp);

% Create SLC_par file for the polar-grid-based rail-SAR acquisition
create_slc_par_for_gsl_rail(strucPolGrid, ...
strucLocation, ...
fullfile(RAW_DIR,inRawParFilename{i}), ...
fullfile(RAW_cplx_DIR, outPROCparFilename), ...
fullfile(RAW_cplx_DIR, outSARparFilename), ...
fullfile(SLC_DIR, outSLCparFilename));

end

%% azimuth focusing

% Automatically create a DATE_LIST (with date and time) for
% rail-based GAMMA L-band SAR data sets
% based on the SAR_par file available in the raw_cplx folder

s e t e n v ( ' T D B P ' ,
'/home/jhju/test/test2/rsmas_insar_old_svn_r2398_LSAR/3rdparty/gamma/GAMMA_SOFTWARE-20200713/TDBP_
gpu/bin')
getenv('TDBP')
for i = 1:length(dir(RAWFILE))
RAWFILEname = raw_struc(i).name;
[RAWFILEsplit, matches] = strsplit(RAWFILEname, {'_', '.'});
RAWFILEstr = string(RAWFILEsplit);
NewDATELIST = strcat(RAWFILEstr(5), '_', RAWFILEstr(6));
DATELIST{1, length(dir(RAWFILE))} = [];
DATELIST{i} = NewDATELIST;
end

% Set TDBP processing parameters
traj_format='0'    %  0 : two ascii files (1. for pos., 2. for vel.)
grid_type='2'      %     type of image reconstruction grid:
%         0: SLC geometry (default)
%         1: DEM reconstruction grid
%         2: custom reconstruction grid (used to process directly
%            into the geometry of another SLC-type of reconstruction)

% SLC_format
SLC_format='0'     %    SLC output format (enter - for default, from PROC_par):
%     0: FCOMPLEX, pairs of 4-byte float values
```

```
%     1: SCOMPLEX, pairs of 2-byte short integers

% FFT upsampling ratio
upsamp_ratio='4'

dopcen_azi = '-'
phase_corr = '-'

%flag indicating whether start-stop-approximation holds
startstop_flg='0'       % 0: start-stop approximation invalid (FMCW case)
% 1: start-stop approximation valid (default)

% indices of first and last echo to consider:
first_az_ind='first_az_ind'
last_az_ind='last_az_ind'

plist='-'
pRC_bandpass1='-'
pRC_bandpass2='-'

% elec_delay=5.042 % input

for i = 1:numel(DATELIST)
j = DATELIST(i);
strDATELIST = string(j);
FILE_BASENAME = strcat(prefix, strDATELIST, pattern);

SAR_PAR = fullfile(RAW_cplx_DIR, strcat(FILE_BASENAME, '.', 'SAR_par'));
PROC_PAR = fullfile(RAW_cplx_DIR, strcat(FILE_BASENAME, '.', 'PROC_par'));
RAW_CPLX = fullfile(RAW_cplx_DIR, strcat(FILE_BASENAME, '.', 'raw_cplx'));

SLC_FILE = strcat(FILE_BASENAME, '.', 'slc');

sensor_pos = fullfile(MOCO_DIR, strcat(prefix, strDATELIST, pattern, '.', 'pos'));
sensor_vel = fullfile(MOCO_DIR, strcat(prefix, strDATELIST, pattern, '.', 'vel'));

grid_file = fullfile(RAW_cplx_DIR, strcat(FILE_BASENAME, '_pol_rec_grid.XYZ'));

SLC = fullfile(SLC_DIR, SLC_FILE);

SLC_par = fullfile(SLC_DIR, strcat(SLC_FILE, '.', 'par'));

% size of first dimension of the reconstruction grid
SLC_par_read = readGammaParFile(SLC_par);
grid_dim_orig = SLC_par_read.range_samples;
grid_dim = string(grid_dim_orig);

% TDBP focusing to SAR image in polar coordinates
command = append('az_proc_tdbp_gpu', ' ', SAR_PAR, ' ', PROC_PAR, ' ', RAW_CPLX, ' ', traj_format, ' ',
sensor_pos, ' ', sensor_vel, ' ', grid_type, ' ', grid_file, ' ', grid_dim, ' ',    SLC, ' ', SLC_format, ' ',
upsamp_ratio, ' ', dopcen_azi, ' ', phase_corr, ' ', startstop_flg, ' ', first_az_ind, ' ', last_az_ind, ' ', plist, ' ',
pRC_bandpass1, ' ', pRC_bandpass2, ' ', elec_delay)

[status, cmdout] = unix(command);
status
end

%% slc_postproc

suffix = '.slc'
filtsuffix = '.slc_filt'
suffix_par = '.slc.par'

valtype = 'float32'

% Read list of SLC file from the slc directory
slc_struc = dir(fullfile(SLC_DIR,['*' suffix]));
slcPar_struc = dir(fullfile(SLC_DIR,['*' suffix_par]));


% Set azimuth fraction of total azimuth bandwidth
% for which the azimuth spectral window is oversampling for angularly sampled grid.
%az_bw_frac = 0.1
az_bw_frac = 0.1

%betaKaiser = 2.120;     % passband ripple   .27    , stoppband ripple = -30 dB, PSLR: -19.0   dB
%betaKaiser = 3.384;     % passband ripple   .0864 , stoppband ripple = -40 dB, PSLR: -26.5   dB
betaKaiser = 4.538;      % passband ripple   .0274 , stoppband ripple = -50 dB, PSLR: -34.1   dB

% Apply azimuth spectral windowing
for i=1:length(slc_struc)

slcPar = readGammaParFile(fullfile(SLC_DIR,slcPar_struc(i).name));
rgDim=slcPar.range_samples;
```

```
azDim=slcPar.azimuth_lines;
slc = readMatrixNoHeader(fullfile(SLC_DIR,slc_struc(i).name),rgDim, azDim, valtype,1, 1,rgDim,1,azDim);

az_spec = fft(slc,[],2);
azWin = kaiserWindow(azDim,az_bw_frac, 0,betaKaiser);
slc_filtered = ifft(az_spec.*(ones(rgDim,1)*fftshift(azWin')),[],2);

regexprep(fullfile(SLC_DIR,slc_struc(i).name),suffix,filtsuffix)

writeMatrixNoHeader(fullfile(SLC_DIR,regexprep(slc_struc(i).name,suffix,filtsuffix)),slc_filtered,valtype);

end

%% intf_proc

intsuffix = '.int'
mlisuffix = '.mli'

% Read list of SLC file from the slc directory
slc_struc = dir(fullfile(SLC_DIR,['*' filtsuffix]));
slcPar_struc = dir(fullfile(SLC_DIR,['*' suffix_par]));

% generate filter for interferogram calculation
fsize = 3
sigma = round(fsize/2);
filter2D = fspecial('gaussian', fsize, sigma);

slcPar = readGammaParFile(fullfile(SLC_DIR,slcPar_struc(1).name));
rgDim=slcPar.range_samples;
azDim=slcPar.azimuth_lines;

% read reference slc
slc1 = readMatrixNoHeader(fullfile(SLC_DIR,slc_struc(1).name),rgDim, azDim, valtype,1, 1,rgDim,1,azDim);

%calculate MLI1 image

mli1 = calcFloatIntensity(slc1, filter2D);

% create MLI1 file name
mli1Filename = fullfile(MLI_DIR, regexprep(slc_struc(1).name,suffix,mlisuffix))

% write MLI1 to disk
writeMatrixNoHeader(mli1Filename,mli1,valtype);

% Toggle plotting of interferogram in HSV mode (to plot set verbose = 1 )
verbose = 0

for i=2:length(slc_struc)

slcPar = readGammaParFile(fullfile(SLC_DIR,slcPar_struc(i).name));
rgDim=slcPar.range_samples;
azDim=slcPar.azimuth_lines;

% read corresponding slc pair
slc2 = readMatrixNoHeader(fullfile(SLC_DIR,slc_struc(i).name),rgDim, azDim, valtype,1, 1,rgDim,1,azDim);

% calculate complex coherence (complex interferogram)
complexCoherence = calcComplexCoherence(slc1,slc2,filter2D);

% detect NaNs in the interferogram and set to 0.0 + j*0.0
nidx = isnan(complexCoherence);
fprintf('     *** number of NaNs in the array: %d₩n',nnz(nidx));
complexCoherence(nidx) = 0.0 + j*0.0;

% create interferogram file name
intFilename = fullfile(INT_DIR, regexprep(slc_struc(i).name,suffix,intsuffix))

% write interferogram to disk
writeMatrixNoHeader(intFilename,complexCoherence,valtype);

% calculate MLI2
mli2 = calcFloatIntensity(slc2,filter2D);

% create MLI2 file name
mli2Filename = fullfile(MLI_DIR, regexprep(slc_struc(i).name,suffix,mlisuffix))

% write MLI2 to disk
writeMatrixNoHeader(mli2Filename,mli2,valtype);

if(verbose)
%figure, imagesc(20*log10(abs(slc1))), axis xy, colorbar, colormap jet

pixCoordRefDim1 = 1626
pixCoordRefDim2 = 270
```

```
extDim1 = (1:rgDim);
extDim2 = 1:azDim
axLabelDim1 = 'Range pix'
axLabelDim2 = 'Azimuth pix'

sizeOfFont = 16
dBThreshold = 10

gcf                                                                                                                 =
plotPhaseDifferenceHSVScheme(complexCoherence(1:2500,:),slc1(1:2500,:),pixCoordRefDim1,pixCoordRefDim2,
extDim1(1:2500),extDim2,axLabelDim1,axLabelDim2,sizeOfFont,dBThreshold);
end
end
```

- 구글어스 기반 가시화 프로그램

```
#!/usr/bin/env python3
############################################################
# Program is part of LSAR code                           #
# Copyright (c) 2021, Sang-Hoon Hong                      #
# Author: Sang-Hoon Hong 2021                             #
############################################################
import os
import sys
import argparse
import numpy as np
import subprocess
from lxml import etree
from zipfile import ZipFile
import matplotlib as mpl
from matplotlib import pyplot as plt
try:
    from pykml.factory import KML_ElementMaker as KML
except ImportError:
    raise ImportError('Can not import pykml!')
############################################################
EXAMPLE = """example:
  export_kmz.py ./190213100847_190213102118.wgs84.disp 190213100847.mli.par
"""
def create_parser():
    parser = argparse.ArgumentParser(description='Export Google Earth KMZ file with gamma geocoded
image.',
                                     formatter_class=argparse.RawTextHelpFormatter,
                                     epilog=EXAMPLE)
    parser.add_argument('file', help='geocoded displacement observation file to be converted, in geo
coordinate.')
    parser.add_argument('dem_par', help='DEM/MAP parameter file')
    parser.add_argument('-amp', '--amp', dest='ampfile', metavar='FILE',
                        help='geocoded multilook image')
    parser.add_argument('-o', '--output', dest='outfile',
                        help='output file base name. Extension is fixed with .kmz')
    # Data
    parser.add_argument('-v','--vlim', dest='vlim', nargs=2, metavar=('MIN', 'MAX'), type=float,
                        help='Y/value limits for plotting.')
    parser.add_argument('-cycle', dest='cycle', type=float, default='0.05',
                        help='meters per color cycle Default: 0.05')

    # Figure
    fig = parser.add_argument_group('Figure')
    fig.add_argument('--dpi', dest='fig_dpi', metavar='NUM', type=int, default=600,
                     help='Figure DPI (dots per inch). Default: 600')

    fig.add_argument('--cbar-loc', dest='cbar_loc', choices=['lower left','lower right','upper left', 'upper
right'],
                     default='lower left', help='Location of colorbar in the screen. Default: lower left.')
    fig.add_argument('--cbar-label', dest='cbar_label', metavar='LABEL', default='Displacement (m)',
                     help='Colorbar label. Default: Displacement (m)')
    fig.add_argument('--cbar-bin-num', dest='cbar_bin_num', metavar='NUM', type=int, default=7,
                     help='Colorbar bin number. Default: 7')

    return parser
def attribute_gamma2roipac(par_dict_in):
    """Convert Gamma metadata into ROI_PAC/MintPy format."""
    par_dict = dict()
    for key, value in iter(par_dict_in.items()):
        par_dict[key] = value
    # LENGTH - number of rows
    for key in par_dict_in.keys():
        if any(key.startswith(i) for i in ['azimuth_lines',
                                           'nlines',
                                           'az_samp',
                                           'interferogram_azimuth_lines']):
            par_dict['LENGTH'] = par_dict[key]
```

```
    # WIDTH - number of columns
    for key in par_dict_in.keys():
        if any(key.startswith(i) for i in ['width',
                                           'range_samp',
                                           'interferogram_width']):
            par_dict['WIDTH'] = par_dict[key]
    # radar_frequency -> WAVELENGTH
    key = 'radar_frequency'
    if key in par_dict_in.keys():
        value = float(par_dict[key])
        par_dict['WAVELENGTH'] = str(SPEED_OF_LIGHT / value)
    # sar_to_earth_center/earth_radius_below_sensor -> HEIGHT/EARTH_RADIUS
    key = 'earth_radius_below_sensor'
    if key in par_dict_in.keys():
        Re = float(par_dict[key])
        par_dict['EARTH_RADIUS'] = str(Re)
        key2 = 'sar_to_earth_center'
        if key2 in par_dict_in.keys():
            value = float(par_dict[key2])
            par_dict['HEIGHT'] = str(value - Re)
    # sensor -> PLATFORM
    key = 'sensor'
    if key in par_dict_in.keys():
        par_dict['PLATFORM'] = par_dict[key]
    # heading -> ORBIT_DIRECTION
    key = 'heading'
    if key in par_dict_in.keys():
        value = float(par_dict[key])
        if (270 < value < 360) or (-90 < value < 90):
            par_dict['ORBIT_DIRECTION'] = 'ascending'
        else:
            par_dict['ORBIT_DIRECTION'] = 'descending'
    # azimuth_angle -> ANTENNA_SIDE
    key = 'azimuth_angle'
    if key in par_dict_in.keys():
        value = float(par_dict[key])
        if 0 < value < 180:
            par_dict['ANTENNA_SIDE'] = '-1'
        else:
            par_dict['ANTENNA_SIDE'] = '1'
    return par_dict
def standardize_metadata(metaDictIn, standardKeys=None):
    """Convert metadata input ROI_PAC/MintPy format (for metadata with the same values)."""
    if standardKeys is None:
        standardKeys = standardMetadataKeys
    # make a copy
    metaDict = dict()
    for key, value in iter(metaDictIn.items()):
        metaDict[key] = value
    # get potential keys to match
    in_keys = [i for i in metaDict.keys() if i not in standardKeys.keys()]
    std_keys = [i for i in standardKeys.keys() if i not in metaDict.keys()]
    # loop to find match and assign values
    for std_key in std_keys:
        cand_keys = standardKeys[std_key]
        cand_keys = [i for i in cand_keys if i in in_keys]
        if len(cand_keys) > 0:
            metaDict[std_key] = metaDict[cand_keys[0]]
    return metaDict
def read_gamma_par(fname, delimiter=':', skiprows=3):
    """Read GAMMA .par/.off file into a python dict structure.
    Parameters: fname : str.
                    File path of .par, .off file.
                delimiter : str, optional
                    String used to separate values.
                skiprows : int, optional
                    Skip the first skiprows lines.
    Returns:    parDict : dict
                    Attributes dictionary
    """
    # Read txt file
    with open(fname, 'r') as f:
        lines = f.readlines()[skiprows:]
    # convert list of str into dict
    parDict = {}
    for line in lines:
        c = [i.strip() for i in line.strip().split(delimiter, 1)]
        if len(c) < 2 or line.startswith(('%', '#')):
            next
        else:
            key = c[0]
            value = str.replace(c[1], '₩n', '').split("#")[0].split()[0].strip()
            parDict[key] = value
```

```
    return parDict
def cmd_line_parse(iargs=None):
    parser = create_parser()
    inps = parser.parse_args(args=iargs)
    par_dict = read_gamma_par(inps.dem_par)
    # Check: file in geo coord
    if 'corner_lat' not in par_dict.keys():
        raise Exception('ERROR: Input file does not have geo coornidate inforamtion.')
    # Check: if datum is wgs84
    if par_dict['datum_name'] != 'WGS':
        raise Exception('ERROR: Input file should have WGS datum.')
    return inps
def plot_colorbar(out_file, vmin, vmax, unit='cm/year', cmap='jet', figsize=(0.18, 3.6),
                  nbins=7, label='Mean LOS velocity'):
    fig, cax = plt.subplots(figsize=figsize)
    norm = mpl.colors.Normalize(vmin=vmin, vmax=vmax)
    cbar = mpl.colorbar.ColorbarBase(cax, cmap=plt.get_cmap(cmap), norm=norm, orientation='vertical')
    cbar.set_label('{} [{}]'.format(label, unit), fontsize=12)
    cbar.locator = mpl.ticker.MaxNLocator(nbins=nbins)
    cbar.update_ticks()
    cbar.ax.tick_params(which='both', labelsize=12)
    fig.patch.set_facecolor('white')
    fig.patch.set_alpha(0.7)
    print('writing', out_file)
    fig.savefig(out_file, bbox_inches='tight', facecolor=fig.get_facecolor(), dpi=300)
    return out_file
def generate_cbar_element(cbar_file, inps):
    # plot colobar and save as an image
    min_val = inps.cycle/2*-1
    max_val = inps.cycle/2
    call_str = "vis_colormap_bar.py " + inps.colormap + ".cm " + inps.file+'_cbar.png ' + str(min_val) + ' ' +
str(max_val) + ' -d ' + str(inps.fig_dpi) + ' -l ₩"' + inps.cbar_label + '₩"'
    print(call_str)
    res = subprocess.Popen(call_str, shell=True).wait()
    if res is not 0:
        err_str = 'ERROR during {}'.format(call_str)
        print(err_str)
        sys.exit(err_str)

    # colobar location
    if inps.cbar_loc.split()[0] == 'lower':
        oy, sy = '0', '0'
    elif inps.cbar_loc.split()[0] == 'upper':
        oy, sy = '1', '1'
    if inps.cbar_loc.split()[1] == 'left':
        ox, sx = '0', '0'
    elif inps.cbar_loc.split()[1] == 'right':
        ox, sx = '1', '1'
    # generate KML screen overlay object
    cbar_overlay = KML.ScreenOverlay(
        KML.name('colorbar'),
        KML.Icon(
            KML.href("{}".format(os.path.basename(cbar_file))),
            KML.viewBoundScale(0.75)
        ),
        KML.overlayXY(x=ox, y=oy, xunits="fraction", yunits="fraction"),
        KML.screenXY(x=sx, y=sy, xunits="fraction", yunits="fraction"),
        KML.size(x="0", y="0.5", xunits="fraction", yunits="fraction"),
        KML.rotation(0),
        KML.visibility(1),
        KML.open(0)
    )
    #print('add colorbar.')
    return cbar_overlay
############################################################
def write_kmz_file(inps):
    """ Generate Google Earth KMZ file for input data matrix.
    Inputs:
        data - 2D np.array in int/float, data matrix to write
        out_file - string, output file name
        metadata  - dict, containing the following attributes:
                WIDTH/LENGTH        : required, file size
                X/Y_FIRST/STEP      : required, for lat/lon spatial converage
                ref_x/y             : optional, column/row number of reference pixel
                PROJECT_NAME        : optional, for KMZ folder name
        inps - Namespace, optional, input options for display
    Output:
        kmz_file - string, output KMZ filename
    Example:
        data, atr = readfile.read(fname)
        out_file = pp.auto_figure_title(fname, None)+'.kmz'
    """
    if not inps:
        inps = cmd_line_parse()
```

```
    par_dict = read_gamma_par(inps.dem_par)
    print(inps)
    call_str = "$GAMMA_BIN/rashgt " + inps.file + " - " + par_dict['width'] + " 1 1 - - - " + str(inps.cycle)
+ " " + inps.file + ".ras"
    print(call_str)
    res = subprocess.Popen(call_str, shell=True).wait()
    if res is not 0:
        err_str = 'ERROR during {}'.format(call_str)
        print(err_str)
        sys.exit(err_str)
    call_str = "convert " + "-transparent " + "₩"rgb(0,0,0)₩" " + inps.file + ".ras " +  inps.file + ".png"
    print(call_str)
    res = subprocess.Popen(call_str, shell=True).wait()
    if res is not 0:
        err_str = 'ERROR during {}'.format(call_str)
        print(err_str)
        sys.exit(err_str)


    # 2.3 Generate KML file
    proj_name = 'PNU LSAR'
    doc = KML.kml(KML.Folder(KML.name(proj_name)))
    north = float(par_dict['corner_lat']) - float(par_dict['post_lat']) / 2.0
    south = north + float(par_dict['post_lat']) * float(par_dict['nlines'])
    west  = float(par_dict['corner_lon']) - float(par_dict['post_lon']) / 2.0
    east  = west + float(par_dict['post_lon']) * float(par_dict['width'])
    # Add data png file
    img_name = os.path.splitext(os.path.basename(inps.file+'.png'))[0]
    img = KML.GroundOverlay(
        KML.name(img_name),
        KML.Icon(
            KML.href(os.path.basename(inps.file+'.png'))
        ),
        KML.altitudeMode('clampToGround'),
        KML.LatLonBox(KML.north(str(north)),
                      KML.east(str(east)),
                      KML.south(str(south)),
                      KML.west(str(west)))
    )
    doc.Folder.append(img)

    cbar_overlay = generate_cbar_element(cbar_file, inps)
    doc.Folder.append(cbar_overlay)
    kmlstr = etree.tostring(doc, pretty_print=True).decode('utf8')
    # Write KML file
    kml_file = '{}.kml'.format(out_name_base)
    print('writing '+kml_file)
    with open(kml_file, 'w') as f:
        f.write(kmlstr)
    # 2.4 Generate KMZ file, by
    # 1) go to the directory of kmz file
    run_dir = os.path.abspath(os.getcwd())
    os.chdir(os.path.abspath(os.path.dirname(inps.outfile)))
    # 2) zip all data files
    kmz_file = '{}.kmz'.format(out_name_base)
    with ZipFile(kmz_file, 'w') as fz:
        for fname in [kml_file, inps.file + '.png', cbar_file]:
            fz.write(os.path.relpath(fname))
            os.remove(fname)
    # 3) go back to the running directory
    os.chdir(run_dir)
    print('merged all files to {}'.format(kmz_file))
    return kmz_file
##############################################################
def main(iargs=None):
    inps = cmd_line_parse(iargs)

    if not inps.outfile:
        inps.outfile = '{}.kmz'.format(inps.file)
    inps.outfile = os.path.abspath(inps.outfile)

    write_kmz_file(inps=inps)
    return inps.outfile
##############################################################
if __name__ == '__main__':
    main(sys.argv[1:])
```

부록 3. 포트홀 탐지기술 소프트웨어 소스코드

○ 포트홀 자동탐지 소프트웨어 소스코드

- 소스 코드 원격 저장소: http://yunsung9503@itgit.cu.ac.kr/SnSLab/Development-of-Drone-Based-for-Detecting-Pothole-Demo-segmentation
- 멀티프로세싱 풀을 관리하기 위한 코드

plagmanager.py

```
"""
2020.09.11 yunsung Han code
"""
from multiprocessing import Process, Queue, current_process
import time
import multiprocessing as mp
import os
import queue # imported for using queue.Empty exception
import inference
import inference_road
import pathlib
import shutil
import ntpath
import timeit
import psutil
#ROOT_DIR confirm
Path = os.getcwd().split('/')[:-2]
Path ='/'.join(Path)
ROOT_DIR=Path
def do_job(tasks_to_accomplish,done_queue):
    """ if data exists running inference
    Args :
        tasks_to_accomplish (queue) : image queue
        done_queue (queue) : end image queue
    Returns :
        None
    """
    count =1
    for data in iter(tasks_to_accomplish.get, "interrupt"):
        count +=1
        road_image = inference_road.image_detecting(data)
        inference.image_detecting(data, road_image)
        done_queue.put(data)
        print("count:", count)
if __name__=="__main__":
    mp.set_start_method('spawn', force =True)  ###this code make you can multi process in this code
    number_of_processes =1 ## process number
    processes = []
    tasks_to_accomplish = Queue() ## queue
    done_queue = Queue() ## queue
    for w in range(number_of_processes):
        p = Process(target =do_job, args =(tasks_to_accomplish,done_queue))
        processes.append(p)
        p.start()
    print("Pool is working")
    count =0
    while True:
        image1 = pathlib.Path(os.path.join(ROOT_DIR, 'media','temp'))
        image_list = list(image1.glob('*'))
        # print(image_list)
        time.sleep(0.1)
        if len(image_list) !=0:
            start = timeit.default_timer()
            count = count +1
            for i in image_list:
                filename = ntpath.basename(i)
                ori_loc = os.path.join(ROOT_DIR, 'media','temp',filename)
                moved_loc = os.path.join(ROOT_DIR,'media','process',filename)
                shutil.move(ori_loc,moved_loc)
                tasks_to_accomplish.put(moved_loc)
```

- 포트홀 추론 코드

inference.py

```
#!/usr/bin/env python
# coding: utf-8
```

```
# # Mask R-CNN Demo
#
# A quick intro to using the pre-trained model to detect and segment objects.
import pymysql
import skimage.io
from mrcnn.config import Config
from mrcnn import utils
from mrcnn import model as modellib
from mrcnn import visualize
from keras import backend as K
import os
path = os.path.abspath(visualize.__file__)
Path = os.getcwd()
Path = Path.split('/')
Path = Path[:-2]
Path ='/'.join(Path)
ORI_DIR=Path
# Root directory of the project
ROOT_DIR = os.path.join(os.getcwd())
# Directory to save logs and trained model
MODEL_DIR = os.path.join(ORI_DIR,'blog','M_DetectingPothole', "logs")
# Local path to trained weights file
'''
Configurations
We'll be using a model trained on the MS -COCO dataset. The configurations of this model
are in the ```CocoConfig``` class in ```coco.py```.
For inferencing, modify the configurations a bit to fit the task. To do so, sub -class
the ```CocoConfig``` class and override the attributes you need to change.
'''
class CocoConfig(Config):
    """Configuration for training on MS COCO.
    Derives from the base Config class and overrides values specific
    to the COCO dataset.
    """
    # Give the configuration a recognizable name
    NAME = "pothole"
    # We use a GPU with 12GB memory, which can fit two images.
    # Adjust down if you use a smaller GPU.
    IMAGES_PER_GPU = 1
    # Uncomment to train on 8 GPUs (default is 1)
    # GPU_COUNT = 8
    # Number of classes (including background) you must change in 120 or 160 model
    NUM_CLASSES = 80 + 1  # COCO has 80 classes
class InferenceConfig(CocoConfig):
    # Set batch size to 1 since we'll be running inference on
    # one image at a time. Batch size = GPU_COUNT * IMAGES_PER_GPU
    GPU_COUNT =1
    IMAGES_PER_GPU =1
config = InferenceConfig()
config.display()
class_names = ['BG', 'POTHOLE'] ## class_name
def get_image_id(image_name):
    '''
    get id in database
    param :
        image_name : inferenced image name
    return
        row : id in database
    '''
    pothole_db = pymysql.connect(
        host='localhost',
        user='root',
        password='20121208',
        db='test_server',
        charset='utf8'
    )
    cursor = pothole_db.cursor(pymysql.cursors.DictCursor)
    while True:
        cursor.execute("select id from blog_image_video where name='"+image_name +"'")
        row = cursor.fetchone()
        cursor.close()
        if row !=None:
            break
        else:
            pass
    return row
def image_detecting(image_list, road_image):
    """Read image and infernece image and then go to visualize module
    params :
        image_list (list) : only one image link
    """
    #TRAIN_MODEL_PATH            =            os.path.join(MODEL_DIR,            "crack",
"mask_rcnn_pothole_test_coco_0120.h5")
```

```
    #      TRAIN_MODEL_PATH      =      os.path.join(MODEL_DIR,      "0812_1253_new_data",
"mask_rcnn_pothole_test_coco_0120.h5")
    TRAIN_MODEL_PATH        =        os.path.join(MODEL_DIR,        "0901_1627_new_data",
"mask_rcnn_pothole_test_coco_0120.h5")
    #        TRAIN_MODEL_PATH        =        os.path.join(MODEL_DIR,        "0923_1135",
"mask_rcnn_pothole_test_coco_0120.h5") ## Train model path
    #           TRAIN_MODEL_PATH           =os.path.join(MODEL_DIR,           "0730_1745",
"mask_rcnn_pothole_test_coco_0060.h5")
    #             TRAIN_MODEL_PATH             =             os.path.join(MODEL_DIR,"road",
"mask_rcnn_pothole_test_coco_0120.h5")
    # Download COCO trained weights from Releases if needed
    if not os.path.exists(TRAIN_MODEL_PATH): ## if TRAIN_MODEL_PATH does not exist
        utils.download_trained_weights(TRAIN_MODEL_PATH)
    image_list = list(str(image_list))
    path =''.join(image_list) ##pathlib data type change to list
    K.clear_session() ## keras session clear
    model = modellib.MaskRCNN(mode ="inference", model_dir =MODEL_DIR, config =config) ##
model setting
# Load weights trained on MS-COCO
    model.load_weights(TRAIN_MODEL_PATH, by_name=True) ## weight load
    temp =str(path).split('/')[-1]
    temp = temp.split('@')
    image = skimage.io.imread(path) ## image read
    results = model.detect([image], verbose =0) ## detecting pothole in image
    r = results[0]
    image_id = get_image_id(temp[1])
    ## visualize instance segmentation result image
    visualize.display_instances(os.path.join(ORI_DIR, 'media', temp[1][:-4]), image_id,
temp[1], image,
                                r['rois'], r['masks'], r['class_ids'], road_image)
    if os.path.isfile(path):
        os.remove(path)
    else:
        pass
```

- 도로 추론 코드

inference_road.py

```
#!/usr/bin/env python
# coding: utf-8
# # Mask R-CNN Demo
#
# A quick intro to using the pre-trained model to detect and segment objects.
# In[29]:
import skimage.io
from mrcnn.config import Config
from mrcnn import utils
from mrcnn import model as modellib
from mrcnn import visualize
from keras import backend as K
import os
path = os.path.abspath(visualize.__file__)
Path = os.getcwd()
Path = Path.split('/')
Path = Path[:-2]
Path ='/'.join(Path)
ORI_DIR=Path
# Root directory of the project
ROOT_DIR = os.path.join(os.getcwd())
# Directory to save logs and trained model
MODEL_DIR = os.path.join(ORI_DIR,'blog','M_DetectingPothole', "logs")
# Local path to trained weights file
# ## Configurations
#
# We'll be using a model trained on the MS-COCO dataset. The configurations of this model
are in the ```CocoConfig``` class in ```coco.py```.
#
# For inferencing, modify the configurations a bit to fit the task. To do so, sub-class
the ```CocoConfig``` class and override the attributes you need to change.
class CocoConfig(Config):
    """Configuration for training on MS COCO.
    Derives from the base Config class and overrides values specific
    to the COCO dataset.
    """
    # Give the configuration a recognizable name
    NAME ="pothole"
    # We use a GPU with 12GB memory, which can fit two images.
    # Adjust down if you use a smaller GPU.
    IMAGES_PER_GPU =1
    # Uncomment to train on 8 GPUs (default is 1)
    # GPU_COUNT = 8
```

```
    # Number of classes (including background) you must change in 120 or 160 model
    NUM_CLASSES =80 +1  # COCO has 80 classes
class InferenceConfig(CocoConfig):
    # Set batch size to 1 since we'll be running inference on
    # one image at a time. Batch size = GPU_COUNT * IMAGES_PER_GPU
    GPU_COUNT =1
    IMAGES_PER_GPU =1
config = InferenceConfig()
config.display()
# COCO Class names
# Index of the class in the list is its ID. For example, to get ID of
# the teddy bear class, use: class_names.index('teddy bear')
class_names = ['BG', 'ROAD']
def image_detecting(image_list):
    '''
    Read image and infernece image and then go to visualize module
    params :
        image_list (list) : only one image link
    '''
    #TRAIN_MODEL_PATH           =            os.path.join(MODEL_DIR,            "crack",
"mask_rcnn_pothole_test_coco_0120.h5")
    #      TRAIN_MODEL_PATH      =      os.path.join(MODEL_DIR,      "0812_1253_new_data",
"mask_rcnn_pothole_test_coco_0120.h5")
    #      TRAIN_MODEL_PATH      =      os.path.join(MODEL_DIR,      "0901_1627_new_data",
"mask_rcnn_pothole_test_coco_0120.h5")
    #           TRAIN_MODEL_PATH           =os.path.join(MODEL_DIR,           "0730_1745",
"mask_rcnn_pothole_test_coco_0060.h5")
    TRAIN_MODEL_PATH                 =                    os.path.join(MODEL_DIR,"road",
"mask_rcnn_pothole_test_coco_0120.h5")
    # Download COCO trained weights from Releases if needed
    if not os.path.exists(TRAIN_MODEL_PATH):
        utils.download_trained_weights(TRAIN_MODEL_PATH)
    image_list = list(str(image_list))
    path =''.join(image_list)
    K.clear_session()
    model = modellib.MaskRCNN(mode ="inference", model_dir =MODEL_DIR, config =config)
# Load weights trained on MS-COCO
    model.load_weights(TRAIN_MODEL_PATH, by_name=True)
    tmp =str(path).split('/')
    temp = tmp[-1]
    temp = temp.split('@')
    image = skimage.io.imread(path)
    results = model.detect([image], verbose =0)
    r = results[0]
    road_image = visualize.display_road(image,
                               r['rois'], r['masks'], r['class_ids'])
    return road_image
```

- 설정 코드

config.py

```
"""
Mask R-CNN
Base Configurations class.
Copyright (c) 2017 Matterport, Inc.
Licensed under the MIT License (see LICENSE for details)
Written by Waleed Abdulla
"""
import math
import numpy as np
# Base Configuration Class
# Don't use this class directly. Instead, sub-class it and override
# the configurations you need to change.
class Config(object):
    """Base configuration class. For custom configurations, create a
    sub-class that inherits from this one and override properties
    that need to be changed.
    """
    # Name the configurations. For example, 'COCO', 'Experiment 3', ...etc.
    # Useful if your code needs to do things differently depending on which
    # experiment is running.
    NAME =None  # Override in sub-classes
    # NUMBER OF GPUs to use. For CPU training, use 1
    GPU_COUNT =1
    # Number of images to train with on each GPU. A 12GB GPU can typically
    # handle 2 images of 1024x1024px.
    # Adjust based on your GPU memory and image sizes. Use the highest
    # number that your GPU can handle for best performance.
    IMAGES_PER_GPU =2
    # Number of training steps per epoch
    # This doesn't need to match the size of the training set. Tensorboard
```

```
    # updates are saved at the end of each epoch, so setting this to a
    # smaller number means getting more frequent TensorBoard updates.
    # Validation stats are also calculated at each epoch end and they
    # might take a while, so don't set this too small to avoid spending
    # a lot of time on validation stats.
    STEPS_PER_EPOCH =1000
    # Number of validation steps to run at the end of every training epoch.
    # A bigger number improves accuracy of validation stats, but slows
    # down the training.
    VALIDATION_STEPS =50
    # Backbone network architecture
    # Supported values are: resnet50, resnet101
    BACKBONE ="resnet101"
    # The strides of each layer of the FPN Pyramid. These values
    # are based on a Resnet101 backbone.
    BACKBONE_STRIDES = [4, 8, 16, 32, 64]
    # Number of classification classes (including background)
    NUM_CLASSES =1  # Override in sub-classes
    # Length of square anchor side in pixels
    RPN_ANCHOR_SCALES = (32, 64, 128, 256, 512)
    # Ratios of anchors at each cell (width/height)
    # A value of 1 represents a square anchor, and 0.5 is a wide anchor
    RPN_ANCHOR_RATIOS = [0.5, 1, 2]
    # Anchor stride
    # If 1 then anchors are created for each cell in the backbone feature map.
    # If 2, then anchors are created for every other cell, and so on.
    RPN_ANCHOR_STRIDE =1
    # Non-max suppression threshold to filter RPN proposals.
    # You can reduce this during training to generate more propsals.
    RPN_NMS_THRESHOLD =0.7
    # How many anchors per image to use for RPN training
    RPN_TRAIN_ANCHORS_PER_IMAGE =256
    # ROIs kept after non-maximum supression (training and inference)
    POST_NMS_ROIS_TRAINING =2000
    POST_NMS_ROIS_INFERENCE =1000
    # If enabled, resizes instance masks to a smaller size to reduce
    # memory load. Recommended when using high-resolution images.
    USE_MINI_MASK =True
    MINI_MASK_SHAPE = (56, 56)  # (height, width) of the mini-mask
    # Input image resing
    # Images are resized such that the smallest side is >= IMAGE_MIN_DIM and
    # the longest side is <= IMAGE_MAX_DIM. In case both conditions can't
    # be satisfied together the IMAGE_MAX_DIM is enforced.
    IMAGE_MIN_DIM =800
    IMAGE_MAX_DIM =1024
    # If True, pad images with zeros such that they're (max_dim by max_dim)
    IMAGE_PADDING =True  # currently, the False option is not supported
    # Image mean (RGB)
    MEAN_PIXEL = np.array([123.7, 116.8, 103.9])
    # Number of ROIs per image to feed to classifier/mask heads
    # The Mask RCNN paper uses 512 but often the RPN doesn't generate
    # enough positive proposals to fill this and keep a positive:negative
    # ratio of 1:3. You can increase the number of proposals by adjusting
    # the RPN NMS threshold.
    TRAIN_ROIS_PER_IMAGE =200
    # Percent of positive ROIs used to train classifier/mask heads
    ROI_POSITIVE_RATIO =0.33
    # Pooled ROIs
    POOL_SIZE =7
    MASK_POOL_SIZE =14
    MASK_SHAPE = [28, 28]
    # Maximum number of ground truth instances to use in one image
    MAX_GT_INSTANCES =100
    # Bounding box refinement standard deviation for RPN and final detections.
    RPN_BBOX_STD_DEV = np.array([0.1, 0.1, 0.2, 0.2])
    BBOX_STD_DEV = np.array([0.1, 0.1, 0.2, 0.2])
    # Max number of final detections
    DETECTION_MAX_INSTANCES =100
    # Minimum probability value to accept a detected instance
    # ROIs below this threshold are skipped
    DETECTION_MIN_CONFIDENCE =0.7
    # Non-maximum suppression threshold for detection
    DETECTION_NMS_THRESHOLD =0.3
    # Learning rate and momentum
    # The Mask RCNN paper uses lr=0.02, but on TensorFlow it causes
    # weights to explode. Likely due to differences in optimzer
    # implementation.
    LEARNING_RATE =0.001
    LEARNING_MOMENTUM =0.9
    # Weight decay regularization
    WEIGHT_DECAY =0.0001
    # Use RPN ROIs or externally generated ROIs for training
    # Keep this True for most situations. Set to False if you want to train
```

```
    # the head branches on ROI generated by code rather than the ROIs from
    # the RPN. For example, to debug the classifier head without having to
    # train the RPN.
    USE_RPN_ROIS =True
    def __init__(self):
        """Set values of computed attributes."""
        # Effective batch size
        self.BATCH_SIZE =self.IMAGES_PER_GPU *self.GPU_COUNT
        # Input image size
        self.IMAGE_SHAPE = np.array(
            [self.IMAGE_MAX_DIM, self.IMAGE_MAX_DIM, 3])
        # Compute backbone size from input image size
        self.BACKBONE_SHAPES = np.array(
            [[int(math.ceil(self.IMAGE_SHAPE[0] / stride)),
              int(math.ceil(self.IMAGE_SHAPE[1] / stride))]
             for stride in self.BACKBONE_STRIDES])
    def display(self):
        """Display Configuration values."""
        print("\nConfigurations:")
        for a in dir(self):
            if not a.startswith("__") and not callable(getattr(self, a)):
                print("{:30} {}".format(a, getattr(self, a)))
        print("\n")
```

- 모델망 설정 코드

model.py

```
"""
Mask R-CNN
The main Mask R-CNN model implemenetation.
Copyright (c) 2017 Matterport, Inc.
Licensed under the MIT License (see LICENSE for details)
Written by Waleed Abdulla
"""
import os
import sys
import glob
import random
import math
import datetime
import itertools
import json
import re
import logging
from collections import OrderedDict
import numpy as np
import scipy.misc
import tensorflow as tf
import keras
import keras.backend as K
import keras.layers as KL
import keras.initializers as KI
import keras.engine as KE
import keras.models as KM
from mrcnn import utils
# Requires TensorFlow 1.3+ and Keras 2.0.8+.
from distutils.version import LooseVersion
assert LooseVersion(tf.__version__) >= LooseVersion("1.3")
assert LooseVersion(keras.__version__) >= LooseVersion('2.0.8')
os.environ["CUDA_VISIBLE_DEVICES"] ="0"
# def get_session(gpu_fraction=0.3):
#     '''Assume that you have 6GB of GPU memory and want to allocate ~2GB'''
#
#
#     gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=gpu_fraction)
#
#     return tf.Session(config=tf.ConfigProto(gpu_options=gpu_options))
# K.set_session(get_session())
# def get_session(gpu_fraction=0.3):
#     '''Assume that you have 6GB of GPU memory and want to allocate ~2GB'''
#
#
#     config = tf.ConfigProto()
#     config.gpu_options.allow_growth =True
#
#     return tf.Session(config=config)
# K.set_session(get_session())
############################################################
#  Utility Functions
############################################################
def log(text, array =None):
```

```
    """Prints a text message. And, optionally, if a Numpy array is provided it
    prints it's shape, min, and max values.
    """
    if array is not None:
        text = text.ljust(25)
        text += ("shape: {:20}  min: {:10.5f}  max: {:10.5f}".format(
            str(array.shape),
            array.min() if array.size else "",
            array.max() if array.size else ""))
    print(text)
class BatchNorm(KL.BatchNormalization):
    """Batch Normalization class. Subclasses the Keras BN class and
    hardcodes training=False so the BN layer doesn't update
    during training.
    Batch normalization has a negative effect on training if batches are small
    so we disable it here.
    """
    def call(self, inputs, training =None):
        return super(self.__class__, self).call(inputs, training =False)
############################################################
#  Resnet Graph
############################################################
# Code adopted from:
# https://github.com/fchollet/deep-learning-models/blob/master/resnet50.py
def identity_block(input_tensor, kernel_size, filters, stage, block,
                   use_bias=True):
    """The identity_block is the block that has no conv layer at shortcut
    # Arguments
        input_tensor: input tensor
        kernel_size: defualt 3, the kernel size of middle conv layer at main path
        filters: list of integers, the nb_filters of 3 conv layer at main path
        stage: integer, current stage label, used for generating layer names
        block: 'a','b'..., current block label, used for generating layer names
    """
    nb_filter1, nb_filter2, nb_filter3 = filters
    conv_name_base ='res'+str(stage) + block +'_branch'
    bn_name_base ='bn'+str(stage) + block +'_branch'
    x = KL.Conv2D(nb_filter1, (1, 1), name =conv_name_base +'2a',
                  use_bias=use_bias)(input_tensor)
    x = BatchNorm(axis =3, name =bn_name_base +'2a')(x)
    x = KL.Activation('relu')(x)
    x = KL.Conv2D(nb_filter2, (kernel_size, kernel_size), padding ='same',
                  name=conv_name_base +'2b', use_bias =use_bias)(x)
    x = BatchNorm(axis =3, name =bn_name_base +'2b')(x)
    x = KL.Activation('relu')(x)
    x = KL.Conv2D(nb_filter3, (1, 1), name =conv_name_base +'2c',
                  use_bias=use_bias)(x)
    x = BatchNorm(axis =3, name =bn_name_base +'2c')(x)
    x = KL.Add()([x, input_tensor])
    x = KL.Activation('relu', name ='res'+str(stage) + block +'_out')(x)
    return x
def conv_block(input_tensor, kernel_size, filters, stage, block,
               strides=(2, 2), use_bias =True):
    """conv_block is the block that has a conv layer at shortcut
    # Arguments
        input_tensor: input tensor
        kernel_size: defualt 3, the kernel size of middle conv layer at main path
        filters: list of integers, the nb_filters of 3 conv layer at main path
        stage: integer, current stage label, used for generating layer names
        block: 'a','b'..., current block label, used for generating layer names
    Note that from stage 3, the first conv layer at main path is with subsample=(2,2)
    And the shortcut should have subsample=(2,2) as well
    """
    nb_filter1, nb_filter2, nb_filter3 = filters
    conv_name_base ='res'+str(stage) + block +'_branch'
    bn_name_base ='bn'+str(stage) + block +'_branch'
    x = KL.Conv2D(nb_filter1, (1, 1), strides =strides,
                  name=conv_name_base +'2a', use_bias =use_bias)(input_tensor)
    x = BatchNorm(axis =3, name =bn_name_base +'2a')(x)
    x = KL.Activation('relu')(x)
    x = KL.Conv2D(nb_filter2, (kernel_size, kernel_size), padding ='same',
                  name=conv_name_base +'2b', use_bias =use_bias)(x)
    x = BatchNorm(axis =3, name =bn_name_base +'2b')(x)
    x = KL.Activation('relu')(x)
    x = KL.Conv2D(nb_filter3, (1, 1), name =conv_name_base +
                  '2c', use_bias =use_bias)(x)
    x = BatchNorm(axis =3, name =bn_name_base +'2c')(x)
    shortcut = KL.Conv2D(nb_filter3, (1, 1), strides =strides,
                         name=conv_name_base +'1', use_bias =use_bias)(input_tensor)
    shortcut = BatchNorm(axis =3, name =bn_name_base +'1')(shortcut)
    x = KL.Add()([x, shortcut])
    x = KL.Activation('relu', name ='res'+str(stage) + block +'_out')(x)
    return x
```

```
def resnet_graph(input_image, architecture, stage5 =False):
    assert architecture in ["resnet50", "resnet101"]
    # Stage 1
    x = KL.ZeroPadding2D((3, 3))(input_image)
    x = KL.Conv2D(64, (7, 7), strides =(2, 2), name ='conv1', use_bias =True)(x)
    x = BatchNorm(axis =3, name ='bn_conv1')(x)
    x = KL.Activation('relu')(x)
    C1 = x = KL.MaxPooling2D((3, 3), strides =(2, 2), padding ="same")(x)
    # Stage 2
    x = conv_block(x, 3, [64, 64, 256], stage =2, block ='a', strides =(1, 1))
    x = identity_block(x, 3, [64, 64, 256], stage =2, block ='b')
    C2 = x = identity_block(x, 3, [64, 64, 256], stage =2, block ='c')
    # Stage 3
    x = conv_block(x, 3, [128, 128, 512], stage =3, block ='a')
    x = identity_block(x, 3, [128, 128, 512], stage =3, block ='b')
    x = identity_block(x, 3, [128, 128, 512], stage =3, block ='c')
    C3 = x = identity_block(x, 3, [128, 128, 512], stage =3, block ='d')
    # Stage 4
    x = conv_block(x, 3, [256, 256, 1024], stage =4, block ='a')
    block_count = {"resnet50": 5, "resnet101": 22}[architecture]
    for i in range(block_count):
        x = identity_block(x, 3, [256, 256, 1024], stage =4, block =chr(98 + i))
    C4 = x
    # Stage 5
    if stage5:
        x = conv_block(x, 3, [512, 512, 2048], stage =5, block ='a')
        x = identity_block(x, 3, [512, 512, 2048], stage =5, block ='b')
        C5 = x = identity_block(x, 3, [512, 512, 2048], stage =5, block ='c')
    else:
        C5 =None
    return [C1, C2, C3, C4, C5]
############################################################
#  Proposal Layer
############################################################
config = tf.ConfigProto()
config.gpu_options.allow_growth =True
session = tf.Session(config =config)
def apply_box_deltas_graph(boxes, deltas):
    """Applies the given deltas to the given boxes.
    boxes: [N, 4] where each row is y1, x1, y2, x2
    deltas: [N, 4] where each row is [dy, dx, log(dh), log(dw)]
    """
    # Convert to y, x, h, w
    height = boxes[:, 2] - boxes[:, 0]
    width = boxes[:, 3] - boxes[:, 1]
    center_y = boxes[:, 0] +0.5 * height
    center_x = boxes[:, 1] +0.5 * width
    # Apply deltas
    center_y += deltas[:, 0] * height
    center_x += deltas[:, 1] * width
    height *= tf.exp(deltas[:, 2])
    width *= tf.exp(deltas[:, 3])
    # Convert back to y1, x1, y2, x2
    y1 = center_y -0.5 * height
    x1 = center_x -0.5 * width
    y2 = y1 + height
    x2 = x1 + width
    result = tf.stack([y1, x1, y2, x2], axis =1, name ="apply_box_deltas_out")
    return result
def clip_boxes_graph(boxes, window):
    """
    boxes: [N, 4] each row is y1, x1, y2, x2
    window: [4] in the form y1, x1, y2, x2
    """
    # Split corners
    wy1, wx1, wy2, wx2 = tf.split(window, 4)
    y1, x1, y2, x2 = tf.split(boxes, 4, axis =1)
    # Clip
    y1 = tf.maximum(tf.minimum(y1, wy2), wy1)
    x1 = tf.maximum(tf.minimum(x1, wx2), wx1)
    y2 = tf.maximum(tf.minimum(y2, wy2), wy1)
    x2 = tf.maximum(tf.minimum(x2, wx2), wx1)
    clipped = tf.concat([y1, x1, y2, x2], axis =1, name ="clipped_boxes")
    clipped.set_shape((clipped.shape[0], 4))
    return clipped
class ProposalLayer(KE.Layer):
    """Receives anchor scores and selects a subset to pass as proposals
    to the second stage. Filtering is done based on anchor scores and
    non-max suppression to remove overlaps. It also applies bounding
    box refinement deltas to anchors.
    Inputs:
        rpn_probs: [batch, anchors, (bg prob, fg prob)]
        rpn_bbox: [batch, anchors, (dy, dx, log(dh), log(dw))]
```

```
    Returns:
        Proposals in normalized coordinates [batch, rois, (y1, x1, y2, x2)]
    """
    def __init__(self, proposal_count, nms_threshold, anchors,
                 config=None, **kwargs):
        """
        anchors: [N, (y1, x1, y2, x2)] anchors defined in image coordinates
        """
        super(ProposalLayer, self).__init__(**kwargs)
        self.config = config
        self.proposal_count = proposal_count
        self.nms_threshold = nms_threshold
        self.anchors = anchors.astype(np.float32)
    def call(self, inputs):
        # Box Scores. Use the foreground class confidence. [Batch, num_rois, 1]
        scores = inputs[0][:, :, 1]
        # Box deltas [batch, num_rois, 4]
        deltas = inputs[1]
        deltas = deltas * np.reshape(self.config.RPN_BBOX_STD_DEV, [1, 1, 4])
        # Base anchors
        anchors =self.anchors
        # Improve performance by trimming to top anchors by score
        # and doing the rest on the smaller subset.
        pre_nms_limit = min(6000, self.anchors.shape[0])
        ix = tf.nn.top_k(scores, pre_nms_limit, sorted =True,
                         name="top_anchors").indices
        scores = utils.batch_slice([scores, ix], lambda x, y: tf.gather(x, y),
                                   self.config.IMAGES_PER_GPU)
        deltas = utils.batch_slice([deltas, ix], lambda x, y: tf.gather(x, y),
                                   self.config.IMAGES_PER_GPU)
        anchors = utils.batch_slice(ix, lambda x: tf.gather(anchors, x),
                                    self.config.IMAGES_PER_GPU,
                                    names=["pre_nms_anchors"])
        # Apply deltas to anchors to get refined anchors.
        # [batch, N, (y1, x1, y2, x2)]
        boxes = utils.batch_slice([anchors, deltas],
                                  lambda x, y: apply_box_deltas_graph(x, y),
                                  self.config.IMAGES_PER_GPU,
                                  names=["refined_anchors"])
        # Clip to image boundaries. [batch, N, (y1, x1, y2, x2)]
        height, width =self.config.IMAGE_SHAPE[:2]
        window = np.array([0, 0, height, width]).astype(np.float32)
        boxes = utils.batch_slice(boxes,
                                  lambda x: clip_boxes_graph(x, window),
                                  self.config.IMAGES_PER_GPU,
                                  names=["refined_anchors_clipped"])
        # Filter out small boxes
        # According to Xinlei Chen's paper, this reduces detection accuracy
        # for small objects, so we're skipping it.
        # Normalize dimensions to range of 0 to 1.
        normalized_boxes = boxes / np.array([[height, width, height, width]])
        # Non-max suppression
        def nms(normalized_boxes, scores):
            indices = tf.image.non_max_suppression(
                normalized_boxes, scores, self.proposal_count,
                self.nms_threshold, name ="rpn_non_max_suppression")
            proposals = tf.gather(normalized_boxes, indices)
            # Pad if needed
            padding = tf.maximum(self.proposal_count - tf.shape(proposals)[0], 0)
            proposals = tf.pad(proposals, [(0, padding), (0, 0)])
            return proposals
        proposals = utils.batch_slice([normalized_boxes, scores], nms,
                                      self.config.IMAGES_PER_GPU)
        return proposals
    def compute_output_shape(self, input_shape):
        return (None, self.proposal_count, 4)
############################################################
#  ROIAlign Layer
############################################################
def log2_graph(x):
    """Implementatin of Log2. TF doesn't have a native implemenation."""
    return tf.log(x) / tf.log(2.0)
class PyramidROIAlign(KE.Layer):
    """Implements ROI Pooling on multiple levels of the feature pyramid.
    Params:
    - pool_shape: [height, width] of the output pooled regions. Usually [7, 7]
    - image_shape: [height, width, channels]. Shape of input image in pixels
    Inputs:
    - boxes: [batch, num_boxes, (y1, x1, y2, x2)] in normalized
             coordinates. Possibly padded with zeros if not enough
             boxes to fill the array.
    - Feature maps: List of feature maps from different levels of the pyramid.
                    Each is [batch, height, width, channels]
```

```
    Output:
    Pooled regions in the shape: [batch, num_boxes, height, width, channels].
    The width and height are those specific in the pool_shape in the layer
    constructor.
    """
    def __init__(self, pool_shape, image_shape, **kwargs):
        super(PyramidROIAlign, self).__init__(**kwargs)
        self.pool_shape = tuple(pool_shape)
        self.image_shape = tuple(image_shape)
    def call(self, inputs):
        # Crop boxes [batch, num_boxes, (y1, x1, y2, x2)] in normalized coords
        boxes = inputs[0]
        # Feature Maps. List of feature maps from different level of the
        # feature pyramid. Each is [batch, height, width, channels]
        feature_maps = inputs[1:]
        # Assign each ROI to a level in the pyramid based on the ROI area.
        y1, x1, y2, x2 = tf.split(boxes, 4, axis =2)
        h = y2 - y1
        w = x2 - x1
        # Equation 1 in the Feature Pyramid Networks paper. Account for
        # the fact that our coordinates are normalized here.
        # e.g. a 224x224 ROI (in pixels) maps to P4
        image_area = tf.cast(
            self.image_shape[0] *self.image_shape[1], tf.float32)
        roi_level = log2_graph(tf.sqrt(h * w) / (224.0 / tf.sqrt(image_area)))
        roi_level = tf.minimum(5, tf.maximum(
            2, 4 + tf.cast(tf.round(roi_level), tf.int32)))
        roi_level = tf.squeeze(roi_level, 2)
        # Loop through levels and apply ROI pooling to each. P2 to P5.
        pooled = []
        box_to_level = []
        for i, level in enumerate(range(2, 6)):
            ix = tf.where(tf.equal(roi_level, level))
            level_boxes = tf.gather_nd(boxes, ix)
            # Box indicies for crop_and_resize.
            box_indices = tf.cast(ix[:, 0], tf.int32)
            # Keep track of which box is mapped to which level
            box_to_level.append(ix)
            # Stop gradient propogation to ROI proposals
            level_boxes = tf.stop_gradient(level_boxes)
            box_indices = tf.stop_gradient(box_indices)
            # Crop and Resize
            # From Mask R-CNN paper: "We sample four regular locations, so
            # that we can evaluate either max or average pooling. In fact,
            # interpolating only a single value at each bin center (without
            # pooling) is nearly as effective."
            #
            # Here we use the simplified approach of a single value per bin,
            # which is how it's done in tf.crop_and_resize()
            # Result: [batch * num_boxes, pool_height, pool_width, channels]
            pooled.append(tf.image.crop_and_resize(
                feature_maps[i], level_boxes, box_indices, self.pool_shape,
                method="bilinear"))
        # Pack pooled features into one tensor
        pooled = tf.concat(pooled, axis =0)
        # Pack box_to_level mapping into one array and add another
        # column representing the order of pooled boxes
        box_to_level = tf.concat(box_to_level, axis =0)
        box_range = tf.expand_dims(tf.range(tf.shape(box_to_level)[0]), 1)
        box_to_level = tf.concat([tf.cast(box_to_level, tf.int32), box_range],
                                 axis=1)
        # Rearrange pooled features to match the order of the original boxes
        # Sort box_to_level by batch then box index
        # TF doesn't have a way to sort by two columns, so merge them and sort.
        sorting_tensor = box_to_level[:, 0] *100000 + box_to_level[:, 1]
        ix = tf.nn.top_k(sorting_tensor, k =tf.shape(
            box_to_level)[0]).indices[::-1]
        ix = tf.gather(box_to_level[:, 2], ix)
        pooled = tf.gather(pooled, ix)
        # Re-add the batch dimension
        pooled = tf.expand_dims(pooled, 0)
        return pooled
    def compute_output_shape(self, input_shape):
        return input_shape[0][:2] +self.pool_shape + (input_shape[1][-1], )
############################################################
#  Detection Target Layer
############################################################
def overlaps_graph(boxes1, boxes2):
    """Computes IoU overlaps between two sets of boxes.
    boxes1, boxes2: [N, (y1, x1, y2, x2)].
    """
    # 1. Tile boxes2 and repeate boxes1. This allows us to compare
    # every boxes1 against every boxes2 without loops.
```

```
    # TF doesn't have an equivalent to np.repeate() so simulate it
    # using tf.tile() and tf.reshape.
    b1 = tf.reshape(tf.tile(tf.expand_dims(boxes1, 1),
                            [1, 1, tf.shape(boxes2)[0]]), [-1, 4])
    b2 = tf.tile(boxes2, [tf.shape(boxes1)[0], 1])
    # 2. Compute intersections
    b1_y1, b1_x1, b1_y2, b1_x2 = tf.split(b1, 4, axis =1)
    b2_y1, b2_x1, b2_y2, b2_x2 = tf.split(b2, 4, axis =1)
    y1 = tf.maximum(b1_y1, b2_y1)
    x1 = tf.maximum(b1_x1, b2_x1)
    y2 = tf.minimum(b1_y2, b2_y2)
    x2 = tf.minimum(b1_x2, b2_x2)
    intersection = tf.maximum(x2 - x1, 0) * tf.maximum(y2 - y1, 0)
    # 3. Compute unions
    b1_area = (b1_y2 - b1_y1) * (b1_x2 - b1_x1)
    b2_area = (b2_y2 - b2_y1) * (b2_x2 - b2_x1)
    union = b1_area + b2_area - intersection
    # 4. Compute IoU and reshape to [boxes1, boxes2]
    iou = intersection / union
    overlaps = tf.reshape(iou, [tf.shape(boxes1)[0], tf.shape(boxes2)[0]])
    return overlaps
def detection_targets_graph(proposals, gt_class_ids, gt_boxes, gt_masks, config):
    """Generates detection targets for one image. Subsamples proposals and
    generates target class IDs, bounding box deltas, and masks for each.
    Inputs:
    proposals: [N, (y1, x1, y2, x2)] in normalized coordinates. Might
               be zero padded if there are not enough proposals.
    gt_class_ids: [MAX_GT_INSTANCES] int class IDs
    gt_boxes: [MAX_GT_INSTANCES, (y1, x1, y2, x2)] in normalized coordinates.
    gt_masks: [height, width, MAX_GT_INSTANCES] of boolean type.
    Returns: Target ROIs and corresponding class IDs, bounding box shifts,
    and masks.
    rois: [TRAIN_ROIS_PER_IMAGE, (y1, x1, y2, x2)] in normalized coordinates
    class_ids: [TRAIN_ROIS_PER_IMAGE]. Integer class IDs. Zero padded.
    deltas: [TRAIN_ROIS_PER_IMAGE, NUM_CLASSES, (dy, dx, log(dh), log(dw))]
            Class-specific bbox refinements.
    masks: [TRAIN_ROIS_PER_IMAGE, height, width). Masks cropped to bbox
           boundaries and resized to neural network output size.
    Note: Returned arrays might be zero padded if not enough target ROIs.
    """
    # Assertions
    asserts = [
        tf.Assert(tf.greater(tf.shape(proposals)[0], 0), [proposals],
                  name="roi_assertion"),
    ]
    with tf.control_dependencies(asserts):
        proposals = tf.identity(proposals)
    # Remove zero padding
    proposals, _ = trim_zeros_graph(proposals, name ="trim_proposals")
    gt_boxes, non_zeros = trim_zeros_graph(gt_boxes, name ="trim_gt_boxes")
    gt_class_ids = tf.boolean_mask(gt_class_ids, non_zeros,
                                   name="trim_gt_class_ids")
    gt_masks = tf.gather(gt_masks, tf.where(non_zeros)[:, 0], axis =2,
                         name="trim_gt_masks")
    # Handle COCO crowds
    # A crowd box in COCO is a bounding box around several instances. Exclude
    # them from training. A crowd box is given a negative class ID.
    crowd_ix = tf.where(gt_class_ids <0)[:, 0]
    non_crowd_ix = tf.where(gt_class_ids >0)[:, 0]
    crowd_boxes = tf.gather(gt_boxes, crowd_ix)
    crowd_masks = tf.gather(gt_masks, crowd_ix, axis =2)
    gt_class_ids = tf.gather(gt_class_ids, non_crowd_ix)
    gt_boxes = tf.gather(gt_boxes, non_crowd_ix)
    gt_masks = tf.gather(gt_masks, non_crowd_ix, axis =2)
    # Compute overlaps matrix [proposals, gt_boxes]
    overlaps = overlaps_graph(proposals, gt_boxes)
    # Compute overlaps with crowd boxes [anchors, crowds]
    crowd_overlaps = overlaps_graph(proposals, crowd_boxes)
    crowd_iou_max = tf.reduce_max(crowd_overlaps, axis =1)
    no_crowd_bool = (crowd_iou_max <0.001)
    # Determine postive and negative ROIs
    roi_iou_max = tf.reduce_max(overlaps, axis =1)
    # 1. Positive ROIs are those with >= 0.5 IoU with a GT box
    positive_roi_bool = (roi_iou_max >=0.5)
    positive_indices = tf.where(positive_roi_bool)[:, 0]
    # 2. Negative ROIs are those with < 0.5 with every GT box. Skip crowds.
    negative_indices = tf.where(tf.logical_and(roi_iou_max <0.5, no_crowd_bool))[:, 0]
    # Subsample ROIs. Aim for 33% positive
    # Positive ROIs
    positive_count =int(config.TRAIN_ROIS_PER_IMAGE *
                        config.ROI_POSITIVE_RATIO)
    positive_indices = tf.random_shuffle(positive_indices)[:positive_count]
    positive_count = tf.shape(positive_indices)[0]
```

```
    # Negative ROIs. Add enough to maintain positive:negative ratio.
    r =1.0 / config.ROI_POSITIVE_RATIO
    negative_count = tf.cast(r * tf.cast(positive_count, tf.float32), tf.int32) -
positive_count
    negative_indices = tf.random_shuffle(negative_indices)[:negative_count]
    # Gather selected ROIs
    positive_rois = tf.gather(proposals, positive_indices)
    negative_rois = tf.gather(proposals, negative_indices)
    # Assign positive ROIs to GT boxes.
    positive_overlaps = tf.gather(overlaps, positive_indices)
    roi_gt_box_assignment = tf.argmax(positive_overlaps, axis =1)
    roi_gt_boxes = tf.gather(gt_boxes, roi_gt_box_assignment)
    roi_gt_class_ids = tf.gather(gt_class_ids, roi_gt_box_assignment)
    # Compute bbox refinement for positive ROIs
    deltas = utils.box_refinement_graph(positive_rois, roi_gt_boxes)
    deltas /= config.BBOX_STD_DEV
    # Assign positive ROIs to GT masks
    # Permute masks to [N, height, width, 1]
    transposed_masks = tf.expand_dims(tf.transpose(gt_masks, [2, 0, 1]), -1)
    # Pick the right mask for each ROI
    roi_masks = tf.gather(transposed_masks, roi_gt_box_assignment)
    # Compute mask targets
    boxes = positive_rois
    if config.USE_MINI_MASK:
        # Transform ROI corrdinates from normalized image space
        # to normalized mini-mask space.
        y1, x1, y2, x2 = tf.split(positive_rois, 4, axis =1)
        gt_y1, gt_x1, gt_y2, gt_x2 = tf.split(roi_gt_boxes, 4, axis =1)
        gt_h = gt_y2 - gt_y1
        gt_w = gt_x2 - gt_x1
        y1 = (y1 - gt_y1) / gt_h
        x1 = (x1 - gt_x1) / gt_w
        y2 = (y2 - gt_y1) / gt_h
        x2 = (x2 - gt_x1) / gt_w
        boxes = tf.concat([y1, x1, y2, x2], 1)
    box_ids = tf.range(0, tf.shape(roi_masks)[0])
    masks = tf.image.crop_and_resize(tf.cast(roi_masks, tf.float32), boxes,
                                     box_ids,
                                     config.MASK_SHAPE)
    # Remove the extra dimension from masks.
    masks = tf.squeeze(masks, axis =3)
    # Threshold mask pixels at 0.5 to have GT masks be 0 or 1 to use with
    # binary cross entropy loss.
    masks = tf.round(masks)
    # Append negative ROIs and pad bbox deltas and masks that
    # are not used for negative ROIs with zeros.
    rois = tf.concat([positive_rois, negative_rois], axis =0)
    N = tf.shape(negative_rois)[0]
    P = tf.maximum(config.TRAIN_ROIS_PER_IMAGE - tf.shape(rois)[0], 0)
    rois = tf.pad(rois, [(0, P), (0, 0)])
    roi_gt_boxes = tf.pad(roi_gt_boxes, [(0, N + P), (0, 0)])
    roi_gt_class_ids = tf.pad(roi_gt_class_ids, [(0, N + P)])
    deltas = tf.pad(deltas, [(0, N + P), (0, 0)])
    masks = tf.pad(masks, [[0, N + P], (0, 0), (0, 0)])
    return rois, roi_gt_class_ids, deltas, masks
class DetectionTargetLayer(KE.Layer):
    """Subsamples proposals and generates target box refinement, class_ids,
    and masks for each.
    Inputs:
    proposals: [batch, N, (y1, x1, y2, x2)] in normalized coordinates. Might
               be zero padded if there are not enough proposals.
    gt_class_ids: [batch, MAX_GT_INSTANCES] Integer class IDs.
    gt_boxes: [batch, MAX_GT_INSTANCES, (y1, x1, y2, x2)] in normalized
              coordinates.
    gt_masks: [batch, height, width, MAX_GT_INSTANCES] of boolean type
    Returns: Target ROIs and corresponding class IDs, bounding box shifts,
    and masks.
    rois: [batch, TRAIN_ROIS_PER_IMAGE, (y1, x1, y2, x2)] in normalized
          coordinates
    target_class_ids: [batch, TRAIN_ROIS_PER_IMAGE]. Integer class IDs.
    target_deltas: [batch, TRAIN_ROIS_PER_IMAGE, NUM_CLASSES,
                    (dy, dx, log(dh), log(dw), class_id)]
                   Class-specific bbox refinements.
    target_mask: [batch, TRAIN_ROIS_PER_IMAGE, height, width)
                 Masks cropped to bbox boundaries and resized to neural
                 network output size.
    Note: Returned arrays might be zero padded if not enough target ROIs.
    """
    def __init__(self, config, **kwargs):
        super(DetectionTargetLayer, self).__init__(**kwargs)
        self.config = config
    def call(self, inputs):
        proposals = inputs[0]
```

```
        gt_class_ids = inputs[1]
        gt_boxes = inputs[2]
        gt_masks = inputs[3]
        # Slice the batch and run a graph for each slice
        # TODO: Rename target_bbox to target_deltas for clarity
        names = ["rois", "target_class_ids", "target_bbox", "target_mask"]
        outputs = utils.batch_slice(
            [proposals, gt_class_ids, gt_boxes, gt_masks],
            lambda w, x, y, z: detection_targets_graph(
                w, x, y, z, self.config),
            self.config.IMAGES_PER_GPU, names =names)
        return outputs
    def compute_output_shape(self, input_shape):
        return [
            (None, self.config.TRAIN_ROIS_PER_IMAGE, 4),  # rois
            (None, 1),  # class_ids
            (None, self.config.TRAIN_ROIS_PER_IMAGE, 4),  # deltas
            (None, self.config.TRAIN_ROIS_PER_IMAGE, self.config.MASK_SHAPE[0],
             self.config.MASK_SHAPE[1])  # masks
        ]
    def compute_mask(self, inputs, mask =None):
        return [None, None, None, None]
############################################################
#  Detection Layer
############################################################
def clip_to_window(window, boxes):
    """
    window: (y1, x1, y2, x2). The window in the image we want to clip to.
    boxes: [N, (y1, x1, y2, x2)]
    """
    boxes[:, 0] = np.maximum(np.minimum(boxes[:, 0], window[2]), window[0])
    boxes[:, 1] = np.maximum(np.minimum(boxes[:, 1], window[3]), window[1])
    boxes[:, 2] = np.maximum(np.minimum(boxes[:, 2], window[2]), window[0])
    boxes[:, 3] = np.maximum(np.minimum(boxes[:, 3], window[3]), window[1])
    return boxes
def refine_detections_graph(rois, probs, deltas, window, config):
    """Refine classified proposals and filter overlaps and return final
    detections.
    Inputs:
        rois: [N, (y1, x1, y2, x2)] in normalized coordinates
        probs: [N, num_classes]. Class probabilities.
        deltas: [N, num_classes, (dy, dx, log(dh), log(dw))]. Class-specific
                bounding box deltas.
        window: (y1, x1, y2, x2) in image coordinates. The part of the image
            that contains the image excluding the padding.
    Returns detections shaped: [N, (y1, x1, y2, x2, class_id, score)] where
        coordinates are in image domain.
    """
    # Class IDs per ROI
    class_ids = tf.argmax(probs, axis =1, output_type =tf.int32)
    # Class probability of the top class of each ROI
    indices = tf.stack([tf.range(probs.shape[0]), class_ids], axis =1)
    class_scores = tf.gather_nd(probs, indices)
    # Class-specific bounding box deltas
    deltas_specific = tf.gather_nd(deltas, indices)
    # Apply bounding box deltas
    # Shape: [boxes, (y1, x1, y2, x2)] in normalized coordinates
    refined_rois = apply_box_deltas_graph(
        rois, deltas_specific * config.BBOX_STD_DEV)
    # Convert coordiates to image domain
    # TODO: better to keep them normalized until later
    height, width = config.IMAGE_SHAPE[:2]
    refined_rois *= tf.constant([height, width, height, width], dtype =tf.float32)
    # Clip boxes to image window
    refined_rois = clip_boxes_graph(refined_rois, window)
    # Round and cast to int since we're deadling with pixels now
    refined_rois = tf.to_int32(tf.rint(refined_rois))
    # TODO: Filter out boxes with zero area
    # Filter out background boxes
    keep = tf.where(class_ids >0)[:, 0]
    # Filter out low confidence boxes
    if config.DETECTION_MIN_CONFIDENCE:
        conf_keep = tf.where(class_scores >= config.DETECTION_MIN_CONFIDENCE)[:, 0]
        keep = tf.sets.set_intersection(tf.expand_dims(keep, 0),
                                        tf.expand_dims(conf_keep, 0))
        keep = tf.sparse_tensor_to_dense(keep)[0]
    # Apply per-class NMS
    # 1. Prepare variables
    pre_nms_class_ids = tf.gather(class_ids, keep)
    pre_nms_scores = tf.gather(class_scores, keep)
    pre_nms_rois = tf.gather(refined_rois,   keep)
    unique_pre_nms_class_ids = tf.unique(pre_nms_class_ids)[0]
    def nms_keep_map(class_id):
```

```
        """Apply Non-Maximum Suppression on ROIs of the given class."""
        # Indices of ROIs of the given class
        ixs = tf.where(tf.equal(pre_nms_class_ids, class_id))[:, 0]
        # Apply NMS
        class_keep = tf.image.non_max_suppression(
               tf.to_float(tf.gather(pre_nms_rois, ixs)),
               tf.gather(pre_nms_scores, ixs),
               max_output_size=config.DETECTION_MAX_INSTANCES,
               iou_threshold=config.DETECTION_NMS_THRESHOLD)
        # Map indicies
        class_keep = tf.gather(keep, tf.gather(ixs, class_keep))
        # Pad with -1 so returned tensors have the same shape
        gap = config.DETECTION_MAX_INSTANCES - tf.shape(class_keep)[0]
        class_keep = tf.pad(class_keep, [(0, gap)],
                            mode='CONSTANT', constant_values =-1)
        # Set shape so map_fn() can infer result shape
        class_keep.set_shape([config.DETECTION_MAX_INSTANCES])
        return class_keep
    # 2. Map over class IDs
    nms_keep = tf.map_fn(nms_keep_map, unique_pre_nms_class_ids,
                         dtype=tf.int64)
    # 3. Merge results into one list, and remove -1 padding
    nms_keep = tf.reshape(nms_keep, [-1])
    nms_keep = tf.gather(nms_keep, tf.where(nms_keep >-1)[:, 0])
    # 4. Compute intersection between keep and nms_keep
    keep = tf.sets.set_intersection(tf.expand_dims(keep, 0),
                                    tf.expand_dims(nms_keep, 0))
    keep = tf.sparse_tensor_to_dense(keep)[0]
    # Keep top detections
    roi_count = config.DETECTION_MAX_INSTANCES
    class_scores_keep = tf.gather(class_scores, keep)
    num_keep = tf.minimum(tf.shape(class_scores_keep)[0], roi_count)
    top_ids = tf.nn.top_k(class_scores_keep, k =num_keep, sorted =True)[1]
    keep = tf.gather(keep, top_ids)
    # Arrange output as [N, (y1, x1, y2, x2, class_id, score)]
    # Coordinates are in image domain.
    detections = tf.concat([
        tf.to_float(tf.gather(refined_rois, keep)),
        tf.to_float(tf.gather(class_ids, keep))[..., tf.newaxis],
        tf.gather(class_scores, keep)[..., tf.newaxis]
        ], axis=1)
    # Pad with zeros if detections < DETECTION_MAX_INSTANCES
    gap = config.DETECTION_MAX_INSTANCES - tf.shape(detections)[0]
    detections = tf.pad(detections, [(0, gap), (0, 0)], "CONSTANT")
    return detections
class DetectionLayer(KE.Layer):
    """Takes classified proposal boxes and their bounding box deltas and
    returns the final detection boxes.
    Returns:
    [batch, num_detections, (y1, x1, y2, x2, class_id, class_score)] where
    coordinates are in image domain
    """
    def __init__(self, config =None, **kwargs):
        super(DetectionLayer, self).__init__(**kwargs)
        self.config = config
    def call(self, inputs):
        rois = inputs[0]
        mrcnn_class = inputs[1]
        mrcnn_bbox = inputs[2]
        image_meta = inputs[3]
        # Run detection refinement graph on each item in the batch
        _, _, window, _ = parse_image_meta_graph(image_meta)
        detections_batch = utils.batch_slice(
            [rois, mrcnn_class, mrcnn_bbox, window],
            lambda x, y, w, z: refine_detections_graph(x, y, w, z, self.config),
            self.config.IMAGES_PER_GPU)
        # Reshape output
        # [batch, num_detections, (y1, x1, y2, x2, class_score)] in pixels
        return tf.reshape(
            detections_batch,
            [self.config.BATCH_SIZE, self.config.DETECTION_MAX_INSTANCES, 6])
    def compute_output_shape(self, input_shape):
        return (None, self.config.DETECTION_MAX_INSTANCES, 6)
# Region Proposal Network (RPN)
def rpn_graph(feature_map, anchors_per_location, anchor_stride):
    """Builds the computation graph of Region Proposal Network.
    feature_map: backbone features [batch, height, width, depth]
    anchors_per_location: number of anchors per pixel in the feature map
    anchor_stride: Controls the density of anchors. Typically 1 (anchors for
                   every pixel in the feature map), or 2 (every other pixel).
    Returns:
        rpn_logits: [batch, H, W, 2] Anchor classifier logits (before softmax)
        rpn_probs: [batch, H, W, 2] Anchor classifier probabilities.
```

```
719         rpn_bbox: [batch, H, W, (dy, dx, log(dh), log(dw))] Deltas to be
720                   applied to anchors.
721     """
722     # TODO: check if stride of 2 causes alignment issues if the featuremap
723     #       is not even.
724     # Shared convolutional base of the RPN
725     shared = KL.Conv2D(512, (3, 3), padding ='same', activation ='relu',
726                        strides=anchor_stride,
727                        name='rpn_conv_shared')(feature_map)
728     # Anchor Score. [batch, height, width, anchors per location * 2].
729     x = KL.Conv2D(2 * anchors_per_location, (1, 1), padding ='valid',
730                   activation='linear', name ='rpn_class_raw')(shared)
731     # Reshape to [batch, anchors, 2]
732     rpn_class_logits = KL.Lambda(
733         lambda t: tf.reshape(t, [tf.shape(t)[0], -1, 2]))(x)
734     # Softmax on last dimension of BG/FG.
735     rpn_probs = KL.Activation(
736         "softmax", name ="rpn_class_xxx")(rpn_class_logits)
737     # Bounding box refinement. [batch, H, W, anchors per location, depth]
738     # where depth is [x, y, log(w), log(h)]
739     x = KL.Conv2D(anchors_per_location *4, (1, 1), padding ="valid",
740                   activation='linear', name ='rpn_bbox_pred')(shared)
741     # Reshape to [batch, anchors, 4]
742     rpn_bbox = KL.Lambda(lambda t: tf.reshape(t, [tf.shape(t)[0], -1, 4]))(x)
743     return [rpn_class_logits, rpn_probs, rpn_bbox]
744 def build_rpn_model(anchor_stride, anchors_per_location, depth):
745     """Builds a Keras model of the Region Proposal Network.
746     It wraps the RPN graph so it can be used multiple times with shared
747     weights.
748     anchors_per_location: number of anchors per pixel in the feature map
749     anchor_stride: Controls the density of anchors. Typically 1 (anchors for
750                    every pixel in the feature map), or 2 (every other pixel).
751     depth: Depth of the backbone feature map.
752     Returns a Keras Model object. The model outputs, when called, are:
753     rpn_logits: [batch, H, W, 2] Anchor classifier logits (before softmax)
754     rpn_probs: [batch, W, W, 2] Anchor classifier probabilities.
755     rpn_bbox: [batch, H, W, (dy, dx, log(dh), log(dw))] Deltas to be
756               applied to anchors.
757     """
758     input_feature_map = KL.Input(shape =[None, None, depth],
759                                  name="input_rpn_feature_map")
760     outputs = rpn_graph(input_feature_map, anchors_per_location, anchor_stride)
761     return KM.Model([input_feature_map], outputs, name ="rpn_model")
762 ############################################################
763 #  Feature Pyramid Network Heads
764 ############################################################
765 def fpn_classifier_graph(rois, feature_maps,
766                          image_shape, pool_size, num_classes):
767     """Builds the computation graph of the feature pyramid network classifier
768     and regressor heads.
769     rois: [batch, num_rois, (y1, x1, y2, x2)] Proposal boxes in normalized
770           coordinates.
771     feature_maps: List of feature maps from diffent layers of the pyramid,
772                   [P2, P3, P4, P5]. Each has a different resolution.
773     image_shape: [height, width, depth]
774     pool_size: The width of the square feature map generated from ROI Pooling.
775     num_classes: number of classes, which determines the depth of the results
776     Returns:
777         logits: [N, NUM_CLASSES] classifier logits (before softmax)
778         probs: [N, NUM_CLASSES] classifier probabilities
779         bbox_deltas: [N, (dy, dx, log(dh), log(dw))] Deltas to apply to
780                      proposal boxes
781     """
782     # ROI Pooling
783     # Shape: [batch, num_boxes, pool_height, pool_width, channels]
784     x = PyramidROIAlign([pool_size, pool_size], image_shape,
785                         name="roi_align_classifier")([rois] + feature_maps)
786     # Two 1024 FC layers (implemented with Conv2D for consistency)
787     x = KL.TimeDistributed(KL.Conv2D(1024, (pool_size, pool_size), padding ="valid"),
788                            name="mrcnn_class_conv1")(x)
789     x = KL.TimeDistributed(BatchNorm(axis =3), name ='mrcnn_class_bn1')(x)
790     x = KL.Activation('relu')(x)
791     x = KL.TimeDistributed(KL.Conv2D(1024, (1, 1)),
792                            name="mrcnn_class_conv2")(x)
793     x = KL.TimeDistributed(BatchNorm(axis =3),
794                            name='mrcnn_class_bn2')(x)
795     x = KL.Activation('relu')(x)
796     shared = KL.Lambda(lambda x: K.squeeze(K.squeeze(x, 3), 2),
797                        name="pool_squeeze")(x)
798     # Classifier head
799     mrcnn_class_logits = KL.TimeDistributed(KL.Dense(num_classes),
800                                             name='mrcnn_class_logits')(shared)
801     mrcnn_probs = KL.TimeDistributed(KL.Activation("softmax"),
```

```
                                    name="mrcnn_class")(mrcnn_class_logits)
    # BBox head
    # [batch, boxes, num_classes * (dy, dx, log(dh), log(dw))]
    x = KL.TimeDistributed(KL.Dense(num_classes *4, activation ='linear'),
                           name='mrcnn_bbox_fc')(shared)
    # Reshape to [batch, boxes, num_classes, (dy, dx, log(dh), log(dw))]
    s = K.int_shape(x)
    mrcnn_bbox = KL.Reshape((s[1], num_classes, 4), name ="mrcnn_bbox")(x)
    return mrcnn_class_logits, mrcnn_probs, mrcnn_bbox
def build_fpn_mask_graph(rois, feature_maps,
                         image_shape, pool_size, num_classes):
    """Builds the computation graph of the mask head of Feature Pyramid Network.
    rois: [batch, num_rois, (y1, x1, y2, x2)] Proposal boxes in normalized
          coordinates.
    feature_maps: List of feature maps from diffent layers of the pyramid,
                  [P2, P3, P4, P5]. Each has a different resolution.
    image_shape: [height, width, depth]
    pool_size: The width of the square feature map generated from ROI Pooling.
    num_classes: number of classes, which determines the depth of the results
    Returns: Masks [batch, roi_count, height, width, num_classes]
    """
    # ROI Pooling
    # Shape: [batch, boxes, pool_height, pool_width, channels]
    x = PyramidROIAlign([pool_size, pool_size], image_shape,
                        name="roi_align_mask")([rois] + feature_maps)
    # Conv layers
    x = KL.TimeDistributed(KL.Conv2D(256, (3, 3), padding ="same"),
                           name="mrcnn_mask_conv1")(x)
    x = KL.TimeDistributed(BatchNorm(axis =3),
                           name='mrcnn_mask_bn1')(x)
    x = KL.Activation('relu')(x)
    x = KL.TimeDistributed(KL.Conv2D(256, (3, 3), padding ="same"),
                           name="mrcnn_mask_conv2")(x)
    x = KL.TimeDistributed(BatchNorm(axis =3),
                           name='mrcnn_mask_bn2')(x)
    x = KL.Activation('relu')(x)
    x = KL.TimeDistributed(KL.Conv2D(256, (3, 3), padding ="same"),
                           name="mrcnn_mask_conv3")(x)
    x = KL.TimeDistributed(BatchNorm(axis =3),
                           name='mrcnn_mask_bn3')(x)
    x = KL.Activation('relu')(x)
    x = KL.TimeDistributed(KL.Conv2D(256, (3, 3), padding ="same"),
                           name="mrcnn_mask_conv4")(x)
    x = KL.TimeDistributed(BatchNorm(axis =3),
                           name='mrcnn_mask_bn4')(x)
    x = KL.Activation('relu')(x)
    x  =  KL.TimeDistributed(KL.Conv2DTranspose(256,  (2,  2),  strides  =2,  activation
="relu"),
                           name="mrcnn_mask_deconv")(x)
    x  =  KL.TimeDistributed(KL.Conv2D(num_classes,  (1,  1),  strides  =1,  activation
="sigmoid"),
                           name="mrcnn_mask")(x)
    return x
############################################################
#  Loss Functions
############################################################
def smooth_l1_loss(y_true, y_pred):
    """Implements Smooth-L1 loss.
    y_true and y_pred are typicallly: [N, 4], but could be any shape.
    """
    diff = K.abs(y_true - y_pred)
    less_than_one = K.cast(K.less(diff, 1.0), "float32")
    loss = (less_than_one *0.5 * diff **2) + (1 - less_than_one) * (diff -0.5)
    return loss
def rpn_class_loss_graph(rpn_match, rpn_class_logits):
    """RPN anchor classifier loss.
    rpn_match: [batch, anchors, 1]. Anchor match type. 1=positive,
               -1=negative, 0=neutral anchor.
    rpn_class_logits: [batch, anchors, 2]. RPN classifier logits for FG/BG.
    """
    # Squeeze last dim to simplify
    rpn_match = tf.squeeze(rpn_match, -1)
    # Get anchor classes. Convert the -1/+1 match to 0/1 values.
    anchor_class = K.cast(K.equal(rpn_match, 1), tf.int32)
    # Positive and Negative anchors contribute to the loss,
    # but neutral anchors (match value = 0) don't.
    indices = tf.where(K.not_equal(rpn_match, 0))
    # Pick rows that contribute to the loss and filter out the rest.
    rpn_class_logits = tf.gather_nd(rpn_class_logits, indices)
    anchor_class = tf.gather_nd(anchor_class, indices)
    # Crossentropy loss
    loss = K.sparse_categorical_crossentropy(target =anchor_class,
                                             output=rpn_class_logits,
```

```
                                        from_logits=True)
    loss = K.switch(tf.size(loss) >0, K.mean(loss), tf.constant(0.0))
    return loss
def rpn_bbox_loss_graph(config, target_bbox, rpn_match, rpn_bbox):
    """Return the RPN bounding box loss graph.
    config: the model config object.
    target_bbox: [batch, max positive anchors, (dy, dx, log(dh), log(dw))].
        Uses 0 padding to fill in unsed bbox deltas.
    rpn_match: [batch, anchors, 1]. Anchor match type. 1=positive,
               -1=negative, 0=neutral anchor.
    rpn_bbox: [batch, anchors, (dy, dx, log(dh), log(dw))]
    """
    # Positive anchors contribute to the loss, but negative and
    # neutral anchors (match value of 0 or -1) don't.
    rpn_match = K.squeeze(rpn_match, -1)
    indices = tf.where(K.equal(rpn_match, 1))
    # Pick bbox deltas that contribute to the loss
    rpn_bbox = tf.gather_nd(rpn_bbox, indices)
    # Trim target bounding box deltas to the same length as rpn_bbox.
    batch_counts = K.sum(K.cast(K.equal(rpn_match, 1), tf.int32), axis =1)
    target_bbox = batch_pack_graph(target_bbox, batch_counts,
                                   config.IMAGES_PER_GPU)
    # TODO: use smooth_l1_loss() rather than reimplementing here
    #       to reduce code duplication
    diff = K.abs(target_bbox - rpn_bbox)
    less_than_one = K.cast(K.less(diff, 1.0), "float32")
    loss = (less_than_one *0.5 * diff **2) + (1 - less_than_one) * (diff -0.5)
    loss = K.switch(tf.size(loss) >0, K.mean(loss), tf.constant(0.0))
    return loss
def mrcnn_class_loss_graph(target_class_ids, pred_class_logits,
                           active_class_ids):
    """Loss for the classifier head of Mask RCNN.
    target_class_ids: [batch, num_rois]. Integer class IDs. Uses zero
        padding to fill in the array.
    pred_class_logits: [batch, num_rois, num_classes]
    active_class_ids: [batch, num_classes]. Has a value of 1 for
        classes that are in the dataset of the image, and 0
        for classes that are not in the dataset.
    """
    target_class_ids = tf.cast(target_class_ids, 'int64')
    # Find predictions of classes that are not in the dataset.
    pred_class_ids = tf.argmax(pred_class_logits, axis =2)
    # TODO: Update this line to work with batch > 1. Right now it assumes all
    #       images in a batch have the same active_class_ids
    pred_active = tf.gather(active_class_ids[0], pred_class_ids)
    # Loss
    loss = tf.nn.sparse_softmax_cross_entropy_with_logits(
        labels=target_class_ids, logits =pred_class_logits)
    # Erase losses of predictions of classes that are not in the active
    # classes of the image.
    loss = loss * pred_active
    # Computer loss mean. Use only predictions that contribute
    # to the loss to get a correct mean.
    loss = tf.reduce_sum(loss) / tf.reduce_sum(pred_active)
    return loss
def mrcnn_bbox_loss_graph(target_bbox, target_class_ids, pred_bbox):
    """Loss for Mask R-CNN bounding box refinement.
    target_bbox: [batch, num_rois, (dy, dx, log(dh), log(dw))]
    target_class_ids: [batch, num_rois]. Integer class IDs.
    pred_bbox: [batch, num_rois, num_classes, (dy, dx, log(dh), log(dw))]
    """
    # Reshape to merge batch and roi dimensions for simplicity.
    target_class_ids = K.reshape(target_class_ids, (-1,))
    target_bbox = K.reshape(target_bbox, (-1, 4))
    pred_bbox = K.reshape(pred_bbox, (-1, K.int_shape(pred_bbox)[2], 4))
    # Only positive ROIs contribute to the loss. And only
    # the right class_id of each ROI. Get their indicies.
    positive_roi_ix = tf.where(target_class_ids >0)[:, 0]
    positive_roi_class_ids = tf.cast(
        tf.gather(target_class_ids, positive_roi_ix), tf.int64)
    indices = tf.stack([positive_roi_ix, positive_roi_class_ids], axis =1)
    # Gather the deltas (predicted and true) that contribute to loss
    target_bbox = tf.gather(target_bbox, positive_roi_ix)
    pred_bbox = tf.gather_nd(pred_bbox, indices)
    # Smooth-L1 Loss
    loss = K.switch(tf.size(target_bbox) >0,
                    smooth_l1_loss(y_true=target_bbox, y_pred =pred_bbox),
                    tf.constant(0.0))
    loss = K.mean(loss)
    loss = K.reshape(loss, [1, 1])
    return loss
def mrcnn_mask_loss_graph(target_masks, target_class_ids, pred_masks):
    """Mask binary cross-entropy loss for the masks head.
```

```
    target_masks: [batch, num_rois, height, width].
        A float32 tensor of values 0 or 1. Uses zero padding to fill array.
    target_class_ids: [batch, num_rois]. Integer class IDs. Zero padded.
    pred_masks: [batch, proposals, height, width, num_classes] float32 tensor
                with values from 0 to 1.
    """
    # Reshape for simplicity. Merge first two dimensions into one.
    target_class_ids = K.reshape(target_class_ids, (-1,))
    mask_shape = tf.shape(target_masks)
    target_masks = K.reshape(target_masks, (-1, mask_shape[2], mask_shape[3]))
    pred_shape = tf.shape(pred_masks)
    pred_masks = K.reshape(pred_masks,
                           (-1, pred_shape[2], pred_shape[3], pred_shape[4]))
    # Permute predicted masks to [N, num_classes, height, width]
    pred_masks = tf.transpose(pred_masks, [0, 3, 1, 2])
    # Only positive ROIs contribute to the loss. And only
    # the class specific mask of each ROI.
    positive_ix = tf.where(target_class_ids >0)[:, 0]
    positive_class_ids = tf.cast(
        tf.gather(target_class_ids, positive_ix), tf.int64)
    indices = tf.stack([positive_ix, positive_class_ids], axis =1)
    # Gather the masks (predicted and true) that contribute to loss
    y_true = tf.gather(target_masks, positive_ix)
    y_pred = tf.gather_nd(pred_masks, indices)
    # Compute binary cross entropy. If no positive ROIs, then return 0.
    # shape: [batch, roi, num_classes]
    loss = K.switch(tf.size(y_true) >0,
                    K.binary_crossentropy(target=y_true, output =y_pred),
                    tf.constant(0.0))
    loss = K.mean(loss)
    loss = K.reshape(loss, [1, 1])
    return loss
############################################################
#  Data Generator
############################################################
def load_image_gt(dataset, config, image_id, augment =False,
                  use_mini_mask=False):
    """Load and return ground truth data for an image (image, mask, bounding boxes).
    augment: If true, apply random image augmentation. Currently, only
        horizontal flipping is offered.
    use_mini_mask: If False, returns full-size masks that are the same height
        and width as the original image. These can be big, for example
        1024x1024x100 (for 100 instances). Mini masks are smaller, typically,
        224x224 and are generated by extracting the bounding box of the
        object and resizing it to MINI_MASK_SHAPE.
    Returns:
    image: [height, width, 3]
    shape: the original shape of the image before resizing and cropping.
    class_ids: [instance_count] Integer class IDs
    bbox: [instance_count, (y1, x1, y2, x2)]
    mask: [height, width, instance_count]. The height and width are those
        of the image unless use_mini_mask is True, in which case they are
        defined in MINI_MASK_SHAPE.
    """
    # Load image and mask
    image = dataset.load_image(image_id)
    mask, class_ids = dataset.load_mask(image_id)
    shape = image.shape
    image, window, scale, padding = utils.resize_image(
        image,
        min_dim=config.IMAGE_MIN_DIM,
        max_dim=config.IMAGE_MAX_DIM,
        padding=config.IMAGE_PADDING)
    mask = utils.resize_mask(mask, scale, padding)
    # Random horizontal flips.
    if augment:
        if random.randint(0, 1):
            image = np.fliplr(image)
            mask = np.fliplr(mask)
    # Note that some boxes might be all zeros if the corresponding mask got cropped out.
    # and here is to filter them out
    _idx = np.sum(mask, axis =(0, 1)) >0
    mask = mask[:, :, _idx]
    class_ids = class_ids[_idx]
    # Bounding boxes. Note that some boxes might be all zeros
    # if the corresponding mask got cropped out.
    # bbox: [num_instances, (y1, x1, y2, x2)]
    bbox = utils.extract_bboxes(mask)
    # Active classes
    # Different datasets have different classes, so track the
    # classes supported in the dataset of this image.
    active_class_ids = np.zeros([dataset.num_classes], dtype =np.int32)
    source_class_ids = dataset.source_class_ids[dataset.image_info[image_id]["source"]]
```

```
    active_class_ids[source_class_ids] =1
    # Resize masks to smaller size to reduce memory usage
    if use_mini_mask:
        mask = utils.minimize_mask(bbox, mask, config.MINI_MASK_SHAPE)
    # Image meta data
    image_meta = compose_image_meta(image_id, shape, window, active_class_ids)
    return image, image_meta, class_ids, bbox, mask
def build_detection_targets(rpn_rois, gt_class_ids, gt_boxes, gt_masks, config):
    """Generate targets for training Stage 2 classifier and mask heads.
    This is not used in normal training. It's useful for debugging or to train
    the Mask RCNN heads without using the RPN head.
    Inputs:
    rpn_rois: [N, (y1, x1, y2, x2)] proposal boxes.
    gt_class_ids: [instance count] Integer class IDs
    gt_boxes: [instance count, (y1, x1, y2, x2)]
    gt_masks: [height, width, instance count] Grund truth masks. Can be full
              size or mini-masks.
    Returns:
    rois: [TRAIN_ROIS_PER_IMAGE, (y1, x1, y2, x2)]
    class_ids: [TRAIN_ROIS_PER_IMAGE]. Integer class IDs.
    bboxes: [TRAIN_ROIS_PER_IMAGE, NUM_CLASSES, (y, x, log(h), log(w))]. Class-specific
            bbox refinements.
    masks: [TRAIN_ROIS_PER_IMAGE, height, width, NUM_CLASSES). Class specific masks
cropped
           to bbox boundaries and resized to neural network output size.
    """
    assert rpn_rois.shape[0] >0
    assert gt_class_ids.dtype == np.int32, "Expected int but got {}".format(
        gt_class_ids.dtype)
    assert gt_boxes.dtype == np.int32, "Expected int but got {}".format(
        gt_boxes.dtype)
    assert gt_masks.dtype == np.bool_, "Expected bool but got {}".format(
        gt_masks.dtype)
    # It's common to add GT Boxes to ROIs but we don't do that here because
    # according to XinLei Chen's paper, it doesn't help.
    # Trim empty padding in gt_boxes and gt_masks parts
    instance_ids = np.where(gt_class_ids >0)[0]
    assert instance_ids.shape[0] >0, "Image must contain instances."
    gt_class_ids = gt_class_ids[instance_ids]
    gt_boxes = gt_boxes[instance_ids]
    gt_masks = gt_masks[:, :, instance_ids]
    # Compute areas of ROIs and ground truth boxes.
    rpn_roi_area = (rpn_rois[:, 2] - rpn_rois[:, 0]) * \
        (rpn_rois[:, 3] - rpn_rois[:, 1])
    gt_box_area = (gt_boxes[:, 2] - gt_boxes[:, 0]) * \
        (gt_boxes[:, 3] - gt_boxes[:, 1])
    # Compute overlaps [rpn_rois, gt_boxes]
    overlaps = np.zeros((rpn_rois.shape[0], gt_boxes.shape[0]))
    for i in range(overlaps.shape[1]):
        gt = gt_boxes[i]
        overlaps[:, i] = utils.compute_iou(
            gt, rpn_rois, gt_box_area[i], rpn_roi_area)
    # Assign ROIs to GT boxes
    rpn_roi_iou_argmax = np.argmax(overlaps, axis =1)
    rpn_roi_iou_max = overlaps[np.arange(
        overlaps.shape[0]), rpn_roi_iou_argmax]
    # GT box assigned to each ROI
    rpn_roi_gt_boxes = gt_boxes[rpn_roi_iou_argmax]
    rpn_roi_gt_class_ids = gt_class_ids[rpn_roi_iou_argmax]
    # Positive ROIs are those with >= 0.5 IoU with a GT box.
    fg_ids = np.where(rpn_roi_iou_max >0.5)[0]
    # Negative ROIs are those with max IoU 0.1-0.5 (hard example mining)
    # TODO: To hard example mine or not to hard example mine, that's the question
#     bg_ids = np.where((rpn_roi_iou_max >= 0.1) & (rpn_roi_iou_max < 0.5))[0]
    bg_ids = np.where(rpn_roi_iou_max <0.5)[0]
    # Subsample ROIs. Aim for 33% foreground.
    # FG
    fg_roi_count =int(config.TRAIN_ROIS_PER_IMAGE * config.ROI_POSITIVE_RATIO)
    if fg_ids.shape[0] > fg_roi_count:
        keep_fg_ids = np.random.choice(fg_ids, fg_roi_count, replace =False)
    else:
        keep_fg_ids = fg_ids
    # BG
    remaining = config.TRAIN_ROIS_PER_IMAGE - keep_fg_ids.shape[0]
    if bg_ids.shape[0] > remaining:
        keep_bg_ids = np.random.choice(bg_ids, remaining, replace =False)
    else:
        keep_bg_ids = bg_ids
    # Combine indicies of ROIs to keep
    keep = np.concatenate([keep_fg_ids, keep_bg_ids])
    # Need more?
    remaining = config.TRAIN_ROIS_PER_IMAGE - keep.shape[0]
    if remaining >0:
```

```
        # Looks like we don't have enough samples to maintain the desired
        # balance. Reduce requirements and fill in the rest. This is
        # likely different from the Mask RCNN paper.
        # There is a small chance we have neither fg nor bg samples.
        if keep.shape[0] ==0:
            # Pick bg regions with easier IoU threshold
            bg_ids = np.where(rpn_roi_iou_max <0.5)[0]
            assert bg_ids.shape[0] >= remaining
            keep_bg_ids = np.random.choice(bg_ids, remaining, replace =False)
            assert keep_bg_ids.shape[0] == remaining
            keep = np.concatenate([keep, keep_bg_ids])
        else:
            # Fill the rest with repeated bg rois.
            keep_extra_ids = np.random.choice(
                keep_bg_ids, remaining, replace=True)
            keep = np.concatenate([keep, keep_extra_ids])
    assert keep.shape[0] == config.TRAIN_ROIS_PER_IMAGE, \
        "keep doesn't match ROI batch size {}, {}".format(
            keep.shape[0], config.TRAIN_ROIS_PER_IMAGE)
    # Reset the gt boxes assigned to BG ROIs.
    rpn_roi_gt_boxes[keep_bg_ids, :] =0
    rpn_roi_gt_class_ids[keep_bg_ids] =0
    # For each kept ROI, assign a class_id, and for FG ROIs also add bbox refinement.
    rois = rpn_rois[keep]
    roi_gt_boxes = rpn_roi_gt_boxes[keep]
    roi_gt_class_ids = rpn_roi_gt_class_ids[keep]
    roi_gt_assignment = rpn_roi_iou_argmax[keep]
    # Class-aware bbox deltas. [y, x, log(h), log(w)]
    bboxes = np.zeros((config.TRAIN_ROIS_PER_IMAGE,
                       config.NUM_CLASSES, 4), dtype =np.float32)
    pos_ids = np.where(roi_gt_class_ids >0)[0]
    bboxes[pos_ids, roi_gt_class_ids[pos_ids]] = utils.box_refinement(
        rois[pos_ids], roi_gt_boxes[pos_ids, :4])
    # Normalize bbox refinements
    bboxes /= config.BBOX_STD_DEV
    # Generate class-specific target masks.
    masks       =       np.zeros((config.TRAIN_ROIS_PER_IMAGE,       config.MASK_SHAPE[0],
config.MASK_SHAPE[1], config.NUM_CLASSES),
                     dtype=np.float32)
    for i in pos_ids:
        class_id = roi_gt_class_ids[i]
        assert class_id >0, "class id must be greater than 0"
        gt_id = roi_gt_assignment[i]
        class_mask = gt_masks[:, :, gt_id]
        if config.USE_MINI_MASK:
            # Create a mask placeholder, the size of the image
            placeholder = np.zeros(config.IMAGE_SHAPE[:2], dtype =bool)
            # GT box
            gt_y1, gt_x1, gt_y2, gt_x2 = gt_boxes[gt_id]
            gt_w = gt_x2 - gt_x1
            gt_h = gt_y2 - gt_y1
            # Resize mini mask to size of GT box
            placeholder[gt_y1:gt_y2, gt_x1:gt_x2] = \
                np.round(scipy.misc.imresize(class_mask.astype(float), (gt_h, gt_w),
                                             interp='nearest') /255.0).astype(bool)
            # Place the mini batch in the placeholder
            class_mask = placeholder
        # Pick part of the mask and resize it
        y1, x1, y2, x2 = rois[i].astype(np.int32)
        m = class_mask[y1:y2, x1:x2]
        mask = scipy.misc.imresize(
            m.astype(float), config.MASK_SHAPE, interp='nearest') /255.0
        masks[i, :, :, class_id] = mask
    return rois, roi_gt_class_ids, bboxes, masks
def build_rpn_targets(image_shape, anchors, gt_class_ids, gt_boxes, config):
    """Given the anchors and GT boxes, compute overlaps and identify positive
    anchors and deltas to refine them to match their corresponding GT boxes.
    anchors: [num_anchors, (y1, x1, y2, x2)]
    gt_class_ids: [num_gt_boxes] Integer class IDs.
    gt_boxes: [num_gt_boxes, (y1, x1, y2, x2)]
    Returns:
    rpn_match: [N] (int32) matches between anchors and GT boxes.
               1 = positive anchor, -1 = negative anchor, 0 = neutral
    rpn_bbox: [N, (dy, dx, log(dh), log(dw))] Anchor bbox deltas.
    """
    # RPN Match: 1 = positive anchor, -1 = negative anchor, 0 = neutral
    rpn_match = np.zeros([anchors.shape[0]], dtype =np.int32)
    # RPN bounding boxes: [max anchors per image, (dy, dx, log(dh), log(dw))]
    rpn_bbox = np.zeros((config.RPN_TRAIN_ANCHORS_PER_IMAGE, 4))
    # Handle COCO crowds
    # A crowd box in COCO is a bounding box around several instances. Exclude
    # them from training. A crowd box is given a negative class ID.
    crowd_ix = np.where(gt_class_ids <0)[0]
```

```
    if crowd_ix.shape[0] >0:
        # Filter out crowds from ground truth class IDs and boxes
        non_crowd_ix = np.where(gt_class_ids >0)[0]
        crowd_boxes = gt_boxes[crowd_ix]
        gt_class_ids = gt_class_ids[non_crowd_ix]
        gt_boxes = gt_boxes[non_crowd_ix]
        # Compute overlaps with crowd boxes [anchors, crowds]
        crowd_overlaps = utils.compute_overlaps(anchors, crowd_boxes)
        crowd_iou_max = np.amax(crowd_overlaps, axis =1)
        no_crowd_bool = (crowd_iou_max <0.001)
    else:
        # All anchors don't intersect a crowd
        no_crowd_bool = np.ones([anchors.shape[0]], dtype =bool)
    # Compute overlaps [num_anchors, num_gt_boxes]
    overlaps = utils.compute_overlaps(anchors, gt_boxes)
    # Match anchors to GT Boxes
    # If an anchor overlaps a GT box with IoU >= 0.7 then it's positive.
    # If an anchor overlaps a GT box with IoU < 0.3 then it's negative.
    # Neutral anchors are those that don't match the conditions above,
    # and they don't influence the loss function.
    # However, don't keep any GT box unmatched (rare, but happens). Instead,
    # match it to the closest anchor (even if its max IoU is < 0.3).
    #
    # 1. Set negative anchors first. They get overwritten below if a GT box is
    # matched to them. Skip boxes in crowd areas.
    anchor_iou_argmax = np.argmax(overlaps, axis =1)
    anchor_iou_max = overlaps[np.arange(overlaps.shape[0]), anchor_iou_argmax]
    rpn_match[(anchor_iou_max <0.3) & (no_crowd_bool)] =-1
    # 2. Set an anchor for each GT box (regardless of IoU value).
    # TODO: If multiple anchors have the same IoU match all of them
    gt_iou_argmax = np.argmax(overlaps, axis =0)
    rpn_match[gt_iou_argmax] =1
    # 3. Set anchors with high overlap as positive.
    rpn_match[anchor_iou_max >=0.7] =1
    # Subsample to balance positive and negative anchors
    # Don't let positives be more than half the anchors
    ids = np.where(rpn_match ==1)[0]
    extra =len(ids) - (config.RPN_TRAIN_ANCHORS_PER_IMAGE //2)
    if extra >0:
        # Reset the extra ones to neutral
        ids = np.random.choice(ids, extra, replace =False)
        rpn_match[ids] =0
    # Same for negative proposals
    ids = np.where(rpn_match ==-1)[0]
    extra =len(ids) - (config.RPN_TRAIN_ANCHORS_PER_IMAGE -
                       np.sum(rpn_match ==1))
    if extra >0:
        # Rest the extra ones to neutral
        ids = np.random.choice(ids, extra, replace =False)
        rpn_match[ids] =0
    # For positive anchors, compute shift and scale needed to transform them
    # to match the corresponding GT boxes.
    ids = np.where(rpn_match ==1)[0]
    ix =0  # index into rpn_bbox
    # TODO: use box_refinement() rather than duplicating the code here
    for i, a in zip(ids, anchors[ids]):
        # Closest gt box (it might have IoU < 0.7)
        gt = gt_boxes[anchor_iou_argmax[i]]
        # Convert coordinates to center plus width/height.
        # GT Box
        gt_h = gt[2] - gt[0]
        gt_w = gt[3] - gt[1]
        gt_center_y = gt[0] +0.5 * gt_h
        gt_center_x = gt[1] +0.5 * gt_w
        # Anchor
        a_h = a[2] - a[0]
        a_w = a[3] - a[1]
        a_center_y = a[0] +0.5 * a_h
        a_center_x = a[1] +0.5 * a_w
        # Compute the bbox refinement that the RPN should predict.
        rpn_bbox[ix] = [
            (gt_center_y - a_center_y) / a_h,
            (gt_center_x - a_center_x) / a_w,
            np.log(gt_h / a_h),
            np.log(gt_w / a_w),
        ]
        # Normalize
        rpn_bbox[ix] /= config.RPN_BBOX_STD_DEV
        ix +=1
    return rpn_match, rpn_bbox
def generate_random_rois(image_shape, count, gt_class_ids, gt_boxes):
    """Generates ROI proposals similar to what a region proposal network
    would generate.
```

```
    image_shape: [Height, Width, Depth]
    count: Number of ROIs to generate
    gt_class_ids: [N] Integer ground truth class IDs
    gt_boxes: [N, (y1, x1, y2, x2)] Ground truth boxes in pixels.
    Returns: [count, (y1, x1, y2, x2)] ROI boxes in pixels.
    """
    # placeholder
    rois = np.zeros((count, 4), dtype =np.int32)
    # Generate random ROIs around GT boxes (90% of count)
    rois_per_box =int(0.9 * count / gt_boxes.shape[0])
    for i in range(gt_boxes.shape[0]):
        gt_y1, gt_x1, gt_y2, gt_x2 = gt_boxes[i]
        h = gt_y2 - gt_y1
        w = gt_x2 - gt_x1
        # random boundaries
        r_y1 = max(gt_y1 - h, 0)
        r_y2 = min(gt_y2 + h, image_shape[0])
        r_x1 = max(gt_x1 - w, 0)
        r_x2 = min(gt_x2 + w, image_shape[1])
        # To avoid generating boxes with zero area, we generate double what
        # we need and filter out the extra. If we get fewer valid boxes
        # than we need, we loop and try again.
        while True:
            y1y2 = np.random.randint(r_y1, r_y2, (rois_per_box *2, 2))
            x1x2 = np.random.randint(r_x1, r_x2, (rois_per_box *2, 2))
            # Filter out zero area boxes
            threshold =1
            y1y2 = y1y2[np.abs(y1y2[:, 0] - y1y2[:, 1]) >=
                        threshold][:rois_per_box]
            x1x2 = x1x2[np.abs(x1x2[:, 0] - x1x2[:, 1]) >=
                        threshold][:rois_per_box]
            if y1y2.shape[0] == rois_per_box and x1x2.shape[0] == rois_per_box:
                break
        # Sort on axis 1 to ensure x1 <= x2 and y1 <= y2 and then reshape
        # into x1, y1, x2, y2 order
        x1, x2 = np.split(np.sort(x1x2, axis =1), 2, axis =1)
        y1, y2 = np.split(np.sort(y1y2, axis =1), 2, axis =1)
        box_rois = np.hstack([y1, x1, y2, x2])
        rois[rois_per_box * i:rois_per_box * (i +1)] = box_rois
    # Generate random ROIs anywhere in the image (10% of count)
    remaining_count = count - (rois_per_box * gt_boxes.shape[0])
    # To avoid generating boxes with zero area, we generate double what
    # we need and filter out the extra. If we get fewer valid boxes
    # than we need, we loop and try again.
    while True:
        y1y2 = np.random.randint(0, image_shape[0], (remaining_count *2, 2))
        x1x2 = np.random.randint(0, image_shape[1], (remaining_count *2, 2))
        # Filter out zero area boxes
        threshold =1
        y1y2 = y1y2[np.abs(y1y2[:, 0] - y1y2[:, 1]) >=
                    threshold][:remaining_count]
        x1x2 = x1x2[np.abs(x1x2[:, 0] - x1x2[:, 1]) >=
                    threshold][:remaining_count]
        if y1y2.shape[0] == remaining_count and x1x2.shape[0] == remaining_count:
            break
    # Sort on axis 1 to ensure x1 <= x2 and y1 <= y2 and then reshape
    # into x1, y1, x2, y2 order
    x1, x2 = np.split(np.sort(x1x2, axis =1), 2, axis =1)
    y1, y2 = np.split(np.sort(y1y2, axis =1), 2, axis =1)
    global_rois = np.hstack([y1, x1, y2, x2])
    rois[-remaining_count:] = global_rois
    return rois
def data_generator(dataset, config, shuffle =True, augment =True, random_rois =0,
                   batch_size=1, detection_targets =False):
    """A generator that returns images and corresponding target class ids,
    bounding box deltas, and masks.
    dataset: The Dataset object to pick data from
    config: The model config object
    shuffle: If True, shuffles the samples before every epoch
    augment: If True, applies image augmentation to images (currently only
            horizontal flips are supported)
    random_rois: If > 0 then generate proposals to be used to train the
                network classifier and mask heads. Useful if training
                the Mask RCNN part without the RPN.
    batch_size: How many images to return in each call
    detection_targets: If True, generate detection targets (class IDs, bbox
        deltas, and masks). Typically for debugging or visualizations because
        in trainig detection targets are generated by DetectionTargetLayer.
    Returns a Python generator. Upon calling next() on it, the
    generator returns two lists, inputs and outputs. The containtes
    of the lists differs depending on the received arguments:
    inputs list:
    - images: [batch, H, W, C]
```

```
    - image_meta: [batch, size of image meta]
    - rpn_match: [batch, N] Integer (1=positive anchor, -1=negative, 0=neutral)
    - rpn_bbox: [batch, N, (dy, dx, log(dh), log(dw))] Anchor bbox deltas.
    - gt_class_ids: [batch, MAX_GT_INSTANCES] Integer class IDs
    - gt_boxes: [batch, MAX_GT_INSTANCES, (y1, x1, y2, x2)]
    - gt_masks: [batch, height, width, MAX_GT_INSTANCES]. The height and width
                are those of the image unless use_mini_mask is True, in which
                case they are defined in MINI_MASK_SHAPE.
    outputs list: Usually empty in regular training. But if detection_targets
        is True then the outputs list contains target class_ids, bbox deltas,
        and masks.
    """
    b =0  # batch item index
    image_index =-1
    image_ids = np.copy(dataset.image_ids)
    error_count =0
    # Anchors
    # [anchor_count, (y1, x1, y2, x2)]
    anchors = utils.generate_pyramid_anchors(config.RPN_ANCHOR_SCALES,
                                             config.RPN_ANCHOR_RATIOS,
                                             config.BACKBONE_SHAPES,
                                             config.BACKBONE_STRIDES,
                                             config.RPN_ANCHOR_STRIDE)
    # Keras requires a generator to run indefinately.
    while True:
        try:
            # Increment index to pick next image. Shuffle if at the start of an epoch.
            image_index = (image_index +1) % len(image_ids)
            if shuffle and image_index ==0:
                np.random.shuffle(image_ids)
            # Get GT bounding boxes and masks for image.
            image_id = image_ids[image_index]
            image, image_meta, gt_class_ids, gt_boxes, gt_masks = \
                load_image_gt(dataset, config, image_id, augment=augment,
                              use_mini_mask=config.USE_MINI_MASK)
            # Skip images that have no instances. This can happen in cases
            # where we train on a subset of classes and the image doesn't
            # have any of the classes we care about.
            if not np.any(gt_class_ids >0):
                continue
            # RPN Targets
            rpn_match, rpn_bbox = build_rpn_targets(image.shape, anchors,
                                                    gt_class_ids, gt_boxes, config)
            # Mask R-CNN Targets
            if random_rois:
                rpn_rois = generate_random_rois(
                    image.shape, random_rois, gt_class_ids, gt_boxes)
                if detection_targets:
                    rois, mrcnn_class_ids, mrcnn_bbox, mrcnn_mask =\
                        build_detection_targets(
                            rpn_rois, gt_class_ids, gt_boxes, gt_masks, config)
            # Init batch arrays
            if b ==0:
                batch_image_meta = np.zeros(
                    (batch_size,) + image_meta.shape, dtype =image_meta.dtype)
                batch_rpn_match = np.zeros(
                    [batch_size, anchors.shape[0], 1], dtype =rpn_match.dtype)
                batch_rpn_bbox = np.zeros(
                    [batch_size,     config.RPN_TRAIN_ANCHORS_PER_IMAGE,     4],     dtype
=rpn_bbox.dtype)
                batch_images = np.zeros(
                    (batch_size,) + image.shape, dtype =np.float32)
                batch_gt_class_ids = np.zeros(
                    (batch_size, config.MAX_GT_INSTANCES), dtype=np.int32)
                batch_gt_boxes = np.zeros(
                    (batch_size, config.MAX_GT_INSTANCES, 4), dtype =np.int32)
                if config.USE_MINI_MASK:
                    batch_gt_masks   =   np.zeros((batch_size,   config.MINI_MASK_SHAPE[0],
config.MINI_MASK_SHAPE[1],
                                                   config.MAX_GT_INSTANCES))
                else:
                    batch_gt_masks = np.zeros(
                        (batch_size,            image.shape[0],            image.shape[1],
config.MAX_GT_INSTANCES))
                if random_rois:
                    batch_rpn_rois = np.zeros(
                        (batch_size, rpn_rois.shape[0], 4), dtype =rpn_rois.dtype)
                    if detection_targets:
                        batch_rois = np.zeros(
                            (batch_size,) + rois.shape, dtype =rois.dtype)
                        batch_mrcnn_class_ids = np.zeros(
                            (batch_size,)       +       mrcnn_class_ids.shape,       dtype
=mrcnn_class_ids.dtype)
```

```
                        batch_mrcnn_bbox = np.zeros(
                            (batch_size,) + mrcnn_bbox.shape, dtype =mrcnn_bbox.dtype)
                        batch_mrcnn_mask = np.zeros(
                            (batch_size,) + mrcnn_mask.shape, dtype =mrcnn_mask.dtype)
            # If more instances than fits in the array, sub-sample from them.
            if gt_boxes.shape[0] > config.MAX_GT_INSTANCES:
                ids = np.random.choice(
                    np.arange(gt_boxes.shape[0]), config.MAX_GT_INSTANCES, replace =False)
                gt_class_ids = gt_class_ids[ids]
                gt_boxes = gt_boxes[ids]
                gt_masks = gt_masks[:, :, ids]
            # Add to batch
            batch_image_meta[b] = image_meta
            batch_rpn_match[b] = rpn_match[:, np.newaxis]
            batch_rpn_bbox[b] = rpn_bbox
            batch_images[b] = mold_image(image.astype(np.float32), config)
            batch_gt_class_ids[b, :gt_class_ids.shape[0]] = gt_class_ids
            batch_gt_boxes[b, :gt_boxes.shape[0]] = gt_boxes
            batch_gt_masks[b, :, :, :gt_masks.shape[-1]] = gt_masks
            if random_rois:
                batch_rpn_rois[b] = rpn_rois
                if detection_targets:
                    batch_rois[b] = rois
                    batch_mrcnn_class_ids[b] = mrcnn_class_ids
                    batch_mrcnn_bbox[b] = mrcnn_bbox
                    batch_mrcnn_mask[b] = mrcnn_mask
            b +=1
            # Batch full?
            if b >= batch_size:
                inputs = [batch_images, batch_image_meta, batch_rpn_match, batch_rpn_bbox,
                          batch_gt_class_ids, batch_gt_boxes, batch_gt_masks]
                outputs = []
                if random_rois:
                    inputs.extend([batch_rpn_rois])
                    if detection_targets:
                        inputs.extend([batch_rois])
                        # Keras requires that output and targets have the same number of
dimensions
                        batch_mrcnn_class_ids = np.expand_dims(
                            batch_mrcnn_class_ids, -1)
                        outputs.extend(
                            [batch_mrcnn_class_ids, batch_mrcnn_bbox, batch_mrcnn_mask])
                yield inputs, outputs
                # start a new batch
                b =0
        except (GeneratorExit, KeyboardInterrupt):
            raise
        except:
            # Log it and skip the image
            logging.exception("Error processing image {}".format(
                dataset.image_info[image_id]))
            error_count +=1
            if error_count >5:
                raise
############################################################
#  MaskRCNN Class
############################################################
class MaskRCNN():
    """Encapsulates the Mask RCNN model functionality.
    The actual Keras model is in the keras_model property.
    """
    def __init__(self, mode, config, model_dir):
        """
        mode: Either "training " or "inference "
        config: A Sub-class of the Config class
        model_dir: Directory to save training logs and trained weights
        """
        assert mode in ['training', 'inference']
        self.mode = mode
        self.config = config
        self.model_dir = model_dir
        self.set_log_dir()
        self.keras_model =self.build(mode =mode, config =config)
    def build(self, mode, config):
        """Build Mask R-CNN architecture.
            input_shape: The shape of the input image.
            mode: Either "training " or "inference ". The inputs and
                outputs of the model differ accordingly.
        """
        assert mode in ['training', 'inference']
        # Image size must be dividable by 2 multiple times
        h, w = config.IMAGE_SHAPE[:2]
        if h /2 **6 !=int(h /2 **6) or w /2 **6 !=int(w /2 **6):
```

```
            raise Exception("Image size must be dividable by 2 at least 6 times "
                            "to avoid fractions when downscaling and upscaling."
                            "For example, use 256, 320, 384, 448, 512, ... etc. ")
        # Inputs
        input_image = KL.Input(
            shape=config.IMAGE_SHAPE.tolist(), name ="input_image")
        input_image_meta = KL.Input(shape =[None], name ="input_image_meta")
        if mode =="training":
            # RPN GT
            input_rpn_match = KL.Input(
                shape=[None, 1], name ="input_rpn_match", dtype =tf.int32)
            input_rpn_bbox = KL.Input(
                shape=[None, 4], name ="input_rpn_bbox", dtype =tf.float32)
            # Detection GT (class IDs, bounding boxes, and masks)
            # 1. GT Class IDs (zero padded)
            input_gt_class_ids = KL.Input(
                shape=[None], name ="input_gt_class_ids", dtype =tf.int32)
            # 2. GT Boxes in pixels (zero padded)
            # [batch, MAX_GT_INSTANCES, (y1, x1, y2, x2)] in image coordinates
            input_gt_boxes = KL.Input(
                shape=[None, 4], name ="input_gt_boxes", dtype =tf.float32)
            # Normalize coordinates
            h, w = K.shape(input_image)[1], K.shape(input_image)[2]
            image_scale = K.cast(K.stack([h, w, h, w], axis =0), tf.float32)
            gt_boxes = KL.Lambda(lambda x: x / image_scale)(input_gt_boxes)
            # 3. GT Masks (zero padded)
            # [batch, height, width, MAX_GT_INSTANCES]
            if config.USE_MINI_MASK:
                input_gt_masks = KL.Input(
                    shape=[config.MINI_MASK_SHAPE[0],
                           config.MINI_MASK_SHAPE[1], None],
                    name="input_gt_masks", dtype =bool)
            else:
                input_gt_masks = KL.Input(
                    shape=[config.IMAGE_SHAPE[0], config.IMAGE_SHAPE[1], None],
                    name="input_gt_masks", dtype =bool)
        # Build the shared convolutional layers.
        # Bottom-up Layers
        # Returns a list of the last layers of each stage, 5 in total.
        # Don't create the thead (stage 5), so we pick the 4th item in the list.
        _, C2, C3, C4, C5 = resnet_graph(input_image, config.BACKBONE, stage5 =True)
        # Top-down Layers
        # TODO: add assert to varify feature map sizes match what's in config
        P5 = KL.Conv2D(256, (1, 1), name ='fpn_c5p5')(C5)
        P4 = KL.Add(name ="fpn_p4add")([
            KL.UpSampling2D(size=(2, 2), name ="fpn_p5upsampled")(P5),
            KL.Conv2D(256, (1, 1), name ='fpn_c4p4')(C4)])
        P3 = KL.Add(name ="fpn_p3add")([
            KL.UpSampling2D(size=(2, 2), name ="fpn_p4upsampled")(P4),
            KL.Conv2D(256, (1, 1), name ='fpn_c3p3')(C3)])
        P2 = KL.Add(name ="fpn_p2add")([
            KL.UpSampling2D(size=(2, 2), name ="fpn_p3upsampled")(P3),
            KL.Conv2D(256, (1, 1), name ='fpn_c2p2')(C2)])
        # Attach 3x3 conv to all P layers to get the final feature maps.
        P2 = KL.Conv2D(256, (3, 3), padding ="SAME", name ="fpn_p2")(P2)
        P3 = KL.Conv2D(256, (3, 3), padding ="SAME", name ="fpn_p3")(P3)
        P4 = KL.Conv2D(256, (3, 3), padding ="SAME", name ="fpn_p4")(P4)
        P5 = KL.Conv2D(256, (3, 3), padding ="SAME", name ="fpn_p5")(P5)
        # P6 is used for the 5th anchor scale in RPN. Generated by
        # subsampling from P5 with stride of 2.
        P6 = KL.MaxPooling2D(pool_size =(1, 1), strides =2, name ="fpn_p6")(P5)
        # Note that P6 is used in RPN, but not in the classifier heads.
        rpn_feature_maps = [P2, P3, P4, P5, P6]
        mrcnn_feature_maps = [P2, P3, P4, P5]
        # Generate Anchors
        self.anchors = utils.generate_pyramid_anchors(config.RPN_ANCHOR_SCALES,
                                                      config.RPN_ANCHOR_RATIOS,
                                                      config.BACKBONE_SHAPES,
                                                      config.BACKBONE_STRIDES,
                                                      config.RPN_ANCHOR_STRIDE)
        # RPN Model
        rpn = build_rpn_model(config.RPN_ANCHOR_STRIDE,
                              len(config.RPN_ANCHOR_RATIOS), 256)
        # Loop through pyramid layers
        layer_outputs = []  # list of lists
        for p in rpn_feature_maps:
            layer_outputs.append(rpn([p]))
        # Concatenate layer outputs
        # Convert from list of lists of level outputs to list of lists
        # of outputs across levels.
        # e.g. [[a1, b1, c1], [a2, b2, c2]] => [[a1, a2], [b1, b2], [c1, c2]]
        output_names = ["rpn_class_logits", "rpn_class", "rpn_bbox"]
        outputs = list(zip(*layer_outputs))
```

```
        outputs = [KL.Concatenate(axis =1, name =n)(list(o))
                   for o, n in zip(outputs, output_names)]
        rpn_class_logits, rpn_class, rpn_bbox = outputs
        # Generate proposals
        # Proposals are [batch, N, (y1, x1, y2, x2)] in normalized coordinates
        # and zero padded.
        proposal_count = config.POST_NMS_ROIS_TRAINING if mode =="training"\
            else config.POST_NMS_ROIS_INFERENCE
        rpn_rois = ProposalLayer(proposal_count =proposal_count,
                                 nms_threshold=config.RPN_NMS_THRESHOLD,
                                 name="ROI",
                                 anchors=self.anchors,
                                 config=config)([rpn_class, rpn_bbox])
        if mode =="training":
            # Class ID mask to mark class IDs supported by the dataset the image
            # came from.
            _, _, _, active_class_ids = KL.Lambda(lambda x: parse_image_meta_graph(x),
                                                  mask=[None,           None,           None,
None])(input_image_meta)
            if not config.USE_RPN_ROIS:
                # Ignore predicted ROIs and use ROIs provided as an input.
                input_rois = KL.Input(shape =[config.POST_NMS_ROIS_TRAINING, 4],
                                      name="input_roi", dtype =np.int32)
                # Normalize coordinates to 0-1 range.
                target_rois = KL.Lambda(lambda x: K.cast(
                    x, tf.float32) / image_scale[:4])(input_rois)
            else:
                target_rois = rpn_rois
            # Generate detection targets
            # Subsamples proposals and generates target outputs for training
            # Note that proposal class IDs, gt_boxes, and gt_masks are zero
            # padded. Equally, returned rois and targets are zero padded.
            rois, target_class_ids, target_bbox, target_mask =\
                DetectionTargetLayer(config, name="proposal_targets")([
                    target_rois, input_gt_class_ids, gt_boxes, input_gt_masks])
            # Network Heads
            # TODO: verify that this handles zero padded ROIs
            mrcnn_class_logits, mrcnn_class, mrcnn_bbox =\
                fpn_classifier_graph(rois, mrcnn_feature_maps, config.IMAGE_SHAPE,
                                     config.POOL_SIZE, config.NUM_CLASSES)
            mrcnn_mask = build_fpn_mask_graph(rois, mrcnn_feature_maps,
                                              config.IMAGE_SHAPE,
                                              config.MASK_POOL_SIZE,
                                              config.NUM_CLASSES)
            # TODO: clean up (use tf.identify if necessary)
            output_rois = KL.Lambda(lambda x: x *1, name ="output_rois")(rois)
            # Losses
            rpn_class_loss   =   KL.Lambda(lambda   x:   rpn_class_loss_graph(*x),   name
="rpn_class_loss")(
                [input_rpn_match, rpn_class_logits])
            rpn_bbox_loss  =  KL.Lambda(lambda  x:  rpn_bbox_loss_graph(config,  *x),  name
="rpn_bbox_loss")(
                [input_rpn_bbox, input_rpn_match, rpn_bbox])
            class_loss    =    KL.Lambda(lambda    x:    mrcnn_class_loss_graph(*x),    name
="mrcnn_class_loss")(
                [target_class_ids, mrcnn_class_logits, active_class_ids])
            bbox_loss    =    KL.Lambda(lambda    x:    mrcnn_bbox_loss_graph(*x),    name
="mrcnn_bbox_loss")(
                [target_bbox, target_class_ids, mrcnn_bbox])
            mask_loss    =    KL.Lambda(lambda    x:    mrcnn_mask_loss_graph(*x),    name
="mrcnn_mask_loss")(
                [target_mask, target_class_ids, mrcnn_mask])
            # Model
            inputs = [input_image, input_image_meta,
                      input_rpn_match, input_rpn_bbox, input_gt_class_ids, input_gt_boxes,
input_gt_masks]
            if not config.USE_RPN_ROIS:
                inputs.append(input_rois)
            outputs = [rpn_class_logits, rpn_class, rpn_bbox,
                       mrcnn_class_logits, mrcnn_class, mrcnn_bbox, mrcnn_mask,
                       rpn_rois, output_rois,
                       rpn_class_loss, rpn_bbox_loss, class_loss, bbox_loss, mask_loss]
            model = KM.Model(inputs, outputs, name ='mask_rcnn')
        else:
            # Network Heads
            # Proposal classifier and BBox regressor heads
            mrcnn_class_logits, mrcnn_class, mrcnn_bbox =\
                fpn_classifier_graph(rpn_rois, mrcnn_feature_maps, config.IMAGE_SHAPE,
                                     config.POOL_SIZE, config.NUM_CLASSES)
            # Detections
            # output is [batch, num_detections, (y1, x1, y2, x2, class_id, score)] in
image coordinates
            detections = DetectionLayer(config, name ="mrcnn_detection")(
```

```
                [rpn_rois, mrcnn_class, mrcnn_bbox, input_image_meta])
            # Convert boxes to normalized coordinates
            # TODO: let DetectionLayer return normalized coordinates to avoid
            #       unnecessary conversions
            h, w = config.IMAGE_SHAPE[:2]
            detection_boxes = KL.Lambda(
                lambda x: x[..., :4] / np.array([h, w, h, w]))(detections)
            # Create masks for detections
            mrcnn_mask = build_fpn_mask_graph(detection_boxes, mrcnn_feature_maps,
                                              config.IMAGE_SHAPE,
                                              config.MASK_POOL_SIZE,
                                              config.NUM_CLASSES)
            model = KM.Model([input_image, input_image_meta],
                             [detections, mrcnn_class, mrcnn_bbox,
                                 mrcnn_mask, rpn_rois, rpn_class, rpn_bbox],
                             name='mask_rcnn')
        # Add multi-GPU support.
        if config.GPU_COUNT >1:
            from parallel_model import ParallelModel
            model = ParallelModel(model, config.GPU_COUNT)
        return model
    def find_last(self):
        """Finds the last checkpoint file of the last trained model in the
        model directory.
        Returns:
            log_dir: The directory where events and weights are saved
            checkpoint_path: the path to the last checkpoint file
        """
        # Get directory names. Each directory corresponds to a model
        dir_names =next(os.walk(self.model_dir))[1]
        key =self.config.NAME.lower()
        dir_names = filter(lambda f: f.startswith(key), dir_names)
        dir_names = sorted(dir_names)
        if not dir_names:
            return None, None
        # Pick last directory
        dir_name = os.path.join(self.model_dir, dir_names[-1])
        # Find the last checkpoint
        checkpoints =next(os.walk(dir_name))[2]
        checkpoints = filter(lambda f: f.startswith("mask_rcnn"), checkpoints)
        checkpoints = sorted(checkpoints)
        if not checkpoints:
            return dir_name, None
        checkpoint = os.path.join(dir_name, checkpoints[-1])
        return dir_name, checkpoint
    def load_weights(self, filepath, by_name =False, exclude =None):
        """Modified version of the correspoding Keras function with
        the addition of multi-GPU support and the ability to exclude
        some layers from loading.
        exlude: list of layer names to excluce
        """
        import h5py
        from keras.engine import saving
        if exclude:
            by_name =True
        if h5py is None:
            raise ImportError('`load_weights` requires h5py.')
        f = h5py.File(filepath, mode ='r')
        if 'layer_names'not in f.attrs and 'model_weights'in f:
            f = f['model_weights']
        # In multi-GPU training, we wrap the model. Get layers
        # of the inner model because they have the weights.
        keras_model =self.keras_model
        layers = keras_model.inner_model.layers if hasattr(keras_model, "inner_model")\
            else keras_model.layers
        # Exclude some layers
        if exclude:
            layers = filter(lambda l: l.name not in exclude, layers)
        if by_name:
            saving.load_weights_from_hdf5_group_by_name(f, layers)
        else:
            saving.load_weights_from_hdf5_group(f, layers)
        if hasattr(f, 'close'):
            f.close()
        # Update the log directory
        self.set_log_dir(filepath)
    def get_imagenet_weights(self):
        """Downloads ImageNet trained weights from Keras.
        Returns path to weights file.
        """
        from keras.utils.data_utils import get_file
        TF_WEIGHTS_PATH_NO_TOP ='https://github.com/fchollet/deep-learning-models/'\
                                'releases/download/v0.2/'\
```

```
                                'resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5'
        weights_path = get_file('resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5',
                                TF_WEIGHTS_PATH_NO_TOP,
                                cache_subdir='models',
                                md5_hash='a268eb855778b3df3c7506639542a6af')
        return weights_path
    def compile(self, learning_rate, momentum):
        """Gets the model ready for training. Adds losses, regularization, and
        metrics. Then calls the Keras compile() function.
        """
        # Optimizer object
        optimizer = keras.optimizers.SGD(lr =learning_rate, momentum =momentum,
                                         clipnorm=5.0)
        # Add Losses
        # First, clear previously set losses to avoid duplication
        self.keras_model._losses = []
        self.keras_model._per_input_losses = {}
        loss_names = ["rpn_class_loss", "rpn_bbox_loss",
                      "mrcnn_class_loss", "mrcnn_bbox_loss", "mrcnn_mask_loss"]
        for name in loss_names:
            layer =self.keras_model.get_layer(name)
            if layer.output in self.keras_model.losses:
                continue
            self.keras_model.add_loss(
                tf.reduce_mean(layer.output, keep_dims=True))
        # Add L2 Regularization
        # Skip gamma and beta weights of batch normalization layers.
        reg_losses      =      [keras.regularizers.l2(self.config.WEIGHT_DECAY)(w)      /
tf.cast(tf.size(w), tf.float32)
                        for w in self.keras_model.trainable_weights
                        if 'gamma'not in w.name and 'beta'not in w.name]
        self.keras_model.add_loss(tf.add_n(reg_losses))
        # Compile
        self.keras_model.compile(optimizer =optimizer, loss =[
                                 None] *len(self.keras_model.outputs))
        # Add metrics for losses
        for name in loss_names:
            if name in self.keras_model.metrics_names:
                continue
            layer =self.keras_model.get_layer(name)
            self.keras_model.metrics_names.append(name)
            self.keras_model.metrics_tensors.append(tf.reduce_mean(
                layer.output, keep_dims=True))
    def set_trainable(self, layer_regex, keras_model =None, indent =0, verbose =1):
        """Sets model layers as trainable if their names match
        the given regular expression.
        """
        # Print message on the first call (but not on recursive calls)
        if verbose >0 and keras_model is None:
            log("Selecting layers to train")
        keras_model = keras_model or self.keras_model
        # In multi-GPU training, we wrap the model. Get layers
        # of the inner model because they have the weights.
        layers = keras_model.inner_model.layers if hasattr(keras_model, "inner_model")\
            else keras_model.layers
        for layer in layers:
            # Is the layer a model?
            if layer.__class__.__name__ =='Model':
                print("In model: ", layer.name)
                self.set_trainable(
                    layer_regex, keras_model=layer, indent =indent +4)
                continue
            if not layer.weights:
                continue
            # Is it trainable?
            trainable = bool(re.fullmatch(layer_regex, layer.name))
            # Update layer. If layer is a container, update inner layer.
            if layer.__class__.__name__ =='TimeDistributed':
                layer.layer.trainable = trainable
            else:
                layer.trainable = trainable
            # Print trainble layer names
            if trainable and verbose >0:
                log("{}{:20}   ({})".format(" "* indent, layer.name,
                                            layer.__class__.__name__))
    def set_log_dir(self, model_path =None):
        """Sets the model log directory and epoch counter.
        model_path: If None, or a format different from what this code uses
            then set a new log directory and start epochs from 0. Otherwise,
            extract the log directory and the epoch counter from the file
            name.
        """
        # Set date and epoch counter as if starting a new model
```

```
        self.epoch =0
        now = datetime.datetime.now()
        # If we have a model path with date and epochs use them
        if model_path:
            # Continue from we left of. Get epoch and date from the file name
            # A sample model path might look like:
            # /path/to/logs/coco20171029T2315/mask_rcnn_coco_0001.h5
            regex                                    =                                                    r
".*/\w+(\d{4})(\d{2})(\d{2})T(\d{2})(\d{2})/mask\_rcnn\_\w+(\d{4})\.h5"
            m = re.match(regex, model_path)
            if m:
                now = datetime.datetime(int(m.group(1)), int(m.group(2)), int(m.group(3)),
                                        int(m.group(4)), int(m.group(5)))
                self.epoch =int(m.group(6)) +1
        # Directory for training logs
        self.log_dir = os.path.join(self.model_dir, "{}{:%Y%m%dT%H%M}".format(
            self.config.NAME.lower(), now))
        # Path to save after each epoch. Include placeholders that get filled by Keras.
        self.checkpoint_path                 =                  os.path.join(self.log_dir,
"mask_rcnn_{}_*epoch*.h5".format(
            self.config.NAME.lower()))
        self.checkpoint_path =self.checkpoint_path.replace(
            "*epoch*", "{epoch:04d}")
    def train(self, train_dataset, val_dataset, learning_rate, epochs, layers):
        """Train the model.
        train_dataset, val_dataset: Training and validation Dataset objects.
        learning_rate: The learning rate to train with
        epochs: Number of training epochs. Note that previous training epochs
                are considered to be done alreay, so this actually determines
                the epochs to train in total rather than in this particaular
                call.
        layers: Allows selecting wich layers to train. It can be:
            - A regular expression to match layer names to train
            - One of these predefined values:
              heaads: The RPN, classifier and mask heads of the network
              all: All the layers
              3+: Train Resnet stage 3 and up
              4+: Train Resnet stage 4 and up
              5+: Train Resnet stage 5 and up
        """
        assert self.mode =="training", "Create model in training mode."
        # Pre-defined layer regular expressions
        layer_regex = {
            # all layers but the backbone
            "heads": r "(mrcnn\_.*)|(rpn\_.*)|(fpn\_.*)",
            # From a specific Resnet stage and up
            "3+":                                                                                    r
"(res3.*)|(bn3.*)|(res4.*)|(bn4.*)|(res5.*)|(bn5.*)|(mrcnn\_.*)|(rpn\_.*)|(fpn\_.*)",
            "4+": r "(res4.*)|(bn4.*)|(res5.*)|(bn5.*)|(mrcnn\_.*)|(rpn\_.*)|(fpn\_.*)",
            "5+": r "(res5.*)|(bn5.*)|(mrcnn\_.*)|(rpn\_.*)|(fpn\_.*)",
            # All layers
            "all": ".*",
        }
        if layers in layer_regex.keys():
            layers = layer_regex[layers]
        # Data generators
        train_generator = data_generator(train_dataset, self.config, shuffle =True,
                                         batch_size=self.config.BATCH_SIZE)
        val_generator = data_generator(val_dataset, self.config, shuffle =True,
                                       batch_size=self.config.BATCH_SIZE,
                                       augment=False)
        # Callbacks
        callbacks = [
            keras.callbacks.TensorBoard(log_dir=self.log_dir,
                                        histogram_freq=0,  write_graph  =True,  write_images
=False),
            keras.callbacks.ModelCheckpoint(self.checkpoint_path,
                                            verbose=0, save_weights_only =True),
        ]
        # Train
        log("\nStarting at epoch {}. LR={}\n".format(self.epoch, learning_rate))
        log("Checkpoint Path: {}".format(self.checkpoint_path))
        self.set_trainable(layers)
        self.compile(learning_rate, self.config.LEARNING_MOMENTUM)
        # Work-around for Windows: Keras fails on Windows when using
        # multiprocessing workers. See discussion here:
        # https://github.com/matterport/Mask_RCNN/issues/13#issuecomment-353124009
        if os.name is 'nt':
            workers =0
        else:
            workers = max(self.config.BATCH_SIZE //2, 2)
        self.keras_model.fit_generator(
            train_generator,
```

```
            initial_epoch=self.epoch,
            epochs=epochs,
            steps_per_epoch=self.config.STEPS_PER_EPOCH,
            callbacks=callbacks,
            validation_data=next(val_generator),
            validation_steps=self.config.VALIDATION_STEPS,
            max_queue_size=100,
            workers=1,
            use_multiprocessing=False,
        )
        self.epoch = max(self.epoch, epochs)
    def mold_inputs(self, images):
        """Takes a list of images and modifies them to the format expected
        as an input to the neural network.
        images: List of image matricies [height,width,depth]. Images can have
            different sizes.
        Returns 3 Numpy matricies:
        molded_images: [N, h, w, 3]. Images resized and normalized.
        image_metas: [N, length of meta data]. Details about each image.
        windows: [N, (y1, x1, y2, x2)]. The portion of the image that has the
            original image (padding excluded).
        """
        molded_images = []
        image_metas = []
        windows = []
        for image in images:
            # Resize image to fit the model expected size
            # TODO: move resizing to mold_image()
            molded_image, window, scale, padding = utils.resize_image(
                image,
                min_dim=self.config.IMAGE_MIN_DIM,
                max_dim=self.config.IMAGE_MAX_DIM,
                padding=self.config.IMAGE_PADDING)
            molded_image = mold_image(molded_image, self.config)
            # Build image_meta
            image_meta = compose_image_meta(
                0, image.shape, window,
                np.zeros([self.config.NUM_CLASSES], dtype =np.int32))
            # Append
            molded_images.append(molded_image)
            windows.append(window)
            image_metas.append(image_meta)
        # Pack into arrays
        molded_images = np.stack(molded_images)
        image_metas = np.stack(image_metas)
        windows = np.stack(windows)
        return molded_images, image_metas, windows
    def unmold_detections(self, detections, mrcnn_mask, image_shape, window):
        """Reformats the detections of one image from the format of the neural
        network output to a format suitable for use in the rest of the
        application.
        detections: [N, (y1, x1, y2, x2, class_id, score)]
        mrcnn_mask: [N, height, width, num_classes]
        image_shape: [height, width, depth] Original size of the image before resizing
        window: [y1, x1, y2, x2] Box in the image where the real image is
                excluding the padding.
        Returns:
        boxes: [N, (y1, x1, y2, x2)] Bounding boxes in pixels
        class_ids: [N] Integer class IDs for each bounding box
        scores: [N] Float probability scores of the class_id
        masks: [height, width, num_instances] Instance masks
        """
        # How many detections do we have?
        # Detections array is padded with zeros. Find the first class_id == 0.
        zero_ix = np.where(detections[:, 4] ==0)[0]
        N = zero_ix[0] if zero_ix.shape[0] >0 else detections.shape[0]
        # Extract boxes, class_ids, scores, and class-specific masks
        boxes = detections[:N, :4]
        class_ids = detections[:N, 4].astype(np.int32)
        scores = detections[:N, 5]
        masks = mrcnn_mask[np.arange(N), :, :, class_ids]
        # Compute scale and shift to translate coordinates to image domain.
        h_scale = image_shape[0] / (window[2] - window[0])
        w_scale = image_shape[1] / (window[3] - window[1])
        scale = min(h_scale, w_scale)
        shift = window[:2]  # y, x
        scales = np.array([scale, scale, scale, scale])
        shifts = np.array([shift[0], shift[1], shift[0], shift[1]])
        # Translate bounding boxes to image domain
        boxes = np.multiply(boxes - shifts, scales).astype(np.int32)
        # Filter out detections with zero area. Often only happens in early
        # stages of training when the network weights are still a bit random.
        exclude_ix = np.where(
```

```
            (boxes[:, 2] - boxes[:, 0]) * (boxes[:, 3] - boxes[:, 1]) <=0)[0]
        if exclude_ix.shape[0] >0:
            boxes = np.delete(boxes, exclude_ix, axis =0)
            class_ids = np.delete(class_ids, exclude_ix, axis =0)
            scores = np.delete(scores, exclude_ix, axis =0)
            masks = np.delete(masks, exclude_ix, axis =0)
            N = class_ids.shape[0]
        # Resize masks to original image size and set boundary threshold.
        full_masks = []
        for i in range(N):
            # Convert neural network mask to full size mask
            full_mask = utils.unmold_mask(masks[i], boxes[i], image_shape)
            full_masks.append(full_mask)
        full_masks = np.stack(full_masks, axis =-1)\
            if full_masks else np.empty((0,) + masks.shape[1:3])
        return boxes, class_ids, scores, full_masks
    def detect(self, images, verbose =0):
        """Runs the detection pipeline.
        images: List of images, potentially of different sizes.
        Returns a list of dicts, one dict per image. The dict contains:
        rois: [N, (y1, x1, y2, x2)] detection bounding boxes
        class_ids: [N] int class IDs
        scores: [N] float probability scores for the class IDs
        masks: [H, W, N] instance binary masks
        """
        assert self.mode =="inference", "Create model in inference mode."
        assert len(
            images) ==self.config.BATCH_SIZE, "len(images) must be equal to BATCH_SIZE"
        if verbose:
            log("Processing {} images".format(len(images)))
            for image in images:
                log("image", image)
        # Mold inputs to format expected by the neural network
        molded_images, image_metas, windows =self.mold_inputs(images)
        if verbose:
            log("molded_images", molded_images)
            log("image_metas", image_metas)
        # Run object detection
        detections, mrcnn_class, mrcnn_bbox, mrcnn_mask, \
            rois, rpn_class, rpn_bbox =\
            self.keras_model.predict([molded_images, image_metas], verbose =0)
        # Process detections
        results = []
        for i, image in enumerate(images):
            final_rois, final_class_ids, final_scores, final_masks =\
                self.unmold_detections(detections[i], mrcnn_mask[i],
                                       image.shape, windows[i])
            results.append({
                "rois": final_rois,
                "class_ids": final_class_ids,
                "scores": final_scores,
                "masks": final_masks,
            })
        return results
    def ancestor(self, tensor, name, checked =None):
        """Finds the ancestor of a TF tensor in the computation graph.
        tensor: TensorFlow symbolic tensor.
        name: Name of ancestor tensor to find
        checked: For internal use. A list of tensors that were already
                 searched to avoid loops in traversing the graph.
        """
        checked = checked if checked is not None else []
        # Put a limit on how deep we go to avoid very long loops
        if len(checked) >500:
            return None
        # Convert name to a regex and allow matching a number prefix
        # because Keras adds them automatically
        if isinstance(name, str):
            name = re.compile(name.replace("/", r "(\_\d+)*/"))
        parents = tensor.op.inputs
        for p in parents:
            if p in checked:
                continue
            if bool(re.fullmatch(name, p.name)):
                return p
            checked.append(p)
            a =self.ancestor(p, name, checked)
            if a is not None:
                return a
        return None
    def find_trainable_layer(self, layer):
        """If a layer is encapsulated by another layer, this function
        digs through the encapsulation and returns the layer that holds
```

```
        the weights.
        """
        if layer.__class__.__name__ =='TimeDistributed':
            return self.find_trainable_layer(layer.layer)
        return layer
    def get_trainable_layers(self):
        """Returns a list of layers that have weights."""
        layers = []
        # Loop through all layers
        for l in self.keras_model.layers:
            # If layer is a wrapper, find inner trainable layer
            l =self.find_trainable_layer(l)
            # Include layer if it has weights
            if l.get_weights():
                layers.append(l)
        return layers
    def run_graph(self, images, outputs):
        """Runs a sub-set of the computation graph that computes the given
        outputs.
        outputs: List of tuples (name, tensor) to compute. The tensors are
            symbolic TensorFlow tensors and the names are for easy tracking.
        Returns an ordered dict of results. Keys are the names received in the
        input and values are Numpy arrays.
        """
        model =self.keras_model
        # Organize desired outputs into an ordered dict
        outputs = OrderedDict(outputs)
        for o in outputs.values():
            assert o is not None
        # Build a Keras function to run parts of the computation graph
        inputs = model.inputs
        if model.uses_learning_phase and not isinstance(K.learning_phase(), int):
            inputs += [K.learning_phase()]
        kf = K.function(model.inputs, list(outputs.values()))
        # Run inference
        molded_images, image_metas, windows =self.mold_inputs(images)
        # TODO: support training mode?
        # if TEST_MODE == "training":
        #     model_in = [molded_images, image_metas,
        #                 target_rpn_match, target_rpn_bbox,
        #                 gt_boxes, gt_masks]
        #     if not config.USE_RPN_ROIS:
        #         model_in.append(target_rois)
        #     if model.uses_learning_phase and not isinstance(K.learning_phase(), int):
        #         model_in.append(1.)
        #     outputs_np = kf(model_in)
        # else:
        model_in = [molded_images, image_metas]
        if model.uses_learning_phase and not isinstance(K.learning_phase(), int):
            model_in.append(0.)
        outputs_np = kf(model_in)
        # Pack the generated Numpy arrays into a a dict and log the results.
        outputs_np = OrderedDict([(k, v)
                                  for k, v in zip(outputs.keys(), outputs_np)])
        for k, v in outputs_np.items():
            log(k, v)
        return outputs_np
############################################################
#  Data Formatting
############################################################
def compose_image_meta(image_id, image_shape, window, active_class_ids):
    """Takes attributes of an image and puts them in one 1D array.
    image_id: An int ID of the image. Useful for debugging.
    image_shape: [height, width, channels]
    window: (y1, x1, y2, x2) in pixels. The area of the image where the real
            image is (excluding the padding)
    active_class_ids: List of class_ids available in the dataset from which
        the image came. Useful if training on images from multiple datasets
        where not all classes are present in all datasets.
    """
    meta = np.array(
        [image_id] +           # size=1
        list(image_shape) +     # size=3
        list(window) +         # size=4 (y1, x1, y2, x2) in image cooredinates
        list(active_class_ids)  # size=num_classes
    )
    return meta
def parse_image_meta_graph(meta):
    """Parses a tensor that contains image attributes to its components.
    See compose_image_meta() for more details.
    meta: [batch, meta length] where meta length depends on NUM_CLASSES
    """
    image_id = meta[:, 0]
```

```
    image_shape = meta[:, 1:4]
    window = meta[:, 4:8]   # (y1, x1, y2, x2) window of image in in pixels
    active_class_ids = meta[:, 8:]
    return [image_id, image_shape, window, active_class_ids]
def mold_image(images, config):
    """Takes RGB images with 0-255 values and subtraces
    the mean pixel and converts it to float. Expects image
    colors in RGB order.
    """
    return images.astype(np.float32) - config.MEAN_PIXEL
def unmold_image(normalized_images, config):
    """Takes a image normalized with mold() and returns the original."""
    return (normalized_images + config.MEAN_PIXEL).astype(np.uint8)
############################################################
#  Miscellenous Graph Functions
############################################################
def trim_zeros_graph(boxes, name =None):
    """Often boxes are represented with matricies of shape [N, 4] and
    are padded with zeros. This removes zero boxes.
    boxes: [N, 4] matrix of boxes.
    non_zeros: [N] a 1D boolean mask identifying the rows to keep
    """
    non_zeros = tf.cast(tf.reduce_sum(tf.abs(boxes), axis =1), tf.bool)
    boxes = tf.boolean_mask(boxes, non_zeros, name =name)
    return boxes, non_zeros
def batch_pack_graph(x, counts, num_rows):
    """Picks different number of values from each row
    in x depending on the values in counts.
    """
    outputs = []
    for i in range(num_rows):
        outputs.append(x[i, :counts[i]])
    return tf.concat(outputs, axis =0)
```

- 학습과 추론 간 필요 기능 구현 코드

utils.py

```
"""
Mask R-CNN
Common utility functions and classes.
Copyright (c) 2017 Matterport, Inc.
Licensed under the MIT License (see LICENSE for details)
Written by Waleed Abdulla
"""
import sys
import os
import math
import random
import numpy as np
import tensorflow as tf
import scipy.misc
import skimage.color
import skimage.io
import urllib.request
import shutil
# URL from which to download the latest COCO trained weights
C      O      C      O          _          M      O      D      E      L          _          U      R      L
="https://github.com/matterport/Mask_RCNN/releases/download/v2.0/mask_rcnn_coco.h5"
############################################################
#  Bounding Boxes
############################################################
def extract_bboxes(mask):
    """Compute bounding boxes from masks.
    mask: [height, width, num_instances]. Mask pixels are either 1 or 0.
    Returns: bbox array [num_instances, (y1, x1, y2, x2)].
    """
    boxes = np.zeros([mask.shape[-1], 4], dtype =np.int32)
    for i in range(mask.shape[-1]):
        m = mask[:, :, i]
        # Bounding box.
        horizontal_indicies = np.where(np.any(m, axis =0))[0]
        vertical_indicies = np.where(np.any(m, axis =1))[0]
        if horizontal_indicies.shape[0]:
            x1, x2 = horizontal_indicies[[0, -1]]
            y1, y2 = vertical_indicies[[0, -1]]
            # x2 and y2 should not be part of the box. Increment by 1.
            x2 +=1
            y2 +=1
        else:
            # No mask for this instance. Might happen due to
            # resizing or cropping. Set bbox to zeros
```

```
            x1, x2, y1, y2 =0, 0, 0, 0
        boxes[i] = np.array([y1, x1, y2, x2])
    return boxes.astype(np.int32)
def compute_iou(box, boxes, box_area, boxes_area):
    """Calculates IoU of the given box with the array of the given boxes.
    box: 1D vector [y1, x1, y2, x2]
    boxes: [boxes_count, (y1, x1, y2, x2)]
    box_area: float. the area of 'box'
    boxes_area: array of length boxes_count.
    Note: the areas are passed in rather than calculated here for
          efficency. Calculate once in the caller to avoid duplicate work.
    """
    # Calculate intersection areas
    y1 = np.maximum(box[0], boxes[:, 0])
    y2 = np.minimum(box[2], boxes[:, 2])
    x1 = np.maximum(box[1], boxes[:, 1])
    x2 = np.minimum(box[3], boxes[:, 3])
    intersection = np.maximum(x2 - x1, 0) * np.maximum(y2 - y1, 0)
    union = box_area + boxes_area[:] - intersection[:]
    iou = intersection / union
    return iou
def compute_overlaps(boxes1, boxes2):
    """Computes IoU overlaps between two sets of boxes.
    boxes1, boxes2: [N, (y1, x1, y2, x2)].
    For better performance, pass the largest set first and the smaller second.
    """
    # Areas of anchors and GT boxes
    area1 = (boxes1[:, 2] - boxes1[:, 0]) * (boxes1[:, 3] - boxes1[:, 1])
    area2 = (boxes2[:, 2] - boxes2[:, 0]) * (boxes2[:, 3] - boxes2[:, 1])
    # Compute overlaps to generate matrix [boxes1 count, boxes2 count]
    # Each cell contains the IoU value.
    overlaps = np.zeros((boxes1.shape[0], boxes2.shape[0]))
    for i in range(overlaps.shape[1]):
        box2 = boxes2[i]
        overlaps[:, i] = compute_iou(box2, boxes1, area2[i], area1)
    return overlaps
def compute_overlaps_masks(masks1, masks2):
    '''Computes IoU overlaps between two sets of masks.
    masks1, masks2: [Height, Width, instances]
    '''
    # flatten masks
    masks1 = np.reshape(masks1 > .5, (-1, masks1.shape[-1])).astype(np.float32)
    masks2 = np.reshape(masks2 > .5, (-1, masks2.shape[-1])).astype(np.float32)
    area1 = np.sum(masks1, axis =0)
    area2 = np.sum(masks2, axis =0)
    # intersections and union
    intersections = np.dot(masks1.T, masks2)
    union = area1[:, None] + area2[None, :] - intersections
    overlaps = intersections / union
    return overlaps
def non_max_suppression(boxes, scores, threshold):
    """Performs non-maximum supression and returns indicies of kept boxes.
    boxes: [N, (y1, x1, y2, x2)]. Notice that (y2, x2) lays outside the box.
    scores: 1-D array of box scores.
    threshold: Float. IoU threshold to use for filtering.
    """
    assert boxes.shape[0] >0
    if boxes.dtype.kind !="f":
        boxes = boxes.astype(np.float32)
    # Compute box areas
    y1 = boxes[:, 0]
    x1 = boxes[:, 1]
    y2 = boxes[:, 2]
    x2 = boxes[:, 3]
    area = (y2 - y1) * (x2 - x1)
    # Get indicies of boxes sorted by scores (highest first)
    ixs = scores.argsort()[::-1]
    pick = []
    while len(ixs) >0:
        # Pick top box and add its index to the list
        i = ixs[0]
        pick.append(i)
        # Compute IoU of the picked box with the rest
        iou = compute_iou(boxes[i], boxes[ixs[1:]], area[i], area[ixs[1:]])
        # Identify boxes with IoU over the threshold. This
        # returns indicies into ixs[1:], so add 1 to get
        # indicies into ixs.
        remove_ixs = np.where(iou > threshold)[0] +1
        # Remove indicies of the picked and overlapped boxes.
        ixs = np.delete(ixs, remove_ixs)
        ixs = np.delete(ixs, 0)
    return np.array(pick, dtype =np.int32)
def apply_box_deltas(boxes, deltas):
```

```
    """Applies the given deltas to the given boxes.
    boxes: [N, (y1, x1, y2, x2)]. Note that (y2, x2) is outside the box.
    deltas: [N, (dy, dx, log(dh), log(dw))]
    """
    boxes = boxes.astype(np.float32)
    # Convert to y, x, h, w
    height = boxes[:, 2] - boxes[:, 0]
    width = boxes[:, 3] - boxes[:, 1]
    center_y = boxes[:, 0] +0.5 * height
    center_x = boxes[:, 1] +0.5 * width
    # Apply deltas
    center_y += deltas[:, 0] * height
    center_x += deltas[:, 1] * width
    height *= np.exp(deltas[:, 2])
    width *= np.exp(deltas[:, 3])
    # Convert back to y1, x1, y2, x2
    y1 = center_y -0.5 * height
    x1 = center_x -0.5 * width
    y2 = y1 + height
    x2 = x1 + width
    return np.stack([y1, x1, y2, x2], axis =1)
def box_refinement_graph(box, gt_box):
    """Compute refinement needed to transform box to gt_box.
    box and gt_box are [N, (y1, x1, y2, x2)]
    """
    box = tf.cast(box, tf.float32)
    gt_box = tf.cast(gt_box, tf.float32)
    height = box[:, 2] - box[:, 0]
    width = box[:, 3] - box[:, 1]
    center_y = box[:, 0] +0.5 * height
    center_x = box[:, 1] +0.5 * width
    gt_height = gt_box[:, 2] - gt_box[:, 0]
    gt_width = gt_box[:, 3] - gt_box[:, 1]
    gt_center_y = gt_box[:, 0] +0.5 * gt_height
    gt_center_x = gt_box[:, 1] +0.5 * gt_width
    dy = (gt_center_y - center_y) / height
    dx = (gt_center_x - center_x) / width
    dh = tf.log(gt_height / height)
    dw = tf.log(gt_width / width)
    result = tf.stack([dy, dx, dh, dw], axis =1)
    return result
def box_refinement(box, gt_box):
    """Compute refinement needed to transform box to gt_box.
    box and gt_box are [N, (y1, x1, y2, x2)]. (y2, x2) is
    assumed to be outside the box.
    """
    box = box.astype(np.float32)
    gt_box = gt_box.astype(np.float32)
    height = box[:, 2] - box[:, 0]
    width = box[:, 3] - box[:, 1]
    center_y = box[:, 0] +0.5 * height
    center_x = box[:, 1] +0.5 * width
    gt_height = gt_box[:, 2] - gt_box[:, 0]
    gt_width = gt_box[:, 3] - gt_box[:, 1]
    gt_center_y = gt_box[:, 0] +0.5 * gt_height
    gt_center_x = gt_box[:, 1] +0.5 * gt_width
    dy = (gt_center_y - center_y) / height
    dx = (gt_center_x - center_x) / width
    dh = np.log(gt_height / height)
    dw = np.log(gt_width / width)
    return np.stack([dy, dx, dh, dw], axis =1)
############################################################
#  Dataset
############################################################
class Dataset(object):
    """The base class for dataset classes.
    To use it, create a new class that adds functions specific to the dataset
    you want to use. For example:
    class CatsAndDogsDataset(Dataset):
        def load_cats_and_dogs(self):
            ...
        def load_mask(self, image_id):
            ...
        def image_reference(self, image_id):
            ...
    See COCODataset and ShapesDataset as examples.
    """
    def __init__(self, class_map =None):
        self._image_ids = []
        self.image_info = []
        # Background is always the first class
        self.class_info = [{"source": "", "id": 0, "name": "BG"}]
        self.source_class_ids = {}
```

```
    def add_class(self, source, class_id, class_name):
        assert "."not in source, "Source name cannot contain a dot"
        # Does the class exist already?
        for info in self.class_info:
            if info['source'] == source and info["id"] == class_id:
                # source.class_id combination already available, skip
                return
        # Add the class
        self.class_info.append({
            "source": source,
            "id": class_id,
            "name": class_name,
        })
    def add_image(self, source, image_id, path, **kwargs):
        image_info = {
            "id": image_id,
            "source": source,
            "path": path,
        }
        image_info.update(kwargs)
        self.image_info.append(image_info)
    def image_reference(self, image_id):
        """Return a link to the image in its source Website or details about
        the image that help looking it up or debugging it.
        Override for your dataset, but pass to this function
        if you encounter images not in your dataset.
        """
        return ""
    def prepare(self, class_map =None):
        """Prepares the Dataset class for use.
        TODO: class map is not supported yet. When done, it should handle mapping
              classes from different datasets to the same class ID.
        """
        def clean_name(name):
            """Returns a shorter version of object names for cleaner display."""
            return ",".join(name.split(",")[:1])
        # Build (or rebuild) everything else from the info dicts.
        self.num_classes =len(self.class_info)
        self.class_ids = np.arange(self.num_classes)
        self.class_names = [clean_name(c["name"]) for c in self.class_info]
        self.num_images =len(self.image_info)
        self._image_ids = np.arange(self.num_images)
        self.class_from_source_map = {"{}.{}".format(info['source'], info['id']): id
                                      for info, id in zip(self.class_info, self.class_ids)}
        # Map sources to class_ids they support
        self.sources = list(set([i['source'] for i in self.class_info]))
        self.source_class_ids = {}
        # Loop over datasets
        for source in self.sources:
            self.source_class_ids[source] = []
            # Find classes that belong to this dataset
            for i, info in enumerate(self.class_info):
                # Include BG class in all datasets
                if i ==0 or source == info['source']:
                    self.source_class_ids[source].append(i)
    def map_source_class_id(self, source_class_id):
        """Takes a source class ID and returns the int class ID assigned to it.
        For example:
        dataset.map_source_class_id("coco.12 ") -> 23
        """
        return self.class_from_source_map[source_class_id]
    def get_source_class_id(self, class_id, source):
        """Map an internal class ID to the corresponding class ID in the source
dataset."""
        info =self.class_info[class_id]
        assert info['source'] == source
        return info['id']
    def append_data(self, class_info, image_info):
        self.external_to_class_id = {}
        for i, c in enumerate(self.class_info):
            for ds, id in c["map"]:
                self.external_to_class_id[ds +str(id)] = i
        # Map external image IDs to internal ones.
        self.external_to_image_id = {}
        for i, info in enumerate(self.image_info):
            self.external_to_image_id[info["ds"] +str(info["id"])] = i
    @property
    def image_ids(self):
        return self._image_ids
    def source_image_link(self, image_id):
        """Returns the path or URL to the image.
        Override this to return a URL to the image if it's availble online for easy
        debugging.
```

```
        """
        return self.image_info[image_id]["path"]
    def load_image(self, image_id):
        """Load the specified image and return a [H,W,3] Numpy array.
        """
        # Load image
        image = skimage.io.imread(self.image_info[image_id]['path'])
        # If grayscale. Convert to RGB for consistency.
        if image.ndim !=3:
            image = skimage.color.gray2rgb(image)
        return image
    def load_mask(self, image_id):
        """Load instance masks for the given image.
        Different datasets use different ways to store masks. Override this
        method to load instance masks and return them in the form of am
        array of binary masks of shape [height, width, instances].
        Returns:
            masks: A bool array of shape [height, width, instance count] with
                a binary mask per instance.
            class_ids: a 1D array of class IDs of the instance masks.
        """
        # Override this function to load a mask from your dataset.
        # Otherwise, it returns an empty mask.
        mask = np.empty([0, 0, 0])
        class_ids = np.empty([0], np.int32)
        return mask, class_ids
def resize_image(image, min_dim =None, max_dim =None, padding =False):
    """
    Resizes an image keeping the aspect ratio.
    min_dim: if provided, resizes the image such that it's smaller
        dimension == min_dim
    max_dim: if provided, ensures that the image longest side doesn't
        exceed this value.
    padding: If true, pads image with zeros so it's size is max_dim x max_dim
    Returns:
    image: the resized image
    window: (y1, x1, y2, x2). If max_dim is provided, padding might
        be inserted in the returned image. If so, this window is the
        coordinates of the image part of the full image (excluding
        the padding). The x2, y2 pixels are not included.
    scale: The scale factor used to resize the image
    padding: Padding added to the image [(top, bottom), (left, right), (0, 0)]
    """
    # Default window (y1, x1, y2, x2) and default scale == 1.
    h, w = image.shape[:2]
    window = (0, 0, h, w)
    scale =1
    # Scale?
    if min_dim:
        # Scale up but not down
        scale = max(1, min_dim / min(h, w))
    # Does it exceed max dim?
    if max_dim:
        image_max = max(h, w)
        if round(image_max * scale) > max_dim:
            scale = max_dim / image_max
    # Resize image and mask
    if scale !=1:
        image = scipy.misc.imresize(
            image, (round(h * scale), round(w * scale)))
    # Need padding?
    if padding:
        # Get new height and width
        h, w = image.shape[:2]
        top_pad = (max_dim - h) //2
        bottom_pad = max_dim - h - top_pad
        left_pad = (max_dim - w) //2
        right_pad = max_dim - w - left_pad
        padding = [(top_pad, bottom_pad), (left_pad, right_pad), (0, 0)]
        image = np.pad(image, padding, mode ='constant', constant_values =0)
        window = (top_pad, left_pad, h + top_pad, w + left_pad)
    return image, window, scale, padding
def resize_mask(mask, scale, padding):
    """Resizes a mask using the given scale and padding.
    Typically, you get the scale and padding from resize_image() to
    ensure both, the image and the mask, are resized consistently.
    scale: mask scaling factor
    padding: Padding to add to the mask in the form
            [(top, bottom), (left, right), (0, 0)]
    """
    h, w = mask.shape[:2]
    mask = scipy.ndimage.zoom(mask, zoom =[scale, scale, 1], order =0)
    mask = np.pad(mask, padding, mode ='constant', constant_values =0)
```

```
    return mask
def minimize_mask(bbox, mask, mini_shape):
    """Resize masks to a smaller version to cut memory load.
    Mini-masks can then resized back to image scale using expand_masks()
    See inspect_data.ipynb notebook for more details.
    """
    mini_mask = np.zeros(mini_shape + (mask.shape[-1],), dtype =bool)
    for i in range(mask.shape[-1]):
        m = mask[:, :, i]
        y1, x1, y2, x2 = bbox[i][:4]
        m = m[y1:y2, x1:x2]
        if m.size ==0:
            raise Exception("Invalid bounding box with area of zero")
        m = scipy.misc.imresize(m.astype(float), mini_shape, interp ='bilinear')
        mini_mask[:, :, i] = np.where(m >=128, 1, 0)
    return mini_mask
def expand_mask(bbox, mini_mask, image_shape):
    """Resizes mini masks back to image size. Reverses the change
    of minimize_mask().
    See inspect_data.ipynb notebook for more details.
    """
    mask = np.zeros(image_shape[:2] + (mini_mask.shape[-1],), dtype =bool)
    for i in range(mask.shape[-1]):
        m = mini_mask[:, :, i]
        y1, x1, y2, x2 = bbox[i][:4]
        h = y2 - y1
        w = x2 - x1
        m = scipy.misc.imresize(m.astype(float), (h, w), interp ='bilinear')
        mask[y1:y2, x1:x2, i] = np.where(m >=128, 1, 0)
    return mask
# TODO: Build and use this function to reduce code duplication
def mold_mask(mask, config):
    pass
def unmold_mask(mask, bbox, image_shape):
    """Converts a mask generated by the neural network into a format similar
    to it's original shape.
    mask: [height, width] of type float. A small, typically 28x28 mask.
    bbox: [y1, x1, y2, x2]. The box to fit the mask in.
    Returns a binary mask with the same size as the original image.
    """
    threshold =0.5
    y1, x1, y2, x2 = bbox
    mask = scipy.misc.imresize(
        mask, (y2 - y1, x2 - x1), interp ='bilinear').astype(np.float32) /255.0
    mask = np.where(mask >= threshold, 1, 0).astype(np.uint8)
    # Put the mask in the right location.
    full_mask = np.zeros(image_shape[:2], dtype =np.uint8)
    full_mask[y1:y2, x1:x2] = mask
    return full_mask
############################################################
#  Anchors
############################################################
def generate_anchors(scales, ratios, shape, feature_stride, anchor_stride):
    """
    scales: 1D array of anchor sizes in pixels. Example: [32, 64, 128]
    ratios: 1D array of anchor ratios of width/height. Example: [0.5, 1, 2]
    shape: [height, width] spatial shape of the feature map over which
           to generate anchors.
    feature_stride: Stride of the feature map relative to the image in pixels.
    anchor_stride: Stride of anchors on the feature map. For example, if the
        value is 2 then generate anchors for every other feature map pixel.
    """
    # Get all combinations of scales and ratios
    scales, ratios = np.meshgrid(np.array(scales), np.array(ratios))
    scales = scales.flatten()
    ratios = ratios.flatten()
    # Enumerate heights and widths from scales and ratios
    heights = scales / np.sqrt(ratios)
    widths = scales * np.sqrt(ratios)
    # Enumerate shifts in feature space
    shifts_y = np.arange(0, shape[0], anchor_stride) * feature_stride
    shifts_x = np.arange(0, shape[1], anchor_stride) * feature_stride
    shifts_x, shifts_y = np.meshgrid(shifts_x, shifts_y)
    # Enumerate combinations of shifts, widths, and heights
    box_widths, box_centers_x = np.meshgrid(widths, shifts_x)
    box_heights, box_centers_y = np.meshgrid(heights, shifts_y)
    # Reshape to get a list of (y, x) and a list of (h, w)
    box_centers = np.stack(
        [box_centers_y, box_centers_x], axis=2).reshape([-1, 2])
    box_sizes = np.stack([box_heights, box_widths], axis =2).reshape([-1, 2])
    # Convert to corner coordinates (y1, x1, y2, x2)
    boxes = np.concatenate([box_centers -0.5 * box_sizes,
                            box_centers +0.5 * box_sizes], axis =1)
```

```
    return boxes
def generate_pyramid_anchors(scales, ratios, feature_shapes, feature_strides,
                             anchor_stride):
    """Generate anchors at different levels of a feature pyramid. Each scale
    is associated with a level of the pyramid, but each ratio is used in
    all levels of the pyramid.
    Returns:
    anchors: [N, (y1, x1, y2, x2)]. All generated anchors in one array. Sorted
        with the same order of the given scales. So, anchors of scale[0] come
        first, then anchors of scale[1], and so on.
    """
    # Anchors
    # [anchor_count, (y1, x1, y2, x2)]
    anchors = []
    for i in range(len(scales)):
        anchors.append(generate_anchors(scales[i], ratios, feature_shapes[i],
                                        feature_strides[i], anchor_stride))
    return np.concatenate(anchors, axis =0)
############################################################
#  Miscellaneous
############################################################
def trim_zeros(x):
    """It's common to have tensors larger than the available data and
    pad with zeros. This function removes rows that are all zeros.
    x: [rows, columns].
    """
    assert len(x.shape) ==2
    return x[~np.all(x ==0, axis =1)]
def compute_ap(gt_boxes, gt_class_ids, gt_masks,
               pred_boxes, pred_class_ids, pred_scores, pred_masks,
               iou_threshold=0.5):
    """Compute Average Precision at a set IoU threshold (default 0.5).
    Returns:
    mAP: Mean Average Precision
    precisions: List of precisions at different class score thresholds.
    recalls: List of recall values at different class score thresholds.
    overlaps: [pred_boxes, gt_boxes] IoU overlaps.
    """
    # Trim zero padding and sort predictions by score from high to low
    # TODO: cleaner to do zero unpadding upstream
    gt_boxes = trim_zeros(gt_boxes)
    gt_masks = gt_masks[..., :gt_boxes.shape[0]]
    pred_boxes = trim_zeros(pred_boxes)
    pred_scores = pred_scores[:pred_boxes.shape[0]]
    indices = np.argsort(pred_scores)[::-1]
    pred_boxes = pred_boxes[indices]
    pred_class_ids = pred_class_ids[indices]
    pred_scores = pred_scores[indices]
    pred_masks = pred_masks[..., indices]
    # Compute IoU overlaps [pred_masks, gt_masks]
    overlaps = compute_overlaps_masks(pred_masks, gt_masks)
    # Loop through ground truth boxes and find matching predictions
    match_count =0
    pred_match = np.zeros([pred_boxes.shape[0]])
    gt_match = np.zeros([gt_boxes.shape[0]])
    for i in range(len(pred_boxes)):
        # Find best matching ground truth box
        sorted_ixs = np.argsort(overlaps[i])[::-1]
        for j in sorted_ixs:
            # If ground truth box is already matched, go to next one
            if gt_match[j] ==1:
                continue
            # If we reach IoU smaller than the threshold, end the loop
            iou = overlaps[i, j]
            if iou < iou_threshold:
                break
            # Do we have a match?
            if pred_class_ids[i] == gt_class_ids[j]:
                match_count +=1
                gt_match[j] =1
                pred_match[i] =1
                break
    # Compute precision and recall at each prediction box step
    precisions = np.cumsum(pred_match) / (np.arange(len(pred_match)) +1)
    recalls = np.cumsum(pred_match).astype(np.float32) /len(gt_match)
    # Pad with start and end values to simplify the math
    precisions = np.concatenate([[0], precisions, [0]])
    recalls = np.concatenate([[0], recalls, [1]])
    # Ensure precision values decrease but don't increase. This way, the
    # precision value at each recall threshold is the maximum it can be
    # for all following recall thresholds, as specified by the VOC paper.
    for i in range(len(precisions) -2, -1, -1):
        precisions[i] = np.maximum(precisions[i], precisions[i +1])
```

```
    # Compute mean AP over recall range
    indices = np.where(recalls[:-1] != recalls[1:])[0] +1
    mAP = np.sum((recalls[indices] - recalls[indices -1]) *
                 precisions[indices])
    return mAP, precisions, recalls, overlaps
def compute_recall(pred_boxes, gt_boxes, iou):
    """Compute the recall at the given IoU threshold. It's an indication
    of how many GT boxes were found by the given prediction boxes.
    pred_boxes: [N, (y1, x1, y2, x2)] in image coordinates
    gt_boxes: [N, (y1, x1, y2, x2)] in image coordinates
    """
    # Measure overlaps
    overlaps = compute_overlaps(pred_boxes, gt_boxes)
    iou_max = np.max(overlaps, axis =1)
    iou_argmax = np.argmax(overlaps, axis =1)
    positive_ids = np.where(iou_max >= iou)[0]
    matched_gt_boxes = iou_argmax[positive_ids]
    recall =len(set(matched_gt_boxes)) / gt_boxes.shape[0]
    return recall, positive_ids
# ## Batch Slicing
# Some custom layers support a batch size of 1 only, and require a lot of work
# to support batches greater than 1. This function slices an input tensor
# across the batch dimension and feeds batches of size 1. Effectively,
# an easy way to support batches > 1 quickly with little code modification.
# In the long run, it's more efficient to modify the code to support large
# batches and getting rid of this function. Consider this a temporary solution
def batch_slice(inputs, graph_fn, batch_size, names =None):
    """Splits inputs into slices and feeds each slice to a copy of the given
    computation graph and then combines the results. It allows you to run a
    graph on a batch of inputs even if the graph is written to support one
    instance only.
    inputs: list of tensors. All must have the same first dimension length
    graph_fn: A function that returns a TF tensor that's part of a graph.
    batch_size: number of slices to divide the data into.
    names: If provided, assigns names to the resulting tensors.
    """
    if not isinstance(inputs, list):
        inputs = [inputs]
    outputs = []
    for i in range(batch_size):
        inputs_slice = [x[i] for x in inputs]
        output_slice = graph_fn(*inputs_slice)
        if not isinstance(output_slice, (tuple, list)):
            output_slice = [output_slice]
        outputs.append(output_slice)
    # Change outputs from a list of slices where each is
    # a list of outputs to a list of outputs and each has
    # a list of slices
    outputs = list(zip(*outputs))
    if names is None:
        names = [None] *len(outputs)
    result = [tf.stack(o, axis =0, name =n)
              for o, n in zip(outputs, names)]
    if len(result) ==1:
        result = result[0]
    return result
def download_trained_weights(coco_model_path, verbose =1):
    """Download COCO trained weights from Releases.
    coco_model_path: local path of COCO trained weights
    """
    if verbose >0:
        print("Downloading pretrained model to "+ coco_model_path +" ...")
    with urllib.request.urlopen(COCO_MODEL_URL) as resp, open(coco_model_path, 'wb') as
out:
        shutil.copyfileobj(resp, out)
    if verbose >0:
        print("... done downloading pretrained model!")
```

- 추론 결과 시각화 코드

visualize.py

```
"""
Mask R-CNN
Display and Visualization Functions.
Copyright (c) 2017 Matterport, Inc.
Licensed under the MIT License (see LICENSE for details)
Written by Waleed Abdulla
"""
import math
import os
import random
```

```
import itertools
import colorsys
import numpy as np
from skimage.measure import find_contours
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import matplotlib.lines as lines
from matplotlib.patches import Polygon
from decimal import Decimal
import IPython.display
import cv2
from pixel2area import modul
from gps import start
from gps import rotate
from mrcnn import utils
import pymysql
Path = os.getcwd()
Path = Path.split('/')
Path = Path[:-2]
Path ='/'.join(Path)
ORI_DIR = Path
ed = (0, 0, 255)
green = (0, 255, 0)
blue = (255, 0, 0)
white = (255, 255, 255)
yellow = (0, 255, 255)
cyan = (255, 255, 0)
magenta = (255, 0, 255)
############################################################
#  Visualization
############################################################
def random_colors(N, bright =True):
    """
    Generate random colors.
    To get visually distinct colors, generate them in HSV space then
    convert to RGB.
    params :
        N : object number
        bright : Default(True)
    return :
        colors (list) : colors list
    """
    brightness =1.0 if bright else 0.7
    hsv = [(i / N, 1, brightness) for i in range(N)]
    colors = list(map(lambda c: colorsys.hsv_to_rgb(*c), hsv))
    random.shuffle(colors)
    return colors
def apply_mask(image, mask, color, road, alpha =0.5):
    """Apply the given mask to the image.
    params :
        image : aerial image
        mask : pothole area mask
        color : random color value
        road : road area mask
        alpha (float) : 0.5
    return :
        image : masked aerial image
    """
    for c in range(3):
        image[:, :, c] = np.where(mask ==1,
                                  image[:, :, c] *
                                  (1 - alpha) + alpha * color[c] *255,
                                  image[:, :, c])
    return image
def apply_mask_binary(image, mask, road =None, flag =1):
    """Apply the given mask to the image.
    params :
        image : aerial image
        mask : pothole area mask
        road : default(None) road area make
        flag : 1 is pothole and 0 is road
    return :
        crop_3 : binary mask aerial image
    """
    for c in range(3):
        image[:, :, c] = np.where(mask ==1,
                                  255,
                                  image[:, :, c])
        image[:, :, c] = np.where(mask ==0,
                                  0,
                                  image[:, :, c])
        if flag ==0:
            pass
```

```
        else:
            image[:, :, c] = np.where(road ==0,
                                      0,
                                      image[:, :, c])
    crop_3 = image.astype(np.uint8).copy()
    # crop_2 = cv2.cvtColor(crop, cv2.COLOR_BGR2RGB)
    crop_2 = cv2.cvtColor(crop_3, cv2.COLOR_BGR2RGB)
    return crop_2
def contour_test(image):
    '''
    search contours in binary image
    param
        image : binary image
    return
        x, y, w, h : x, y is bounding rectangle point w, h is bounding rectangle width and
height
    '''
    imgray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    ret, thr = cv2.threshold(imgray, 127, 255, 0)
    contours, _ = cv2.findContours(thr, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    if not contours:
        x =0
        y =0
        w =0
        h =0
        pass
    else:
        contour = contours[0]
        x, y, w, h = cv2.boundingRect(contour)
    return x, y, w, h
def display_road(image, boxes, masks, class_ids):
    '''
    be called in inference_road.py and save road image in aerial image
    params
        path : "media/" + "aerial image name"
        image_name : image name
        image : aerial image
        boxes : rois in image
        masks : masks in image
        class_ids : class_ids in image
    '''
    # Confirm instances exist
    N = boxes.shape[0]
    if not N:
        print("\n*** No road instances to display *** \n")
    else:
        assert boxes.shape[0] == masks.shape[-1] == class_ids.shape[0]
    # Generate random colors
    colors = random_colors(N)
    road_list = []
    root_image = image.astype(np.uint8).copy()
    root_image = cv2.cvtColor(root_image, cv2.COLOR_BGR2RGB)
    # Except to make directory
    # N is instance number
    for i in range(N):
        color = colors[0]
        # Bounding box
        if not np.any(boxes[i]):
            # Skip this instance. Has no bbox. Likely lost in image cropping.
            continue
        # boxes's points
        y1, x1, y2, x2 = boxes[i]
        file_name =str(y1) +"_"+str(y1 + (y2 - y1)) +"_"+str(x1) +"_"+str(x1 + (x2 - x1))
+".jpg"
        mask = masks[:, :, i]
        # image copy
        binary_image = image.astype(np.uint8).copy()
        # make road's binary mask
        binary_image = apply_mask_binary(binary_image, mask, flag =0)  ## binary mask
        # binary roads contour images append
        road_list.append(binary_image)
    # Merge binary roads contour
    road_image = merge_road(road_list, root_image)
    return road_image
def merge_road(road_list, root_image):
    '''
    merge road contours image and if road_list don't have value
    param
        road_list : road contours list
        path : media link
        image_name : aerial image's name
        root_image : original image
    '''
```

```
177     if not road_list:
178         whites = np.ones(root_image.shape[0:2], dtype ="uint8") *255
179         return whites
180         # cv2.imwrite(path + '/binary_mask_road/' + image_name, root_image)
181         # cv2.imwrite(path + '/binary_mask_road/' + "result" + image_name, whites)
182     else:
183         for index, road in enumerate(road_list):
184             if index ==0:
185                 temp = road
186             else:
187                 temp = cv2.bitwise_or(temp, road)
188         result = temp
189         result = cv2.cvtColor(result, cv2.COLOR_BGR2GRAY)
190         print(result.shape)
191         return result
192         # cv2.imwrite(path + '/binary_mask_road/' + "result" + image_name, result)
193         # cv2.imwrite(path + '/binary_mask_road/' + image_name, temp)
194 def crop_image(ori_crop, image, cx, cy, w, h):
195     """crop a pothole in image
196         params
197             ori_crop : aerial image and masked image
198             image : original image
199             cx : x point
200             cy : y point
201             w : width
202             h : height
203         Retrun
204             ROI : ROI(crop) area
205     """
206     if cy <=50 or cx <=50:
207         ROI = ori_crop[cy:cy + h +5, cx:cx + w +5]
208     elif cy >= image.shape[0] -50 or cx >= image.shape[1] -50:
209         ROI = ori_crop[cy -50:cy + h, cx -50:cx + w]
210     else:
211         ROI = ori_crop[cy -50:cy + h +50, cx -50:cx + w +50]
212     return ROI
213 def make_image(based_image, image, cx, cy, w, h):
214     """make crop image
215         param
216             based_image : aerial image and masked image
217             image : original image
218             cx : x point
219             cy : y point
220             w : width
221             h: height
222         Return
223             ROI : ROI(crop) area
224     """
225     based_image = cv2.cvtColor(based_image, cv2.COLOR_BGR2RGB)
226     # based_image = cv2.bitwise_or(based_image, image)
227     ROI = crop_image(based_image, image, cx, cy, w, h)
228     # ori_ROI = cv2.cvtColor(ROI, cv2.COLOR_BGR2BGRA)
229     return ROI
230 def display_instances(path, image_id, count, image, boxes, masks, class_ids, road_image):
231     """pothole mask and post processing in inference result image
232         param :
233             path (string) : image directory (in pothole, mask_pothole, ...)
234             image_id : image_id in blog_image_video table
235             count : image name
236             image (ndarray) : aerial image
237             boxes : r['rois'] bounding box ,[num_instance, (y1, x1, y2, x2, class_id)] in
238 image coordinates.
239             masks : r['masks'] mask , [height, width, num_instances]
240             class_ids : r['class_ids'] class index, [num_instances]
241     """
242     # db connect
243     pothole_db = pymysql.connect(
244         host='localhost',
245         user='root',
246         password='20121208',
247         db='test_server',
248         charset='utf8'
249     )
250     cursor = pothole_db.cursor(pymysql.cursors.DictCursor)
251     # Number of instances
252     N = boxes.shape[0]
253     if not N:
254         print("\n*** No pothole instances to display *** \n")
255     else:
256         assert boxes.shape[0] == masks.shape[-1] == class_ids.shape[0]
257     # Generate random colors
258     colors = random_colors(N)
259     angle =0
```

```
    # make directory if not exist
    try:
        if not (os.path.isdir(path)):
            os.makedirs(os.path.join(path, 'pothole'))
            os.makedirs(os.path.join(path, 'mask_pothole'))
            os.makedirs(os.path.join(path, 'ruler_pothole'))
    except OSError as e:
        if e.errno != errno.EEXIST:
            print("Failed to create directory!!!!!")
            raise
    # instance number
    for i in range(N):
        color = colors[0]
        # Bounding box
        if not np.any(boxes[i]):
            # Skip this instance. Has no bbox. Likely lost in image cropping.
            continue
        y1, x1, y2, x2 = boxes[i]
        file_name =str(y1) +"_"+str(y1 + (y2 - y1)) +"_"+str(x1) +"_"+str(x1 + (x2 - x1))
+".jpg"
        road = road_image
        # Mask
        mask = masks[:, :, i]
        masked_image = image.astype(np.uint8).copy()
        masked_image = apply_mask(masked_image, mask, color, road)  ## mask image in road
delete
        masked_image = masked_image.astype(np.uint8).copy()
        binary_image = image.astype(np.uint8).copy()
        binary_image = apply_mask_binary(binary_image, mask, road)  ## binary mask
        binary_image = binary_image.astype(np.uint8).copy()
        root_image = image.astype(np.uint8).copy()
        images_list = [root_image, masked_image, binary_image]
        cx, cy, w, h = contour_test(binary_image)  ## contour's x,y and width, height
        if cx ==0 and cy ==0 and w ==0 and h ==0:
            pass
        else:
            # crop pothole image output
            for i, image_idx in enumerate(images_list):
                image_result = make_image(image_idx, image, cx, cy, w, h)
                ####### original pothole image output
                if i ==0:
                    cv2.imwrite(path +'/pothole/'+ file_name, image_result)
                ####### mask pothole image output
                elif i ==1:
                    cv2.imwrite(path +'/mask_pothole/'+ file_name, image_result)
            ####### ruler line write (ruler body make)
            ruler = cv2.line(root_image, (cx, cy), (cx, cy + h), (0, 0, 0), 2)
            ruler = cv2.line(ruler, (cx, cy), (cx, cy + h), (255, 255, 255), 1)
            ruler = cv2.line(ruler, (cx, cy), (cx + w, cy), (0, 0, 0), 2)
            ruler = cv2.line(ruler, (cx, cy), (cx + w, cy), (255, 255, 255), 1)
            # Mask Polygon
            # Pad to ensure proper polygons for masks that touch image edges.
            padded_mask = np.zeros((mask.shape[0] +2, mask.shape[1] +2), dtype =np.uint8)
#
            padded_mask[1:-1, 1:-1] = mask
            contours = find_contours(padded_mask, 0.5)
            large_area =0
            for verts in contours:
                # Subtract the padding and flip (y, x) to (x, y)
                c = np.expand_dims(verts.astype(np.float32), 1)
                # Convert it to UMat object
                c = cv2.UMat(c)
                area = cv2.contourArea(c)
                if large_area < area:
                    large_area = area
            name = count.split('.')[0]
            folder_name = os.path.join(ORI_DIR, 'media', count.split('.')[0])
            tile_location    =    os.path.join(ORI_DIR,    'media',    count.split('.')[0],
'pothole', file_name)
            # modul.operation(ori,image_path,pixel,cmosW,cmosH,altitude)
            area_result,  pixel_area,  relative_altitude  =  modul.operation(path  +'.JPG',
path +'.JPG', int(large_area))
            print(relative_altitude)
            ######################ruler code###################################
            real_width = float("%0.2f" % (math.sqrt(pixel_area) * w))
            real_height = float("%0.2f" % (math.sqrt(pixel_area) * h))
            modular_width = (real_width - (real_width % 10)) /10
            modular_height = (real_height - (real_height % 10)) /10
            for i in range(1, int(modular_width +1)):
                check_point = i *10
                check_point =int(check_point / math.sqrt(pixel_area))
                ruler = cv2.line(ruler, (cx + check_point, cy -3), (cx + check_point, cy
+3), (255, 255, 255), 2)
```

```
343                 ruler = cv2.line(ruler, (cx + check_point, cy -3), (cx + check_point, cy
344 +3), (0, 0, 0), 1)
345             for i in range(1, int(modular_height +1)):
346                 check_point = i *10
347                 check_point =int(check_point / math.sqrt(pixel_area))
348                 ruler  =  cv2.line(ruler,  (cx  -3,  cy  +  check_point),  (cx  +3,  cy  +
349 check_point), (255, 255, 255), 2)
350                 ruler  =  cv2.line(ruler,  (cx  -3,  cy  +  check_point),  (cx  +3,  cy  +
351 check_point), (0, 0, 0), 1)
352             ruler = make_image(ruler, image, cx, cy, w, h)
353             cv2.imwrite(path +'/ruler_pothole/'+ file_name, ruler)
354             # here
355             ######################################################################3
356             # if os.path.isfile(path+'/pothole/'+file_name):
357             #                     edge_area    =    edge.ckdEdge(path+'/pothole/'+file_name,
358 path+'/binary_mask_pothole/'+file_name, path+'/edge_pothole/')
359             #     print(edge_area)
360             #  prepr_area_result  =  modul.operation(path  +  '.JPG',  path  +  '.JPG',
361 int(area_result))
362             prepr_area_result =0
363             area_result = round(area_result, 2)
364             ######################################################################
365             pothole_lat, pothole_long = start.operation(float(area_result), path +'.JPG',
366 tile_location,
367                                                         folder_name,
368                                                         int((x1  +  x2)  /2),  int((y1  +
369 y2) /2), name)
370             sql   ="INSERT   INTO   blog_tile   (_object,   origin_image,   tile_location,
371 axis_x,axis_y,accuracy,pothole_size, real_width, real_height, prepr_pothole_size, " \
372                    "pothole_lat, pothole_long," \
373                    "img_show,axis_w,axis_h,m_tile_location,ruler_image)              VALUES
374 (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
375             val = (
376             1,  image_id['id'],  os.path.join('media',  count.split('.')[0],  'pothole',
377 file_name), int(x1), int(y1),
378             Decimal(area_result), 0, real_width, real_height, Decimal(prepr_area_result),
379 pothole_lat, pothole_long, 0,
380             int(x1 + (x2 - x1)), int(y1 + (y2 - y1)),
381             os.path.join('media', count.split('.')[0], 'mask_pothole', file_name),
382             os.path.join('media', count.split('.')[0], 'ruler_pothole', file_name))
383             cursor.execute(sql, val)
384             pothole_db.commit()
385     if not N:
386         folder_name = os.path.join(ORI_DIR, 'media', count.split('.')[0])
387         path_nonpothole = path.split('/')[-1]
388         rotated = []
389         rotate.rotate_nonpothole(rotated, path +'.JPG', folder_name, path_nonpothole)
390     sql_2 ="UPDATE blog_image_video SET gps_image=%s WHERE name=%s"
391     val_2     =     (os.path.join('media',     count.split('.')[0],     count.split('.')[0]
392 +"rotate_check"+'.png'),
393              os.path.join(path.split('/')[-1] +'.JPG'))
394     cursor.execute(sql_2, val_2)
395     pothole_db.commit()
396     plt.close()
397     cursor.close()
398     # ax.imshow(masked_image.astype(np.uint8))
399     # plt.savefig(path+"/%s" % (str(count[1:50])),bbox_inches='tight',pad_inches=0.0)
```

– 이미지 헤더 정보를 이용한 면적 계산 프로그램

modul.py

```
 1 import imageNameRead
 2 import binarizationAndPixelCount as bap
 3 import math
 4 import cv2
 5 def myFormat(k):
 6     '''
 7     소수점 둘째자리까지만 남긴다
 8     :param k: 전달된 값
 9     :return a/100: 소수점 둘째자리 이후 제거된 값
10     '''
11     a = k *100
12     a =int(a)
13     return a /100
14 def otsu(src):
15     '''
16     전달된 grayscale 영상을 otsu 이진화 기법을 통해 이진화한다.
17     :param src: grayscale 영상
18     :return: dst 이진화된 영상
19     '''
20     ret, dst = cv2.threshold(src, 0, 255, cv2.THRESH_BINARY +cv2.THRESH_OTSU)
```

```
    #dst = cv2.threshold(src, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    return dst
def operation(ori,image_path,pixel,cmosW,cmosH,altitude):
    '''
    :param      ori:      이진화된      마스크      영역의      영상의      경로      ex)
c:\\Desktop\\mask\\IX-12-82312_0101_0044.jpg"
    :param  image_path:  원본  영상의  경로,  원본  영상에서  exif  데이터를  추출한다
c:\\Desktop\\src\\IX-12-82312_0101_0044.jpg"
    :param pixel: 파라미터로 전달된 ori에서 명도값이 255인 픽셀 수를 센다
    :param cmosW: cmos의 폭(mm)
    :param cmosH: cmos의 높이mm)
    :param altitude: 영상이 촬영된 고도
    :return result: 마스크영역의 추정 면적
    '''
    name=[] #이미지 이름과 확장자 ex) DJI_045.jpg
    name2=[] #이미지 이름과 확장자 ex) DJI_045.JPG
    pixel=cv2.imread(image_path,cv2.IMREAD_GRAYSCALE)
    pixel=otsu(pixel)
    asd=cv2.imread(image_path,cv2.IMREAD_GRAYSCALE)
    rows,cols=asd.shape
    pixel=bap.pixelRead(pixel,rows,cols)
    cmosD=math.sqrt((cmosW **2)+(cmosH **2))
    cmosD=myFormat(cmosD)
    cmosD=cmosD /1000
    cmosW=cmosW /1000
    cmosH=cmosH /1000
    imageNameRead.nameRead(image_path,name)
    imageNameRead.nameRead2(image_path,name)
    results = bap.binarizationAndPixelCount(ori,pixel,name,cmosD,cmosW,cmosH,altitude)
    return results
```

binarizationAndPixelCount.py

```

import getCalData
def pixelRead(src,rows,cols):
    '''
    전달된 src 영상에서 명도값이 255인 픽셀 수를 센다.
    :param src: grayscale 영상
    :param rows: 영상의 행
    :param cols: 영상의 열
    :return:
    '''
    white=0
    black=0
    for i in range(0,rows):
        for j in range(0,cols):
            x=src[i,j]
            if x ==255:
                white+=1
    return white
def binarizationAndPixelCount(ori,pixel,name,cmosD,cmosW,cmosH,altitude):
    '''
    각 파라미터를 전달 받아 전달된 이진화된 영상 내의 명도값 255를 가지는 픽셀들의 면적을 계산한
다.
    :param      ori:      이진화된      마스크      영역의      영상의      경로      ex)
c:\\Desktop\\mask\\IX-12-82312_0101_0044.jpg"
    :param pixel: ori에서 명도값 255를 가지는 픽셀수
    :param name 영상의 이름
    :param cmosD: cmos의 대각 길이(mm)
    :param cmosW: cmos의 폭(mm)
    :param cmosH: cmos의 높이mm)
    :param altitude: 영상이 촬영된 고도
    :return result: 마스크영역의 추정 면적
    '''
    results=0
    for i in range(0, len(name)):
        count = pixel #operation(name[i])
        pixel = getCalData.getArea(ori, name[i], cmosD, cmosW, cmosH, altitude)
        print(count *pixel)

        results=count *pixel
    return results
```

getCalData.py

```
import math
from PIL import Image
from PIL.ExifTags import TAGS
def myFormat(k):
    '''
    소수점 둘째자리까지만 남긴다
```

```
    :param k: 전달된 값
    :return a/100: 소수점 둘째자리 이후 제거된 값
    '''
    a=k *100
    a=int(a)
    return a /100
def getArea(ori,name,cmosD,cmosW,cmosH,altitude):
    '''
    전달된 파라미터를 기반으로 한 픽셀이 가지는 단위 면적을 계산한다
    :param ori: 원본 영상
    :param name: 마스크 영상
    :param cmosD: cmos의 대각 길이(mm)
    :param cmosW: cmos의 폭(mm)
    :param cmosH: cmos의 높이mm)
    :param altitude: 영상이 촬영된 고도
    :return pixel: 한 픽셀의 단위 며적
    '''
    imgFile = Image.open(name)
    oriFile = Image.open(ori)
    info = oriFile._getexif()
    #print(info)
    imageW=0
    imageH=0
    fl=0
    z=0
    for tag, value in info.items():
        decoded = TAGS.get(tag, tag)
        #print(decoded, value)
        if 'FocalLength'in decoded:
            z +=1
            fl=value
        if 'ExifImageWidth'in decoded:
            z +=1
            imageW = value
        if 'ExifImageHeight'in decoded:
            z +=1
            imageH = value
        if z >=3:
            break
    fov = math.degrees(math.atan(cmosD *1000 / (2 * fl)))
    A = float(90 - fov)
    A=myFormat(A)
    c=myFormat(altitude *100 /int(math.sin(math.radians(A))*100))  # 66.48 to radian
    c *=2
    ratio = c / cmosD
    w = ratio * cmosW *100
    h = ratio * cmosH *100
    pixel = (w / imageW) * (h / imageH)
    return pixel
```

imageNameRead.py

```
'''
@param string path 이미지 폴더가 있는 경로
@param list name 이미지 파일의 이름을 저장
'''
def nameRead(image_path,name):
    if '.jpg'in image_path:
        name.append(image_path)
        print(name)
'''
@param string path 이미지 폴더가 있는 경로
@param list name 이미지 파일의 이름을 저장
'''
def nameRead2(image_path,name):
    if '.JPG'  in image_path:
        name.append(image_path)
        print(name)
```

- 항공 영상 기반 상대 위치 계산 프로그램

start.py

```
import meta
from PIL import Image, ExifTags
import rotate
import cv2
import math as m
def centerPos(rotated,cPos):
    '''
    회전된 원본 이미지의 중점 좌표 값을 찾는다
```

```
    :param rotated: 회전된 원본 이미지
    :param cPos: 원본이미지의 x,y 좌표를 저장할 리스트
    '''
    rotated = cv2.cvtColor(rotated, cv2.COLOR_BGR2GRAY)
    rows,cols=rotated.shape
    cPos.append(round(rows/2))
    cPos.append(round(cols/2))
def cal(cPos,potPos,result,loc):
    '''
    이미지 중점과 포트홀 과의 거리 차이를 계산하는 메소드
    :param cPos: 원본이미지의 좌표
    :param potPos: 포트홀 이미지의 좌표
    :param result: [0] 위도 차이
                   [1] 경도 차이
                   [2] 위도 차이를 km로 저장
                   [3] 경도 차이를 km로 저장
    :param loc: 포트홀이 있는 위치를 저장함
    :return 위치 문자열 : 객체가 있는 곳의 문자열을 리턴한다
    '''
    result.append(abs(cPos[0]-potPos[0]))
    result.append(abs(cPos[1]-potPos[1]))
    result.append(result[0]*0.82 /100000)
    result.append(result[1]*0.82 /100000)
    if cPos[0]>=potPos[0]:
        loc.append("위")
        if cPos[1]<=potPos[1]:
            loc.append("오른")
        else:
            loc.append("왼")
        return "위"
    else:
        loc.append("아래")
        if cPos[1]<=potPos[1]:
            loc.append("오른")
        else:
            loc.append("왼")
        return "아래"
#위도 경도 초 계산
def cal2(meta,di,sec):
    '''
    위도 경도의  초단위 값을 이용해서 중심부로 부터 객체가 얼마나 차이 나는지 계산한다.
    :param meta: 에미지의 메타데이터가 저장된 리스트
    :param di: 중점 기준 방향
    :param sec: 초 단위 값
    '''
    #위도부터 계산함
    if meta[0]=="N":
        cal2_N(meta[5],di,sec)
    else:
        cal2_S(meta[5],di,sec)
    #경도 계산
    cal2_longitude(meta[5],sec)
#북반구 초 계산
def cal2_N(lati,di,sec):
    '''
    영상을 촬영한 위치가 북반구 일 떄 초단위 값을 이용해서 중심부로 부터 객체가 얼마나 차이 나는
지 계산한다.
    :param lati: 위도
    :param di:  방향
    :param sec:  초단위 값
    '''
    if di =="위":
        x1=6378 * m.cos(m.radians(int(lati)))
        y1=6357 * m.sin(m.radians(int(lati)))
        x2=6378 * m.cos(m.radians(int(lati)+1))
        y2=6357 * m.sin(m.radians(int(lati)+1))
        a = x1 - x2
        b = y1 - y2
        c = m.sqrt((a * a) + (b * b))
        sec.append(c/3600)
    else:
        x1=6378 * m.cos(m.radians(int(lati)))
        y1=6357 * m.sin(m.radians(int(lati)))
        x2=6378 * m.cos(m.radians(int(lati)-1))
        y2=6357 * m.sin(m.radians(int(lati)-1))
        a = x1 - x2
        b = y1 - y2
        c = m.sqrt((a * a) + (b * b))
        #print(c/3600)
        sec.append(c/3600)
#남반구 초 계산
def cal2_S(lati,di,sec):
    '''
```

```
   영상을 촬영한 위치가 남반구 일 떄 초단위 값을 이용해서 중심부로 부터 객체가 얼마나 차이 나는지 계산한다.
   :param lati: 위도
   :param di:  방향
   :param sec:  초단위 값
   '''
    if di =="위":
        x1=6378 * m.cos(m.radians(int(lati)))
        y1=6357 * m.sin(m.radians(int(lati)))
        x2=6378 * m.cos(m.radians(int(lati)-1))
        y2=6357 * m.sin(m.radians(int(lati)-1))
        a = x1 - x2
        b = y1 - y2
        c = m.sqrt((a * a) + (b * b))
        sec.append(c/3600)
    else:
        #print(6378 * m.cos(m.radians(int(lati))))
        x1=6378 * m.cos(m.radians(int(lati)))
        y1=6357 * m.sin(m.radians(int(lati)))
        x2=6378 * m.cos(m.radians(int(lati)+1))
        y2=6357 * m.sin(m.radians(int(lati)+1))
        a = x1 - x2
        b = y1 - y2
        c = m.sqrt((a * a) + (b * b))
        sec.append(c/3600)
#경도 초 계산
def cal2_longitude(longi,sec):
    '''
    경도의 도분초 표기법의 초를 계산한다
    :param longi: 경도
    :param sec: 경도 초단위 값
    '''
    c=m.cos(m.radians(int(longi)))*2 *m.pi *6378 /360
    #print(c/3600)
    sec.append(c/3600)
#포트홀 좌표 출력
def getPotoleGps(image_meta,x,y,loc):
    '''
    객체의 좌표를 출력한다
    :param image_meta: 계산된 좌표 값을 담은 리스트
    :param x: 영장 중심부와 객체간의 차이나는 위도 값
    :param y: 영상 중심부와 객체간의 차이나는 경도 값
    :param loc: 영상 중심부 기준 좌표를 구할 객체가 위치한 곳 ex) 영상 중심부 기준 아래 오른
    :return latitude+longitude: 객체의 위치를 문자열의 형태로 리턴한다
    '''
    lati1=image_meta[5]
    lati2=image_meta[6]
    lati3=1
    lati4=image_meta[0]
    longi1=image_meta[8]
    longi2=image_meta[9]
    longi3=1
    longi4=image_meta[1]
    if loc[0]=="위":
        if image_meta[0]=="N":
            lati3=float(image_meta[7])+x
        else:
            lati3=float(image_meta[7])-x
    else:
        if image_meta[0]=="N":
            lati3=float(image_meta[7])-x
        else:
            lati3=float(image_meta[7])+x
    if loc[1]=="오른":
        if image_meta[1]=="E":
            longi3=float(image_meta[10])+y
        else:
            longi3=float(image_meta[10])-y
    else:
        if image_meta[0]=="E":
            longi3=float(image_meta[10])-y
        else:
            longi3=float(image_meta[10])+y
    latitude=lati1 +lati2 +str(lati3)+" "+lati4 +" "
    longitude=longi1 +longi2 +str(longi3)+" "+longi4
    return latitude +longitude
def pixelRead(src, potPos):
    '''
    객체의 위치를 담은 영상을 파라미터로 전달 받아 해당 객체의
    픽셀 위치를 potPos 리스트에 저장한다
    :param src: 상대좌표를 구할 객체가 있는 영상
    :param potPos: 빈 리스트
    '''
```

```
175     src = otsu(src)
176     rows, cols = src.shape
177     k =0
178     for i in range(0, rows):
179         if k ==1:
180             break
181         for j in range(0, cols):
182             x = src[i, j]
183             if x ==255:
184                 potPos.append(i)
185                 potPos.append(j)
186                 k =1
187                 break
188 def otsu(src):
189     '''
190     전달된 grayscale 영상을 otsu 이진화 기법을 통해 이진화한다.
191     :param src: grayscale 영상
192     :return: dst 이진화된 영상
193     '''
194     ret, dst = cv2.threshold(src, 0, 255, cv2.THRESH_BINARY +cv2.THRESH_OTSU)
195     return dst
196 def operation(unit,path1,path2,path3 ="none",name ="roated"):
197     '''
198     path2의 마스크 영역의 gps 값을 출력하는 메소드
199     :param unit: 영상 내에서 한 픽셀이 가지는 실제 단위 면적
200     :param path1: 원본 이미지 "E:\\source\\DJI_0086.JPG"
201     :param path2: 원본 이미지와 동일한 크기의 영상이나 상대 좌표를 구할 곳의 픽셀을 제외한 곳의
202 명도 값이 0이여아함
203     :param path3: 이미지를 저장할 경로 지정하지 않았다면 path1의 위치에 저장된다.
204     :param name: 이미지를 저장할 이름
205     '''
206     #이미지 회전 및 저장
207     rotated=[]
208     rotate.rotate(rotated,path1,path2,path3,name)
209     #회전된 이미지 중점 찾기
210     cPos=[]
211     potPos=[]
212     centerPos(rotated[0],cPos)
213     pixelRead(rotated[1],potPos)
214     print("이미지 중심 좌표",cPos[0],cPos[1])
215     print("임의 좌표",potPos[0],potPos[1])
216     distance=[]
217     loc=[]
218     #이미지 중심부와 포트홀의 거리 차이(픽셀) 확인
219     direction=cal(cPos,potPos,distance,loc)
220     print("포트홀은 중심부에서",loc[0],loc[1],"에 있음")
221     print("위도 차: ",distance[0],"pixel","경도 차",distance[1],"pixel")
222     print("위도 차",distance[0]*0.82 /100000,"km","경도 차",distance[1]*0.82 /100000,"km")
223     #print("위도 차",distance[2],"km","경도 차",distance[3],"km")
224
225     #이미지 읽어서 메타데이터 읽기
226     img = Image.open(path1)
227     img_exif = img.getexif()
228     image_meta=[]
229     meta.exifRead(img_exif,img,image_meta)
230     #위도 경도 초 단위 계산 그러니까 경도 위도 1초가 얼마인지 계산하는거임
231     sec=[]
232     cal2(image_meta,direction,sec)
233     #print(sec[0],sec[1])
234     #초차이 계산 중심부에서 포트홀과 얼마나 차이나는지 계싼하는거임
235     x=distance[2]/sec[0]
236     y=distance[3]/sec[1]
237     print("초차이",x,y)
238     print("중심부 gps: ",image_meta[4])
239     print("포트홀위치: ",getPotoleGps(image_meta,x,y,loc))
```

xmpRead.py

```
1  def xmpRead(path):
2      '''
3      이미지 내 XMP(Extensible Metadata Platform)을 읽어오기 위한 함수
4      해당 메소드는 yaw 데이터만 읽어온 후 값을 리턴한다
5      :param path: 이미지의 경로
6      :return: 해당 이미지의 yaw값
7      '''
8      #erros='ignore'을 안쓰면 에러 발생해서 코드가 안 읽힘
9      fd =open(path, encoding ="utf8", errors ='ignore')
10     d= fd.read()
11     xmp_start = d.find('<x:xmpmeta')
12     #비짠 드론 yaw 찾을 때
13     if d.find('Camera:Yaw') !=-1:
```

```
        xmp_yaw_start=d.find('Camera:Yaw')
        xmp_yaw_end=d.find('Camera:GPSXYAccuracy')
        xmp_end = d.find('</x:xmpmeta')
        xmp_str = d[xmp_start:xmp_end +13]
        #print(xmp_str)
        xmp_yaw_str=d[xmp_yaw_start:xmp_yaw_end]
        yaw=xmp_yaw_str.replace("Camera:Yaw=","")
        yaw = yaw[1:-2]
        #print(yaw)
        return yaw
    #싼 드론  yaw 찾을 때
    elif d.find('FlightYawDegree') !=-1:
        xmp_yaw_start=d.find('FlightYawDegree')
        xmp_yaw_end=d.find('drone-dji:FlightPitchDegree')
        xmp_end = d.find('</x:xmpmeta')
        xmp_str = d[xmp_start:xmp_end +13]
        #print(xmp_str)
        xmp_yaw_str=d[xmp_yaw_start:xmp_yaw_end]
        yaw=xmp_yaw_str.replace("FlightYawDegree=","")
        yaw = yaw[1:-2]
        if yaw.find('"')!=-1:
            yaw=yaw[:-3]
        #print(yaw)
        return yaw
```

meta.py

```
from PIL import Image, ExifTags
from PIL.ExifTags import TAGS
def exifRead(info,x,image_meta):
    '''
    이미지의 gps 정보를 읽어와 리스트에 저장하는 메소드
    :param info: 이미지 exif 정보
    :param x: 원본 이미지
    :param image_meta: 이미지의 메타데이터가 제장될 리스트
        0: North, South
        1: East, West
        2: 위도
        3: 경도
        4: 이미지 중점 gps 4번 값이 이미지 중점의 gps 좌표 값
        5: 위도의 도만
        6: 위도의 분만
        7: 위도의 초만
        8: 경도의 도만
        9: 경도의 분만
        10: 경도의 초만
    '''
    for tag,value in info.items():
        decoded=TAGS.get(tag,tag)

        if 'GPSInfo'in decoded:
            #print(value.get(1)[0])
            #print(value.get(3)[0])
            #print(value.get(2))
            x1=str(int(value.get(2)[0]))+" "
            x2=str(int((value.get(2)[1])))+" "
            x3=str((value.get(2)[2]))
            x=x1 +x2 +x3
            #x=round(x,2)
            #print(value.get(4))
            y1=str(int(value.get(4)[0]))+" "
            y2=str(int(value.get(4)[1]))+" "
            y3=str(value.get(4)[2])
            y=y1 +y2 +y3
            #print(x,y)
            image_meta.append(value.get(1)[0])
            image_meta.append(value.get(3)[0])
            image_meta.append(x)
            image_meta.append(y)
            image_meta.append(x+" "+value.get(1)[0]+" "+y +" "+value.get(3)[0])
            image_meta.append(x1)
            image_meta.append(x2)
            image_meta.append(x3)
            image_meta.append(y1)
            image_meta.append(y2)
            image_meta.append(y3)
```

rotate.py

```
import xmpRead
```

```
import cv2
import numpy as np
import imutils
def otsu(src):
    '''
    전달된 grayscale 영상을 otsu 이진화 기법을 통해 이진화한다.
    :param src: grayscale 영상
    :return: dst 이진화된 영상
    '''
    ret, dst = cv2.threshold(src, 0, 255, cv2.THRESH_BINARY +cv2.THRESH_OTSU)
    return dst
def rotate_bound(image, angle,h,w):
    '''
    imutils.rotate_bound와 동일하게 동작하나 이미지의 height, width만 추가적으로
    입력받는 점이 다릅니다
    :param image: 회전할 영상
    :param angle: 회전할 각도
    :param h: height
    :param w: widht
    :return cv2.warpAffine(image, M, (nW, nH)): 회전된 영상
    '''
    # grab the dimensions of the image and then determine the
    # center
    (cX, cY) = (w /2, h /2)
    # grab the rotation matrix (applying the negative of the
    # angle to rotate clockwise), then grab the sine and cosine
    # (i.e., the rotation components of the matrix)
    M = cv2.getRotationMatrix2D((cX, cY), -angle, 1.0)
    #cv2.imshow("as",M)
    #cv2.waitKey(0)
    cos = np.abs(M[0, 0])
    sin = np.abs(M[0, 1])
    # compute the new bounding dimensions of the image
    nW =int((h * sin) + (w * cos))
    nH =int((h * cos) + (w * sin))
    # adjust the rotation matrix to take into account translation
    M[0, 2] += (nW /2) - cX
    M[1, 2] += (nH /2) - cY
    # perform the actual rotation and return the image
    return cv2.warpAffine(image, M, (nW, nH))
def rotate(rotated,path1,path2,path3,name):
    '''
    :param rotated: 빈 리스트
    :param path1: 읽어올 이미지의 경로 ex) "E:\\source\\DJI_0086.JPG"
    :param path2: 포트홀이 이진화된 이미지
    :param path3: 이미지를 저장할 경로 ex) "E:\\source\\
    :param name: 저장될 이미지의 이름 ex) "저장용"
    '''
    if path2 =="none":
        path2=path1

    angle=xmpRead.xmpRead(path1)
    #print(type(angle))
    angle=angleConvert(angle)
    image = cv2.imread(path1)
    image2= cv2.imread(path2,cv2.IMREAD_GRAYSCALE)
    image2=otsu(image2)
    rotated.append(imutils.rotate_bound(image, angle))
    (h, w) = image.shape[:2]
    rotated.append(rotate_bound(image2, angle,h,w))
    cv2.imwrite(path3+name +".JPG", rotated[0])
def angleConvert(angle):
    '''
    xmp 리턴 값의 경우 +123.45와 같이 앞에 + 또는 -가 붙어 나오기 때문에
    이 값을 제거하면서 실수값을 리턴한다.
    :param angle: +123.56 또는 - 223,34 등 부호가 붙은 문자열
    :return angle: 123.56 또는 (-223.34+360) 값
    '''
    # +인 경우
    if angle.find('+') !=-1:
        angle=float(angle)
        return angle
    # -인 경우
    else:
        angle=float(angle)
        #print(angle)
        return 360 +angle
```

– 항공 영상기반 오브젝트 외곽선 추출 프로그램

ckdtree.py

```
import cv2
import numpy as np
import mlist
def setArray(src):
    '''
    파라미터로 전달된 영상 내에서 명도값 255인 픽셀들의 행렬 내 위치를 리스트에 저장한다
    :param src: 이진화된 영상
    :return: arr 내에서 명도 값 255인 곳의 위치를 가진 리스트
    '''
    rows, cols = src.shape
    arr = np.empty((0, 2), int)
    z =-1
    for i in range(0, rows):
        for j in range(0, cols):
            z = z +1
            x = src[i, j]
            if (x ==255):
                arr = np.append(arr, np.array([[i, j]]), axis =0)
    return arr
def RemoveExternalAreas(gray,mask):
    '''
    원본 영상에서 canny edge를 통해 외곽선을 추출한 영상과 mask R-CNN으로 처리된 마스크영역을
가진 영상을 파라미터로 전달 받아
    canny 영상과 mask 영상 대조를 통해 mask 영상에서 255인 명도값을 가지지 않는 canny 영상 내
픽셀들의 값을 0으로 만든다
    :param gray: canny edge로 처리된 영상
    :param mask: mask R-CNN으로 처리된 영상
    :return: gray mask 영역과 중복되지 않은 픽셀들의 값을 0으로 만든 영상
    '''
    rows, cols = gray.shape
    for i in range(0, rows):
        for j in range(0, cols):
            otValue = gray[i, j]
            maskValue=mask[i,j]
            if otValue ==255 and maskValue ==0:
                gray[i,j]=0
    return gray
def otsu(src):
    '''
    전달된 grayscale 영상을 otsu 이진화 기법을 통해 이진화한다.
    :param src: grayscale 영상
    :return: dst 이진화된 영상
    '''
    ret, dst = cv2.threshold(src, 0, 255, cv2.THRESH_BINARY +cv2.THRESH_OTSU)
    return dst
def operationStart(gray,mask,low =100,high =160,per =10):
    '''
    ckdtree가 적용된 영상을 리턴해주는 메소드이다.
    필수적으로 grayscale된 영상과 mask R-CNN으로 탐지된 이진화된 마스크 영역을 전달 받아야한다
    gray 영상과 mask 영상 대조를 통해 mask 영상 내에서 255인 명도 값을 가지지 않는 픽셀
    제거를 통해 해당 메소드의 결과가 추론된 mask 보다 나빠지지 않게 제한한다.
    :param gray: grayscale 영상
    :param mask: Mssk R-CNN으로 탐지된 영역이 이진화된 영상
    :param low:  cannyEdge를 적용하기 위한 low 임계값, 실험상 최적 low는 100이다
    :param high: cannyEdge를 적용하기 위한 high 임계값 실험상 최적 high는 160이다
    :param per: ckdtree적용할 횟수 최소 1, 최대 영상내에서 255인 명도 값을 가지는 픽셀 수 이다
    :return result : ckdtree가 적용된 영상
    '''
    # canny edge를 적용한다
    ot = cv2.Canny(gray, low, high)
    #전달받은 mask 영상을 이진화
    mask = otsu(mask)
    # 외부 영역 제거
    ot = RemoveExternalAreas(ot, mask)
    # 영상 내에서 edge가 하나도 검출 되지 않을 수 있다
    # 그러면 실행 중 오류가 날 위험이 있기에 그럴 경우 mask 영상을 리턴한다
    black = mask
    # 255인 명도값을 가지는 픽셀들의 위치를 저장한다
    newlist = setArray(ot)
    m = mlist.mlist()
    m.baseDataInIt(ot,black, low, high, newlist, per)
    #영상 내에서 255값을 가지는 픽셀들의 수가 100개 미만이라면
    #모든 픽셀의 위치에서 ckdtree를 적용하더라도 시간이 적게 걸리므로
    #모든 위치에서 ckdtree 적용을 통해 최적의 위치를 찾는다
    #100개 이상이라면 랜덤한 위치에서 파라미터로 전달된 per의 수만큼 ckdtree를 적용한다
    if len(newlist) <100:
        result = m.start()
    else:
        result = m.start2()
    return result
```

mlist.py

```
import cv2
import copy
from scipy import spatial
import timeit
import numpy as np
from scipy.spatial import cKDTree
import math
import random
import sys
def floodFill(src):
    '''
    플flood fill) 다차원 배열의 어떤 칸과 연결된 영역을 찾는 알고리즘이다.
    이 알고리즘은 그림 프로그램에서 연결된 비슷한 색을 가지는 영역에 "채우기" 도구와 같이
    전달받은 영상에서 연결된 영역을 찾아 그 내부를 채우는데 사용된다
    :param src: 내부를 채울 영상
    :return src:  floodfill이 적용된 영상
    '''
    im_th = src.copy()
    im_floodfill = im_th.copy()
    h, w = im_th.shape[:2]
    mask = np.zeros((h +2, w +2), np.uint8)
    cv2.floodFill(im_floodfill, mask, (0,0), 255);
    im_floodfill_inv = cv2.bitwise_not(im_floodfill)
    im_out = im_th | im_floodfill_inv
    return im_out
def pixelRead(src):
    '''
    파라미터로 전달된 영상에서 명도값 255를 가지는 픽셀들의 수를 체크한다
    :param src: 이진화된 영상
    :return white: 255인 값을 가지는 픽셀들의 수
    '''
    rows,cols=src.shape
    white=0
    black=0
    for i in range(0,rows):
        for j in range(0,cols):
            x=src[i,j]
            if x ==255:
                white+=1
    return white
class mlist:
    """
    한붓그리기와 동일하다. 영상 내에서 255 명도 값을 가지는 픽셀들을
    최소 이동거리 최대 면적을 가지게 서로 이어주어 floodFill을 적용하여
    면적계산이 가능하도록 한다.
    """
    def __init__(self):
        # 방문 했는지 안했는지
        self.visited = []
        # 각 노드의 거리 값
        self.dlist =[]
        self.route234 =np.empty((0, 2), int)
        self.route567 = np.empty((0, 2), int)
        self.visited2 =[]
    def baseDataInIt(self,src,black,x,y,newlist,per):
        '''
        클래스 실행에 필요한 기본적인 것들을 전달받는다
        :param src: ckdtree를 적용할 영상
        :param black: edge가 검출되지 않았을 때 리턴할 mask
        :param x: low 임계값
        :param y: high 임계값
        :param newlist: src에서 255인 명도값을 가지는 픽셀들의 위치
        :param per: ckdtree를 적용할 횟수
        '''
        self.rows, self.cols = src.shape
        self.src =src
        self.tempSrc =src
        self.black =black
        self.x =x
        self.y =y
        self.newlist =newlist
        tempList = []
        for i in range(len(newlist)):
            tempList.append(i)
        count=per
        if len(tempList)!=0:
            if count >len(tempList):
                count=len(tempList)
            self.perList =random.sample(tempList,count)
    def ckdtreeFunc(self,startNode):
        '''
```

```
        stratnode를 기준으로 ckdtree를 적용한다
        :param startNode: ckdtree를 적용할 시작 노드
        :return dis: 시작노드에서 ckdtree를 적용하여 계산된 이동거리
        '''
        dis=0
        templist=copy.deepcopy(self.newlist)
        pt=templist[startNode]
        asdf=templist[startNode]
        templist=np.delete(templist,(startNode),axis =0)
        #print(pt)
        self.route234 = np.append(self.route234, np.array([asdf]), axis =0)
        #instart = timeit.default_timer()
        for i in range(0,len(templist)):
            tree = cKDTree(templist)
            idx = tree.query(pt, k =1)
            distance = float(idx[0])
            index =int(idx[1])
            dis+=distance
            pt=templist[index]
            self.route234 = np.append(self.route234, np.array([pt]), axis =0)
            if len(templist)==1:
                break
            templist = np.delete(templist, (index), axis =0)
        distance, index = spatial.KDTree(templist).query(asdf)
        self.route234 = np.append(self.route234, np.array([asdf]), axis =0)
        dis += distance
        return dis
    def con2(self,src, a, b):
        '''
        전달받은 영상에서 a 픽셀 위치에서 b 픽셀 위치까지 최단거리로 잇는다
        a to b
        :param src: 이어줄 영상
        :param a: 시작점
        :param b: 도착점
        :return src: 이은 영상
        '''
        ai = a[0]
        aj = a[1]
        bi = b[0]
        bj = b[1]
        # 연결하기
        if (bi < ai and bj < aj):  # 좌상
            src =self.connentLt(src, ai, aj, bi, bj)
        elif (bi < ai and bj == aj):  # 중상
            src =self.connentCt(src, ai, aj, bi, bj)
        elif (bi < ai and bj > aj):  # 우상
            src =self.connentRt(src, ai, aj, bi, bj)
        elif (bi == ai and bj < aj):  # 좌중
            src =self.connentLs(src, ai, aj, bi, bj)
        elif (bi == ai and bj > aj):  # 우중
            src =self.connentRs(src, ai, aj, bi, bj)
        elif (bi > ai and bj < aj):  # 좌하
            src =self.connentLb(src, ai, aj, bi, bj)
        elif (bi > ai and bj == aj):  # 중하
            src =self.connentCb(src, ai, aj, bi, bj)
        elif (bi > ai and bj > aj):  # 우하
            src =self.connentRb(src, ai, aj, bi, bj)
        return src
    def drawPaint2(self,matlist,k,edgelist):
        '''
        floodFill을 적용한 영상을 matlist에
        edgelist에는 floodFill 적용 전 영상을 넣는다
        :param matlist: floodFill을 적용한 영상을 가진 리스트
        :param k: 비교할 거리값
        :param edgelist: edgelist에는 floodFill 적용 전 영상
        :return:
        '''
        copySrc=copy.deepcopy(self.src);
        #print("연결 경로 ",self.route234)
        for i in range(0, len(self.newlist)):
            copySrc =self.con2(copySrc,self.route234[i],self.route234[i +1])
        if len(edgelist)-1 <k:
            edgelist.append(copySrc)
            copySrc = floodFill(copySrc)
            matlist.append(copySrc)
        else:
            edgelist[k]=(copySrc)
            copySrc = floodFill(copySrc)
            matlist[k]=(floodFill(copySrc))
        return pixelRead(copySrc)
    def start(self):
        '''
        해당 클래스의 전체적 동작을 관린한다
```

```
        :return matlist[biggestMatIndex]: ckdtree가 적용되어 최소 이동거리 최대 면적을 가지는 결과를 리턴한다
        '''
        matlist=[]
        edgelist=[]
        biggest=0
        biggestMatIndex=0
        minDistance=sys.maxsize
        x=0
        for i in range(0, len(self.newlist)):
            startNode=i #시작노드 지정
            cur2 =self.ckdtreeFunc(startNode)
            if minDistance >cur2:
                minDistance=cur2
                w=self.drawPaint2(matlist,x,edgelist)
                biggest=0
                if w >biggest:
                    biggest=w
                    biggestMatIndex=x
                    x+=1
            elif minDistance ==cur2:
                w =self.drawPaint2(matlist, x, edgelist)
                if w > biggest:
                    biggest = w
                    biggestMatIndex = x
                    x +=1
            self.route234 = np.empty((0, 2), int)
        if len(matlist) ==0: #윤곽선 검출이 안됐을 때 터지는거 방지
            return self.black
        return matlist[biggestMatIndex]
    def start2(self):
        '''
        해당 클래스의 전체적 동작을 관린한다
        :return matlist[biggestMatIndex]: ckdtree가 적용되어 최소 이동거리 최대 면적을 가지는 결과를 리턴한다
        '''
        matlist=[]
        edgelist=[]
        biggest=0
        biggestMatIndex=0
        minDistance=sys.maxsize
        x=0
        for i in range(0, len(self.perList)):
            startNode=self.perList[i]#시작노드 지정
            cur2 =self.ckdtreeFunc(startNode)
            if minDistance >cur2:
                minDistance=cur2
                w=self.drawPaint2(matlist,x,edgelist)
                biggest=0
                if w >biggest:
                    biggest=w
                    biggestMatIndex=x
                    x+=1
                    xindex=i
            elif minDistance ==cur2:
                w =self.drawPaint2(matlist, x, edgelist)
                if w > biggest:
                    biggest = w
                    biggestMatIndex = x
                    x +=1
                    xindex = i
            self.route234 = np.empty((0, 2), int)
        stop = timeit.default_timer()
        if len(matlist) ==0: #윤곽선 검출이 안됐을 때 터지는거 방지
            return self.black
        return matlist[biggestMatIndex]
    '''
    아래 메소드 들은 위 con2 메소드 동작을 위해 구현된 함수들입니다
    '''
    def connentLt(self,src, ai, aj, bi, bj):
        #print("lt")
        src[ai, aj] =255
        if (ai == bi and aj == bj):
            return src
        elif (bi < ai and bj < aj):
            return self.connentLt(src, ai -1, aj -1, bi, bj)
        elif (bi < ai):
            return self.connentLt(src, ai -1, aj, bi, bj)
        elif (bj < aj):
            return self.connentLt(src, ai, aj -1, bi, bj)
    def connentCt(self,src, ai, aj, bi, bj):
        # print("ct")
        src[ai, aj] =255
```

```
        if (ai == bi and aj == bj):
            return src
        elif (bi < ai):
            return self.connentCt(src, ai -1, aj, bi, bj)
    def connentRt(self,src, ai, aj, bi, bj):
        # print("rt")
        src[ai, aj] =255
        if (ai == bi and aj == bj):
            return src
        elif (bi < ai and bj > aj):
            return self.connentRt(src, ai -1, aj +1, bi, bj)
        elif (bi < ai):
            return self.connentRt(src, ai -1, aj, bi, bj)
        elif (bj > aj):
            return self.connentRt(src, ai, aj +1, bi, bj)
    def connentLs(self,src, ai, aj, bi, bj):
        # print("ls")
        src[ai, aj] =255
        if (ai == bi and aj == bj):
            return src
        elif (bj < aj):
            return self.connentLs(src, ai, aj -1, bi, bj)
    def connentRs(self,src, ai, aj, bi, bj):
        # print("rs")
        # print(type(src))
        src[ai, aj] =255
        if (ai == bi and aj == bj):
            return src
        elif (bj > aj):
            return self.connentRs(src, ai, aj +1, bi, bj)
    def connentLb(self,src, ai, aj, bi, bj):
        # print("lb")
        src[ai, aj] =255
        if (ai == bi and aj == bj):
            return src
        elif (bi > ai and bj < aj):
            return self.connentLb(src, ai +1, aj -1, bi, bj)
        elif (bi > ai):
            return self.connentLb(src, ai +1, aj, bi, bj)
        elif (bj < aj):
            return self.connentLb(src, ai, aj -1, bi, bj)
    def connentCb(self,src, ai, aj, bi, bj):
        # print("cb")
        src[ai, aj] =255
        if (ai == bi and aj == bj):
            return src
        elif (bi > ai):
            return self.connentCb(src, ai +1, aj, bi, bj)
    def connentRb(self,src, ai, aj, bi, bj):
        src[ai, aj] =255
        # print("ai,aj,bi,bj",ai,",",aj,",",bi,",",bj)
        # print("rb")
        if (ai == bi and aj == bj):
            return src
        elif (bi > ai and bj > aj):
            return self.connentRb(src, ai +1, aj +1, bi, bj)
        elif (bi > ai):
            return self.connentRb(src, ai +1, aj, bi, bj)
        elif (bj > aj):
            return self.connentRb(src, ai, aj +1, bi, bj)
    def getRowI(self,x):
        # print(int(x/(rows)))
        return int(x / (self.cols))
    def getColl(self,x):
        # print(x%rows)
        return x % (self.cols)
    def setRowI(self,x):
        # print(int(x/(rows)))
        return int(x / (self.cols))
    def setColl(self,x):
        # print(x%rows)
        return x % (self.cols)
```

부록 4. PCI 측정기술 소프트웨어 소스코드

1. csv 데이터 불러오기 (인식하기) + 데이터 파싱

dataloader.py 파일

```python
class Create_Data:
    def __init__(self, csv_path, txt_path, save_folder):
        self.csv_path = csv_path        # GUI 에서 csv 경로 받아옴
        self.txt_path = txt_path        # GUI 에서 txt 경로 받아옴
        self.save_folder = save_folder

        self.txt_loader()               # txt 파일 불러오는 부분 (Meta Data reader)
                                        # 주어진 meta data로부터 이미지상 1 pixel별 위도, 경도 거리 수식 계산
                                        # 주어진 csv image의 Height와 Width 데이터 로드

        print(f'{csv_path}, {txt_path} load success')

    def load_image(self):
        print('image load start')
        image = Image.open(self.csv_path)
        image = np.array(image, dtype=np.uint8)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        print('image load end')
        return image, self.csv_path.split('/')[-1].split('.')[0]

    def create_image(self):     # GUI 에서 버튼 누를시 작동하는 부분
        name = self.csv_path.split('/')[-1].split('.')[0]
        print('image create start')
        df = pd.read_csv(self.csv_path)         # csv 파일 읽는 부분
        df['dummy'] = df['Line#'].astype(str) + '_' + df['Column#'].astype(str)
        df = df.drop(columns=['Line#', 'Column#'])
        B = df['485.872nm'].values      # Load한 CSV 파일로 부터 RGB wave length 데이터 추출
        G = df['544.146nm'].values
        R = df['669.657nm'].values

        # wave length 값 Normalization 이후 이미 얻어 놓은 Height, Width로 [R, G, B] 이미지 생성
        minmax = MinMaxScaler()
        b = minmax.fit_transform(np.log(B + 1).reshape(-1, 1)) * 255
        g = minmax.fit_transform(np.log(G + 1).reshape(-1, 1)) * 255
        r = minmax.fit_transform(np.log(R + 1).reshape(-1, 1)) * 255
        image = np.concatenate([b, g, r], axis=1).reshape(-1, self.width, 3).astype(np.uint8)

        # 생성된 이미지 저장
        cv2.imwrite(f'{self.save_folder}/{name}.png', image)
        print('image create end')
        return image, name, df
```

2. meta 데이터 불러오기 (인식하기) + 데이터 파싱

dataloader.py

```
def txt_loader(self):
    ## txt 파일 읽어서 Height와 Width, 1 pixel 당 위도 경도 변화 정도 데이터 추출
    with open(self.txt_path, 'r', encoding='utf-8') as f:
        datas = list(map(str.strip, f.readlines()))
        map_info = []
        for data in datas:
            if 'samples' in data:
                data = data.split(' ')
                self.width = int(data[-1])
            elif 'lines' in data:
                data = data.split(' ')
                self.height = int(data[-1])
            elif 'map' in data:
                map_info = data.split(',')
                break
        self.latitude = float(map_info[5])  # 위도
        self.longitude = float(map_info[6])  # 경도


    ## 제일 마지막에 json파일 저장할 때 이용할 변수 미리 선언
    with open(self.txt_path, 'r', encoding='utf-8') as f:
        self.all_txt_data = f.read()
```

3. 차선 검출 알고리즘 (일부) + 차선 인식 검출 보정 알고리즘(곡선 보정 등)

lane_detect.py 전체 프로세싱

```
if event == cv2.EVENT_LBUTTONDOWN:  # 마우스로 시작할 포인트 클릭
    i = 0
    self.points.append((x, y))      # 클릭한 좌표 저장
    self.count += 1
    self.process = True
    if self.count == 2:             # 마우스로 클릭한 좌표가 2개라면 아래 코드 실행
        h, w = self.dummy.shape[:2]
        left_points, right_points = [], []
        # 마우스로 클릭한 좌표 정렬 (기울기 양/음 값에 대한 문제 사전 방지), 무조건 위쪽의 점이 0번 index 아래쪽 점의 1번 index
        self.points = utils.fix_position(self.points)
        # 마우스로 클릭한 2개의 좌표에 대한 기울기 계산
        gradient = gradient1 = gradient2 = utils.calculate_gradient(self.points)    # forward line detect를 위한 계산
        back_grad = back_grad1 = back_grad2 = utils.calculate_gradient(self.points) # backward line detect를 위한 계산
        # sr 는 line을 detect하기 위해서 볼 영역의 넓이, cur_points = forward 시작점, back_points = backward 시작점
        sr, cur_points, back_points = 120, self.points[0], self.points[0]
        while sr < cur_points[0] < w-sr and sr < cur_points[1] < h-sr:
            crop_image, dx, dy = utils.crop_rect(self.image, cur_points, sr)  # 이미지 crop
            ## 왼쪽, 오른쪽 라인을 따로 detect하기 위해서 반대쪽 라인에 대한 이미지 Masking
            mask1, mask2, dir = utils.masking_image(crop_image, gradient)
            ## 왼쪽, 오른쪽 인식된 라인 저장
            ## 인식 기준은 처음에 포인트에서 얻은 기울기를 가지고 기울기가 비슷한 직선 추출
            left_points.extend(utils.detect_line(mask1, back_grad1, dx, dy, dir))
            right_points.extend(utils.detect_line(mask2, back_grad2, dx, dy, dir))
            ## 위에서 얻은 포인트를 가지고 이전 이미지의 포인트와 현재 이미지의 포인트를 연결
            ## 떨어진 라인 연결하는 부분
            left_points, mask1, gradient1, have1 = utils.forward_connect_line(self.background, left_points, i)
            right_points, mask2, gradient2, have2 = utils.forward_connect_line(self.background, right_points, i)
            self.background = cv2.add(self.background, cv2.add(mask1, mask2))
            ## 다음 이미지에서 추출할 라인 기울기 추출
            if have1 and have2: gradient = (gradient1 + gradient2) / 2
            elif have1: gradient = gradient1
            elif have2: gradient = gradient2
            else: gradient = (gradient1 + gradient2) / 2
            if i == 0: have1, have2 = False, False
            ## 이미지 기울기 방향으로 전진
            cur_points = utils.moving_forward_point(gradient, cur_points, sr, left_points[0][:4],
                                                    right_points[0][:4], have1, have2)
```

4. 차선 검출 알고리즘 세부 항목

```
def detect_line(image, gradient, dx, dy, dir):
    ret, labels, stats, centroids = cv2.connectedComponentsWithStats(image)
    line_point = []
    dummy = np.zeros_like(image)
    h, w = dummy.shape[:2]
    pre_dis = float('inf')
    for i in range(1, ret):
        dummy[labels == i] = 255
        contours = cv2.findContours(dummy, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
        cnt = contours[0]
        rect = cv2.minAreaRect(cnt[0])
        box = cv2.boxPoints(rect)
        box = np.int0(box)
        box = fixed_box_point(box)
        check, box = is_line(box, gradient, dir)
        if check:
            x, y = 0, 0
            for j in range(4):
                x += box[j][0]
                y += box[j][1]

            cen_x = x / 4
            cen_y = y / 4

            cur_dis = np.sqrt((cen_x - w / 2) ** 2 + (cen_y - h / 2) ** 2)

            if cur_dis < pre_dis:
                line_point = [box]
                pre_dis = cur_dis
    try:
        cv2.drawContours(dummy, np.array([line_point[0][:-1]]), -1, (100, 100, 100), 3)
        for j in range(4):
            line_point[0][j] = [line_point[0][j][0] + dx, line_point[0][j][1] + dy]
    except:
        pass

    return line_point
```

5. 차선 검출 알고리즘 다음 영역으로 진행 알고리즘

```
def forward_positive_grad(gradient, cur_points, sr, left_points, right_points, left_check, right_check, value):
    x, y = cur_points
    x += sr // 4
    y += gradient * sr // 4
    answer = [int(x), int(y)]
    if left_check and right_check:
        left = sorted(left_points, key=lambda x:x[0])[-1]
        right = sorted(right_points, key=lambda x:x[0])[-1]
        if left[0] > right[0]:  # left something
            if left[0] > cur_points[0]:
                return [left[0], left[1] + value]
            else:
                return answer
        else:
            if right[0] > cur_points[0]:
                return [right[0], right[1] - value]
            else:
                return answer
    elif left_check:
        points = sorted(left_points, key=lambda x:x[0])[-1]
        if points[0] > cur_points[0]:
            return [points[0], points[1] + value]
        else:
            return answer

    elif right_check:
        points = sorted(right_points, key=lambda x:x[0])[-1]
        if points[0] > cur_points[0]:
            return [points[0], points[1] - value]
        else:
            return answer
    else:
        return answer
```

6. 검출 차선에서 도로 마스킹 및 차량 등. 장해물 제거 코드

```
def __init__(self, pad_image, result_image, latitude, longitude, split_road_count, save_folder, name):
    self.save_folder = save_folder
    self.name = name
    d_latitude = 111195   # 경도 meter
    d_longitude = 88804   # 위도 meter
    self.d_lat = latitude * d_latitude
    self.d_long = longitude * d_longitude
    self.split_road_count = split_road_count

    ### 20cm 로 나눌때 도로의 중심에서 2m 거리만큼 그리기 위해서 이용 ###
    ### [1, 3, 2] 도로 분할시 이용되는 변수
    self.sr = ((2 / self.d_lat) + (2 / self.d_long)) // 2

    self.line_image = result_image
    self.pad_image = pad_image

    # 쓰레기 값 제어 이미지 도로 split 할 때 해당 영역은 배제 하기 위해서 활용 #
    self.outside_image = cv2.inRange(pad_image, (0, 0, 0), (1, 1, 1))
    self.interrupt = cv2.inRange(pad_image, (250, 250, 250), (255, 255, 255))
    self.interrupt = cv2.add(self.outside_image,
                             cv2.add(self.interrupt,
                                     cv2.add(cv2.inRange(self.pad_image, (110, 0, 250), (130, 1, 255)),
                                             cv2.inRange(self.pad_image, (110, 110, 250), (130, 130, 255)))))

    self.outside_image = cv2.bitwise_not(self.outside_image)

    # 라인만 흰색 이미지 #
    self.mask_image = cv2.inRange(self.line_image.copy(), (190, 190, 0), (210, 210, 10))
```

7. 20cm단위로 자른 영역 rect point를 추출하는 코드

```python
# 20 cm 단위로 쪼개는 부분 #
def _draw_20cm_box(image, outside_image, dummy, point1, point2, i, latitude, longitude, split_count, data):
    ul_x, ul_y = point1[i-1]
    ur_x, ur_y = point2[i-1]
    dr_x, dr_y = point2[i]
    dl_x, dl_y = point1[i]
    du_x, du_y = (ur_x - ul_x)/6, (ur_y - ul_y)/6
    dd_x, dd_y = (dr_x - dl_x)/6, (dr_y - dl_y)/6

    z = (du_x*latitude)**2 + (du_y*longitude)**2
    alpha = np.sqrt(0.2**2 / z)
    ddu_x, ddu_y = np.round(alpha * du_x, 3), np.round(alpha * du_y, 3)

    z = (dd_x*latitude)**2 + (dd_y*longitude)**2
    alpha = np.sqrt(0.2**2 / z)
    ddd_x, ddd_y = np.round(alpha * dd_x, 3), np.round(alpha * dd_y, 3)

    upper_points = [[ul_x, ul_y], [int(ul_x+du_x*1-ddu_x), int(ul_y+du_y*1-ddu_y)], [int(ul_x+du_x*1+ddu_x), int(ul_y+du_y*1+ddu_y)], [int(ul_x+du_x*2), int(ul_y+du_y*2)],
                    [int(ul_x+du_x*4), int(ul_y+du_y*4)], [int(ul_x+du_x*5-ddd_x), int(ul_y+du_y*5-ddd_y)], [int(ul_x+du_x*5+ddd_x), int(ul_y+du_y*5+ddd_y)], [ur_x, ur_y]]

    lower_points = [[dl_x, dl_y], [int(dl_x+dd_x*1-ddd_x), int(dl_y+dd_y*1-ddd_y)], [int(dl_x+dd_x*1+ddd_x), int(dl_y+dd_y*1+ddd_y)], [int(dl_x+dd_x*2), int(dl_y+dd_y*2)],
                    [int(dl_x+dd_x*4), int(dl_y+dd_y*4)], [int(dl_x+dd_x*5-ddd_x), int(dl_y+dd_y*5-ddd_y)], [int(dl_x+dd_x*5+ddd_x), int(dl_y+dd_y*5+ddd_y)], [dr_x, dr_y]]

    # points = np.array([[upper_points[3], upper_points[-4], lower_points[-4], lower_points[3]]])
    points = np.array([[upper_points[1], upper_points[-2], lower_points[-2], lower_points[1]]])
    cv2.drawContours(dummy, points, -1, (i, i, i), -1)

    data['case1'][f'{(i-1)//split_count+1}-{4}-{(1000//split_count)*((i-1)%split_count+1)}'] = [upper_points[0], upper_points[-1], lower_points[-1], lower_points[0]]

    ########################## 시간 오래 걸리는 부분 ##############################################################
    if np.sum(outside_image[dummy == i]) != 0:   # 디텍트한 영역이 쓰레기 부분이 포함되어 있으면 배제
        return image, data
    ###############################################################################################

    for num, k in enumerate([1, 5, 3]):
        points = np.array([[upper_points[k], upper_points[k+1], lower_points[k+1], lower_points[k]]])
        color = (np.random.randint(0, 255), np.random.randint(0, 255), np.random.randint(0, 255))
        cv2.drawContours(image, points, -1, color, -1)
        data['case1'][f'{(i-1)//split_count+1}-{num+1}-{(1000//split_count)*((i-1)%split_count+1)}'] = [upper_points[k], upper_points[k+1], lower_points[k+1], lower_points[k]]

    return image, data
```

```python
def processing(self):
    ## Line 을 [1, 3, 2] 영역으로 분할하기 위해서 3개로 분할한 좌표값을 가져옴
    line_split_image = utils.triple_line_split(self.mask_image, self.outside_image, self.d_lat,
                                               self.d_long, self.sr, self.split_road_count)

    dummy = self.pad_image.copy()
    dummy[line_split_image == 255] = (0, 0, 0)

    find_center = np.zeros_like(self.mask_image)
    find_center[line_split_image == 255] = 255

    test = cv2.bitwise_and(self.mask_image, line_split_image)
    retval, labels, stats, centroids = cv2.connectedComponentsWithStats(test)
    left_center, right_center = [], []

    ## 실제 필요한 좌표만 가져옴 쓰레기값 버리는 부분 ##
    for i in trange(1, retval):
        y, x = np.where(labels == i)
        left_center.append([int(x[0]), int(y[0])])
        right_center.append([int(x[-1]), int(y[-1])])

    ## 10m 를 self.split_road_count 변수 만큼의 개수로 쪼갤 수 있음 ##
    ## default론 self.split_road_count = 50 이기 때문에 1000cm / 50 = 20cm 단위로 쪼개짐
    utils.draw_20cm_box(dummy, self.interrupt, left_center, right_center, self.d_lat, self.d_long,
                        self.split_road_count, self.save_folder)

    cv2.imwrite(f'{self.save_folder}/{self.name}_split.png', dummy)

    return dummy
```

8. Wave Length 별 CSV 추출 공식 코드

create_use_csv.py

```python
# wave length 별로 영역 가져와서 추출
def create_csv(path):
    new_path = '/'.join(path.split('/')[:-1]) + '/use_csv'
    if 'use_csv' not in os.listdir('/'.join(path.split('/')[:-1])):
        os.mkdir(new_path)

    wave_csv_path = [path + '/' + x for x in os.listdir(path)]

    # 1번은 나누기, 2번은 ABS 나누기, 3번은 역수 빼기, 4번은 곱하기
    use_rules = {'VIS1': [740, 460, 2], 'VIS1R': [740, 460, 1], 'VIS2': [830, 490, 2], 'VIS2R': [830, 490, 1],
                 'VIS3': [420, 595, 2], 'VIS3R': [420, 595, 1], 'NDVI': [802, 638, 2], 'NDVIR': [802, 638, 1],
                 'PVISI1': [836, 688, 2], 'PVISI1R': [836, 688, 1], 'PRI': [570, 531, 2], 'PRIR': [570, 531, 1],
                 'ARI1': [705, 550, 3], 'ARI2': [802, 'ARI1', 4], 'REI': [705, 754, 1], 'WBI': [899, 967, 1]}

    for csv_path in wave_csv_path:
        use_df = pd.DataFrame([], columns=['location'] + list(use_rules.keys()))
        df = pd.read_csv(csv_path, index_col=0)

        use_df['location'] = df['location']

        for col in list(use_rules.keys()):
            front, back, rule = use_rules[col]
            front = str(front)
            back = str(back)
            if rule == 1:
                use_df[col] = df[front] / df[back]
            elif rule == 2:
                use_df[col] = abs(df[front] - df[back]) / abs(df[front] + df[back])
            elif rule == 3:
                use_df[col] = (1/df[front]) - (1/df[back])
            elif rule == 4:
                use_df[col] = df[front] * use_df[back]

        use_df.to_csv(f'{new_path}/{csv_path.split("/")[-1]}')
    return new_path
```

9. 최종 IRI, Rutting, Crack 값 출력

```python
def Rutting(value, start, end, cen_start, type):
    answer = []
    distance = cen_start - start
    for i in range(start, end):
        if not np.isnan(value[i][0]):
            answer.append(abs(value[i+distance] - value[i]))

    if type == 'mean':
        return np.mean(np.array(answer), axis=0)
    return np.max(np.array(answer), axis=0)

def Crack(value, start, end):
    answer = []
    for i in range(start, end):
        if not np.isnan(value[i][0]):
            answer.append(value[i])
    return np.mean(np.array(answer), axis=0)

def create_csv(path, split_road_count):
    new_path = '/'.join(path.split('/')[:-1]) + '/final_csv'
    if 'final_csv' not in os.listdir('/'.join(path.split('/')[:-1])):
        os.mkdir(new_path)

    use_csv_path = [path + '/' + x for x in os.listdir(path)]

    for csv_path in use_csv_path:
        df = pd.read_csv(csv_path, index_col=0).iloc[:, 1:]
        value = df.values
        name = csv_path.split('/')[-1].split('.')[0]
        final_csv = pd.DataFrame([], columns=[name] + list(df.keys()))
        final_csv[name] = ['IRI1', 'IRI2', 'Rutting1_1-3', 'Rutting1_2-3', 'Rutting2_1-3', 'Rutting2_2-3', 'Crack']
        final_csv.iloc[0, 1:] = IRI(value, 0, split_road_count)       # IRI1
        final_csv.iloc[1, 1:] = IRI(value, split_road_count, split_road_count*2)    # IRI2
        final_csv.iloc[2, 1:] = Rutting(value, 0, split_road_count, split_road_count*2, 'mean')    # Rutting1_1-3
        final_csv.iloc[3, 1:] = Rutting(value, split_road_count, split_road_count*2, split_road_count*2, 'mean')    # Rutting1_2-3
        final_csv.iloc[4, 1:] = Rutting(value, 0, split_road_count, split_road_count*2, 'max')    # Rutting2_1-3
        final_csv.iloc[5, 1:] = Rutting(value, split_road_count, split_road_count*2, split_road_count*2, 'max')    # Rutting2_2-3
        final_csv.iloc[6, 1:] = Crack(value, split_road_count*3, split_road_count*4)    # Crack

        final_csv.to_csv(f'{new_path}/{csv_path.split("/")[-1]}')
    return new_path
```

드론 비탈면 관리 활용지침(안)

2021. 04.

목 차

제1장 비탈면 개요 .....

제2장 비탈면 관리 지침 현황 .....

제3장 드론 비탈면 관리 활용지침(안) .....

제1장 비탈면 개요

1.1 비탈면 정의

- 비탈면은 크게 자연비탈면과 인공비탈면으로 크게 분류할 수 있는데, 이 중 인공비탈면은 쌓기비탈면과 깎기비탈면으로 나눌 수 있다.

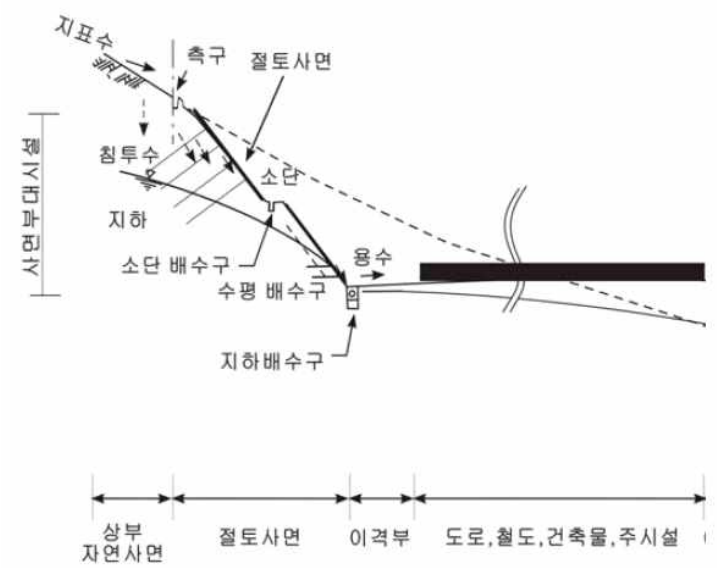
- 쌓기비탈면은 정해진 쌓기재료로 요구되는 단면을 축조하여야 하므로 재료의 구분이 명확하며, 용도에 따라 도로개설을 위한 노반 쌓기비탈면, 제방이나 댐의 제체비탈면, 기타 부지확보를 위해 쌓기비탈면 등을 들 수 있다.

- 깎기비탈면은 자연적으로 이루어져 있는 비탈면을 깎아서 만든 것으로 흙과 암석이 불규칙하게 뒤섞인 불균일한 지층을 이루기도 하고, 때로는 암석이 풍화되거나 단층, 절리 등이 잘 발달되어 있어서 상당히 복잡한 구성을 갖기도 한다.

- 깎기비탈면은 장기간 자연적인 안정화 과정을 거친 자연비탈면이나 구성재료의 품질관리가 가능한 쌓기비탈면에 비하여 인위적인 하부지지력 제거로 급속한 힘의 불균형 상태를 초래하므로 상대적인 붕괴위험이 높아 적절한 보수·보강에 의한 유지관리가 필수적이다.

1.2 비탈면 구성

• 깎기비탈면은 공간적 위치나 기능에 따라 상부자연비탈면, 깎기비탈면, 비탈면시설, 이격부, 소단으로 나눈다.



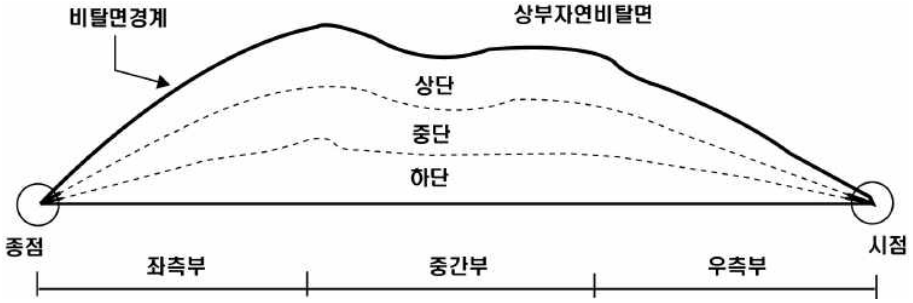
<그림 1.1> 깎기비탈면 구성 (절토사면 유지관리 매뉴얼 보완개정 연구, 2013)

1.2.1 상부자연비탈면

• 상부자연비탈면은 깎기구간의 상부경계인 비탈면경계선 상부의 자연경사지반을 말한다. 상부자연비탈면은 깎기비탈면의 강우 유입량에 따른 안정성에 큰 영향을 미치는 계곡부 및 집수지형 여부를 결정한다. 상부자연비탈면에서는 붕괴의 전조현상인 인장균열 및 침하 등을 관찰할 수 있으며, 인공시설물(봉분, 송전탑, 농경지 등)의 분포여부와 단면경사의 완급에 따라 대책공법 결정에 영향을 주는 깎기비탈면의 주요 구성요소 중 하나이다.

1.2.2 깎기비탈면

• 깎기비탈면은 자연비탈면을 깎아서 노출시킨 경사 지반을 말한다. 깎기비탈면의 각 부위별 명칭은 깎기비탈면의 전방에서 비탈면을 바라보았을 때를 기준으로 하며, 깎기비탈면을 마주볼 때 '좌측부', '중간부', '우측부', 상하 위치에 따라 '상단', '중단', '하단'으로, 우측부의 끝점은 '시점부', 반대 위치는 '종점부'로 칭한다. 또한, 깎기비탈면과 상부자연비탈면과의 경계를 '비탈면경계'라 하며 이중 제일 높은 곳을 '최상부'라 한다.



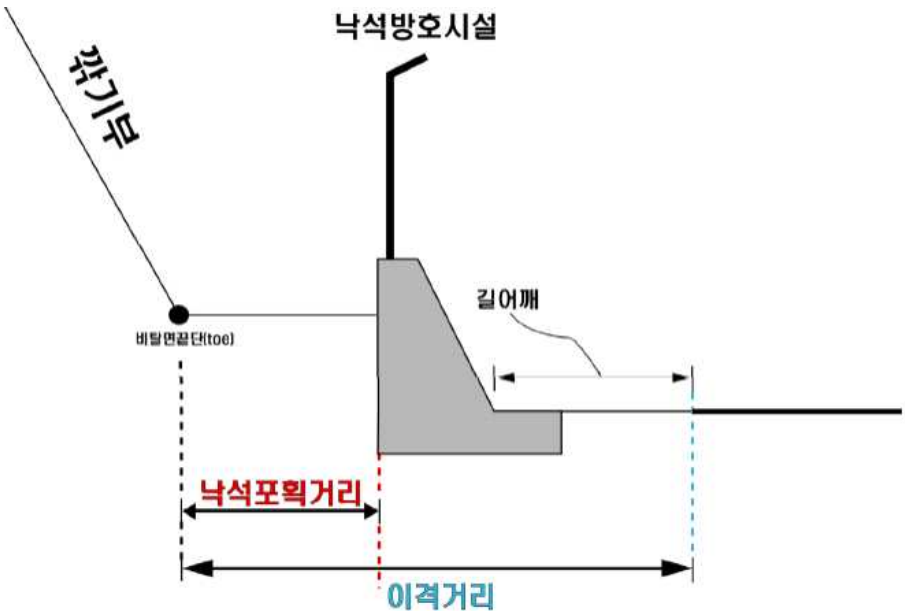
<그림 1.2> 깎기부 부위별 명칭 (도로비탈면 유지관리 실무 매뉴얼, 2018)

1.2.3 비탈면시설

• 비탈면시설은 깎기비탈면의 안정성 유지나 개선을 위해 상부자연비탈면, 깎기비탈면, 이격부, 소단내에 설치한 인공시설물로, 대표적인 비탈면안정공법을 말하며 안전율을 증가시키는 보강공법과 안전율을 유지하는 보호공법으로 분류할 수 있다. 비탈면시설의 목적에 따른 분류는 시설물을 이용한 비탈면의 안전율을 증가시키는 활동억제시설, 낙석이나 유실물이 시설물로 유입되는 것을 차단하는 낙석방호시설, 비탈면의 유실 및 풍화 등을 억제하는 표면보호시설, 지표수 및 지하수에 의한 안정성 저하를 억제하는 수리제어시설 및 점검 · 진단 등 유지관리를 위하여 설치하는 비탈면 점검시설로 나눌 수 있다.

1.2.4 이격부

• 이격부는 깎기비탈면의 상·하단부에서 주시설물까지의 공간을 말한다. 특히, 깎기비탈면 하단부에 위치한 이격부는 낙석이나 붕괴발생시 충격완화 또는 주 시설물로의 직접적인 유입을 방지하기 위한 여유공간을 제공하며, 보수보강을 위한 공사 시행시 작업공간 및 비탈면시설 설치공간 등의 기능을 가진다. 낙석포획거리는 낙석 등 붕괴산물을 포획할 수 있는 거리를 말하며, 이격거리는 낙석 등 붕괴산물이 시설물이용자에게 직접적인 영향을 미치는 거리이다.



<그림 1.2> 이격부 구성 (도로비탈면 유지관리 실무 매뉴얼, 2018)

1.2.5 소단

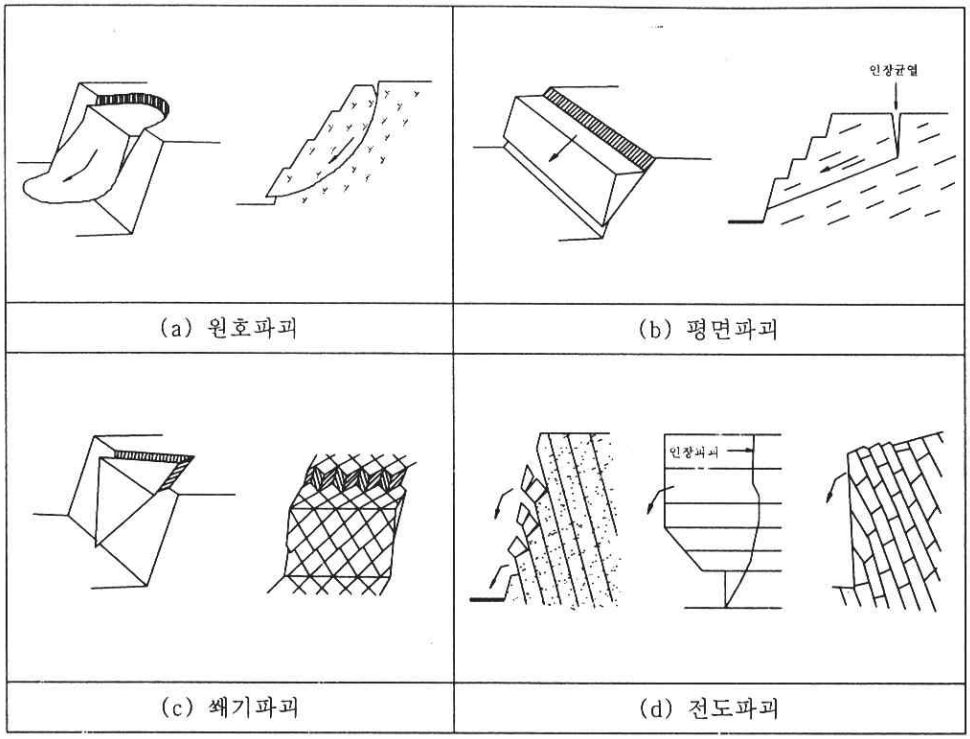
• 소단은 비탈면 중간에 시공하며 완충지대로서의 안정성 확보, 비탈면을 따라 흐르는 지표수의 유속 저감, 유지관리에 있어서 점검로의 역할을 한다. 지표수의 유입과 풍화 등으로 소단의 훼손이 우려될 경우 소단배수구 시공을 고려하여야 한다. 깎기비탈면의 높이가 10m 이상인 비탈면에서는 유지관리를 위한 점검, 배수시설의 설치공간으로 활용하기 위하여 원칙적으로 소단을 설치하며, 비탈면 중간에 5~20m 높이마다 폭은 1~3m의 소단을 설치한다.

1.3 비탈면 붕괴특성

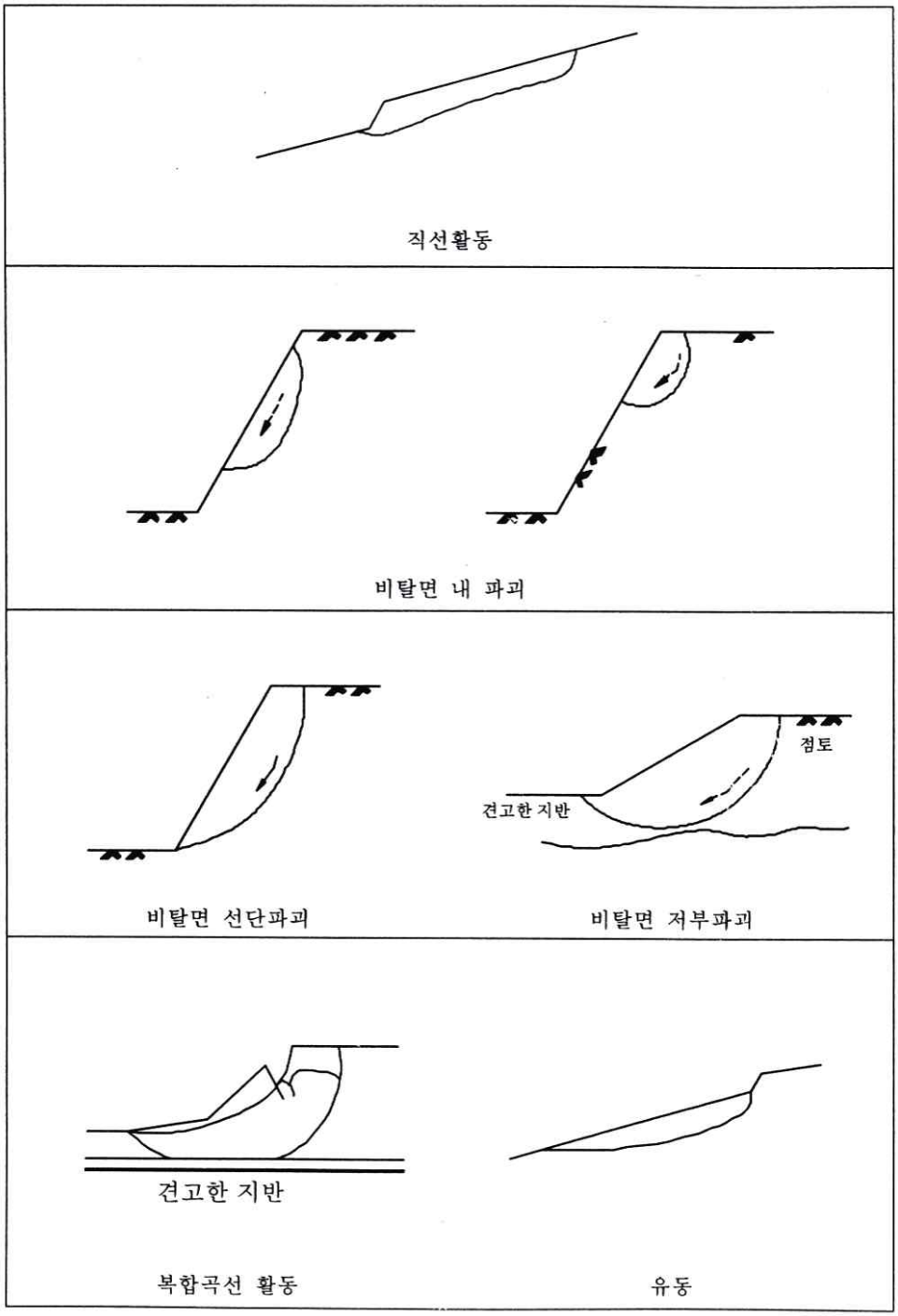
• 비탈면에서 발생하는 파괴유형은 지반종류, 지층조건과 외부적 유발원인에 따라 매우 다양한 형태로 발생하며, 전형적인 파괴유형은 다음과 같다.

(1) 토사비탈면 : 원호파괴, 평면파괴, 유동파괴
(2) 암반비탈면 : 평면파괴, 쐐기파괴, 전도파괴, 원호파괴

• 파괴유형은 지반의 종류, 지층조건, 파괴유발조건에 따라 차이를 나타낸다. 토사지반의 경우, 균질한 점쌓기에서는 원호형태의 깊은 파괴가 발생하고, 균질한 사질토 지반에서는 평면파괴와 유동파괴가 주로 발생한다. 암반의 경우는 암반의 강도, 불연속면의 방향성과 상태(빈도, 길이, 충전물, 거칠기 등), 단층파쇄대의 존재여부에 따라 다양한 파괴형태를 발생한다. 점토와 실트가 많은 지반은 집중호우에 의해 흐름형태의 파괴인 이토류가 발생하고, 모래와 자갈층의 경우에는 토석류가 발생한다. 그림 1.3과 1.4는 암반비탈면과 토사비탈면에서 발생하는 다양한 파괴유형을 나타낸 것이다. 토사와 암반의 중간상태 지반조건에서는 두 가지 지반조건에서 발생하는 파괴형태가 모두 발생하기도 하고 복합적으로 발생할 수도 있다.



<그림 1.3> 암반비탈면 파괴유형(건설공사 비탈면 유지관리지침, 2006)



<그림 1.4> 토사비탈면 파괴유형(건설공사 비탈면 유지관리지침, 2006)

1.3.1 쌓기비탈면 붕괴특성

- 일반적으로 쌓기비탈면의 붕괴 특징은 지하수 및 지표수에 의한 비탈면 표층의 유실 또는 깎기쌓기경계(절성경계)의 경우 간극수압의 상승으로 인한 비탈면 선단부에서의 파괴가 있으며, 쌓기비탈면 시공 시 다짐 불량, 쌓기 재료 불량 등으로 발생한다. 또한, 산지 도로에서 종단경사를 따라 유하하던 표면수가 도로 턱을 넘어 비탈면을 침식하여 일어나기도 한다.

(1) 기초지반을 포함하는 붕괴

- 쌓기비탈면에서 기초지반을 포함하는 붕괴형태는 기초지반이 연약한 경우(이탄, 충적점토 등) 쌓기로 인한 상재하중에 의해 기초지반이 파괴되어 쌓기비탈면이 붕괴하는 사례가 있다. 또, 자연비탈면에 쌓기를 시행한 경우는 원지반 비탈면(자연비탈면)이 붕괴하는 경우가 있다.

(2) 비탈면내 붕괴

- 쌓기비탈면의 비탈면내 붕괴형태는 쌓기재료, 다짐도, 비탈면경사 등의 설계 · 시공의 잘못으로 발생하는 붕괴형태로 일반적으로 비탈면 붕괴 규모가 작은 특징을 보이며, 파괴면이 비탈끝을 지나며 중규모의 붕괴양상으로 나타나는 특징을 보인다.

(3) 기초지반 특성에 따른 붕괴

- 기초지반의 특성에 따른 붕괴유형은 비탈면이 형성된 기초지반의 특성에 따라 붕괴형태와 그 규모가 다르게 나타난다. 불투수성의 견고한 기초지반위에 만들어진 쌓기비탈면은 불투수성의 견고한 지반과 쌓기비탈면과의 경계면(깎기쌓기경계)에서 지하수 침투 등에 의해 쌓기 지반이 연약화 되어 붕괴되는 특성을 보이며, 활동성 지반에 쌓기하여 만들어진 비탈면의 경우는 소규모의 유실 등이 가속화 되어 대규모 비탈면 붕괴로 발달할 수 있다.

(4) 지하수 침투에 의한 붕괴

- 하천과 같은 모래, 자갈 등의 투수층에 접하여 쌓기를 하는 경우는 침투수에 의해 제체 내 수위가 상승하고 쌓기부의 비탈끝 부분의 붕괴를 유발한다. 이러한 붕괴는 비탈면 전역에 걸쳐 발생하지 않고 소규모붕괴로 시작하여 발견이 늦어지는 경우가 많으므로 비탈면 조사에 있어 주의가 필요하다.

1.3.2 깎기비탈면 붕괴특성

- 깎기비탈면의 붕괴특징은 활동면 심도가 얕은 붕괴, 활동면 심도가 깊은 붕괴 그리고 붕괴규모가 광범위한 경우로 발생되는 특징이 있다.

(1) 활동면 심도가 얕은 붕괴

- 토층과 풍화층으로 급경사를 가진 비탈면은 암반층을 경계면으로 하여 암반층 상부의 토사층이 흘러내리기 쉽다. 이러한 형태의 붕괴는 주로 지표수나 침투수에 의해 발생하며, 집중호우시 많이 발생한다.
- 풍화암으로 구성된 비탈면을 깎기한 경우 깎기면의 풍화 및 침식으로 비탈면의 일부가 붕괴하는 경우도 있다.
- 파쇄가 진행된 암석과 균열이 많은 암석 또는 풍화하기 쉬운 암석을 깎기하는 경우, 건설장비에 의한 진동, 깎기에 의한 응력해방, 풍화 등의 영향으로 비탈면이 부분적으로 활동하는 경우가 있으며 이러한 형태의 붕괴는 비탈면 깎기 작업 중 또는 그 직후에 발생하는 수가 많으므로 지속적인 관찰이 필요하다.
- 투수성과 침식에 대한 저항이 다른 지층으로 이루어진 비탈면에서는 차별침식으로 침식되지 않은 층이 붕락(탈락)하는 수가 있다.

(2) 활동면의 심도가 깊은 붕괴

- 편마암의 엽리와 사암, 셰일이 교대로 호층을 이루는 암반에서 층리 방향이 비탈면방향과 유사한 경우 또는 파쇄대를 동반한 단층과 큰 지질구조선이 비탈면방향으로 경사져 있는 경우 대규모의 붕괴가 발생되는 경우가 많다. 특히, 지질구조선을 따라 점토화가 진전된 장소를 깎기하는 경우에 많이 발생한다.
- 붕적토가 두껍게 덮여있는 지반을 깎기하는 경우 하단부의 암반층을 활동면으로 하여 상부퇴적물이 활동하는 경우가 있다. 이러한 유형은 이상징후가 뚜렷하지 않아 예측이 어려우며 순간적으로 발생하는 특징이 있다.
- 이러한 유형의 붕괴는 지하수의 영향을 크게 받아 집중호우뿐만 아니라 호우 종료 후에도 붕괴되는 특징을 갖는다.

(3) 깊고 붕괴지역이 넓은 형태의 붕괴

- 붕적토 퇴적물과 두꺼운 풍화층 및 단층파쇄대, 점토로 충전된 층리로 이루어진 급경사의 비탈면을 깎기하는 경우, 깊고 넓은 지역에 걸쳐 붕괴가 발생될 수 있다.
- 비탈면방향으로 경사진 점토성 지반을 깎기하는 경우, 강우에 의한 지하수위 상승으로 대규모 붕괴가 발생하는 수가 있다.
- 연약지반 상에서 깎기하는 경우, 상부 토사 또는 암괴제거에 따른 지반의 수평저항력 감소로 깊고 넓은 지역에 걸쳐 광범위한 붕괴를 일으킬 수 있다.
- 이상과 같이 깊고 붕괴범위가 넓은 형태의 붕괴유형은 활동면을 따라 천천히 진행되는 경우가 많고 비탈면상에 발생한 균열조사를 통해 붕괴 예상범위와 방향을 조기에 예측할 수 있으나 비탈면 붕괴 전 이상징후를 간과하는 경우 대규모 붕괴로 이어질 수 있으므로 유의하여야 한다.

1.4 이상 징후

- 이상징후는 붕괴 전 또는 붕괴진행 중 외형적으로 나타나는 전조현상으로 붕괴 위치나 규모를 예측하고 예방하는 중요한 단서로 비탈면 점검시 기록하여 그 형태의 규모, 추이 등을 관찰하도록 한다. 다음의 이상징후가 관찰되는 경우, 즉시 전문 기술자에게 의뢰하여 적절한 보강대책을 수립하여야 한다.

1.4.1 인장균열

- 인장균열은 비탈면의 대표적인 붕괴인지 수단이다. 인장균열이 발생하여 붕괴에 이르는 시간은 수일에서 수개월까지 변위가 진행한다. 따라서 점검자는 비탈면뿐만 아니라 상부 자연비탈면의 균열여부를 상세히 살펴야 한다.

- 일반적으로 인장균열은 붕괴부 이동방향에 수직으로 발달하며 붕괴규모에 따라 수 미터에서 수십 미터에 걸쳐 발생한다. 현장조사 시 자연적으로 포행현상에 의한 표층균열이나 터파기 배수로, 등산로 등과 구별하기가 쉽지 않으며, 수풀이나 낙엽으로 관찰이 불가능한 경우가 많으므로 관찰 시 세심한 주의가 필요하다. 인장균열이 발생한 경우에는 단층, 습곡의 존재, 암종이나 염기성 조암광물의 분포상태, 엽리나 기타 불연속면의 분포, 차별풍화 정도 등을 상세히 관찰하고, 균열의 규모, 위치, 깊이, 방향 등을 조사하여 개략적인 붕괴규모나 양상을 예측 할 수 있다. 인장균열이 관찰되고 붕괴위험성이 높다고 판단되는 비탈면에 대해서는 비닐포장, 천막 등으로 인장균열부분과 붕괴예상 비탈면을 피복하고 추가적인 강우의 침투를 방지하여 활동력이 증가되지 않도록 해야 한다.



<그림 1.5> 인장균열(도로비탈면 유지관리 실무매뉴얼, 2018)

1.4.2 배부름 현상과 침하

- 비탈면 이상징후로 상부 자연비탈면의 인장균열이 발생되고 활동면이 형성되면 이 활동면 하단깎기 표면에 균열과 변위가 발생한다. 이를 배부름 현상이라고 한다. 배부름 현상이 발생된 부근에서는 낙석이나 표층붕괴가 발생하는데 이 경우 일반적인 낙석이나 표층붕괴로 인지하고 대수롭지 않게 여기는 수가 있으므로 세심한 주의가 필요하다.



<그림 1.6> 배부름 현상과 침하(도로비탈면 유지관리 실무매뉴얼, 2018)

1.4.3 시설물 변형

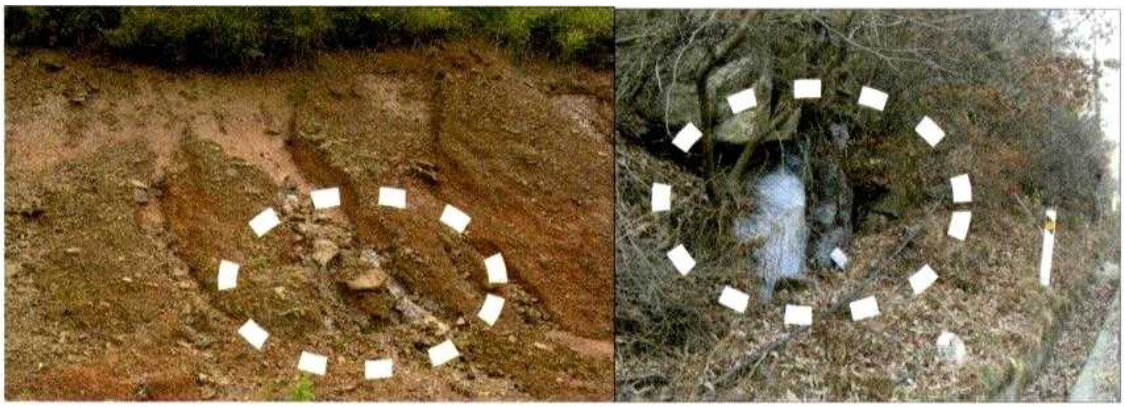
- 비탈면에 설치된 보호 · 보강시설에 발생한 균열, 파손 등의 결함은 자연지반에 비해 뚜렷한 변위양상을 나타내는 특징을 갖는다. 따라서, 점검시 각종 재질로 이루어진 비탈면시설에 발생한 균열의 진행성향, 위치, 깊이, 범위 등의 기록 · 관찰을 통해 비탈면의 이상거동을 추정할 수 있으므로 점검에 있어서 이러한 사항을 세심히 관찰하도록 한다. 특히, 여름철 수풀이 우거진 비탈면에서 이러한 현상을 확인하기는 쉽지 않으므로 해빙기 점검시 기록 · 표시하였다가 수풀이 우거지는 우기대비 점검시 기록자료로 활용하여 유지관리 점검이 효율적으로 이루어 질 수 있도록 한다.



<그림 1.7> 좌: 돌망태 옹벽 배부름 변형, 우: 격자블록 파손(비탈면 가이드북, 2008)

1.4.4 급격한 지하수 용출

- 평상시 비탈면의 지하수 유출지점이나 유출량, 시간에 따른 유출 변동량을 기록하여 두면 급격한 유출량 변동에 대해 대처할 수 있다. 급격한 유출량의 변화나 용출수의 탁도가 짙어지는 경우는 비탈면의 이상징후 일 수 있으므로 신속히 전문가의 소견을 구하고 대처 방안을 강구하여야 한다. 또한, 비가 오지 않은 건기에 지하수의 용출을 지속적으로 발생 하면 적절한 배수시설을 적용하여 지하수를 원활히 배출시킬 수 있도록 해야 한다.



<그림 1.8> 좌: 지하수 용출, 우: 겨울철 동결(비탈면 가이드북, 2008)

1.4.5 비탈면 식생 변화

- 비탈면 식생 변화는 비탈면 손상부위를 대략적으로 알려주고 파괴여부를 시사해 줄 수 있으며, 지표 및 지하수의 배수상태 변화를 알려줄 수 있으므로 점검시 기록하여 차회 점검 시 상대지표로 검토하여야 한다.

- 주로 습한 지역에 분포하는 이끼류, 갈대류의 식생군이 비탈면 상부지역에 새롭게 발생한 경우, 지하수위의 상승이 원인이 될 수 있으므로 이에 대한 추가 조사와 수평 또는 지하 배수 시설을 고려하는 방법을 강구하여야 한다. 응급대책방법으로 유출수를 배제하는 경우에 있어서 손상된 비탈면 방향으로 유출수를 배출하는 것은 피해야 한다.

1.5 현장 조사

• 현장 조사는 비탈면의 지반상태, 비탈면형상 등을 파악하여 물리적, 기능적 결함 및 위험요인을 발견하고 이에 대한 보수 · 보강 방법을 제시하기 위하여 실시한다.

<표 1.1> 비탈면점검 현장조사 내용(건설공사 비탈면 유지관리지침, 2006)

분 류	현장조사내용
상부 자연비탈면	토석류 발생가능성 조사 자연비탈면의 인장균열 자연비탈면상의 폐목 철탑, 묘지, 참호 등의 인공구조물 등
파괴징후	인장균열 또는 이완 암괴의 존재와 상태 비탈면 지반의 변형과 진행성 비탈면시설 또는 인접한 시설물의 이상 등
파괴현황	비탈면에서 발생한 파괴의 규모 및 개소 파괴 발생 위치 발생한 파괴의 유형등
지반상태	비탈면에서 발생 토사비탈면 : 토질조건, 토층 심도율 암반비탈면 : 지반강도, 불연속면(절리)면의 방향성 및 위치, 풍화도, 불연속면의 분포특성 등
비탈면형상	비탈면의 규모 및 경사 집수지형 등
자연적 요인	강우 및 지하수 조건 비탈면내 수목의 기울어짐 비탈면 식생의 변화 등
인위적 요인	표면보호 상태 비탈면 표면 상태 보호 · 보강시설의 상태 배수상태 등
조사 및 시험	물리탐사 시추조사 토질 및 암석역학시험 등

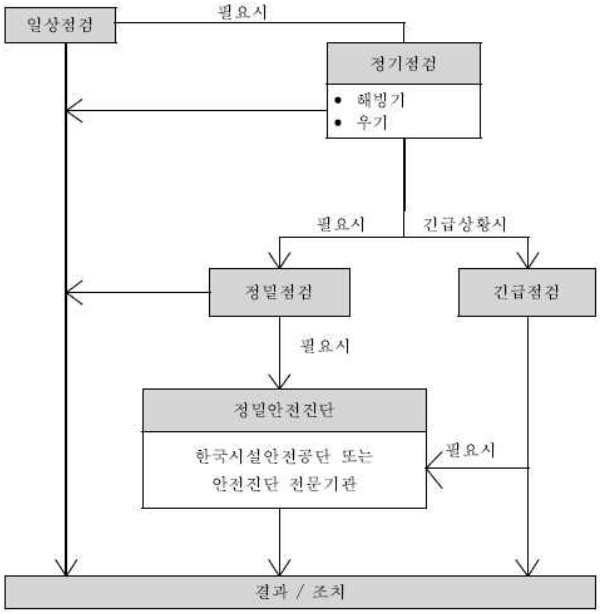
제2장 비탈면 관리 지침 현황

• 비탈면 시공 또는 유지관리시 기존 방법으로는 육안조사, 지표면에 설치된 계측기기 지점만 점검, 계측 관리되는 한계점을 가지고 있다.

2.1 절토사면 유지관리 매뉴얼 보완 · 개정 연구(2013.12), 한국시설안전공단

• 지표면에 설치된 계측기기로 지표 변위량 계측으로 설치 위치, 수량 등 계측 관리 한계가 있다.
• 전문가 육안조사시 접근 불가 위치, 소요 시간, 전문인력 등 점검 한계가 있다.
• 급경사 또는 대깎기 비탈면의 경우 식생, 경사에 의해 비탈면 지표면, 비탈면 상단 계측이 불가하다.

4.4 깎기비탈면의 계측관리
: 지표 변위량은 지표면 신축계, 지표면 경사계, 광파측량기 등



제5장 깎기비탈면의 점검 및 진단 : 정기점검, 정밀점검, 긴급점검, 정밀안전진단으로 나뉨
: 정기점검 - 반기에 1회 이상 실시, 세심한 육안검사 수준의 점검으로서 간단한 점검장비(카메라, 필리도구, 줄자, 망치 등)
: 정밀점검, 긴급점검, 정밀안전진단 : 근접조사를 위한 접근장비 사용 가능, 측량기 사용, 외관 조사

5.3 현장조사 및 시험법
: 깎기비탈면 측량 : 외관 조사를 실시한 결과, 상태 평가 또는 안정성 평가를 위해 정밀한 비탈면 형성이 필요한 경우 비탈면에 대한 측량을 실시
: 광파거리계에 의한 측량기를 사용하며 측정간격은 20~100m 범위에서 비탈면의 형상이나 손상정도에 따라 위치를 선정하여 측량
: 측점별 횡단 측량은 보통 20m마다 1개소씩 측량하며, 단면 변화구간 및 육안조사 단면의 이상 부위는 보다 정확한 측량을 실시

2.2 건설공사 비탈면 설계기준(2011.12), 국토해양부

• 지상 계측, 계측기기에 의한 계측 한계에 대한 언급이 없지만 계측목적에 부합하는 계측기 선정이 중요함을 언급하고 있다.

> 5.8 깎기비탈면의 계측
> : 계측계획 수립 : 비탈면의 변형이나 붕괴형태를 사전에 예상하고 계측목적에 부합하는 계측기 선정이나 배치, 계측방법, 관리기준치 설청 등을 검토하여 계측계획 작성
> : 계측기준 : 계측기의 배치는 비탈면의 붕괴 및 활동 특성, 지형적 위치, 계측기 설치 편의성, 계측기의 관리 편의성, 비용 등을 고려하여야 하며, 비탈면의 변동 상황을 최소화의 계측기로 효과적으로 파악할 수 있도록 배치하여야 한다.

2.3 철도설계지침 및 편람 : KR C-02050 계측 (2017.3), 한국철도시설공단

• 지표면에 설치된 계측기기로 계측, 사진 촬영 측정 등 설치 위치, 수량 등 계측 관리 한계가 있으며, 식생에 의한 육안, 지표 계측 관리 한계도 있다.

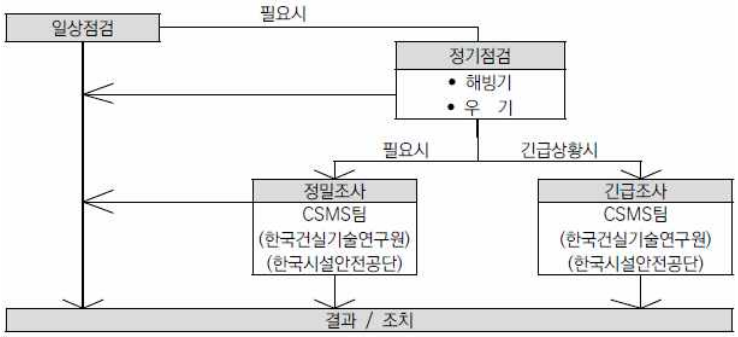
> 2. 깎기 비탈면 계측관리
> : 비탈면의 안정성을 주의 깊게 관찰하기 위해 필요한 개소에 계측을 수행하도록 설계해야 한다.
> : 암반 비탈면 변위에 대한 계측방법은 아래와 같은 방법들이 있다.
> . 단순한 기계적 변위측정 : 지표의 암반변위를 직접계측
> . 광학적 측정 : 광학사진 촬영
> . 측지학적 측정 : 기준선, 거리측량, 삼각법 측량
> . Extensomenter : 2점 사이의 길이 변화를 측정
> . Deflectometer : 시추공의 휨 변형량 측정
> . 기울기 측정 : 기울기 변화 측정

2.4 도로비탈면 유지관리 실무매뉴얼 (2018.12), 국토교통부

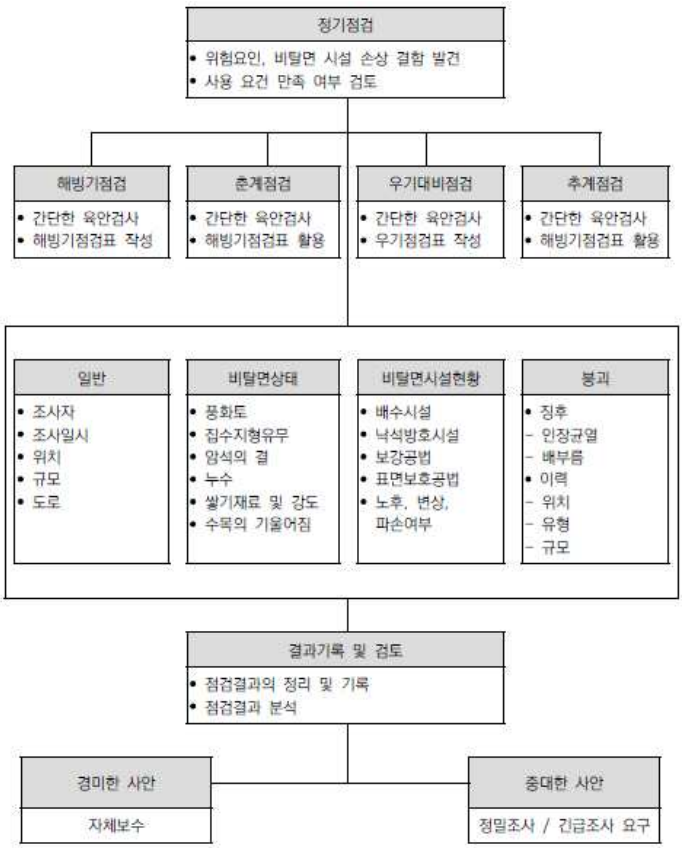
• 육안에 의한 일상점검, 정기점검 요령을 언급하고 있으며, 계측관리로는 지표면변위경사계와 강우량계를 이용한 상시계측시스템을 중요도와 특성에 적합한 비탈면으로 제안하고 있다.

제5장 도로비탈면 점검요령 및 점검요청

: 도로법 적용, 일반적인 정기적인 안전점검 흐름



: 정기점검 업무 흐름



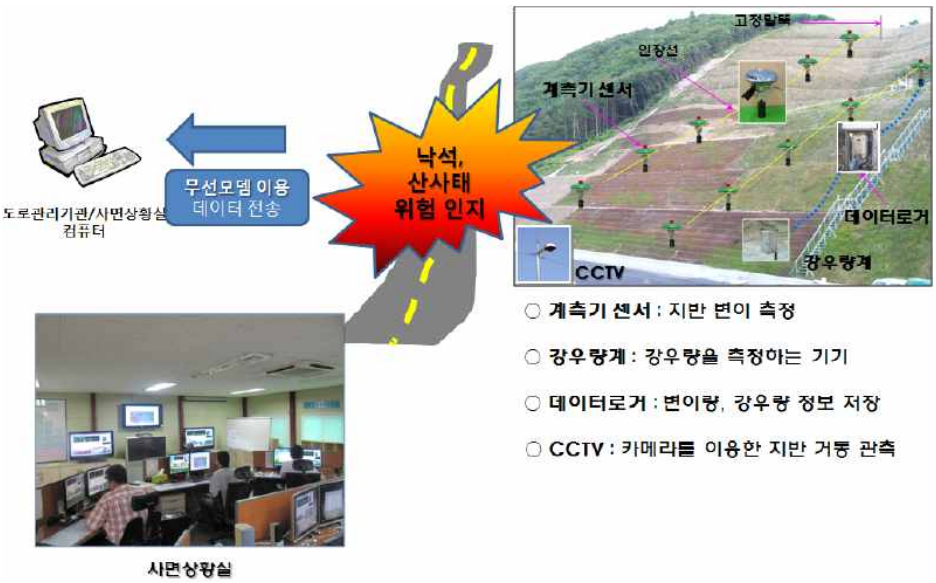
제8장 상시계측시스템

: 상부자연비탈면 경사가 급하고 높이가 40m 이상인 대규모 비탈면

: 국립공원, 문화재 보호구역 등으로 지정되어 대책공사가 어려운 비탈면

: 대책공법 적용에 한계성을 가지며 우회도로 개설이 불가능한 비탈면

: 전문가 의견에 따라 지형, 지질특성상 특별히 필요하다고 판단되는 경우 등



설치위치	장비 내용		설명	비고
현장	계측기기	지표변위경사계	센서	현장에 따라
		강우량	센서	현장에 따라
	데이터로거 (Data logger)		현장데이터 수집장치	현장당 1대
	무선모뎀		계측데이터 무선 전송장치	현장당 1대
	CCTV		현장 영상 정보 제공	현장당 1대
	유선모뎀		영상정보 데이터 유선송신	현장당 1EA
현장사무소	관리자 컴퓨터 (분석프로그램)		각 현장 관리 및 분석시스템	사무소당 1대
	무선 모뎀		무선 데이터 수신장치	사무소당 1대
	인터넷		영상정보 제공	

제3장 드론 비탈면 관리 활용지침(안)

- 비탈면 관리를 위한 방법으로는 통상 육안점검, 지상 계측기 설치를 통한 계측관리가 대표적이며, 육안점검으로는 일상점검, 정기점검으로 나뉘며, 위험도가 큰 비탈면의 경우 지표면에 계측기를 설치하여 실시간 또는 주기적 계측관리로 나뉜다.

- 대깎기 비탈면, 급경사지, 식생이 많은 곳, 계측기가 설치되지 않은 지점 등에서는 비탈면 시공 또는 유지관리시 육안조사, 지표면에 설치된 계측기기 지점만 점검, 계측 관리되는 한계점을 가지고 있다.

- 최근 들어 드론과 관련한 항법, 통신, 센서 기술 등이 비약적으로 발전하면서 광범위한 지역에 산재한 도로시설물의 효율적 점검을 위해 관리 목적에 적합한 센서(RGB카메라, 열화상카메라, 초분광카메라, 라이다, 레이더 등)를 장착한 드론이 적용될 수 있다.

3.1 이상 징후

- 이상징후(인장균열, 배부름 현상과 침하, 시설물 변형, 급격한 지하수 용출, 비탈면 식생 변화)는 붕괴 전 또는 붕괴진행 중 외형적으로 나타나는 전조현상으로 붕괴 위치나 규모를 예측하고 예방하는 중요한 단서로 육안 또는 드론을 활용하여 비탈면 점검시 기록하여 그 형태의 규모, 추이 등을 관찰하도록 한다. 다음의 이상징후가 관찰되는 경우, 즉시 전문 기술자에게 의뢰하여 적절한 보강대책을 수립하여야 한다.

3.2 붕괴가능성이 높은 비탈면 선정

(1) 붕괴이력이 있는 비탈면
- 과거 붕괴이력이 있는 비탈면, 붕괴 재발 가능성이 높음.
- 특히 지질요인이 문제가 높은 현장은 평상시 세심한 주의 필요.

(2) 지형적인 취약부에 위치하는 비탈면
- 계곡이나 집수지형을 포함한 경우, 호우기시 지표수 영향 반복 영향
- 계곡부는 지질학적으로 취약한 단층, 파쇄대 영향 큼.
- 테일러스, 붕적층로 노출되는 지형은 붕괴에 취약
- 지형적으로 취약한 부분은 상대적으로 풍화도가 높음.

(3) 비탈면 주변에서 이상현상이 발견되는 현장
- 비탈면 하단부의 도로포장면이 융기되어 있는 현장
- 비탈면 하단부 L형측구, 옹벽 등에서 균열이나 변위가 있는 현장
- 전신주나 가드레일이 기울어져 있는 경우
- 도로포장체에서 종방향 균열이나 안행상 균열이 발달하는 현장
- 상부 자연비탈면의 지표면 침하, 균열 등의 변위가 관측되는 현장
- 비탈면의 지표면 배부름 현상, 침하 등의 변위가 관측되는 현장

(4) 어두운 색의 암이 다소 포함된 현장
- 어두운 색이 많은 부분은 풍화에 대한 저항성이나 강도가 약함

- 녹색 계열의 암색은 열수변질작용의 영향으로 지반강도가 약함

(5) 주위암과 다른 층의 분포
- 점토층(단층점토), 파쇄대, 관입암, 다른 암의 협재(탄질셰일, 토상흑연 등)와 같이 기반암과 다른 상태의 암이 함께 존재하는 경우

(6) 평소 축축한 지반
- 평상시 지하수의 이동이 빈번한 지역
- 비탈면의 상시로 젖어 있거나 유출수가 흐르는 현장
- 겨울철 수평배수공을 따라 결빙 현상이 심하여 빙벽이 형성되는 현장
- 호습성 식물(대나무, 버드나무, 삼나무, 무화과나무, 이끼류 등)의 풍부하게 자라는 현장
- 열화상과 같은 센서로 물의 흐름에 의한 국부적인 지표면의 열 변화가 있는 비탈면

(7) 흙막이 공사 현장
- 상부 비탈면 아래로 흙막이 공사 또는 급경사 굴착하는 현장
- 옹벽 구조물 아래로 흙막이 공사 또는 급경사 굴착하는 현장

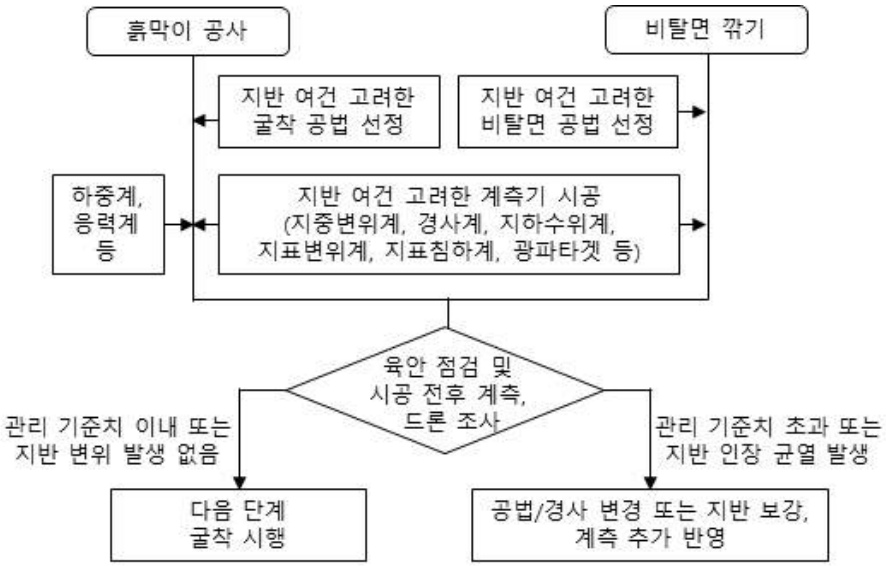
3.3 현장조사

<표 3.1> 비탈면점검 현장조사 내용

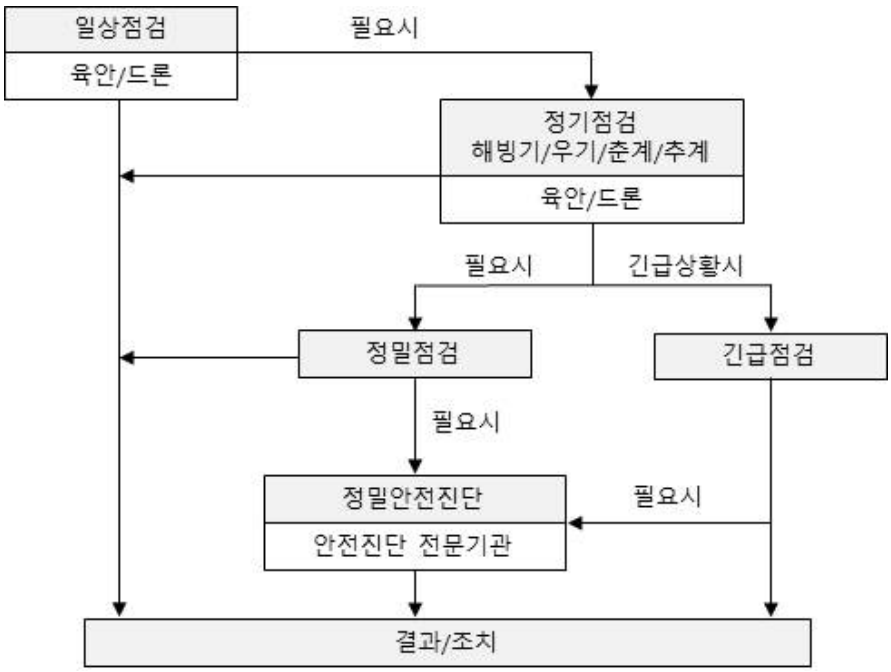
분 류	현장조사내용
상부 자연비탈면	토석류 발생가능성 조사 자연비탈면의 인장균열 자연비탈면상의 폐목 철탑, 묘지, 참호, 옹벽 등의 인공구조물 등
파괴징후	인장균열 또는 이완 암괴의 존재와 상태 비탈면 지반의 변형과 진행성 비탈면시설 또는 인접한 시설물의 이상 등
파괴현황	비탈면에서 발생한 파괴의 규모 및 개소 파괴 발생 위치 발생한 파괴의 유형 등
지반상태	비탈면에서 발생 토사비탈면 : 토질조건, 토층 심도율 암반비탈면 : 지반강도, 불연속면(절리)면의 방향성 및 위치, 풍화도, 불연속면의 분포특성 등
비탈면형상	비탈면의 규모 및 경사 집수지형 등
자연적 요인	강우 및 지하수 조건 비탈면내 수목의 기울어짐 비탈면 식생의 변화 등
인위적 요인	표면보호 상태 비탈면 표면 상태 보호·보강시설의 상태 배수상태 등
조사 및 시험	물리탐사 시추조사 육안조사 드론조사(RGB카메라, 열화상카메라, 초분광카메라, 라이다, 레이더 등) 토질 및 암석역학시험 등

3.4 점검

• 대깎기 비탈면, 급경사지, 식생이 많은 곳, 계측기가 설치되지 않은 지점 등에서는 비탈면 시공 또는 유지관리시 육안조사, 지표면에 설치된 계측기기 지점만 점검, 계측 관리만으로는 확인하지 못한 곳이 발생하므로, 적합한 센서(RGB카메라, 열화상카메라, 초분광카메라, 라이다, 레이더 등)를 장착한 드론을 활용하여 조사 점검하도록 한다.



<그림 3.1> 공사중 관리 점검 업무 흐름도



<그림 3.2> 공사후 관리 점검 업무 흐름도

3.5 계측관리

- 일반적으로 비탈면을 시공하는 공사는 지반의 비선형적인 거동특성, 실험실에서의 현장상황 재현에 따른 제약, 각종 조사 및 실험방법상의 기술적 한계성 등으로 인하여 설계시 많은 불안요소를 내포하게 된다. 따라서 설계단계의 이러한 불확실성으로 인한 문제점을 시공 및 운용 중에 발견하여 설계 및 시공에 반영함으로써 공사의 안정성과 경제성을 도모하고자 현장계측을 수행한다.
- 계측이 필요하다고 인정되는 시설물에 대하여는 위치 및 개소를 선정하여 정기적으로 계측을 시행하고 그 기록(계측위치, 계측기기의 종류, 계측결과의 값, 위치별 개소가 표기된 도면 등)을 보관하여야 한다. 정밀한 센서 드론을 이용하여 전범위의 대상영역 모두를 확인할 수 있다.
- 계측을 실시한 경우에는 시설물 계측지점을 표시 보존함으로써 연계성 있는 계측을 할 수 있도록 도면과 계측지점을 일치시켜 기록 보존한다. 특히, 붕괴징후가 관찰되는 등 안전성에 심각한 문제를 야기할 수 있는 시설물에 대해서는 정기적인 계측결과를 검토 분석함으로써 지속적인 유지관리가 반드시 이루어져야 한다.
- 정밀안전진단 결과 계측이 필요한 것으로 판단되는 깎기비탈면에 대하여서는 계측하고자 하는 항목과 내용에 따라 적절한 계측기를 설치하여야 한다.

<표 3.2> 계측 항목 내용

계측 항목	계측 내용	계측 기기	비고
지표	지표변위량	지표면 신축계 지표면 경사계 드론 측량기 등	드론 센서 (RGB카메라, 열화상카메라, 초분광카메라, 라이다, 레이더 등)
지중수평변위	지층내 수평이동	지중 수평변위계 등	
배수	강우량, 간극수압	우량계 지하수위계 간극수압계 등	

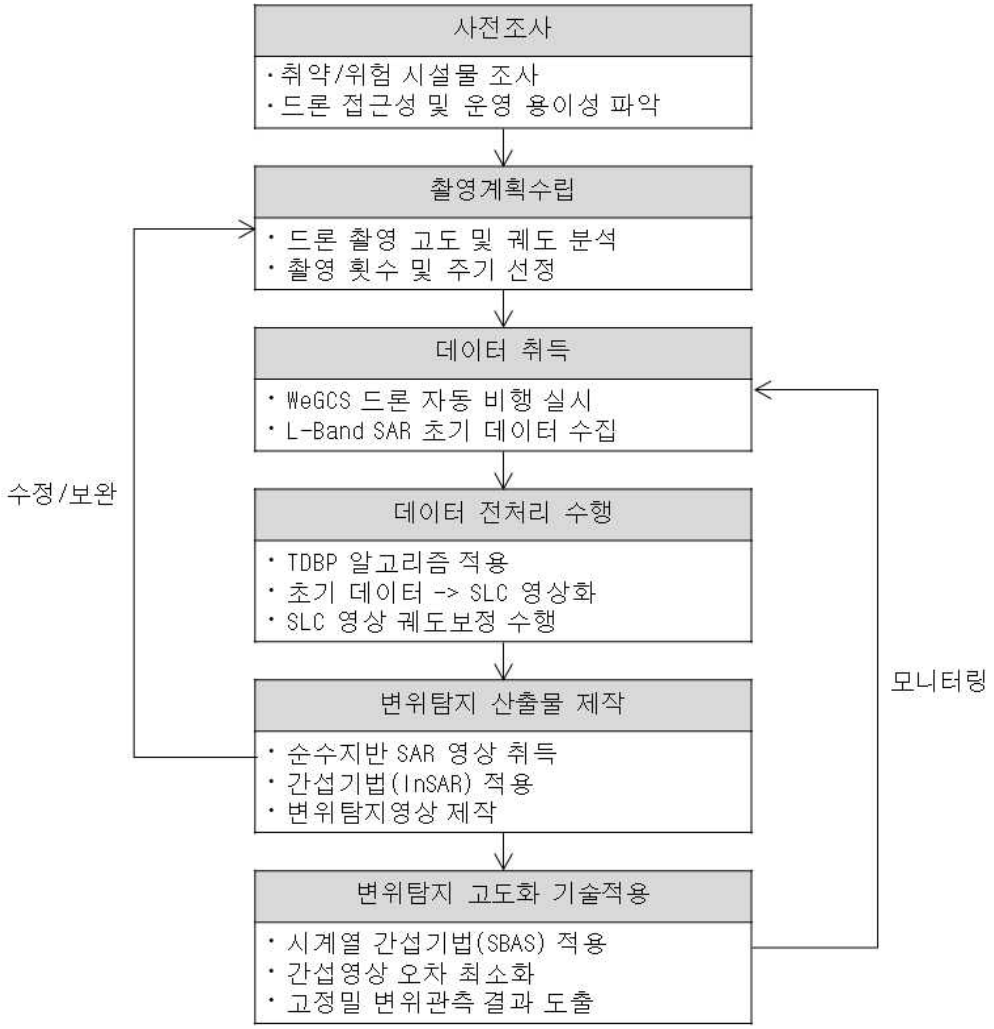
<표 3.3> 붕괴 형태와 계측기기 적합성

계측기기명 및 계측방법 / 비탈면 붕괴형태 / 주요한 계측대상항목	응력개방현상	붕락현상	평면활동붕괴	쐐기변형붕괴	원호활동붕괴	복합활동붕괴	전도붕괴
지표면신축계	인장부 부근의 머리부 변위량			○	○	◎	◎	○
지반경사계	활동머리부, 중간부, 말단부 표면 경사량					◎	◎	△
광파측거법	지표면 변위량(수평, 수직성분)	○		△	△	△	○	△
소폭판측량	〃			△	△	△	△	△
파이프변형계	활동면위치 검출과 변형누적경향			△	△	△	△	
삽입식 지중경사계	활동면위치 검출과 암괴이동량			○	○	○	◎	△
고정식 지중경사계	〃			△	△	△	○	
지중신축계	〃			△	△	△	○	△
다단식 지중신축계	〃			△	△	△	○	△
지중변위계	지중 변위량(수평, 수직성분)	○		○	○	○	○	△
검지선식 낙석감지기	낙석 감지		◎					
전도스위치	지반, 암괴 경사량(전도, 회전)							△
GPS 측량	지표면 변위량(수평, 수직성분)						△	
지하수위계	지하수 동태 관측과 비탈면 안정도 검토	○		○	○	○	○	
간극수압계	비탈면 부군 간극수압 측정			△	△	△	△	
우량계	우량값 수집(비탈면안정과 강우 관계)	○	○	○	○	◎	◎	○
적설량계	적설값 수집(비탈면안정과 적설 관계)	△					△	
유량계	지하수 배제 효과 확인						○	
실화상 드론	식생 포함 지표면 변위량(수평, 수직성분)	△	◎	○	◎	○	○	○
L-SAR 센서 드론	식생 아래 지표면 변위량(수평, 수직성분)	◎	◎	◎	◎	◎	◎	◎

(기호설명) ◎ : 가장 적합한 것, ○ : 약간 적합한 것, △ : 적합한 경우와 적합하지 않은 경우가 있는 것(시행/연구 단계), 무기입 : 적합하지 않는 것

3.6 드론을 활용한 비탈면 변위 측정 환경 및 활용방안(업무프로세스)

• 드론 SAR 시스템을 활용한 업무프로세스는 크게 사전조사, 촬영계획수립, 데이터취득, 전처리, 산출물 제작, 고도화로 구분된다.



<그림 3.3> 드론 SAR 시스템 프로세스

(1) 사전조사

• 붕괴가능성이 높은 비탈면을 대상으로 사전조사를 통해 취약/위험한 시설물 파악함
• 선정된 대상지역의 드론 접근성 및 운영 용이성 파악함
• 사전조사 시 드론을 활용한 광학영상 촬영으로 대상지 분석 및 정밀 DSM 제작을 수행함

(2) 촬영계획 수립

• 선정된 비탈면에 상태에 따라 아래의 표와 같이 드론 촬영 고도, 궤도, 촬영 횟수, 주기 등 구체적인 촬영계획을 수립함

<표 3.4> 드론 SAR 촬영조건

구분	일반 사면 지역	위험 사면 지역
촬영거리	· 대상체와 이격 거리 100m 이상	· 대상체와 이격 거리 100m 이상
촬영횟수	· 매 촬영 당 2회 · 7회 이상 촬영수행 · 정밀 분석 시 10회 이상	· 매 촬영 당 2회 · 10회 이상 촬영수행(시계열분석)
촬영주기	· 월 1회 촬영 · 기상사항 고려	· 월 2회 촬영(2주 간격) · 기상사항 고려
촬영고도	· 지면으로부터 최소 50m 이상 · 촬영 전 드론 DSM 제작필요	· 지면으로부터 최소 50m 이상 · 촬영 전 드론 DSM 제작필요

(3) 데이터 취득

- 드론 SAR 시스템의 비행 및 데이터 취득은 WeGCS 프로그램, GAMMA Remote Sensing 프로그램을 통해 자동화된 프로세스로 적용됨
- 획득된 초기 데이터는 현장에서 노트북으로 실시간으로 확인 가능함

(4) 데이터 전처리 수행

- 원시 데이터의 지정된 채널을 추출하여 힐버트 변환 및 spectral windowing을 수행함
- 드론 궤도에 따라 측정된 각 펄스에서 위치와 속도를 계산하고 ASCII 파일로 위치와 속도를 작성함
- Time-Domain Back Projection (TDBP) SAR processor을 사용하여 영상화 및 궤도보정을 수행함
- TDBP는 제작된 DSM을 활용하여 3차원 그리드를 생성하고 ascii파일로 작성된 센서의 궤도정보와 3차원 그리드 위치정보를 이용하여 원시신호를 3차원 그리드에 계산하여 SAR 신호를 SAR SLC 영상으로 변환하는 알고리즘임

(5) 변위탐지 산출물 제작

- 생성된 SLC 영상에 간섭기법을 적용하여 변위탐지영상을 제작함
- 제작된 산출물을 확인하여 오차, 추가 수정사항 발생 시 다시 촬영계획을 수립함

(6) 변위탐지 고도화 기술적용

- 고도화된 변위탐지는 SAR SLC 영상 산출물에 시계열 변위탐지 기법인 SBAS(Small BAseline Subset)를 적용하여 대기, 지형, 궤도 오차 등을 제거하여 고정밀 변위관측 결과를 도출함
- 이를 통해 간섭영상에서 오차를 최소화하여 고정밀 변위관측 결과 도출이 가능함
- 고도화 기술은 주기적인 데이터 취득을 반복하여 대상지역에 대한 모니터링을 수행함

드론을 활용한 비탈면 및 도로포장 관리 효율화 기술 개발

포트홀 탐지 드론 운용 매뉴얼

2021.05

목차

제 1 장 드론의 구분

1.1 드론의 구분 ····· 3

제 2 장 드론의 신고

2.1 보험 가입 ····· 7
2.2 드론의 신고 ····· 10
2.3 초경량비행장치 조종자 증명 ····· 17

제 3 장 촬영계획 수립

3.1 대상지 선정 ····· 23
3.2 현장 답사 ····· 33
3.3 촬영계획 수립 ····· 36
3.4 항공촬영허가 및 비행승인신청 ····· 43

제 4 장 드론 운용

4.1 드론 운용 및 안전 ····· 61
4.2 단계별 주의사항 ····· 66
4.3 사고의 보고 ····· 77

제 5 장 포트홀 탐지

5.1 포트홀 자동 탐지시스템 ····· 81
5.2 포트홀 가시화 시스템 ····· 88

제 1 장

드론의 구분

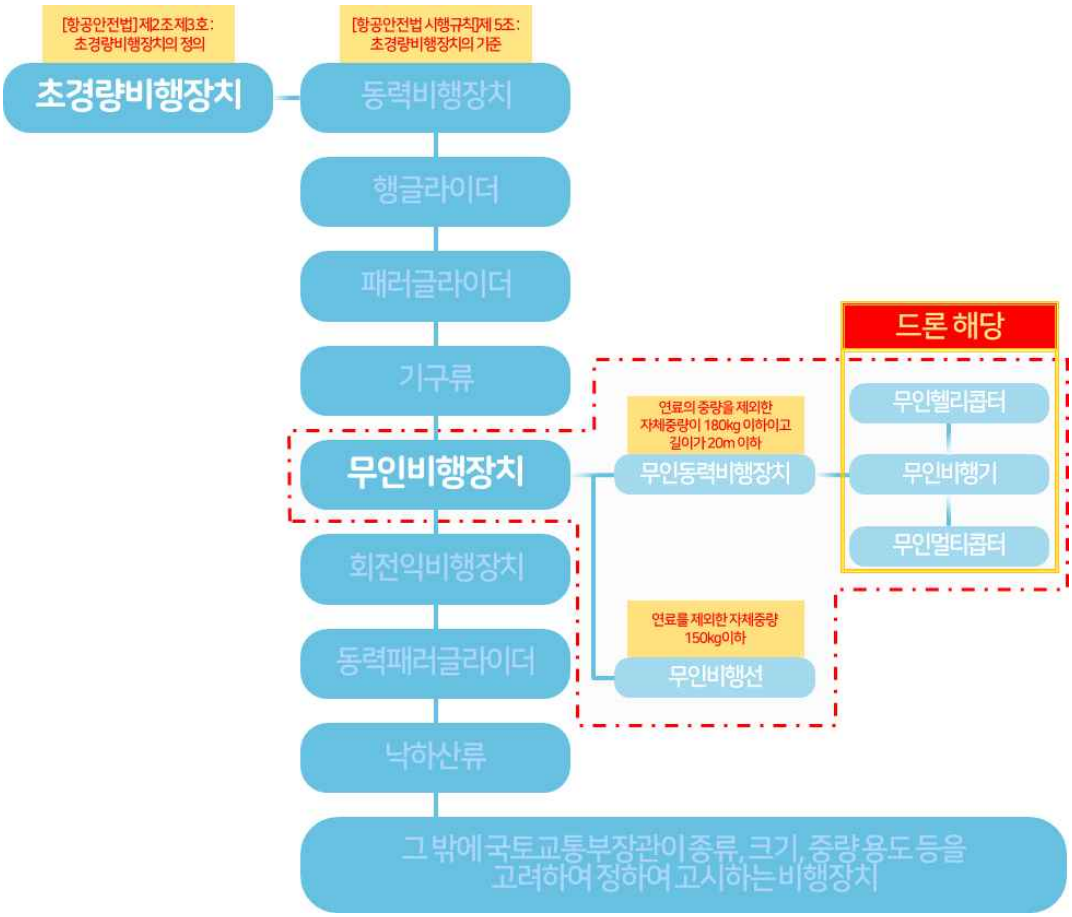
1.1 드론의 구분

제1장 드론의 구분

1.1 드론의 구분

○ 항공안전법(이하 “법”) 제2조 제3호에서 “초경량비행장치”항공기와 경량항공기 외에 공기의 반작용으로 뜰 수 있는 장치로서 자체중량, 좌석 수 등 국토교통부령으로 정하는 기준에 해당하는 동력비행장치, 행글라이더, 패러글라이더, 기구류 및 무인비행장치 등을 말한다.

○ 국토교통부령으로 정하는 “초경량비행장치의 기준”은 항공안전법 시행규칙(이하 “규칙”) 5조에 규정하고 있다. 여기서 드론은 초경량비행장치 중 무인비행장치(사람이 탑승하지 아니하는 것)로 무인동력비행장치(연료의 중량을 제외한 자체중량이 150kg 이하인 무인비행기, 무인헬리콥터 또는 무인멀티콥터)와 무인비행선(연료의 중량을 제외한 자체중량이 180kg 이하이고 길이가 20미터 이하인 무인비행선)에 해당된다.



<그림 1-1> 초경량비행장치의 분류

○ 무인비행장치는 무인동력비행장치, 무인비행선으로 분류되며, 무인동력비행장치의 세부분류로 무인비행기, 무인헬리콥터, 무인멀티콥터가 있다. 일반적으로 드론은 무인동

력비행장치에 해당된다.

○ 본 매뉴얼에서는 포트홀 탐지를 위해 주로 사용되는 무인비행기와 무인멀티콥터를 위주로 설명되었다.

<표 1-1> 무인비행장치 분류

구분	무인동력비행장치			무인비행선
	무인비행기	무인헬리콥터	무인멀티콥터	
사진				
특성	사람이 타지 않고 무선통신 장비를 이용하여 조종하거나, 내장된 프로그램에 의해 자동으로 비행하는 비행체로, 구조적으로 일반 비행기와 거의 같고 정찰, 항공촬영 등에 활용되고 있음 비행속도가 빠르며 장거리 비행 가능 활주로 또는 발사대를 이용하여 이륙	사람이 타지 않고 무선통신 장비를 이용하여 조종하거나, 내장된 프로그램에 의해 자동으로 비행하는 비행체로써, 구조적으로 일반 헬리콥터와 거의 같고, 항공촬영, 농약살포 등에 활용되고 있음 제자리 비행 가능 고정익 대비 속도, 비행거리 등이 짧음 농업방제 또는 영상촬영 등으로 활용	사람이 타지 않고 무선통신 장비를 이용하여 조종하거나, 내장된 프로그램에 의해 자동으로 비행하는 비행체로 4개이상의 전동모터로 구동되며 조종면이 없어 모터의 속도로 기체를 제어하는 것이 특징임 제자리 비행 가능 고정익 대비 속도, 비행거리 등이 짧음 농업방제 또는 영상촬영 등으로 활용 주로 소용 드론에 적용, 멀티콥터 회전익 일종임	가스기구와 같은 기구 비행체에 스스로의 힘으로 움직힐 수 있는 추진장치를 부착하여 이동이 가능하도록 만든 비행체이며 추진장치는 전기식 모터, 가솔린 엔진 등 사용됨 비행선에 가스를 주입하여 양력을 만들며, 비행속도는 느리지만 체공시간이 긴 장점이 있음

제 2 장

드론의 신고

2.1 드론 보험가입

2.2 드론 신고

2.3 초경량비행장치 조종자 증명

제2장 드론의 신고

2.1 드론 보험가입

○ 드론을 영리목적으로 사용하려는 사업자는 시행령 부칙 제 3조(초경량비행장치의 보험 또는 공제 가입에 관한 경과조치)에 따라 무인비행장치를 사용하는 항공기대여업, 초경량비행장치사용사업 및 항공스포츠사업을 등록한 경우 별표 9 제4호에 따라 보험 또는 공제되어 있어야한다. 국가, 지방자치단체, 공공기관 또한 보험 또는 공제 가입의무대상이다.

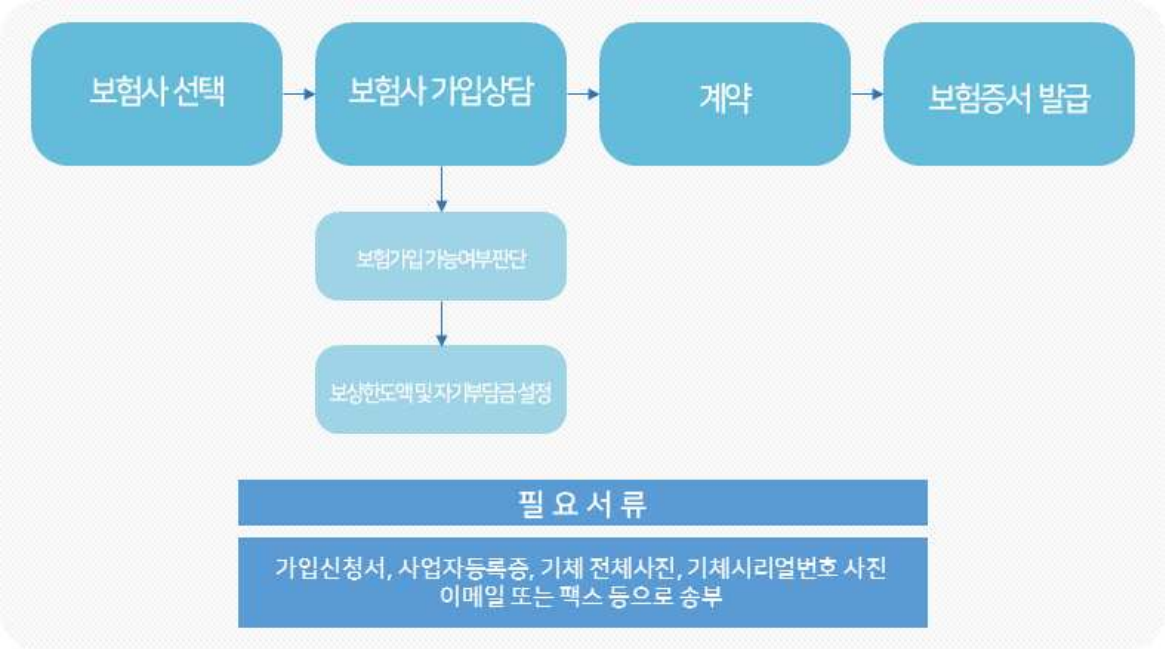
○ 보험가입을 통해 제3자 피해에 대하여 대인사고 발생 시 최소 1.5억원의 보상 금액이 보장되어야하며, 대물 피해에 대해서는 최소 2천만 원의 보상 금액이 보장되어야 한다.

<표 2-1> 항공사업법 시행령 [별표 9]

구분	기준
1. 자본금 또는 자산평가액	가. 법인 : 납입자본금 3천만원 이상 나. 개인 : 자산평가액 3천만원 이상
2. 조종자	1명 이상
3. 장치	초경량비행장치(무인비행장치로 한정한다) 1대 이상
4. 보험 또는 공제 가입	○ 다른 사람이 사망하거나 부상한 경우 피해자(피해자가 사망한 경우에는 손해배상을 받을 권리를 가진 자를 말한다. 이하 이 호에서 같다)에게 「자동차손해배상 보장법 시행령」 제3조제1항 각 호에 따른 금액 이상을 보장하는 보험 또는 공제 ○ 다른 사람의 재물이 멸실되거나 훼손된 경우 피해자에게 「자동차손해배상 보장법 시행령」제3조제3항에 따른 금액 이상을 보장하는 보험 또는 공제

2.1.1. 보험 가입 절차

○ 보험가입절차는 ①보험사 선택 ②보험사 가입상담(보험료, 보험가입가능여부, 보상한도액 및 자기부담금 설정) ③계약 ④보험증서 발급 순으로 진행된다. 보험가입 시 필요서류는 가입신청서, 사업자등록증, 기체 전체사진, 기체시리얼번호 사진으로 구성된다.



<그림 2-1> 보험가입 절차

가. 영업배상책임보험

○ 드론 보험의 한 종류인 영업배상책임보험은 드론 운용 중 다른 사람의 신체나 재물에 손해를 끼침으로써 법률상의 책임을 졌을 때 입힌 손해를 배상하는 보험이다.

○ 드론 비행 중 추락이나 충돌로 인해 대인 또는 대물 피해를 입혔을 경우 최소 이천만 원에서 최대 이십억 원까지 피해에 대한 법률상 배상책임 한도를 설정할 수 있다. 드론은 기종에 따라 다르지만 사고의 위험성이 존재하므로 보험가입 금액을 가급적 높게 설정할 필요가 있다.

1) 보험 피해 보상

○ 드론이 비행 중 추락이나 불시착하여 제3자에게 피해를 야기한 부분에 대한 대인, 대물 보상이 가능하다.

○ 사고 발생 시 드론이 파손되는 경우가 발생할 수 있으나 영업배상보험은 자산의 손실에 대한 부분은 보상되지 않는다.

2) 보험 보장 내용 예시

○ 영업배상책임보험의 보장범위 예시는 아래의 <표 2-1>과 같다.

○ 보상한도는 가입조건에 따라 달라질 수 있으며, 최소 보상한도에 대한 확인이 필요하다.

<표 2-2> 보장범위 예시

구분	보상한도	내용
대물배상	1사고당 2천~20억	1사고당 보상받을 수 있는 최고 한도액
대인배상	1인당 1.5억~20억	1명당 보상 받을 수 있는 최고 한도액
자기부담금	최소 10만원	사고가 발생한 경우 손해액의 일부를 보험회사가 아닌 보험계약자가 부담하는 금액

※ 보상한도는 가입조건에 따라 달라질 수 있으며, 단 최소금액의 확인은 필요함

나. 드론종합보험

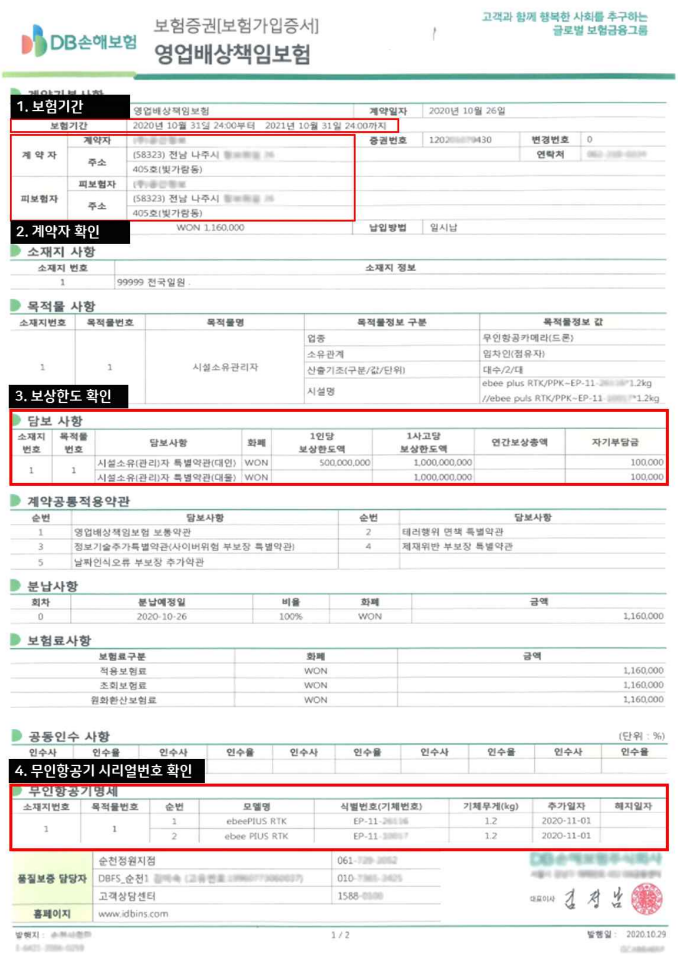
○ 드론 종합보험의 경우 영업배상책임보험의 배상뿐만아니라 기체 자체에 대한 파손을 보상받을 수 있는 보험이다. 영업배상책임보험보다 보험료가 비싸며, 보험가입 조건이 까다롭다. 그리고 현재까지는 드론종합보험을 가입하기 위한 드론 사용용도는 촬영용으로 한정되어 있으며, 각 보험사마다 자체중량 조건, 카메라 화소, 드론안전기능 탑재유무, 조종사 교육수료증 등과 같은 가입조건이 붙을 수 있다.

2.1.2 보험 가입 유무 확인

○ 드론을 운용하고자 하면 먼저 드론의 보험가입유무를 확인해야하고 현장에 투입되는 모든 드론은 보험에 필수적으로 가입되어야 하며, 이로 인해 사고 발생 시 드론조종자의 사고책임을 최소화할 수 있다. 따라서 현장책임자 및 담당자는 투입하려는 드론의 보험가입 유무를 꼭 확인해야한다.

○ 보험가입 유무 확인방법으로 보험가입증서에 기재된 S/N와 사용하려는 드론의 S/N가 일치여부와 보험종목, 보험기간, 보상한도 등을 확인해야한다. 항공사업법 2조에 따라 초경량비행장치(드론포함)를 이용하여 영리목적으로 사용하려는 사람에 대하여 보험가입 등의 의무사항을 규정되어있으므로 보험가입 완료 후에 현장에 투입되도록 한다. 이를 준수하지 않고 비행을 한다면 항공사업법 제80조2항에 의거하여 6개월 이하의 징역 또는 500만원 이하의 벌금에 처한다.

<표 2-3> 보험가입확인

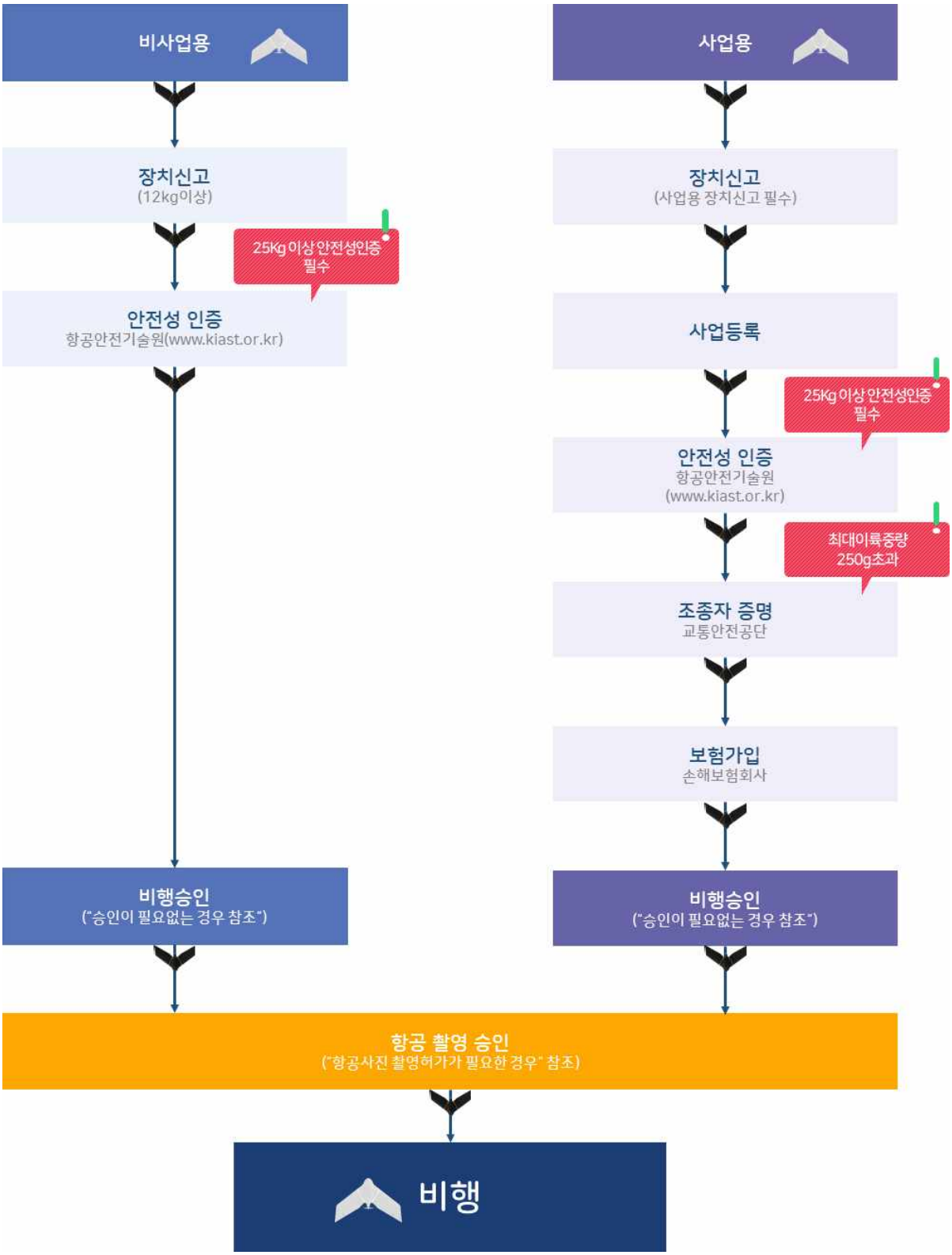
드론 S/N(시리얼넘버) 확인	보험가입증서 확인
	
	
	

2.2 드론 신고

○ 초경량비행장치 신고 목적은 체계적인 장치 신고 관리로 안전 관련 위법행위 예방 및 국민 인명 재산 보호와 기체별 고유번호 관리를 통한 향후 무인기 교통 관리, 드론 택시 택배 사용화에 기여하기 위함이다.

○ 드론 신고는 사업용(영리)과 비사업용(비영리)에 따라 장치신고 및 사업등록의 필요성이 구분되고 영리목적은 항공사업법 제2조에 따른 항공기대여업, 초경량비행장치사용사업, 항공레저스포츠사업이 사용되는 초경량 비행장치를 말한다.

○ 드론 비행 및 항공촬영허가를 신청하기 위해서는 비사업용의 경우 12kg이상, 사업용의 경우 장치신고가 필수로 되어있어야 한다.



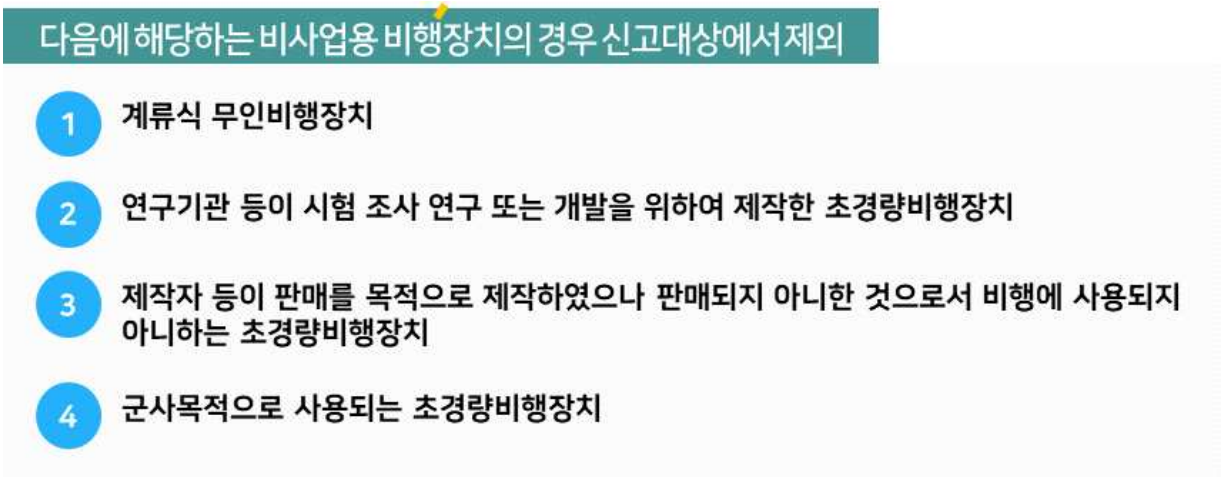
<그림 2-2> 드론 비행/항공촬영허가 절차(출처:드론 원스탑 민원서비스)

○ 초경량비행장치를 소유하거나 사용할 권리가 있는 자가 소유자 및 비행장치 정보 등을 사전에 신고하는 제도를 말하며, 신고 종류는 아래의 <표 2-4>와 같다.

○ 드론은 무게별로 완구형 모형비행장치(250g 이하), 저위험 무인비행장치(250g~7kg), 중위험 무인비행장치(7kg~25kg), 고위험 무인비행장치(25kg초과)로 분류된다.



<그림 2-3> 드론분류체계



<그림 2-4> 초경량비행장치 신고제외대상

<표 2-4> 드론 이륙중량별 비행절차

비행절차		*최대이륙중량 기준					담당기관
		250g이하	250g초과 2kg이하	2kg초과 7kg이하	7kg초과 25kg이하	25kg초과	
①장치 신고	비사업	X	X	O	O	O	한국교통안전공단 ('21.1.1 시행)
	사업	O	O	O	O	O	
②사업등록		O	O	O	O	O	지방항공청
③안전성인증		X	X	X	X	O	항공안전기술원
④조종자증명		X	O (4종)	O (3종)	O (2종)	O (1종)	한국교통안전공단 ('21.3.1 시행)
⑤비행승인**		△	△	△	△	O	지방항공청 또는 국방부
⑥항공촬영승인		O	O	O	O	O	국방부
⑦비행		조종자 준수사항에 따라 비행					

* 최대이륙중량 : 항공기가 이륙함에 있어서 설계상 또는 운영상의 한계를 벗어나지 않는 한도 내에서 최대 적재 가능한 중량

**비행제한구역 및 비행금지구역, 관제권, 고도 150m 이상 비행시에는 무게와 상관없이 비행승인 필요

최대이륙중량 25kg 초과 기체는 상시승인 필요 (단, 초경량비행장치 비행공역에서는 승인 불필요)

<표 2-5> 초경량비행장치 신고

구분	내용
신규신고	**초경량비행장치를 소유하거나 사용할 권리가 있는 자가 최초로 행하는 신고**
변경신고	**비행장치의 용도, 소유자 등의 성명이나 명칭 또는 주소, 보관 장소 등이 변경된 경우 행하는 신고**
이전신고	**비행장치의 소유권이 이전된 경우 행하는 신고**
말소신고	**비행장치의 멸실 또는 해체(정비 등, 수송 또는 보관하기 위한 해체는 제외)등의 사유가 발생한 경우 행하는 신고**

○ 초경량 비행장치 신규신고는 한국교통안전공단에서 관할하고 있으며, 신고업무는 드론 원스탑 민원서비스로 신청할 수 있다.

○ 이외의 방법으로 접수 시 사전에 공단에 문의(초경량비행장치 신고(054-459-7942~8))하면 되며, e-mail, Fax(0502-384-5453), 우편, 방문 등을 통해서도 신청 가능하다. 방문접수의 경우 공단 본사 및 전국 14개 지역본부에서 접수 할 수 있으며, 본부명과 주소는 아래의 표와 같다.

<표 2-6> 초경량비행장치 관할 지역본부

본부명	주소
본사	경북 김천시 혁신6로 17, 한국교통안전공단 드론관리처
서울	서울 마포구 월드컵로 220(성신동)
경기남부	경기 수원시 권선구 수인로 24(서둔동)
대전충남	대전 대덕구 대덕대로 1417번길 31(문평동)
대구경북	대구 수성구 노변로 33(노변동)
부산	부산 사상구 확장로 256(주례동)
광주전남	광주 남구 송암로 96(송하동)
경기북부	경기 의정부시 평화로 285(호원동)
인천	인천 남동구 백범로 357, 한국교직원공제회관 3F(간석동)
강원	강원 춘천시 동내면 춘천순환로 70(거두리) 만호빌딩 3F
충북	충북 충주시 흥덕구 사운로 386번길 21(신봉동)
전북	전북 전주시 덕진구 신행로 44(팔복동)
경남	경남 창원시 의창구 차룡로 48번길 44 창원스마트업타워 2층(팔용동)

○ 초경량비행장치 신고를 위한 구비서류는 초경량비행장치 신고서와 그 외 첨부서류인 초경량비행장치를 소유하거나 사용할 수 있는 권리가 있음을 증명하는 서류, 초경량비행장치의 제원 및 성능표, 초경량비행장치의 사진(가로 15cmX세로10cm)가 필요하다.

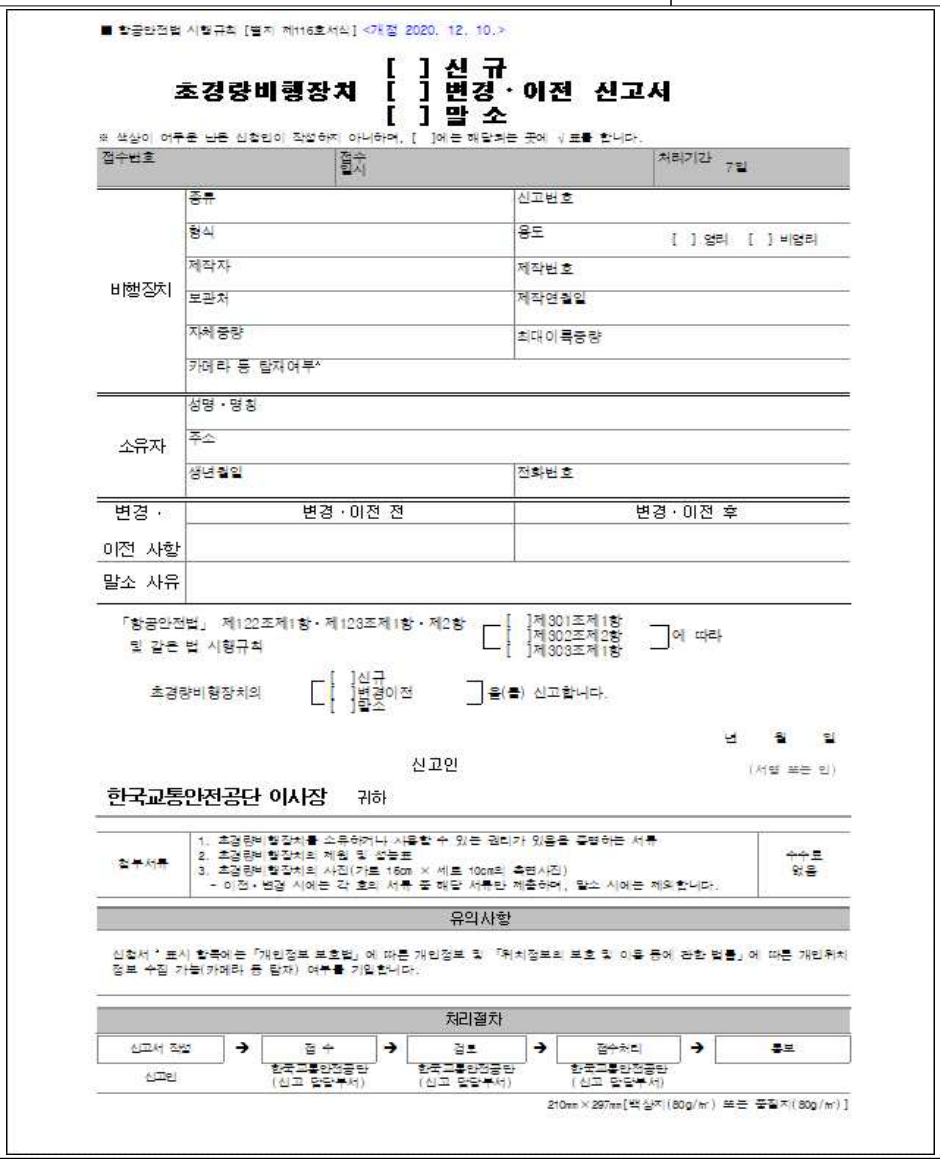
○ 신고방법은 초경량비행장치 신고서(항공안전법 시행규칙 별지 제116호 서식)와 첨부서류를 팩스 또는 드론원스탑민원서비스(https://drone.onestop.go.kr)를 이용하여 제출할 수 있다.

○ 드론 원스탑 민원 포털 서비스 홈페이지에서 회원가입 후 로그안하여 화면 중앙 상단의 민원신청 탭의 비행장치 신고메뉴를 통해 진행할 수 있다.

* 초경량비행장치 신고제외 대상은 최대이륙중량 2kg이하 비영리용도 무인동력비행장치와 연료중량을 제외한 자체중량이 12kg이하이고, 길이가 7m 이하인 비영리 용도 무인비행선이 있다.

<표 2-7> 신고 시 제출서류 및 신고시기

신고업무	제출서류	신고시기
신규신고	○ 항공안전법 시행규칙 별지 제116호 서식 신고서, 초경량비행장치를 소유하거나 사용할 수 있는 권리가 있음을 증명하는 서류(매매계약서, 거래명세서, 견적서 포함 영수증, 제작증명서 등을 첨부) ○ 초경량비행장치의 제원 및 성능표 ○ 초경량비행장치의 사진(가로15cmX세로10cm)	○ 신규신고 사유가 있는 날부터 30일 이내 *안전성인증 대상은 안전성인증 받기 전 신고
변경 및 이전 신고	○ 초경량비행장치 변경·이전 신고서 *변경 및 이전 사유를 증명할 수 있는 서류 첨부	○ 변경·이전사유가 발생한 날부터 30일 이내
말소신고	○ 초경량비행장치 말소 신고서	○ 말소 사유가 발생한 날부터 15일 이내



■ 항공안전법 시행규칙 [별지 제116호서식] <개정 2020. 12. 10.>

초경량비행장치 [ ] 신규 [ ] 변경·이전 [ ] 말소 신고서

※ 색상이 어두운 난은 신청인이 작성하지 아니하며, [ ]에는 해당하는 곳에 √표를 합니다.

접수번호	접수일시	처리기간 7일

비행장치	종류	신고번호
	형식	용도 [ ] 영리 [ ] 비영리
	제작자	제작번호
	보관처	제작연월일
	자체중량	최대이륙중량
	카메라 등 탑재여부^	
소유자	성명·명칭	
	주소	
	생년월일	전화번호

변경·이전 사항	변경·이전 전	변경·이전 후
말소 사유		

「항공안전법」 제122조제1항·제123조제1항·제2항 및 같은 법 시행규칙 [ ]제301조제1항 [ ]제302조제2항 [ ]제303조제1항 에 따라

초경량비행장치의 [ ]신규 [ ]변경이전 [ ]말소 을(를) 신고합니다.

년 월 일

신고인 (서명 또는 인)

한국교통안전공단 이사장 귀하

첨부서류	수수료
1. 초경량비행장치를 소유하거나 사용할 수 있는 권리가 있음을 증명하는 서류 2. 초경량비행장치의 제원 및 성능표 3. 초경량비행장치의 사진(가로 15cm × 세로 10cm의 측면사진) - 이전·변경 시에는 각 호의 서류 중 해당 서류만 제출하며, 말소 시에는 제외합니다.	수수료 없음

유의사항

신청서 ^ 표시 항목에 는 「개인정보 보호법」에 따른 개인정보 및 「위치정보의 보호 및 이용 등에 관한 법률」에 따른 개인위치정보 수집 가능(카메라 등 탑재) 여부를 기입합니다.

처리절차

신고서 작성	→	접수	→	검토	→	결수처리	→	통보
신고인		한국교통안전공단 (신고 담당부서)		한국교통안전공단 (신고 담당부서)		한국교통안전공단 (신고 담당부서)		

210mm×297mm[백상지(80g/m²) 또는 중질지(80g/m²)]

○ 초경량비행장치 신고 관련 위반사항으로 무인비행장치를 신고 및 변경 신고를 하지 않고 비행한 경우, 신고번호를 표시하지 않거나 거짓으로 표시한 경우, 말소 신고를 하지 않는 경우가 있다. 신고 관련 벌칙조항은 <그림 2-5>와 같다.



<그림 2-5> 신고관련 벌칙조항

2.3 초경량비행장치 조종자 증명

○ 항공안전법 제125조 1항에 따라 초경량비행장치를 사용하여 비행하려는 사람은 국토교통부령으로 정하는 기관 또는 단체의 장으로부터 그가 정한 해당 초경량비행장치별 자격기준 및 시험의 절차·방법에 따라 해당 초경량비행장치의 조종을 위하여 발급하는 증명(이하 “초경량비행장치 조종자 증명”이라 한다)을 받아야 한다. 이 경우 해당 초경량비행장치별 자격기준 및 시험의 절차 방법 등에 관하여는 국토교통부령으로 정하는 바에 따라 국토교통부장관의 승인을 받아야 햐며, 변경할 때에도 또한 같다.

○ 무인비행장치 조종자 증명은 최대이륙중량에 따라 구분되며, 운용을 위해 해당 조종자 증명이 되어야한다.

○ 항공안전법 시행규칙 제306조 제4호에서 무인비행기, 무인헬리콥터 또는 무인멀티콥터 중에서 연료의 중량을 포함한 최대이륙중량이 250그램 이하인 것, 무인비행선 중에서 연료의 중량을 제외한 자체중량이 12킬로그램 이하이고, 길이가 7미터이면 조종자 증명에서 제외한다.

○ 자격증명 업무범위는 항공안전법 제125조 초경량비행장치를 사용하여 비행 또는 조종하는 행위를 말하며, 무인동력비행장치 중 무인비행기, 무인멀티콥터, 무인헬리콥터는 각각 1종에서 4종으로 분류된다.

1. 1종 무인동력비행장치: 최대이륙중량이 25킬로그램을 초과하고 연료의 중량을 제외한 자체중량이 150킬로그램 이하인 무인동력비행장치
2. 2종 무인동력비행장치: 최대이륙중량이 7킬로그램을 초과하고 25킬로그램 이하인 무인동력비행장치
3. 3종 무인동력비행장치: 최대이륙중량이 2킬로그램을 초과하고 7킬로그램 이하인 무인동력비행장치
4. 4종 무인동력비행장치: 최대이륙중량이 250그램을 초과하고 2킬로그램 이하인 무인동력비행장치

○ 항공안전법 시행규칙 제306조에 따라 응시자격은 아래의 <표>와 같다.

○ 항공안전법 제166조제2항3조에서 125조제1항을 위반하여 초경량비행장치 조종자 증명을 받지 아니하고 초경량비행장치를 사용하여 비행을 한 사람(제161조제2항이 적용되는 경우는 제외한다)은 300만원 이하의 과태료를 부과한다.

<표 2-8> 무인비행기, 무인헬리콥터, 무인멀티콥터 조종자 응시자격

자격		연령	비행경력 또는 관련 자격 보유자		전문교육기관 이수
공통사항			- 비행경력은 안정성 인증검사, 비행승인 등의 적법한 기준 및 절차를 따른 경력을 말함 - 보통2종이상 운전면허 신체검사 증명서 또는 항공신체검사증명서를 소지해야 함		
초경량비행장치조종자	무인비행기	14세이상	1종	해당종류 비행시간 20시간 이상 (2종 무인비행기 자격소지자 15시간 이상, 3종 무인비행기 자격소지자는 17시간 이상) * 최대이륙중량 25kg 초과 ~ 자체중량 150kg 이하 무인비행기 비행경력	전문교육기관 해당 과정 이수
			2종	1종 또는 2종 무인비행기 비행시간 10시간 이상 (3종 무인비행기 자격소지자 7시간 이상) * 최대이륙중량 7kg 초과 ~ 최대이륙중량 25kg 이하 무인비행기 비행경력	
			3종	1종, 2종, 3종 무인비행기 중 어느 하나의 비행시간 6시간 이상 * 최대이륙중량 2kg 초과 ~ 최대이륙중량 7kg 이하 무인비행기 비행경력	
			4종	해당 종류 온라인 교육과정 이수로 대체 * 최대이륙중량 2kg 이하의 무인비행기에 해당 (10세이상)	해당없음
	무인헬리콥터	14세이상	1종	해당종류 비행시간 20시간 이상 (2종 무인헬리콥터 자격소지자 15시간 이상, 3종 무인헬리콥터 자격소지자는 17시간 이상, 1종 무인멀티콥터 자격소지자 10시간 이상) * 최대이륙중량 25kg 초과 ~ 자체중량 150kg 이하 무인헬리콥터 비행경력	전문교육기관 해당 과정 이수
			2종	1종 또는 2종 무인헬리콥터 비행시간 10시간 이상 (3종 무인헬리콥터 자격소지자 7시간 이상, 2종 무인멀티콥터 자격소지자 5시간 이상) * 최대이륙중량 7kg 초과 ~ 최대이륙중량 25kg 이하 무인헬리콥터 비행경력	
			3종	1종, 2종, 3종 무인헬리콥터 중 어느 하나의 비행시간 6시간 이상(3종 무인멀티콥터 자격소지자 3시간 이상) * 최대이륙중량 2kg 초과 ~ 최대이륙중량 7kg 이하 무인헬리콥터 비행경력	
			4종	해당 종류 온라인 교육과정 이수로 대체 * 최대이륙중량 2kg 이하의 무인헬리콥터에 해당 (10세이상)	해당없음
	무인멀티콥터	14세이상	1종	해당종류 비행시간 20시간 이상 (2종 무인멀티콥터 자격소지자 15시간 이상, 3종 무인멀티콥터 자격소지자는 17시간 이상, 1종 무인헬리콥터 자격소지자 10시간 이상) * 최대이륙중량 25kg 초과 ~ 자체중량 150kg 이하 무인멀티콥터 비행경력	전문교육기관 해당 과정 이수
			2종	1종 또는 2종 무인멀티콥터 비행시간 10시간 이상 (3종 무인멀티콥터 자격소지자 7시간 이상, 2종 무인헬리콥터 자격소지자 5시간 이상) * 최대이륙중량 7kg 초과 ~ 최대이륙중량 25kg 이하 무인멀티콥터 비행경력	
			3종	1종, 2종, 3종 무인멀티콥터 중 어느 하나의 비행시간 6시간 이상 (3종 무인헬리콥터 자격소지자 3시간 이상) * 최대이륙중량 2kg 초과 ~ 최대이륙중량 7kg 이하 무인멀티콥터 비행경력	
			4종	해당 종류 온라인 교육과정 이수로 대체 * 최대이륙중량 2kg 이하의 무인멀티콥터에 해당 (10세이상)	해당없음

○ 4종 무인동력비행장치(무인비행기, 무인헬리콥터, 무인멀티콥터) 조종자 증명은 항공교육훈련포털(http://www.kaa.atims.kr)에 접속하여 로그인 후 온라인 강의실을 통해 수강신청을 하고 해당 교육과정에 대한 신청 및 수강을 통해 교육이수증명서를 발급받을 수 있다.

○ 최대이륙중량 25kg 이하의 무인동력비행장치는 관제권 및 비행금지 공역을 제외한 공역에서는 150m미만의 고도에서는 비행승인 없이 비행 가능하고, 비행가능 공역, 관제권 및 비행금지구역 현황은 국토교통부에서 제작한 스마트폰 어플 Ready to Fly, 브이월드(http://map.vwold.kr/map/mps.do)지도서비스에서 확인할 수 있다.

제 3 장

촬영계획수립

3.1 대상지 선정

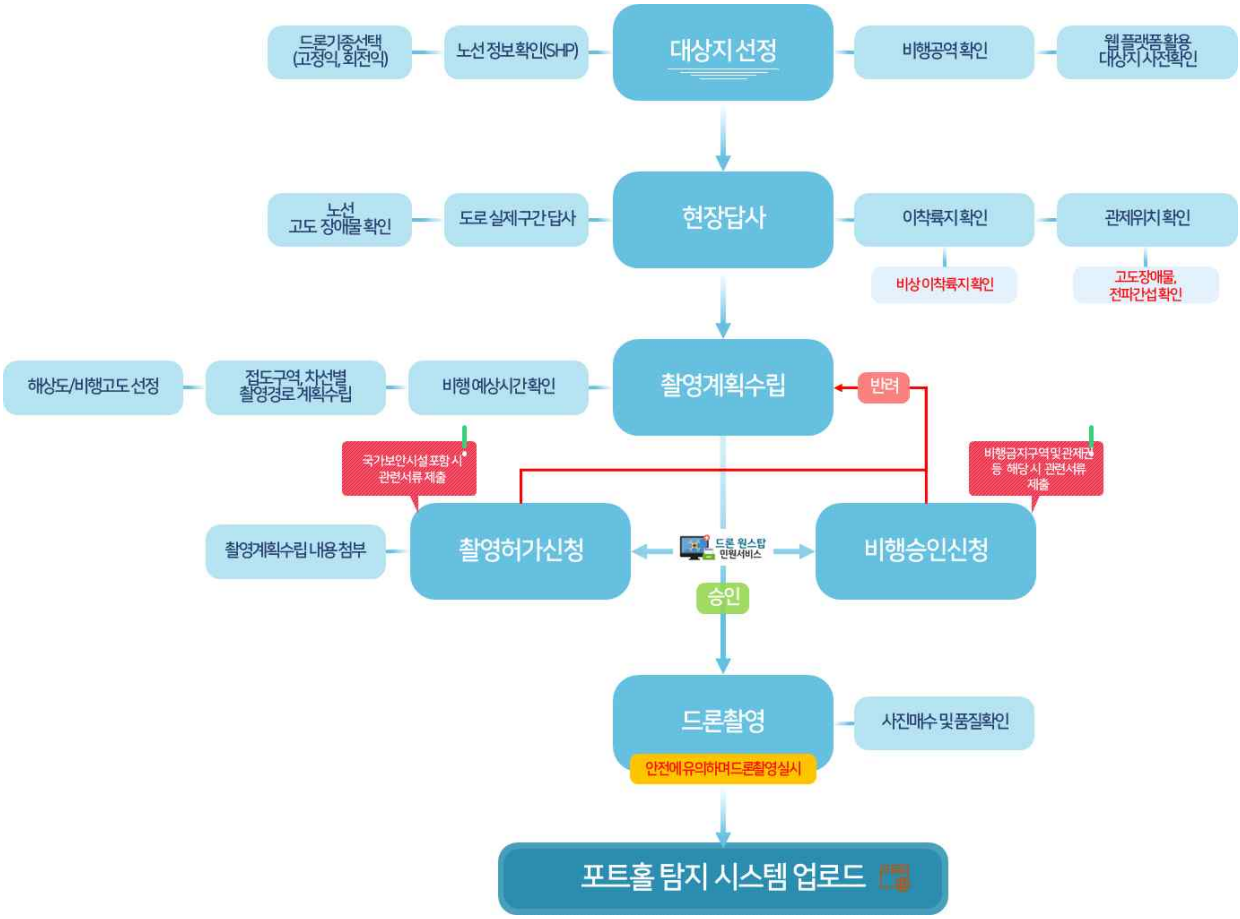
3.2 현장 답사

3.3 촬영계획 수립

3.4 항공촬영허가 및 비행승인신청

제3장 촬영계획수립

○ 포트홀 탐지를 위한 촬영은 주로 무인비행기 형식인 고정익 드론과 무인멀티콥터 형식인 회전익 드론을 활용한다. 촬영계획은 <그림 3-1>과 같이 대상지선정, 현장답사, 촬영계획수립, 촬영 및 비행승인신청, 드론촬영 순으로 이루어진다.



촬영계획수립 프로세스

3.1 대상지 선정

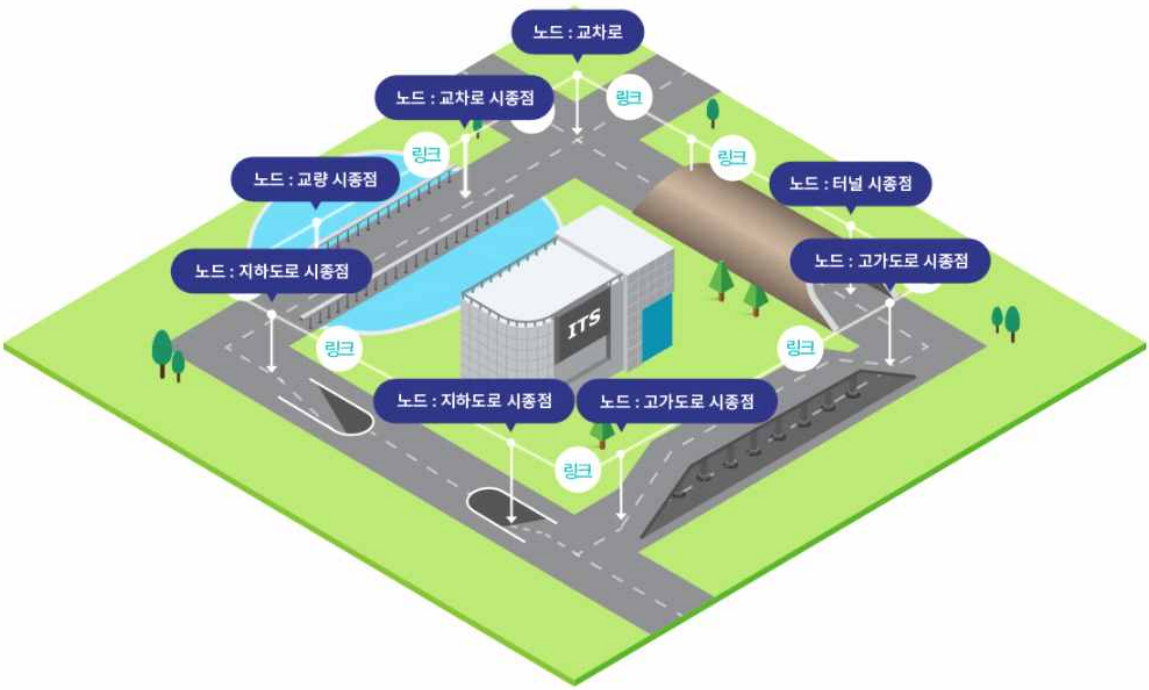
○ 포트홀 탐지를 위한 대상지 선정은 매우 중요한 업무로 대상지 선정 시 활용성, 시급성, 안정성, 지속성 등을 고려해야 한다. 또한 비행경로와 비행조건 그리고 드론 작업에 있어 대상지 선정은 매우 중요한 업무이며 활용성, 시급성, 안전성, 지속성 등을 고려해 우선순위를 선정하고, 노선길이, 비행조건, 이·착륙장소, 기상 등을 고려하고 선정하여야 한다.

3.1.1 노선정보확인

○ 대상지 선정 시 노선정보와 위치 등을 확인을 통하여 드론 비행 가능여부 판단해야하고 대상지의 정확한 위치를 인지하고 있어야 한다. 대상지의 위치와 드론 가능여부를 판단하기 위해 정확한 노선정보를 확인할 수 있어야 한다.

가. 노드링크

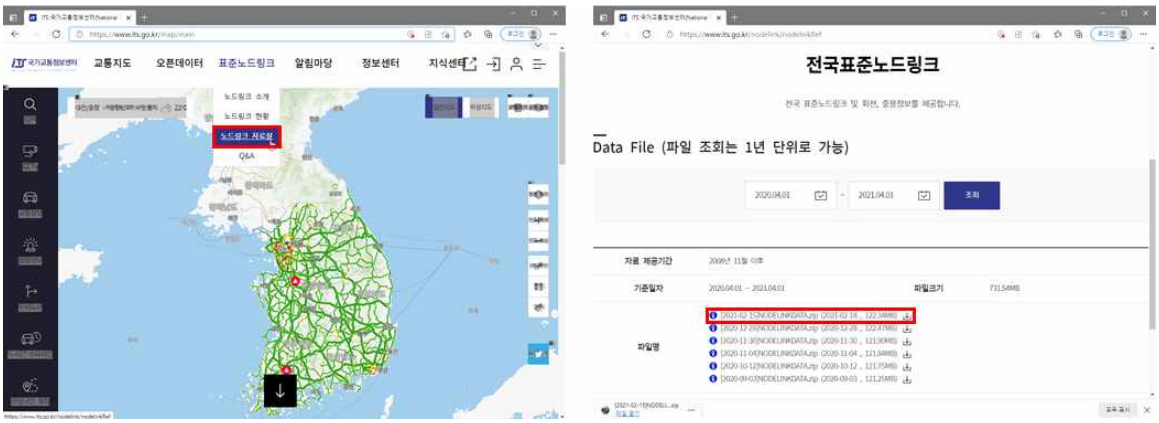
○ 노드링크 중 노드란 차량이 도로를 주행함에 있어 속도의 변화가 발생되는 곳을 표현한 곳을 말하며 예로들어 교차로, 교량시종점, 고가도로시종점, 도로의 시종점, 지하차도시종점, 터널시종점, 행정경계, IC/JC등이 있다. 링크란 속도변화 발생점인 노드와 노드를 연결한 선을 의미하며 실세계에서의 도로를 말한다. 예로들어 도로, 교량, 고가도로, 지하차도, 터널 등이 있다.



노드링크의 이해(출처 : https://its.go.kr ITS 국가교통정보센터)

나. 노드링크를 통한 노선정보 확인방법

○ 대상지에 대한 정확한 노선정보를 확인하기 위해 ITS 국가교통정보센터 홈페이지(its.go.kr)에 접속하여 전국표준노드링크를 다운받을 수 있다.



전국표준노드링크 다운로드

○ 국가교통정보통센터 홈페이지의 상단탭의 표준노드링크 탭에서 노드링크 자료실을 클릭한다. 노드링크 자료실의 가장 최신날짜 '[2021-02-15]NODELINKDATA.zip (2021-02-18, 122.34MB)의 파일을 다운받는다. 다운받은 압축파일을 해제하면 SHP가 생성되며, SHP파일은 QGIS 소프트웨어를 통해 확인할 수 있다. 또한 kml 포맷으로 변환하여 구글어스와 같은 프로그램에서도 사용할 수 있다.

<표 3-1> 전국도로 노드링크

전국도로 노드,링크	노드,링크 확대
	

○ 노드는 링크를 구분하는 점(링크, 시작점, 링크, 종료점)으로 표준노드/링크 구축 운영 지침에 따라 도로교차점, 도로 시종점, 교통통제점, 도로구조 변환점, 행정구역 변환점, 도로 운영 변환점, 교통 진출입점. 그 외에 ITS사업 주체가 필요에 따라 정하는 지점이다. 노드에 대한 테이블 설명은 아래의 표와 같다.

<표 3-2> 노드 테이블 설명

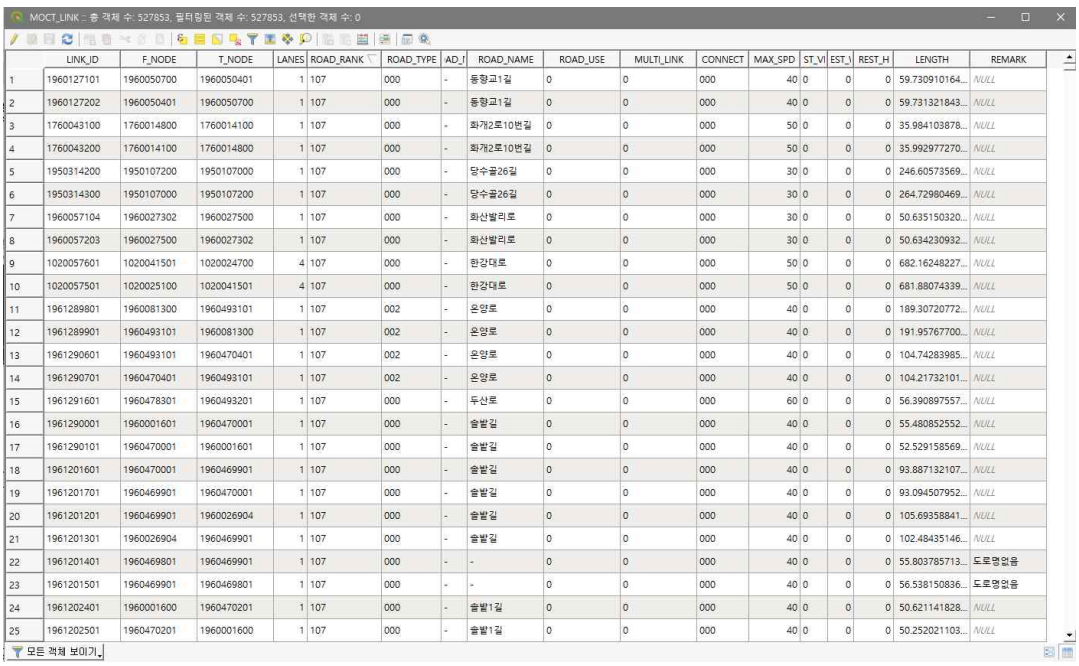
필드명	NODE_ID	NODE_TYPE	NODE_NAME	TURN_P	REMARK
속성명	노드ID 노드유형 교차로명칭 회전제한유무 비고	노드유형	교차로명칭	회전제한유무	비고

○ 링크는 노드와 노드를 이은 도로중심선을 방향별로 일정간격 이격시켜 생성한 선으로서 실제 도로구간에 대한 정보를 담게된다. 링크에 대한 테이블 설명은 아래의 표와 같다.

<표 3-3> 노드 테이블 설명

필드명	NODE_ID	F_NODE	T_NODE	LANES	ROAD_RANK
속성명	링크ID	시작노드ID	종료노드ID	차로수	도로등급
필드명	**ROAD_TYPE**	**ROAD_NO**	**ROAD_NAME**	**MULTI_LINK**	**CONNECT**
속성명	도로유형	도로번호	도로명	중용구간여부	연결로코드
필드명	**MAX_SPD**	**REST_VEH**	**REST_W**	**REST_H**	**REMARK**
속성명	최고제한속도	통과제한차량	통과제한하중	통과제한높이	`비고

○ SHP일 정보에는 각 노드에 대한 도로번호, 도로명과 길이 등을 확인할 수 있으며, 필터를 통해 원하는 도로명만 추출할 수 있다.



	LINK_ID	F_NODE	T_NODE	LANES	ROAD_RANK	ROAD_TYPE	AD_T	ROAD_NAME	ROAD_USE	MULTI_LINK	CONNECT	MAX_SPD	ST_VI	EST_N	REST_H	LENGTH	REMARK
1	1960127101	1960050700	1960050401	1	107	000	-	동향교1길	0	0	000	40	0	0	0	59.730910164...	NULL
2	1960127202	1960050401	1960050700	1	107	000	-	동향교1길	0	0	000	40	0	0	0	59.731321843...	NULL
3	1760043100	1760014800	1760014100	1	107	000	-	화개2로10번길	0	0	000	50	0	0	0	35.984103878...	NULL
4	1760043200	1760014100	1760014800	1	107	000	-	화개2로10번길	0	0	000	50	0	0	0	35.992977270...	NULL
5	1950314200	1950107200	1950107000	1	107	000	-	당수골26길	0	0	000	30	0	0	0	246.60573569...	NULL
6	1950314300	1950107000	1950107200	1	107	000	-	당수골26길	0	0	000	30	0	0	0	264.72980469...	NULL
7	1960057104	1960027302	1960027500	1	107	000	-	화산발리로	0	0	000	30	0	0	0	50.635150320...	NULL
8	1960057203	1960027500	1960027302	1	107	000	-	화산발리로	0	0	000	30	0	0	0	50.634230932...	NULL
9	1020057601	1020041501	1020024700	4	107	000	-	한강대로	0	0	000	50	0	0	0	682.16248227...	NULL
10	1020057501	1020025100	1020041501	4	107	000	-	한강대로	0	0	000	50	0	0	0	681.88074339...	NULL
11	1961289801	1960081300	1960493101	1	107	002	-	온양로	0	0	000	40	0	0	0	189.30720772...	NULL
12	1961289901	1960493101	1960081300	1	107	002	-	온양로	0	0	000	40	0	0	0	191.95767700...	NULL
13	1961290601	1960493101	1960470401	1	107	002	-	온양로	0	0	000	40	0	0	0	104.74283985...	NULL
14	1961290701	1960470401	1960493101	1	107	002	-	온양로	0	0	000	40	0	0	0	104.21732101...	NULL
15	1961291601	1960478301	1960493201	1	107	000	-	두산로	0	0	000	60	0	0	0	56.390897557...	NULL
16	1961290001	1960001601	1960470001	1	107	000	-	솔밭길	0	0	000	40	0	0	0	55.480852552...	NULL
17	1961290101	1960470001	1960001601	1	107	000	-	솔밭길	0	0	000	40	0	0	0	52.529158569...	NULL
18	1961201601	1960470001	1960469901	1	107	000	-	솔밭길	0	0	000	40	0	0	0	93.887132107...	NULL
19	1961201701	1960469901	1960470001	1	107	000	-	솔밭길	0	0	000	40	0	0	0	93.094507952...	NULL
20	1961201201	1960469901	1960026904	1	107	000	-	솔밭길	0	0	000	40	0	0	0	105.69358841...	NULL
21	1961201301	1960026904	1960469901	1	107	000	-	솔밭길	0	0	000	40	0	0	0	102.48435146...	NULL
22	1961201401	1960469801	1960469901	1	107	000	-	-	0	0	000	40	0	0	0	55.803785713...	도로명없음
23	1961201501	1960469901	1960469801	1	107	000	-	-	0	0	000	40	0	0	0	56.538150836...	도로명없음
24	1961202401	1960001600	1960470201	1	107	000	-	솔밭1길	0	0	000	40	0	0	0	50.621141828...	NULL
25	1961202501	1960470201	1960001600	1	107	000	-	솔밭1길	0	0	000	40	0	0	0	50.252021103...	NULL

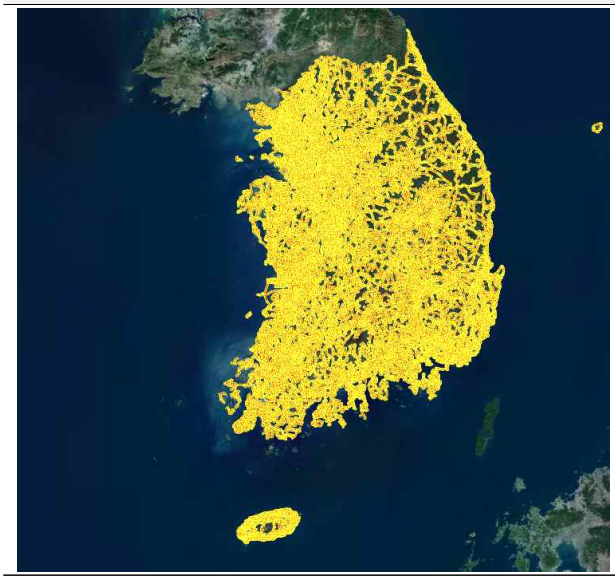
노드, 링크 테이블

○ 예로들어 GIS 소프트웨어의 필터기능을 통해 해당 도로번호, 도로명 등을 검색하고 필터링할 수 있다. 전국 노드, 링크를 통해 정확한 노선의 위치 및 방향과 길이를 확인해 볼 수 있으며, 대상지 선정에 유용하게 활용될 수 있다.



쿼리작성기를 이용한 필터링

<표 3-4> 전국도로 노드링크

필터링 전(전국도)	필터링 후(21번국도, 7번국도)
	

3.1.2 드론 기종 선택

○ 포트홀 탐지에 위해 투입되는 드론의 종류는 다양하며 일반적으로 드론은 고정익 드론과 회전익이 있다. 형식과 기종 특성별 비행거리, 비행시간, 해상도 등 사용자가 요구하는 조건에 따라 기종을 선택해야한다.

○ 앞서 선정된 노선의 길이와 대상지의 특성(지형고도, 시설물 등)을 파악 후 신속하게 투입될 수 있는 드론으로 선택해야한다. 또한 복합운용을 통해 효율적으로 포트홀 탐지에 활용할 수 있다.

○ 포트홀 탐지를 위해 중요한 여건 중 해상도의 경우 일정 해상도 이상의 요구조건을

충족시켜야 포트홀 탐지가 가능하다. 포트홀 탐지를 위한 드론 권장사양을 아래의 표에 제시하였으며, 기본성능과 탑재기능 등이 효율적으로 활용될 수 있어야 한다.

<표 3-5> 드론기종선택 권장사양

구분	권장사양	
형식	고정익	회전익
비행거리	10km이상 권장	10km미만 권장
이착륙지 여건	대상지 3km내에 40mX80m이상의 고도장애물이 없으며, 평평한 개활지 또는 운동장과 같은 공간확보	10mx10m 이상의 공간확보
카메라	20MP 이상의 기계식 셔터방식 디지털카메라(4:3 종횡비율) 사진 포맷 : JPEG 또는 PNG	
비행시간	50분 이상	20분 이상
비행방식	지상관제장치를 통해 지정된 웨이포인트 입력 또는 자동경로비행 기능 탑재	
드론 송수신거리	최소 1km이상(가시영역 내에서는 100% 연결)	
GNSS	GPS(미국), GLONASS(러시아), GALILEO(유럽연합) Beiduou(중국) 등 2개이상 수신	
자동복귀 시스템	비상 시 자동으로 착륙지점으로 복귀 또는 착륙	
무게	추락 시 안전을 위해 최대이륙중량 2kg 미만으로 권장	
지상관제장치	휴대하기 용이하고 위성맵 또는 맵을 통해 자동비행경로 지정이 가능해야하고 KML 또는 KMZ파일을 입력할 수 있어야 함	
기타	KC 전파인증받은 드론 또는 모뎀	

<표 3-6> 고정익과 회전익 드론의 비교

종류	고정익 드론	회전익 드론
사진		
특징	▸ 추력 및 양력발생장치가 분리 전진방향으로 기속을 얻으면 날개에서 양력을 발생하여 비행 ▸ 드론 구조가 단순 고속, 고효율 비행	▸ 수직이·착륙 및 정점 체공에 적합 ▸ 로터형, 덕트형, 여러개의 로터를 대칭으로 배치한 멀티콥터형 등으로 구분함
속도	▸ 비교적 높음	▸ 비교적 낮음
촬영면적	▸ 비교적 넓음	▸ 비교적 좁음
비행시간	▸ 비교적 긴 비행시간	▸ 비교적 짧은 비행시간
정밀	▸ 보통~양호	▸ 양호~우수
장점	▸ 1회 비행에 넓은 지역을 촬영 가능 ▸ 추락으로 인한 손상규모가 작음 ▸ 수리 및 관리 용이	▸ 이·착륙에 많은 공간을 차지하지 않음 ▸ 감시나 관찰을 위한 호버링이 가능 ▸ 수직비행이 가능하여 도시지역과 해안 수직암반에서도 사용이 용이 ▸ 물체에 근접시켜서 근접촬영이나 비행이 가능 ▸ 다양하고 중량감 있는 탑재체를 장착 가능
단점	▸ 넓은 이·착륙 공간 필요 ▸ 경사사진 취득 불가 ▸ 다양한 탑재체의 장착이 다소 어려움 ▸ 자유로운 각도조절이 어려움 ▸ 사진중복촬영 시 바람의 영향이 받음	▸ 복잡하고 많은 부품들이 사용되어 관리 힘듬 ▸ 짧은 배터리 수명 ▸ 체공시간 제한적 ▸ 넓은 지역 촬영이 어려움 ▸ 추락 시 기체구조와 무게, 재질로 인한 위험성
성과물	▸ 스틸사진, 정사영상, DSM, 3차원 모델 등	▸ 스틸사진, 경사사진, 정사영상, DSM, 3차원 모델, 3차원 건물모델, 동영상 등

3.1.3 비행공역의 확인

○ 비행 전 항공촬영 및 비행승인 신청을 위해 비행공역을 확인하여야 하며, 공역확인 방법은 스마트폰으로 확인하는 방법, 비행지도를 통하여 확인하는 방법 등이 있다.

가. 스마트폰 공역확인

○ 스마트폰 앱(Ready to Fly)을 통하여 비행공역 확인이 가능하다. 한국 드론 협회에서 제작한 'Ready To Fly' 앱은 공역정보 조회(전국에 설정된 공역 현황), 지역정보(비행승인 기관 연락처 등), FAQ(드론 관련 필요정보, 질의응답 등), 조종자 준수사항, 기상정보(일출·일몰시각, 풍속, 지구자기장 정보 등), 비행승인신청서 등 비행허가에 필요한 자료를 제공한다.

○ 'Google Play', '앱스토어'에서 'Ready to Fly'를 검색 후 다운받아 설치할 수 있다.



Ready to Fly 앱 설치 및 기능(출처 : 국토교통부)

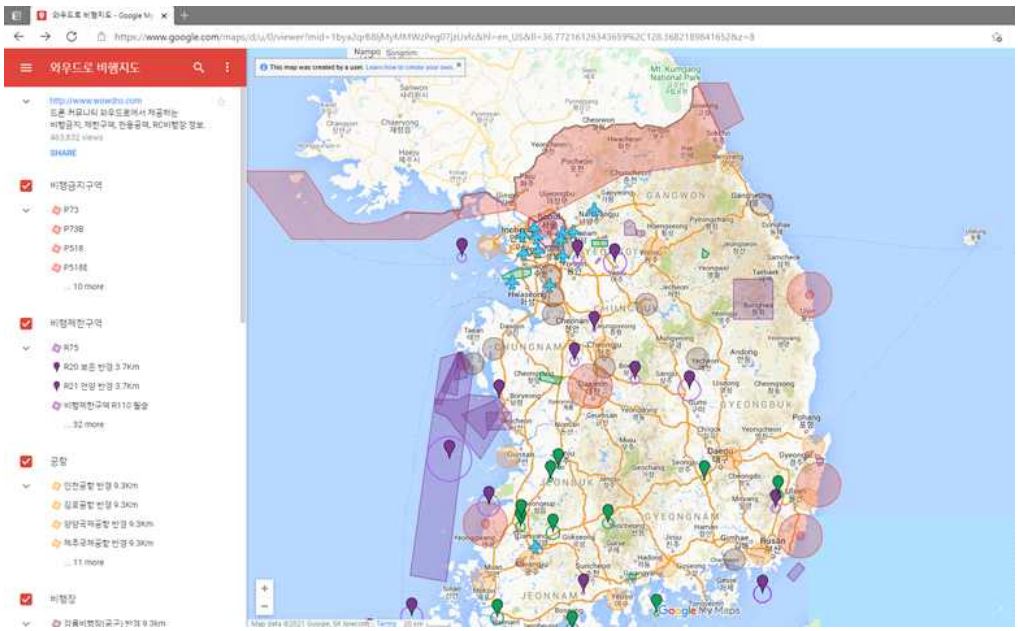
나. 비행지도 공역확인

1) 와우드로 비행지도

○ 공역에 대한 간편한 검색은 '와우드로 비행지도'를 활용한다. 비행금지구역 및 비행제한구역 그리고 관제권 등을 확인할 수 있다.

○ 와우드로 비행지도를 확인하는 방법은 검색엔진에 와우드로 비행지도를 검색하여 접속할 수 있으며, 'https://www.google.com/maps/d/viewer?mid=zXKS1gnPHvvg.ksl6JGBmFkok'에 접속하여 확인할 수 있다.

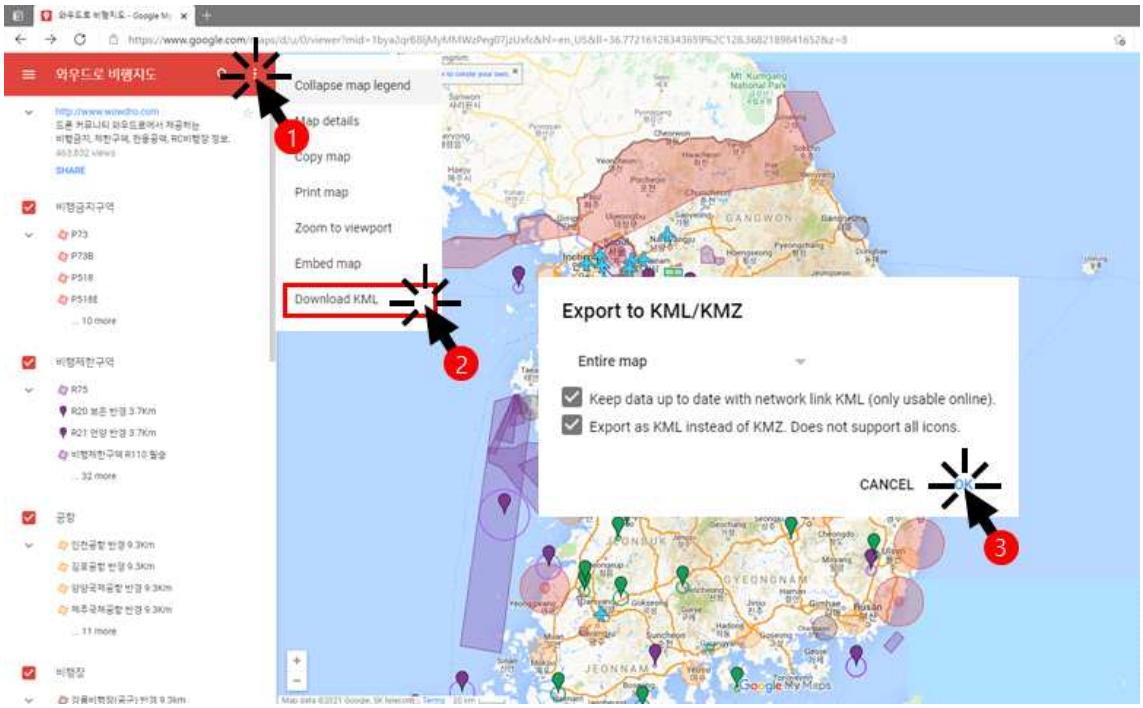
○ 와우드로 비행지도의 왼쪽에는 비행금지구역, 비행제한구역, 공항(관제권), 비행장(관제권) 등과 같이 레이어로 구성되어있으며, 필요에 따라 레이어를 끌 수 있다.



와우드로 비행지도

○ 또한 와우드로 비행지도를 KML파일로 다운받을 수 있으며, 구글어스 또는 GIS소프트웨어 함께 공역을 확인해볼 수 있다.

○ 와우드로 비행지도를 다운받는 방법은 아래의 표와 같이 와우드로 비행지도에 접속 → 오른쪽 상단 돋보기 아이콘 옆의 ':' 아이콘을 클릭 → Download KML을 클릭한다. GIS 소프트웨어에 입력된 비행공역표시는 아래의 <그림 3-9>과 같다.



와우드로 비행지도 QGIS 입력

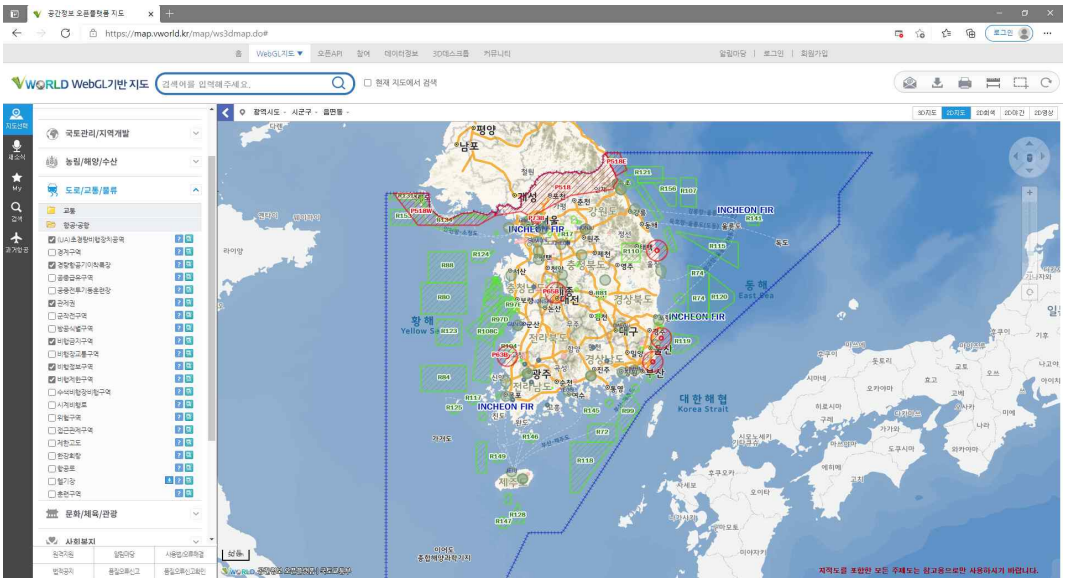


와우드로 비행지도 QGIS 입력

2) 브이월드 비행지도

○ 브이월드 지도를 확인하는 방법은 포털사이트에서 '브이월드'를 검색하거나 웹브라우저의 주소입력창에 https://map.vworld.kr/map/ws3dmap.do를 입력하여 접속할 수 있다.

○ 브이월드 오른쪽 상단의 2D지도를 선택 후 화면왼쪽 메뉴에서 도로/교통/물류 탭을 클릭한다. 세부 레이어로 항공.공항을 선택하여 체크박스를 클릭 후 원하는 공역을 선택하여 확인할 수 있다.



브이월드 비행지도

3.2 현장답사

○ 드론 운용에 앞서 현장답사는 필수사항으로 대상지 선정과 안전한 촬영이 될 수 있도록 현장답사를 실시한다. 현장답사는 주로 이·착륙지의 위치와 환경, 그리고 대상지 주변의 고도 장애물, 위급상황 시 착륙할 수 있는 비상착륙지 등을 파악해야한다. 비행 또는 운용 여건에 따라 드론 기종을 선택해야한다.

3.2.1 이·착륙지 및 관제위치 확인

○ 현장답사에서 가장 먼저 확보해야 할 것은 드론의 이·착륙 공간으로 안전한 이·착륙 공간의 확보가 중요하다. 드론의 가장 많은 사고가 이·착륙 시 발생되므로 최적의 공간으로 선점해야한다. 그리고 드론 비행 시 원활한 관제가 가능한 위치를 확인해야한다.

○ 먼저 안전한 이착륙지는 개활지, 공원, 운동장과 같은 넓으면 넓을수록 좋으며, 이착륙지 주변 고도 장애물이 존재하는지 파악해야한다.

○ 드론관제위치의 경우 GPS신호 단절, 드론과 지상관제국의 수신 단절 등의 제약사항이 생기므로 관제위치 주변에는 드론이 시야에 들어올 수 있도록 고도장애물이 존재하지 않아야 한다.

<표 3-7> 관제위치

관제원활 위치	관제불가 위치
	

가. 고정익 드론

○ 모든 방향으로 40m×80m거리 이상 범위 내에 장애물(건물이나 바위, 전력선, 언덕이나 나무 등)이 없는 곳을 이·착륙지로 선정한다. 고정익 드론은 바람의 영향을 많이 받으므로 이·착륙 시 바람에 의해 추돌할 가능성을 최소화하기 위해 충분한 공간의 확보가 필수적이다.

○ 이·착륙지로는 학교 운동장, 대형 주차장, 공원, 개활지, 수확이 끝난 논이나 밭 등 평평한 지형이 적절하다.

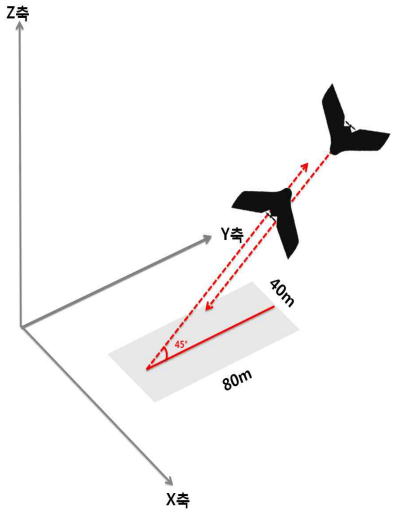
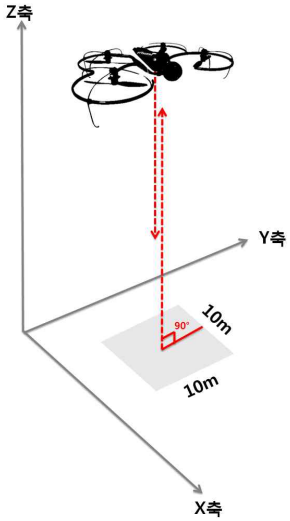
○ 착륙지로는 가급적 표면이 잔디나 수풀이 있는 곳이 좋다.

나. 회전익 드론

○ 모든 방향으로 10m×10m거리 이상 범위 내에 장애물(건물이나 바위, 전력선, 언덕이나 나무 등)이 없는 곳을 이·착륙지로 선정한다. 고정익에 비해 좁은 면적의 이·착륙 장소를 요구하지만 추락 시 안전에 대비하여 여분의 공간 확보가 필요하다.

○ 학교 운동장, 대형 주차장, 수확이 끝난 논이나 밭 등이 적합한 이·착륙 장소이지만 불가피할 경우 차량통행이 없는 도로나 단층건물의 옥상 등 하늘이 보이는 지역을 활용할 수 있다. 단, 조종자의 숙련된 비행능력과 경험이 요구된다.

<표 3-8> 이·착륙지 방향 및 공간

고정익 이·착륙	회전익 이·착륙
	
• 모든 방향 또는 풍향의 반대방향에 고도장애물(나무, 펜스, 고층건물 등)이 없는 40m×80m이상 공간 • 표면조건은 잔디, 수풀, 아스팔트 등 평평한 곳	• 모든 방향으로 고도장애물(나무, 펜스, 고층건물 등)이 없는 10m×10m이상 공간 • 표면조건은 평평한 모든 곳 단 먼지가 발생하지 않아야 함

<표 3-9> 이·착륙장소

		
운동장	공원	개활지

3.2.2 비상 착륙지 확인

○ 드론이 비행 중 고장이나 예기치 못한 사고로 인하여 이·착륙지로 복귀하지 못하는 상황이 발생할 수 있기 때문에 대상 노선 주변의 비상 착륙지를 확인한다.

○ 비상 착륙지 또한 이·착륙지와 동일한 조건을 충족해야하고, 비행경로와 근접한 곳으로 선점해야하며, 비상사태 발생 시 인명과 재산의 피해가 발생하지 않도록 운용 안정성을 확보하여야 한다.

3.3 촬영계획수립

○ 촬영계획수립 전 확인된 현장답사를 통해 선정한 이착륙 장소와 관제위치를 확인하고 대상 노선 주변에 고도장애물(아파트, 송전탑, 나무 등)이 존재하는지 확인해야한다. 또한 접도구역의 안전한 비행을 위해 비행경로 설정법을 제시하였다.

3.3.1 해상도에 따른 비행고도 산정

○ 비행 전 원활한 촬영을 진행하기 위해 세부 비행 및 촬영계획 수립 전 비행고도에 따른 해상도 산정을 실시해야한다. 포트홀 탐지를 위해 적정 해상도는 약 1cm 내외로 촬영해야 포트홀 탐지가 용이하며, 주변여건을 고려 후 최대한 고도를 낮춰 고해상도로 촬영해야한다.

○ 비행고도에 따른 해상도 산정방법은 2,000만화소 카메라 기준으로 정리하였으며, 아래의 <표 3-10>에 제시하였다.

○ 드론의 비행고도는 ATO(Above Take Off)를 기준으로 나타나며, 지형의 고저차가 일정하지 않는 경우 구글어스나 수치지형도 등을 참고하여 비행고도를 산정해야한다.

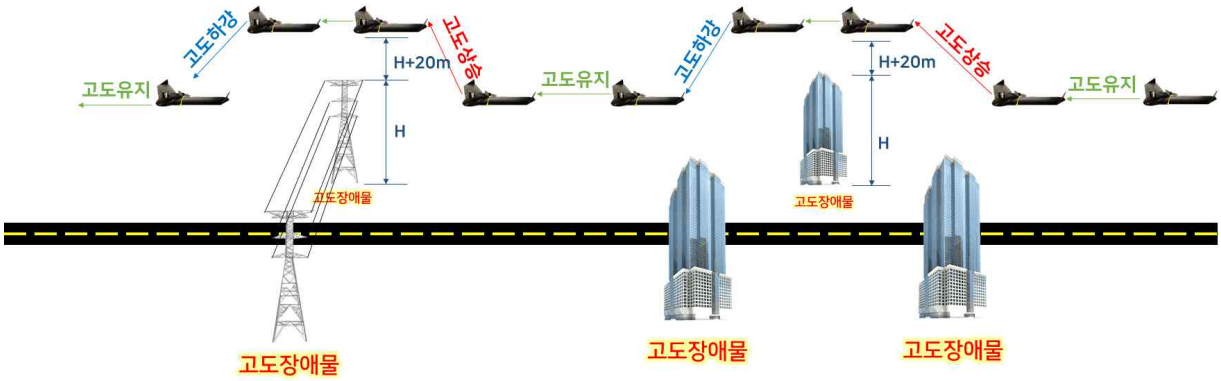
<표 3-10> 비행고도에 따른 촬영영역 및 해상도 산정(20MP 기준)

해상도(cm/px)	고도(m)	가로(m)	세로(m)	면적(㎡)
1	37	54.72	36.48	1,996.19
1.5	55	82.08	57.72	4,737.66
2	73	109.44	72.94	7,982.12
2.5	92	136.80	91.20	12,476.16
3	110	164.16	109.44	17,965.67

Sensor Height	7.22mm	Focal Length	8.80mm	Image Height	3,078

해상도(GSD)=(Sensor Height*Flight Height*100)/(Focal Length*Image Height)

○ 드론 비행고도가 고도 장애물의 높이보다 낮을 경우 고도 장애물에 부딪혀 추락하는 사고가 발생할 수 있으며, 재산 또는 인명피해로 이어질 수 있다. 고도 장애물로 인한 사고를 막기 위해서 고도 장애물의 높이보다 20m 더 높게 고도를 산정해야하고 고도 장애물이 없는 경우 최대한 고도를 낮춰 고해상도 사진을 얻어야 한다.



고도장애물 유무에 따른 비행고도 산정

나. 도로 폭 및 접도구역에 따른 비행경로 설정

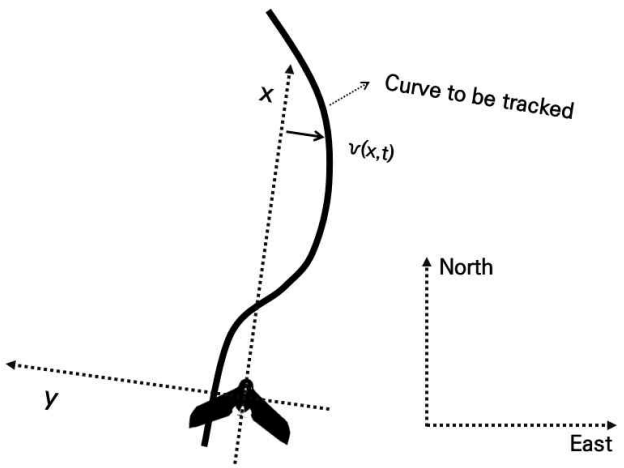
○ ‘가. 해상도에 따른 비행고도 산정’을 통해 앞서 선정한 해상도를 적용하여 도로 폭과 접도구역에 따른 안전한 비행경로를 설정해야한다.

○ 드론추락을 대비하는 비행경로 설정은 도로 갓길의 상공을 비행하여 추락시에도 주변 피해를 최소화 할 수 있도록 안전을 최우선으로 해야한다.

○ 비행경로를 설정하기 전 비행고도에 따른 카메라의 촬영면적을 확인해야한다. 또한 도로의 폭에 따라 비행방법을 단일경로(1Line) 또는 다중경로(2Line)으로 비행한다.

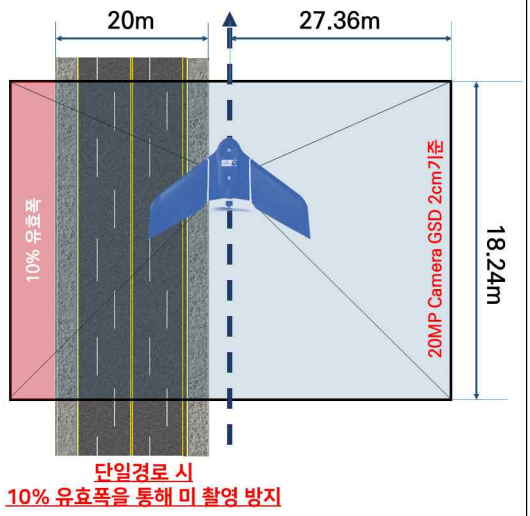
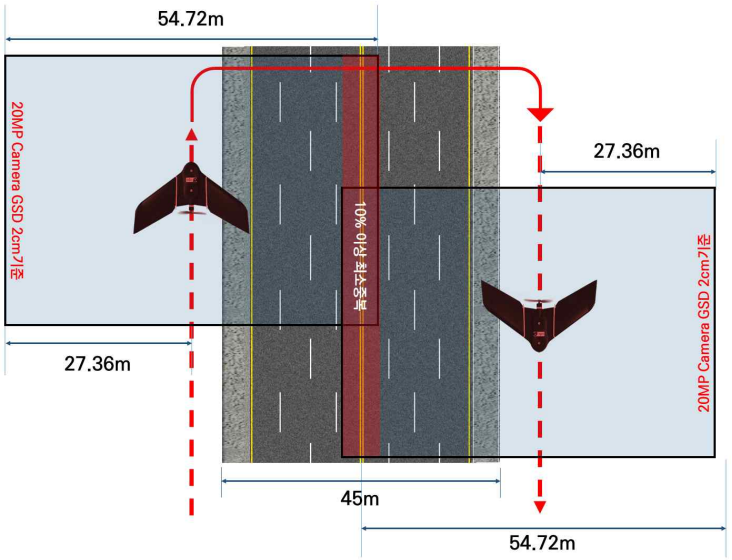
○ 단일경로(1Line)일때는 비행방향의 우측통행을 원칙으로하며, 최대한 도로와 이격하여 비행하도록 한다. 다중경로(2Line)일때에도 우측통행을 원칙으로하며, 촬영영역은 최소 10%이상 중첩되도록 한다.

○ 도로의 구조·시설 기준에 관한 규칙 제10조 2호를 참고하여 도로 폭 기준을 확인하였으며, 비행경로 설정 시 도로의 폭을 참고하여야한다.



드론 접도구역 비행

<표 3-11> 도로폭에 따른 비행경로

1~6차선 비행경로(단일경로)	8~16차선 비행경로(다중경로)
	

<표 3-12> 도로폭에 따른 비행경로

차선	도로 폭	길어깨 폭 포함	해상도	비행경로	비고
1	3.5	없음	1cm/px 기준	1	
2	7	13		1	
4	14	20		1	
6	21	27		2	
8	28	34		2	
10	35	38.5		2	
12	42	45		2	

3.3.2 촬영 및 비행계획서 작성

○ 촬영 및 비행계획서 작성은 대상지 선정과 현장답사 이후 촬영목적, 일시·장소, 촬영계획, 장비제원, 조종자 정보, 안전대책, 대상지, 담당자, 촬영일정 등을 포함하여 촬영계획을 수립하며 <표 4-5>와 같이 작성한다.

○ 촬영계획은 효율적인 업무 추진을 위하여 촬영계획과 촬영허가 및 비행승인신청을 동시에 수행하는 것이 좋다.

<표 3-13> 촬영 및 비행계획서 작성 예시

"포트홀 탐지를 위한"

드론 촬영 및 비행 계획

□ **촬영 및 비행 개요**

○ 목 적 : 포트홀 탐지를 위한 드론촬영

○ 일 시 : 2021.00.00. 09:00~2021.00.00. 18:00 중 0일(예비일 별도)

촬영일시는 여유 있도록 기간을 설정

○ 장 소 : 대상지 2곳(촬영 대상지현황 참조) *법위는 성과물 법위보다 넓게 설정*

□ **촬영 및 비행 계획**

○ 비행고도 : 150m

○ 비행속도 : 15m/s 내외

○ 소요시간 : 2시간 내외

□ **장비 제원**

○ 항공기 기종 : 무인비행장치 중 무인비행기, 무인멀티콥터

○ 항공기 명칭 : 고정익 드론, 회전익 드론

○ 촬 영 장 비 : S.O.D.A Camera(20MP), 내장카메라(20MP)

제품명	고정익 드론	회전익 드론
형식	고정익 무인비행장치	회전익 무인비행장치
이미지		
크기	96×82×10cm	80×56×17cm
무게	700g	1800g

☐ 조종사 정보

제품명	고정익 드론	회전익 드론
조종사	홍길동	홍길서
소속/직책	도로관리청/부장	도로관리청/대리
연락처	010-1234-5678	010-8765-4321

☐ 안전 대책

○ 안전반경 설정으로 촬영지역 밖으로 벗어나지 않음

○ 위급상황 발생(강풍, 기체이상 등)시 자동으로 회귀 및 착륙

○ 안전보험(대인, 대물) 가입 및 확인

☐ 대상지

연번 및 지역		대상지 주소	이착륙 장소	노선길이
1	국도 2호선	00시~00시	운동장	377.9km
2	국도 20호선	00시~00시	밭	300km

☐ 담당자

○ 부서 / 성명 : 000부서 / 홍길북

○ 082-123-4567 / 010-7890-1234 / Korea@KK.go.kr

☐ 대상지별 촬영 일정

대상지	00월					
	25	26	27	28	29	30
1번국도						
5번국도						
10번국도						

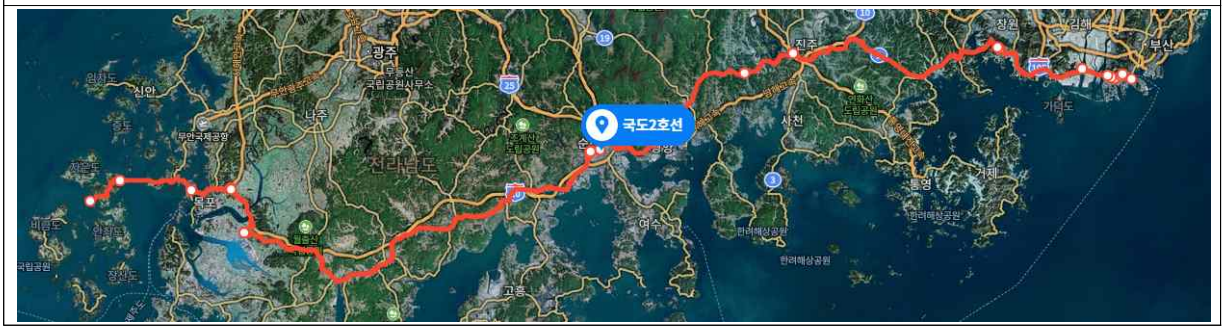
: 1차 촬영 예정일

: 예비 촬영 예정일

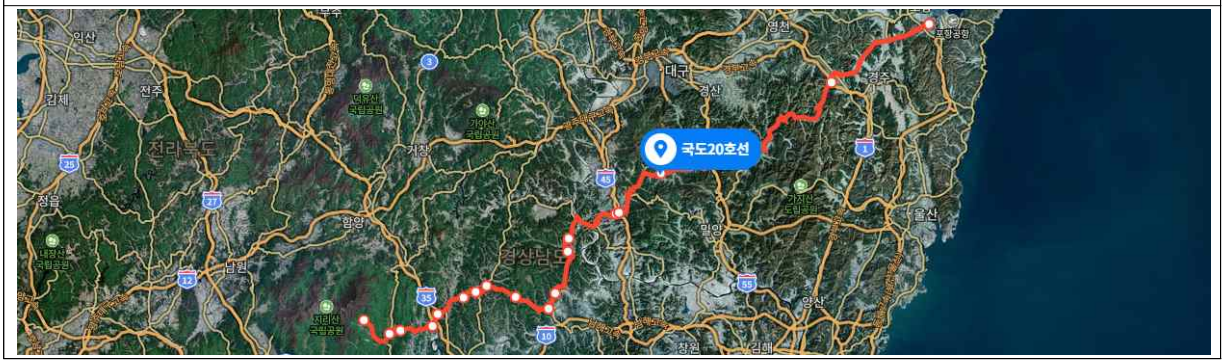
☐ 촬영 대상지현황

연번 및 지역		대상지 주소	이착륙 장소	비고
1	국도 2호선	00시~00시	운동장	
2	국도 20호선	00시~00시	밭	

2번국도



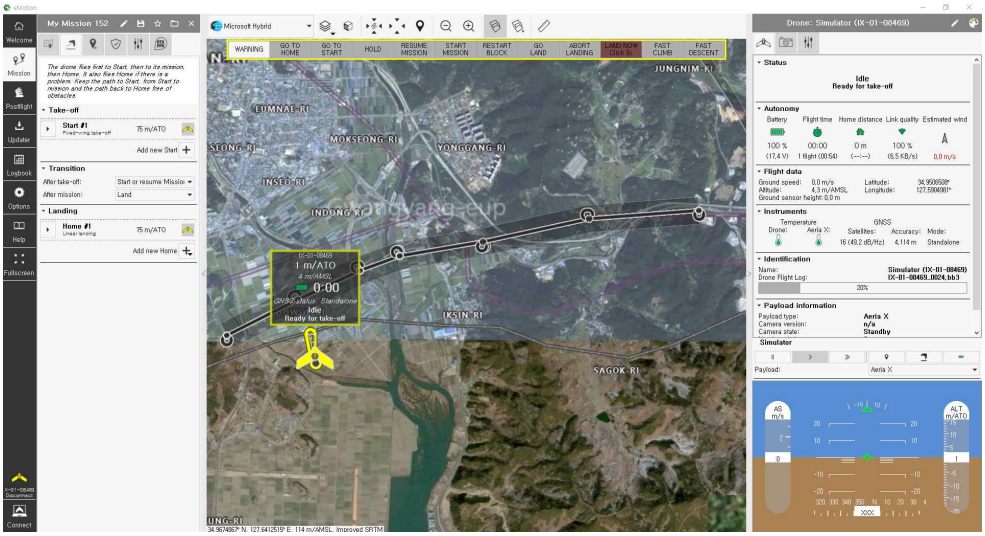
20번국도



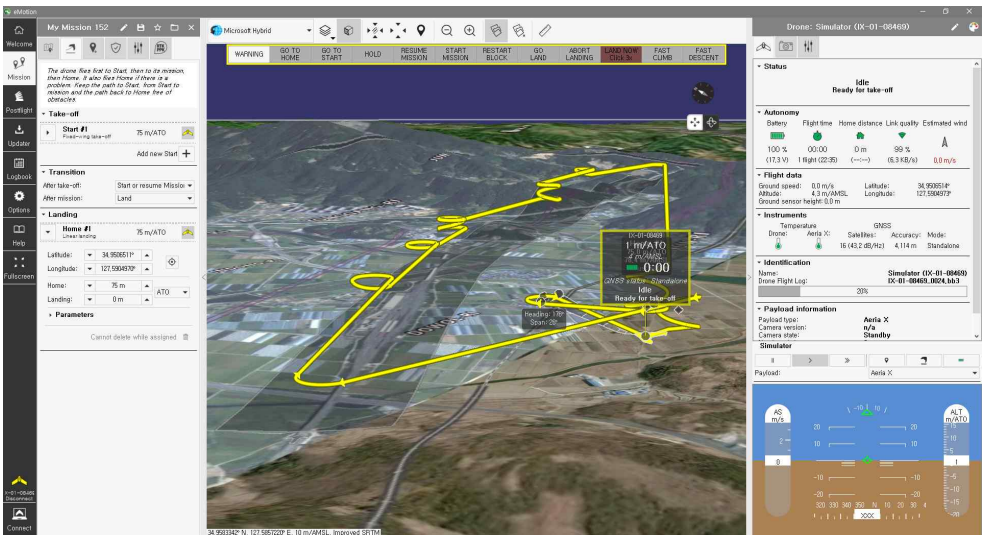
3.3.3 비행시뮬레이션

○ 비행 전 촬영계획을 바탕으로 비행시뮬레이션을 실시함으로 대상지의 비행고도, 해상도, 중복도, 촬영매수, 비행시간 등에 대한 비행조건을 산출할 수 있으며, 보다 안전한 비행을 실시할 수 있다.

○ 자동경로 비행시뮬레이션은 드론운용소프트웨어를 이용하여 시뮬레이션을 실시할 수 있다. 드론 제조사에 따라 비행시뮬레이션이 불가능할 수도 있다.



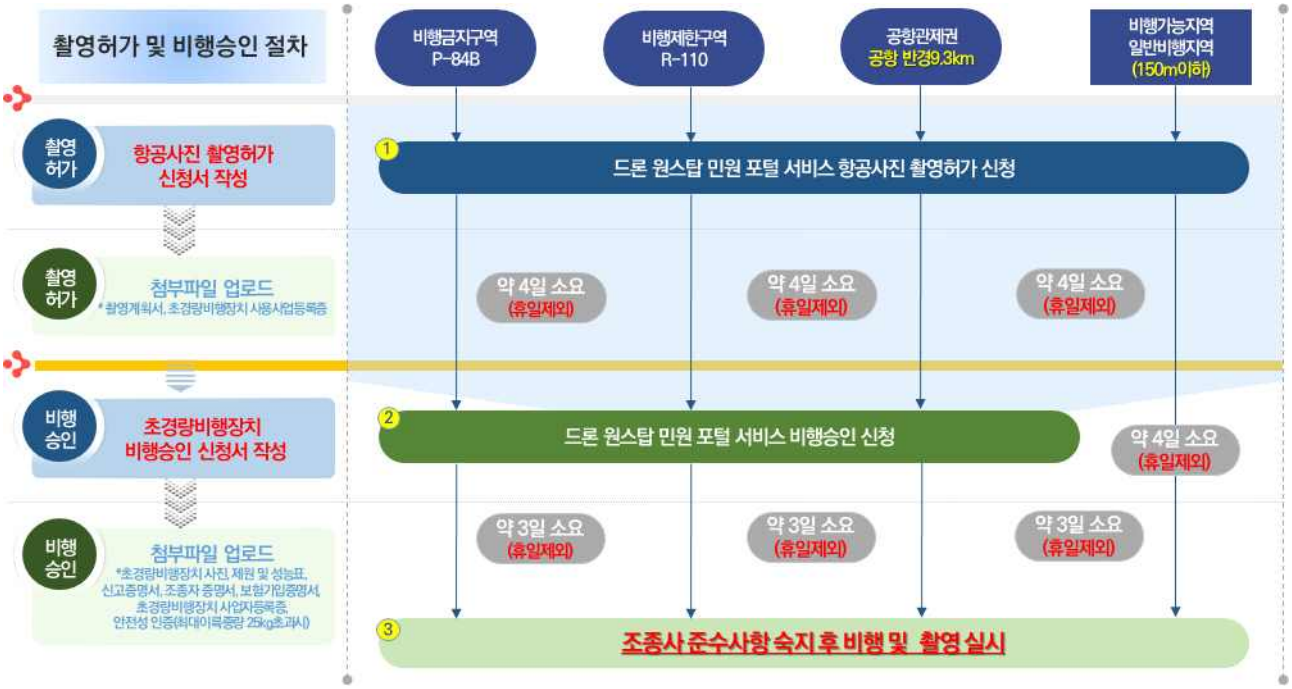
비행경로 설정



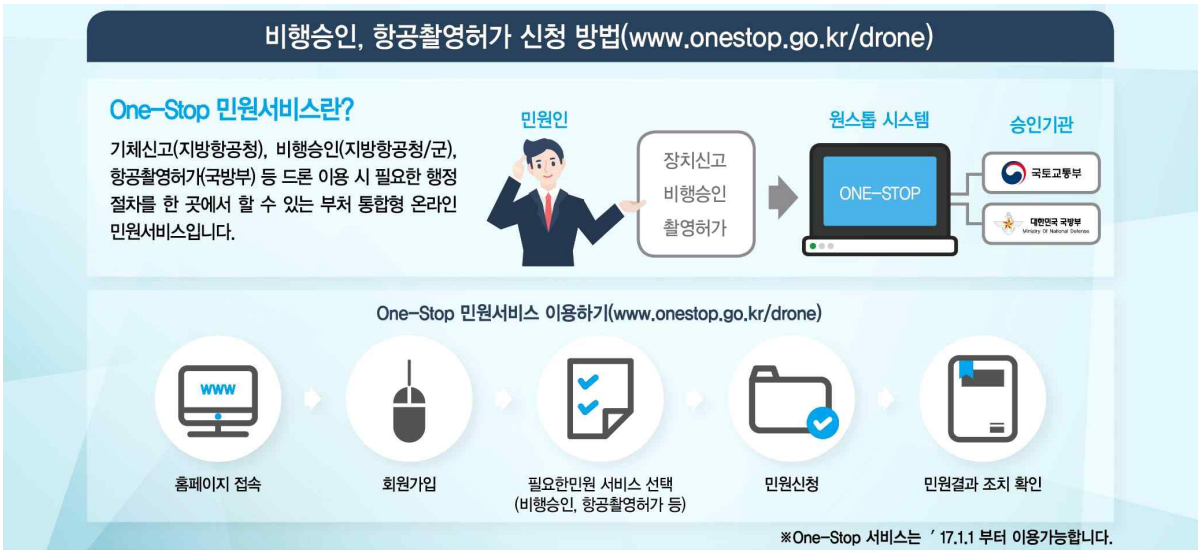
비행경로 시뮬레이션

3.4 항공사진 촬영허가 및 비행승인신청

○ 드론 촬영 전 항공사진 촬영허가신청 및 비행승인신청은 드론 원스탑 민원 포털 서비스(http://www.onestop.go.kr/drone)를 통해 신청하며 항공사진 촬영허가는 국방부에서 허가하고, 비행승인은 공역에 따라 지방항공청, 군공항 등 관련승인 기관에서 승인한다.

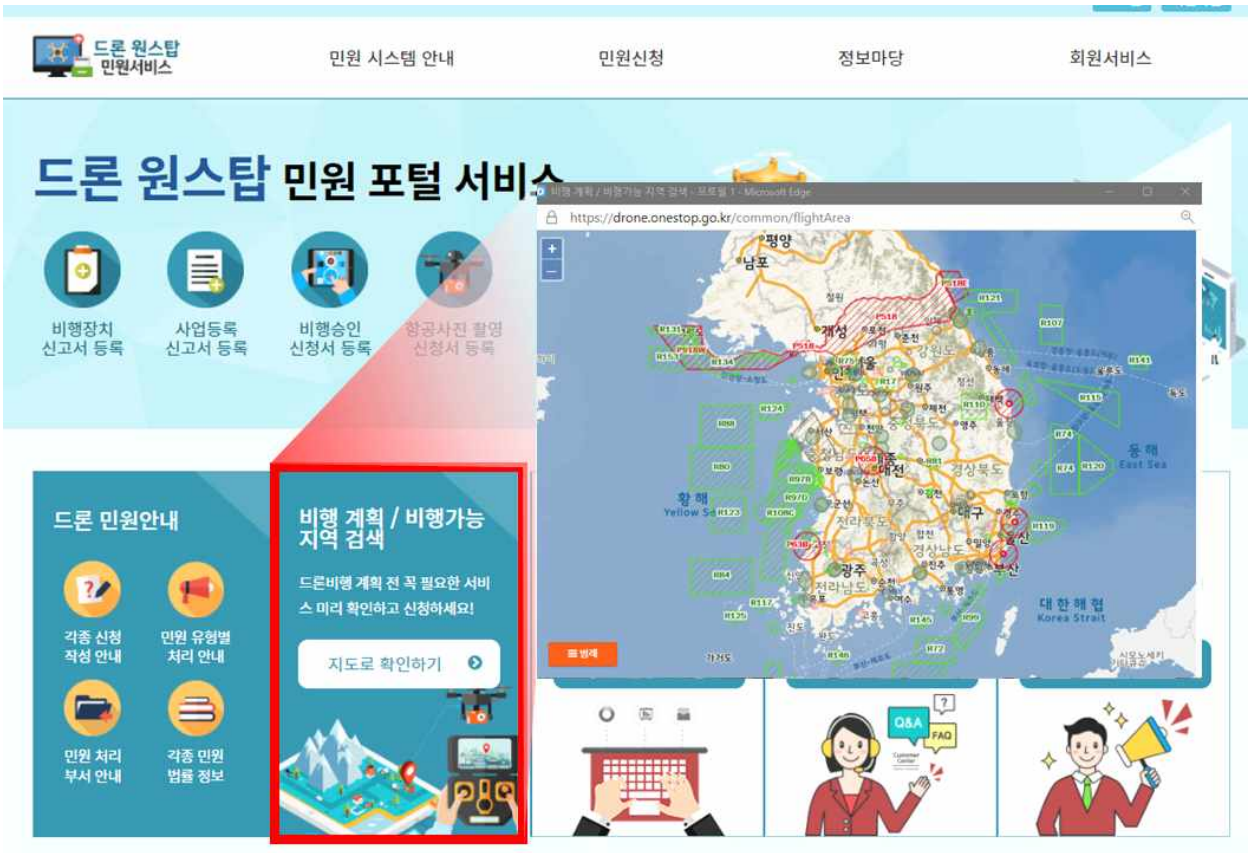


항공사진 촬영허가 및 비행승인신청 절차



One-Stop 민원서비스(출처 : 국토교통부)

○ 항공사진 촬영허가 및 비행승인 신청 전 대상지 노선에 포함된 비행공역을 확인해야 하며, 드론 원스탑 민원 포털 서비스 홈페이지에서 비행 계획/비행 가능지역 검색(지도로 확인하기)를 클릭하여 간단하게 확인할 수 있다.



드론 원스탑 민원 포탈서비스 공역확인

3.4.1 초경량비행장치 비행승인

가. 비행승인 절차

○ 초경량 비행장치의 비행승인은 비행금지구역, 비행제한구역, 민간관제권, 군관제권, 그 외 기타지역에 대한 공역에서 비행하고자 할 때 비행승인이 필요하다. 초경량비행장치 비행승인 대상은 아래의 표와 같다.

<표 3-14> 초경량비행장치 비행승인 대상

-사업에 사용되는 행글라이더, 패러글라이더, 계류식 무인비행장치, 낙하산류 - 고도 150m 이상으로 비행하는 경우 - 최대 이륙중량 25kg 초과하는 경우의 무인동력비행장치(농업용은 관제권.비행금지구역 내 비행만 포함) - 자체중량(연료제외) 12kg 초과 길이 7m 초과하는 비행선

<표 3-15> 비행고도에 따른 촬영영역 및 해상도 산정(20MP 기준)

구분	비행금지구역 (P73, P65 등)	비행제한구역 (R-75)	민간관제권 (반경 9.3kg)	군관제권 (반경 9.3km)	기타지역 (고도 150m 이하)
촬영허가 (국방부)	O	O	O	O (공역 승인 포함)	△
촬영허가 (수방사)	O	O (150m 이하 불필요)	X	X	X
촬영승인 (국토부)	X	X	O	X	X
공통사항	① 25kg 초과 기체로 비행시 고도에 상관없이 비행승인 필요 ② 공역이 2개 이상 겹칠 경우 각 기관 허가 사항 모두 적용 ③ 비행제한구역 및 기타 지역에서 150m 초과 고도에서 비행할 경우 비행승인 필요 ④ △ : 국가/군사 시설 유무에 따라 달라질 수 있어 국방부에 문의 필요				

나. 드론 비행 협의

○ 비행장 중심으로부터 반경 3km내에서 비행하고자 하는 경우[1] 또는 이착륙장 중심으로부터 반경 3km내에서 비행하고자 하는 경우[2] 관련 기관과 협의해야하며, 협의 절차는 비행장의 경우 비행장을 관할하는 기관에 방문하여 협의하고, 이착륙장(항공기, 경량항공기)은 운영자와 방문협의 해야한다.

다. 공역구분

○ <표 3-14>의 공역 분류는 국토교통부장관이 공역을 체계적이고 효율적으로 관리하기 위하여 법 제 38조 제3항 및 규칙 116조의 2에서 정한 “공역의 사용목적에 따른 구분”을 적용하여 구분한 것이다.

1) 비행장은 인천국제공항 등 민간공항과 육군, 해군, 공군이 운영하는 군비행장, 한서대학교와 같은 사설비행장이 있음

2) 이착륙장은 경량항공기 및 초경량비행장치가 이륙할 수 있도록 허가 받은 장소를 말함

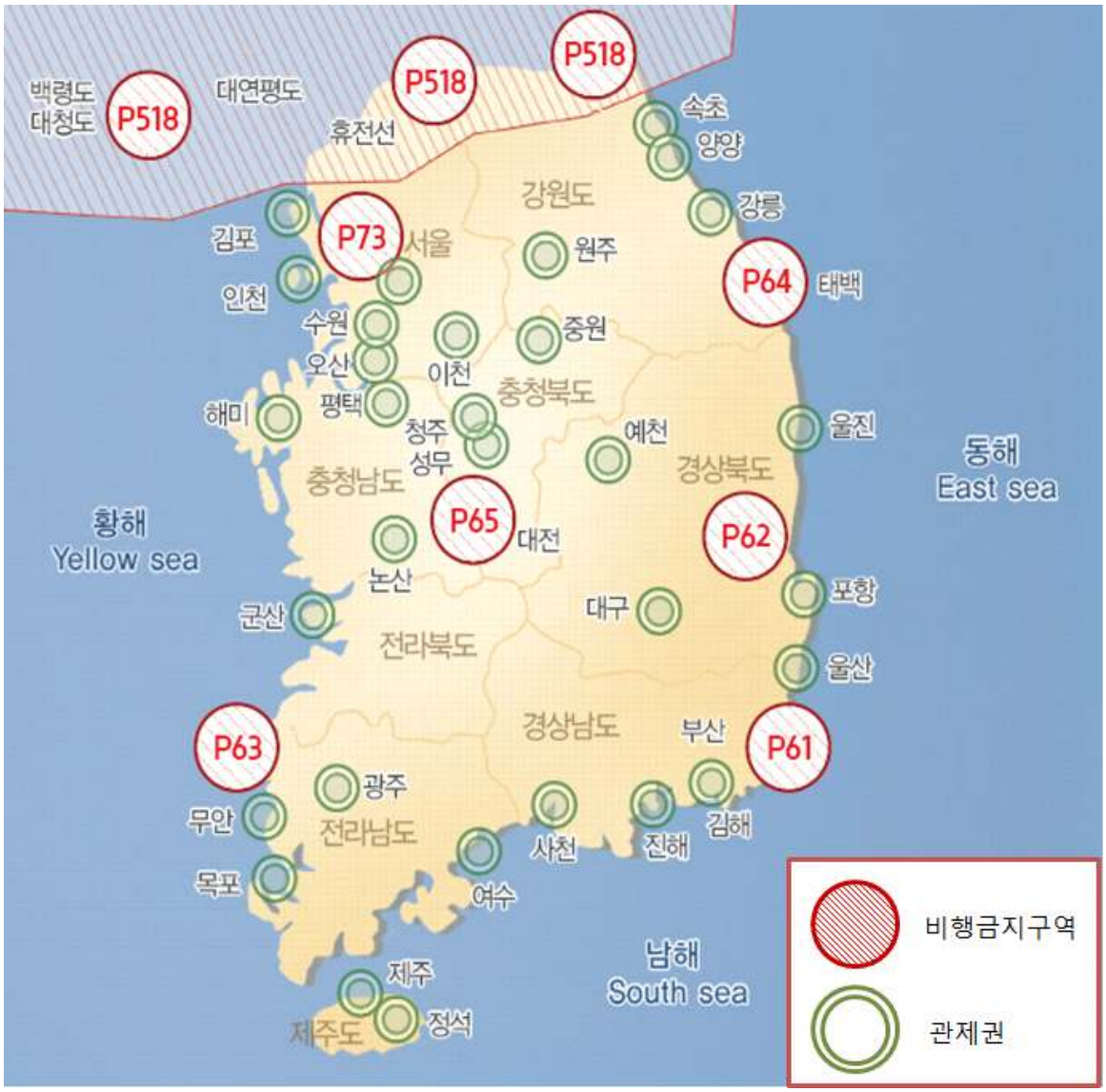
<표 3-16> 공역의 사용목적에 따른 구분

구분		내용	비고
관제공역	관제권	비행장과 그 주변 공역으로 시계비행 및 계기비행 항공기에 대하여 항공교통관제업무를 제공하는 공역	김포공항 서울공항
	관제구	인천비행정보구역의 300m(1,000ft)이상의 공역 중 시계비행 및 계기비행 항공기에 대하여 항공교통관제업무를 제공하는 공역	
	비행장 교통구역	교통량이 많은 헬기장 등과 그 주변 공역으로 시계비행을 하는 항공기간에 교통정보를 제공하는 공역	
비관제공역	조언구역	항공교통조언업무가 제공되도록 지정된 비관제공역	
	정보구역	비행정보업무가 제공되도록 지정된 비관제공역	
통제공역	비행금지구역	안전, 국방상, 그 밖의 이유로 비행을 금지하는 공역	P-73A공역 P-73B공역
	비행제한구역	항공사격·대공사격 등으로 인한 위험으로부터 항공기의 안전을 보호하거나 그 밖의 이유로 비행허가를 받지 않은 항공기의 비행을 제한하는 공역	R-75
	초경량비행장치 비행제한구역	초경량비행장치의 비행안전을 확보하기 위하여 초경량비행장치의 비행활동에 대한 제한이 필요한 공역 * 국토교통부장관이 지정한 초경량비행장치 비행구역 이외의 공역	
주의공역	훈련구역	민간항공기의 훈련공역으로서 계기비행항공기로부터 분리를 유지할 필요가 있는 공역	
	군작전구역	군사작전을 위하여 설정된 공역으로서 계기비행항공기로부터 분리를 유지할 필요가 있는 공역	
	위험구역	항공기의 비행 시 항공기 또는 지상시설물에 대한 위험이 예상되는 공역	
	경계구역	대규모 조종사의 훈련이나 비정상 형태의 항공활동이 수행되는 공역	

※ 출처 : 항공법 시행규칙[별표 20] 및 국방부 초경량비행장치 비행승인업무 지침서

나. 전국관제권 및 비행금지구역

○ 전국의 관제권과 비행금지구역은 공항과 국가 중요시설로 <그림 5-6>과 같다. 관제권, 비행금지구역 내에서 비행을 하고자 한다면 촬영계획서 및 관련 기관 협조공문 등을 첨부하여야 한다.



전국관제권 및 비행금지구역

1) 비행금지구역

○ P73, P518, P61, P62, P63, P64, P65로 분포되어있으며, 항공안전법 시행규칙 제308조에 의거 초경량비행장치 비행 제한구역에서 비행하려는 사람은 국토교통부령으로 정하는 바에 따라 미리 국방부와 국토교통부장관으로부터 촬영허가 및 비행승인을 받아야 한다.

○ 비행금지구역 비행승인기관 연락처는 다음 표를 참고한다.

<표 3-17> 지방 비행금지구역 승인기관 연락처

구분		관할기관	연락처
1	P61 A (고리원전)	합동참모본부 (공중종심작전과)	전화 : 02-748-3435 팩스 : 02-796-0369
2	P62 A (월성원전)		
3	P63 A (한빛원전)		
4	P64 A (한울원전)		
5	P65 A (원자력연구소)		
6	P61 B (고리원전)	부산지방항공청 (항공운항과)	전화 : 051-974-2154 팩스 : 051-971-1219
7	P62 B (월성원전)		
8	P63 B (한빛원전)		
9	P64 B (한울원전)		
10	P65 B (원자력연구소)	서울지방항공청 (항공안전과)	전화 : 032-740-2153 팩스 : 032-740-2159

2) 관제권

○ 관제권내의 드론 비행을 위해서는 승인기관에 사전승인을 받아야 한다. <표 5-3>는 초경량 비행장치 비행승인 관할 기관 연락처이다.

<표 3-18> 관제권 승인기관 연락처

구분		관할기관	연락처
1	인천, 경기 서부 (화성, 시흥, 의왕, 군포, 과천, 수원, 오산, 평택, 강화)	서울지방항공청 (항공운항과)	TP : 032-740-2157~8 FAX : 032-740-2159
2	서울, 경기 동부 (부천, 광명, 김포, 고양, 구리, 여주, 이천, 성남, 광주, 용인, 안성, 가평, 양평, 의정부, 남양주)	김포항공관리사무소 (안전운항과)	TP : 02-2660-5734
3	충청도	청주공항출장소	TP : 043-210-6202
4	전라북도	군산공항출장소	TP : 063-471-5820
5	강원 영동지역 (고성, 속초, 양양, 강릉, 동해, 삼척, 태백)	양양공항출장소	TP : 033-670-7206
6	강원 영서지역 (철원, 화천, 양구, 인제, 춘천, 홍천, 원주, 횡성, 평창, 영월, 정선)	원주공항출장소	TP : 033-340-8202
7	울진, 울산, 여수, 무안, 광주 (관제권 외 지역)	부산지방항공청 (항공운항과)	TP : 051-974-2153 FAX : 051-971-1219

구분			관할기관	연락처
8	제주		제주지방항공청 (안전운항과)	TP : 064-797-1745 FAX : 064-797-1759
			정석비행장	TP : 064-780-0353
9	1전비	광 주	광주기지 (계획처)	TP : 062-940-1111
10	15비	서 울	서울기지 (작전과)	TP : 031-720-3232 FAX : 031-720-4459
11	8전비	원 주	원주기지 (작전과)	TP : 033-730-4221~2 FAX : 033-747-7801
12	10전비	수 원	수원기지 (계획처)	TP : 031-220-1012 FAX : 031-220-1167
13	11전비	대 구	대구기지	TP : 053-989-3203
14	16전비	예 천	예천기지 (계획처)	TP : 054-650-4517 FAX : 054-650-5757
15	17전비	청 주	청주기지 (계획처)	TP : 043-200-2111~2 FAX : 043-200-3747
16	18전비	강 릉	강릉기지 (항공작전과)	TP : 033-649-3012 FAX : 033-649-3790
17	19전비	충 주	중원기지 (작전과)	TP : 043-849-3084~5 FAX : 043-849-5599
18	20전비	해 미	서산기지 (작전과)	TP : 041-689-2020~3 FAX : 041-689-4455
19	3훈비	진주, 사천	성무기지 (작전과)	TP : 055-850-3111
20	성 무		성무기지 (작전과)	TP : 043-290-5230 FAX : 043-297-0479
21	이천/논산/속초 비행장교통구역(가평/양평/홍천/현리/전주/덕소/용인/춘천/영천/금왕/조치원)		항공작전사령부(비행정보반)	TP : 031-644-3704~6
22	오 산		미공군 오산기지	TP : 0505-784-4222 (문의 후 신청)
23	군 산		군산기지	TP : 063-470-4422 (문의 후 신청)
24	평 택		미육군 평택기지	TP : 0503-355-2497 (문의 후 신청)
25	P73/R75(서울도심)		수도방위사령부	TP : 02-524-3346
26	P518(휴전선지역)		합동참모본부	TP : 02-748-3294
27	공군사격장		공군작전사령부	TP : 031-669-3014 / 7095
28	육군사격장		육군본부 훈련과	TP : 042-550-3321

3.4.2 항공사진 촬영허가

가. 목적

○ 이 절차는 국가공간정보 기본법 제35조, 국가공간정보 보안 관리지침(비공개 공간정보), 보안업무규정 제32조, 국가보안시설 및 보호장비 관리지침 제29조・제30조, 국방부 항공사진촬영 지침서에 의한 드론촬영 허가업무 수행 절차에 관하여 필요한 사항을 안내함을 목적으로 합니다.

나. 활용범위

○ 이 절차는 軍 책임지역 부대장과 민간 촬영 신청자・드론촬영 업체 및 기관이 드론 항공사진촬영 업무를 수행하는 때에 활용합니다.

다. 보안책임

1) 이 절차를 적용을 받는 드론촬영 신청자는 개인정보보호의 책임이 있으며, 업체 및 기관의 대표는 보안업무규정을 적용받는 분야에 대한 전반적인 보안책임이 있습니다. 또한 소속 인원은 부여된 업무와 관련하여 보안책임을 지며 비밀사항을 지득하거나 점유 시 이를 보호할 책임이 있습니다.

2) 책임지역 부대장은 국방부장관의 명을 받아 이 절차의 적용을 받는 개인,업체 및 기관의 효율적인 보안업무 수행에 필요한 지원임무를 수행할 책임이 있습니다.

라. 항공촬영 신청 및 허가

1) 드론 항공사진촬영 신청은 우리나라 全 지역이 대상 지역입니다. 관제권, 비행금지구역, 비행제한구역 등은 항공사진촬영 허가가 아닌 비행승인과 관련된 구역이며, '다'항의 항공촬영 금지구역을 지도에 표시할 수 없기 때문에 촬영대상 시설과 지역의 보안상 중요도를 검토하는 것이 항공사진촬영 신청 및 허가 승인 절차입니다.

2) 드론촬영 신청자는 촬영 4일전(천재지변에 의한 긴급보도 등 부득이한 경우는 제외)까지 드론원스탑 민원처리 포털서비스(https://drone.onestop.go.kr)의 항공사진촬영 허가 신청(촬영대상·일시·목적·촬영자 인적사항 등)를 이용하여 신청합니다.

※ 항공사진 촬영 허가관련 문의 : 국방부 정보본부 보안암호정책과(02-748-2344)

※ 지역별 항공촬영 승인업무 책임부대 연락처는 항공사진 촬영 허가 신청란에 게시

3) 지역별 항공촬영 승인업무 책임부대 부대장은 촬영목적.용도 및 대상시설.지역의 보안상 중요도 등을 검토하여 항공촬영 허가여부를 결정하되, 다음 각 호에 해당되는 시설에 대하여는 드론촬영을 금지합니다.

가) 「국가공간정보 보안관리 지침」의 세부 분류기준에서 촬영을 금지한 국가보안시설 및 군사보안시설

나) 비행장, 군항, 유도탄 기지 등 군사시설

다) 기타 군수산업시설 등 국가안보 상 중요한 시설.지역

4) 국가보안시설(국가중요시설) 감독기관의 장은 촬영금지 시설에 대하여 시설점검・오염감시・각종 측량 등 국익목적 또는 국가이익 상 촬영이 필요할 때에는 그 사유를 첨부하여 국방부장관(해당 책임부대장)에게 촬영협조를 요청할 수 있습니다. 이 경우 국방부장관(정보본부장)에게 보고하고 국방부장관(정보본부장)은 국가정보원장과 협의하여 그 제한을 완화할 수 있습니다. 단, 제한 완화로 촬영된 촬영물은 최초 촬영목적 이외의 용도로 사용을 금지합니다.

5) 지자체 등 공공기관과 신문방송사 등에서 사업목적으로 항공촬영을 할 경우에는 대행업체(촬영업체 등)가 아닌 직접 신청해야 합니다. 건설사 등 일반업체의 경우 원발주처의 신청을 원칙으로 하되, 촬영업체가 신청하는 경우에는 계약서 등을 첨부하여 신청합니다.

6) 국방부장관(드론촬영 승인업무 지역부대장)은 항공촬영 허가 시 관련 기관 및 업체의 업무를 고려하여 촬영허가 기간을 관공서(최장 3개월), 촬영업체 및 개인 (최장 1개월) 이내에서 허가할 수 있습니다. 특히, 국가중요기반시설의 경우 시설점검・오염감시・각종 측량 목적으로 드론촬영 이력관리 및 관제 시스템 구축 시에는 보안측정 후 후속조치에 대한 보고(국방부)가 완료되면 1년 단위로 협의에 의해 허가가 가능합니다.

7) 전국단위(도로 또는 강・해안을 따라 수 개의 지역을 촬영하는 경우) 드론촬영 허가 신청 접수 및 승인은 육군 제 17보병사단 (정보참모처 : 032-510-9216)에서 실시하며 해당 지역의 보안조치는 해당 책임지역 부대장이 실시합니다.

마. 항공촬영 보안조치

1) 지역별 항공촬영 승인업무 책임부대 부대장은 드론촬영 허가 시 드론 조종사의 신원과 촬영 분량을 확인하고 촬영금지지역 고지 등 보안조치를 실시하고 있으니 드론촬영 신청자는 이에 협조해야 합니다.

2) 지역별 항공촬영 승인업무 책임부대 부대장은 촬영물에 대하여 촬영금지시설과 지형정보 삭제 등 적절한 보안조치를 하고 필요한 경우 촬영물 취급기관(업체)으로 하여금 적정등급의 비밀로 분류.관리토록 하여야 하며, 촬영물 취급기관(업체)는 이를 준수해야 합니다.

3) 각 지역 책임부대장은 초경량비행장치(드론, 무인UAV, 열기구 등)를 이용하여 촬영 시 조정 및 촬영자의 신원을 항공사진 촬영허가 신청 인원과 동일 여부를 확인하고 촬영 전 보안교육 및 촬영 후 영상(사진)에 대해서 보안조치를 실시해야 합니다.

바. 비행승인

1) 드론 운용자가 항공촬영을 위한 비행 시에는 항공촬영 허가와 별도로 국토교통부에 신고하여야 한다. 다만, 비행금지구역을 비행할 경우 항공촬영 신청자는 해당 지역의 공역(空域) 관리기관(합참・수방사, 공군 등)의 별도 승인을 얻은 후 국토교통부에 신고하여야 합니다.

2) 드론촬영 신청자가 군사작전 지역 내 비행 및 군 시설 이용이 필요할 경우 사전에 관할 군부대와 협조하여야 합니다.

사. 행정사항

1) 드론촬영 신청자(개인)와 항공촬영 업체 및 기관에서 부득이한 사정으로 촬영일정, 촬영대상, 촬영관계자, 항공기 이착륙지 등 촬영 허가된 내용을 변경할 경우에는 촬영을 승인한 부대에 재 허가를 받아야 합니다.

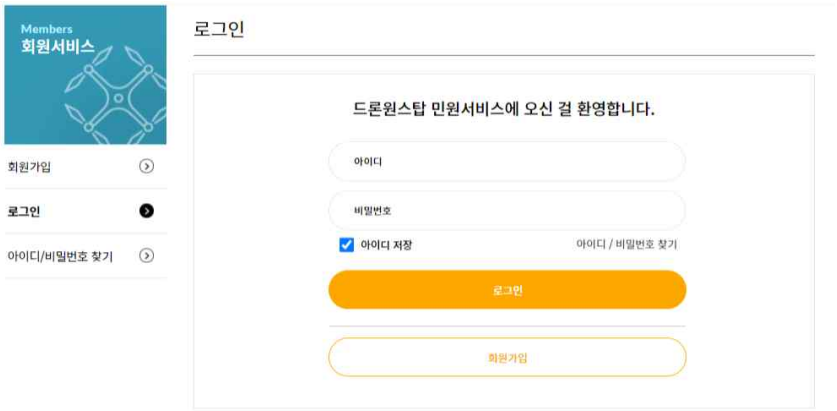
<표 3-19> 항공사진 촬영허가 담당부대 연락처

	구분	연락처
1	서울특별시	TP : 02-524-3345, 3346
2	강원도 춘천시(서면, 사북면 제외), 화천군(간동면)	TP : 033-249-6066
3	강원도 춘천시(서면, 사북면), 화천군(사내면)	TP : 033-440-8221
4	강원도 화천군(간동면, 사내면 제외), 철원군(원동면, 원남면)	TP : 033-440-5204
5	강원도 철원군(근남면, 근동면)	TP : 033-440-6203
6	강원도 철원군(임남면)	TP : 033-480-6205
7	강원도 인제군, 양구군	TP : 033-461-5102 →교환 983-2223
8	강원도 고성군(간성읍, 거진읍, 현내읍, 죽왕면, 수동면)	TP : 033-639-6229
9	강원도 고성군(토성면), 속초시, 양양군(양양읍·강현면)	TP : 033-671-6661
10	강원도 양양군(서면·현북면·현남면), 강릉시, 동해시, 삼척시	TP : 033-571-6214
11	강원도 양양군(손양면)	TP : 033-670-6221
12	강원도(원주시, 횡성군, 평창군, 홍천군, 영월군, 정선군, 태백시)	TP : 033-741-6206
13	강원도 철원군 갈말읍(문혜리, 토성리), 근북면 강원도 철원군 김화읍(도창리, 학사리, 생창리), 철원군 서면(자등리, 와수리)	TP : 033-450-6211
14	강원도 철원군 동송읍(내포리, 외촌리, 관우리, 사요리, 관전리, 중리, 월하리, 대위리, 화지리, 오덕리, 이평리, 오지리, 장흥리), 철원읍 강원도 철원군 갈말읍(상사리, 동막리, 지포리, 군탄리, 내대리) 경기도 포천시(관인면 - 사정리, 삼율리, 초과리), (영북면-야미리, 자일리)	TP : 031-539-6221
15	광주광역시, 전라남도	TP : 062-260-6204
16	대전광역시, 충청남도, 세종특별자치시	TP : 042-829-6205
17	전라북도	TP : 063-640-9222
18	충청북도	TP : 043-835-6205
19	경상남도(창원시 진해구 양산시, 거제시 중 장목면만 제외)	TP : 055-259-6205
20	대구광역시, 경상북도(울릉도, 독도, 경주시 양북면 제외)	TP : 053-320-6204~5
21	부산광역시(부산 강서구 성북동, 다덕도동 제외), 울산광역시, 경상남도 양산시	TP : 051-730-6205
22	경상남도 창원시 진해구, 거제시 장목면, 부산광역시 (강서구 성북동, 가덕도동)	TP : 055-549-4185, 4173
23	울릉도, 독도	TP : 033-539-4222
24	경기도 포천시(이동면, 일동면)	TP : 031-531-0555 → 교환 985-2215
25	경기도 포천시(내촌면), 남양주시(진접읍, 오남읍)	TP : 031-528-7110
26	경기도 포천시(영북면)	TP : 031-531-0555 → 교환 985-7624, 7628

3.4.3 드론 원스탑 민원서비스 신청 절차

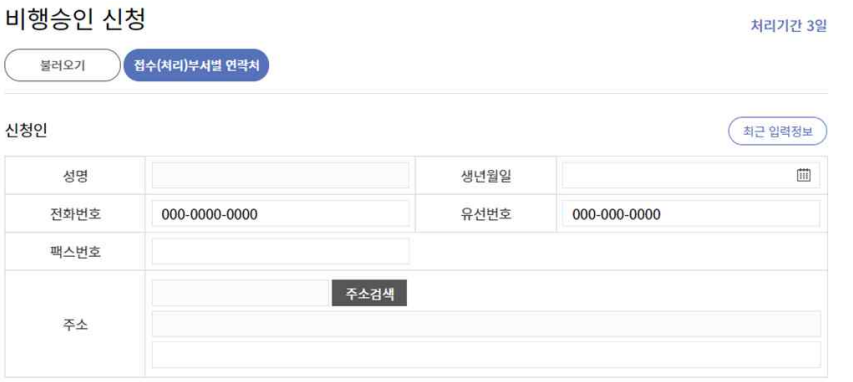
○ 아래의 <표 3-18>~<표 3-19>와 같이 항공사진 촬영허가 신청 및 비행승인 신청을 실시할 수 있다.

<표 3-20> 항공사진 촬영허가 신청 절차

절차	이미지
①드론 원스탑 민원서비스 사이트에 로그인 한다.	
②민원신청란에서 항공사진 촬영허가 신청을 클릭한다.	
③신청인 정보를 입력한다. 구분에 따라 첨부서류가 다르므로 잘 선택한다.	
④촬영 계획정보를 입력한다. (오른쪽 상단 비행구역 설정을 통해 주소 및 비행고도 등 입력가능)	

절차	이미지
⑤비행장치의 정보를 입력하고 추가 버튼을 이용하여 추가한다. (다수의 드론 추가 가능)	
⑥조종자 정보 및 경력를 입력하고 추가를 클릭한다. (조종자와 부조종자 모두 입력)	
⑦업체의 경우 업등록, 보험가입, 기체정보, 신고서 등을 첨부해야 한다.	

<표 3-21> 비행승인 신청 절차

절차	이미지
①민원신청란에서 비행승인 및 항공사진 촬영허가 신청을 클릭한다.	
②신청인 정보를 입력한다.	
③비행 계획정보를 입력한다. (오른쪽 상단 비행구역 설정을 통해 주소 및 비행고도 등 입력가능)	
④비행장치의 정보를 입력하고 추가 버튼을 이용하여 추가한다. (다수의 드론 추가 가능)	

절차	이미지
⑥조종자 정보 및 경력를 입력하고 추가를 클릭한다. (조종자와 부조종자 모두 입력)	
⑦해당 사항의 첨부파일을 모두 업로드 한다.	

제 4 장

드론 운용

4.1 드론 운용 및 안전

4.2 단계별 주의사항

4.3 사고의 보고

제4장 드론 운용

4.1 드론 운용 및 안전

○ 드론 비행을 위해서는 조종자, 부조종자, 안전요원, 발주기관 담장자로 인력을 구성하여 시민들과 차별화된 복장을 착용하고, 드론 운용 인력은 조종자 준수사항 등을 사전에 숙지하고 준수하여야 한다.

○ 조종자는 경로설정, 비행제어 프로그램 모니터링 등의 임무를 수행하며, 부조종자는 이륙, 비행위치 확인 등을 수행한다. 안전요원은 드론의 이·착륙 시 안전범위 내에 인원을 통제 하는 등 현장에서 발생할 수 있는 사고에 대비해야 하며 발주기관 담당자는 현장 운용 및 안전관리를 총괄하여야 한다.

4.1.1 조종자 준수사항(항공안전법 제129조, 시행규칙 제310조)

○ 조종자는 초경량비행장치로 인하여 인명이나 재산에 피해가 발생하지 않도록 조종자 준수사항에 따라 비행하여야 한다. 조종자는 다음 각 호의 어느 하나에 해당하는 행위를 해서는 아니 된다. 단, 무인비행장치의 조종자에 대하여는 제4호 및 제 5호를 적용하지 아니한다.

① 인명이나 재산에 위험을 초래할 우려가 있는 낙하물을 투하하는 행위

② 인구가 밀집된 지역이나 그 밖에 사람이 많이 모인 장소의 상공에서 인명 또는 재산에 위험을 초래할 우려가 있는 방법으로 비행하는 행위

③ 관제공역, 통제공역, 주의공역에서 비행하는 행위. 단, 다음 각 목의 행위와 지방항공청장의 허가를 받은 경우에는 제외한다.

1.1.1.1.1.1. - 군사목적으로 사용되는 초경량비행장치를 비행하는 행위

1.1.1.1.1.2. - 무인비행기 및 무인회전익 비행장치 중 최대이륙중량이 25킬로그램 이하인 것, 무인비행선 중 연료의 무게를 제외한 자체 무게가 12킬로그램 이하이고, 길이가 7미터 이하인 것의 관제권 또는 비행금지구역이 아닌 곳에서 최저비행고도(150미터) 미만의 고도에서 비행하는 행위

④ 안개 등으로 인하여 지상목표물을 육안으로 식별할 수 없는 상태에서 비행하는 행위

⑤ 비행시정 및 구름으로부터의 거리기준을 위반하여 비행하는 행위

⑥ 일몰시부터 일출시까지의 야간에 비행하는 행위(허가를 받아 비행하는 행위 제외)

⑦ 주류, 마약류, 환각물질 등의 영향으로 조종업무를 정상적으로 수행할 수 없는 상태에서 조종하는 행위 또는 비행 중 주류 등을 섭취하거나 사용하는 행위

⑧ 그 밖에 비정상적인 방법으로 비행하는 행위

1.1.1.1.1.3. - 초경량비행장치 조종자는 항공기 또는 경량항공기를 육안으로 식별하여 미리 피할 수 있도록 주의하여 비행하여야 한다.

1.1.1.1.1.4. - 동력을 이용하는 초경량비행장치 조종자는 모든 항공기, 경량항공기 및 동력을 이용하지 아니하는 초경량비행장치에 대하여 진로를 양보하여야 한다.

1.1.1.1.1.5. - 무인비행장치 조종자는 해당 무인비행장치를 육안으로 확인할 수 있는 범위 내에서 조종하여야 한다. 다만, 허가를 받아 비행하는 경우는 제외한다.

1.1.1.1.1.6. - 항공레저스포츠사업에 종사하는 초경량비행장치 조종자는 다음 각 호의 사항을 준수하여야 한다.

1.1.1.1.1.7. ㅇ 비행 전에 해당 초경량비행장치의 이상 유무를 점검하고, 이상이 있을 경우에는 비행을 중단할 것

1.1.1.1.1.8. ㅇ 비행 전에 비행안전을 위한 주의사항에 대하여 동승자에게 충분히 설명할 것

1.1.1.1.1.9. ㅇ 해당 초경량비행장치의 제작자가 정한 최대이륙중량을 초과하지 아니하도록 비행할 것

1.1.1.1.1.10. ㅇ 동승자에 관한 인적사항(성명, 생년월일 및 주소)을 기록하고 유지할 것



<그림 6-1> 드론 조종자 체크리스트(출처 : 국토교통부)

4.1.2 드론 비행규칙

○ 안전비행을 위하여 교통안전공단에서 발간한 조종자 안전가이드를 따라 다음과 같은 비행규칙을 준수하여야 한다.

① 드론의 운영자, 소유자, 임대인 또는 지정된 대리인으로부터 권한을 부여받을 것

② 드론을 운용할 수 있는 충분한 능력을 보유할 것

③ 무선통신이 필요한 경우 무선통신기 운용자격을 갖출 것

④ 드론의 이·착륙장소 배치, 이와 관련된 신호, 절차 등에 대한 교육을 받고 이·착륙지에서 드론을 안전하게 이동시키는데 필요한 절차를 준수할 것

○ 드론 조종자는 이륙을 시작하기 전 아래의 사항을 확인해야 한다.

① 이·착륙지의 기상과 상태가 드론을 안전하게 이륙 및 출발시킬 수 있는 지의 여부

② 드론의 이륙방향, 가시범위 또는 시정이 적용 이륙기상 최저치와 같거나 그 이상임을 확인해야 한다.

○ 충돌사고를 발생시킬 정도로 다른 초경량비행장치에 근접하여 비행해서는 안된다.

○ 각 초경량비행장치의 조종자간의 사전협의 없이 편대비행을 해서는 안된다.

○ 통행우선권

① 조종자는 비행 중 다른 드론의 발견과 회피를 위해 외부 경계를 지속적으로 유지하여야 한다. 규칙에 따른 통행우선권을 준수하여야 하며 통행우선권을 가진 드론이 있을 때 그의 위, 아래 또는 앞을 근접하여 통과해서는 안 된다.

② 드론이 정면으로 서로 접근할 경우 각 조종자는 운항경로를 오른쪽으로 변경한다.

③ 추월되는 드론은 통행 우선권을 가지게 되고 추월하는 드론은 오른쪽으로 운항경로를 변경하여 추월하여야 한다.

④ 착륙 중이거나 착륙을 위한 최종 접근로 상에 있는 드론은 비행 중인 다른 드론 또는 지상에서 이동 중인 드론 보다 통행우선권을 가진다.

⑤ 동일한 종류의 드론이 거의 동일고도에서 집중되는 경우에는 우측 드론이 우선권을 가진다.

⑥ 집중되는 드론들의 종류가 상이하다면 다음과 같이 통행 우선권이 부여된다.

- 기구는 다른 종류의 드론들에 대하여 통행우선권을 가진다.
- 글라이더는 비행선, 동력비행장치 또는 회전익 드론에 대하여 통행 우선권을 가진다.
- 비행선(Airship)은 동력비행장치 또는 회전익 드론에 대하여 우선권을 가진다.

⑦ 2대 이상의 드론이 착륙을 위하여 비행장으로 접근중일 때에는 더 낮은 고도에 있는 드론이 통행 우선권을 갖는다.

4.1.3 현장 운용 임무

가. 조종자

○ 조종자는 비행경로 입력 등 드론비행에 대한 상태정보를 실시간으로 모니터링 해야 하며, 주로 체크할 사항으로 배터리 상태, 모니터링위치와 드론과의 거리, 드론 온도, 풍향·풍속 등이 있다.

<표 6-1> 드론 조종자 임무

	
비행경로 입력	**드론모니터링**

나. 부조종자

○ 부조종자는 이륙 또는 착륙 방향을 설정하여야 하며, 비행 중 육안으로 드론의 상태 및 위치 등을 파악하여야 한다. 드론 비행 중 이상한 움직임이 포착되면 즉시 조종자에게 보고하여야 하며, 비상시 비상대처방안에 따라 비상착륙을 실시하여야 한다. 드론의 추락은 즉 인명사고로 직결할 수 있음을 인지하여야 한다.

<표 6-1> 드론 부조종자 임무

	
이륙	**착륙**

다. 안전요원

○ 안전요원은 안전조끼, 안전모(필요에 따라 착용), 경광봉, 호루라기 등을 착용하고 이·착륙 시 안전범위 내에 외부인 출입을 통제하여야 한다. 드론의 사고는 이·착륙 시에 가장 많이 발생하므로 상시 경계를 확보하여 내·외부인통제에 힘써야 한다.

<표 6-1> 안전요원 착용물품

		
안전조끼	**경광봉**	**호루라기**

라. 발주기관 담당자

○ 발주기관 담당자는 현장에 상주하여 드론운용 및 안전관리를 총괄하고, 원활한 포트홀 영상데이터를 취득하기 위해 각 담당자의 임무를 지원한다.

4.2 단계별 주의사항

4.2.1 드론 비행 전

○ 모든 드론은 비행 전 안전점검을 통하여 비행 가능유무를 확인하고, 이상 발견 시에는 신속한 조취를 통하여 대처방안을 마련하여야 한다.

가. 기체 점검

○ 기체에 크랙이 있는지 확인하거나 날개 등 부품이 파손되었는지 확인한다.

○ 기체의 GPS, 모터, 날개 등의 작동을 확인한다.

○ 자동으로 캘리브레이션을 실시하는 기체는 수평을 유지하며, 뒤집거나 이동시키지 않는다.

○ 드론에 탑재된 GPS/INS는 자기(자석 및 전자파)간섭에 민감함으로 모든 전자 장치로부터 멀리 떨어져 있어야 한다. INS 캘리브레이션 중에는 전자장비(휴대폰, PDA, 스마트기기 등)를 소지하면 안 된다.

나. 모터점검

○ 육안으로 체크한 후 모터를 회전시켜 모터를 점검한다.

○ 모터가 부드럽게 돌아가는지와 주변에 이물질이 없는지 확인한다.

○ 모터가 부드럽게 회전하지 않는다면, 모터 안의 이물질을 확인하고 제거한다.

○ 압축 공기를 사용하여 이물질을 빼내면 모터내부에 손상이 생길 수 있으므로 주의하여야 한다.

다. 날개 및 프로펠러 점검

○ 날개는 비행체의 속도와 방향에 지대한 영향을 끼치므로 날개결함이 발견되면 반드시 수리 후 비행하여야 한다.

○ 프로펠러의 결함이 발견되면 비행이 불가능하며, 비행을 실시 할 시에는 추락의 원이 될 수 있으므로 여분의 프로펠러를 항상 보유하여야 한다.

○ 육안으로 프로펠러의 휨, 크랙 등을 확인한다.

○ 느슨하게 결합된 프로펠러는 진동을 일으킬 수 있으므로 프로펠러의 조임부를 확인한다.

라. 라디오 모뎀 및 송수신기 점검

○ USB 라디오모뎀의 안테나는 사용자의 부주의로 인하여 쉽게 파손되기 때문에 안테나의 단자를 항상 확인하여야 한다.

○ 안테나의 단자파손으로 인하여 비행 중 라디오 모뎀의 수신감도가 저하되거나 단절될 수 있으므로 비행 전 점검하여야 한다.

마. 카메라 점검

○ 카메라 종류에 따른 SD카드 및 배터리 삽입 유무를 확인한다.

○ 카메라 렌즈의 파손 및 균열을 확인한다.

○ 카메라 렌즈에 이물질이 있다면 반드시 제거해주어야 한다.

○ 비행제어 프로그램에서 카메라 설정이 완료되었는지 확인 후 촬영이 정상적으로 이루어지는지 확인한다.

바. GPS 점검

○ GPS의 신호가 7개 이상 수신되는지 확인해야 하며, 수신되는 위성의 수가 7개 이상이어도 위성 데이터가 깜박일 경우 비행하지 않는다. (Fixed가 되지 않는 상태)

1.1.1.1.2.

사. 배터리 점검

○ 모두 완충된 배터리만 사용하여야 하며, 충전상태 및 전압을 확인한다.

○ 배터리 연결케이블 단선 및 부식, 커넥터 손상유무 확인하고, 이상이 발견된 배터리는 비행에 절대 사용하지 말고 즉각 폐기 처분하여야 한다.

아. 비행체크리스트

○ <표 6-4>와 같이 비행 전 체크리스트를 작성하여 효율적으로 드론의 상태를 점검하도록 한다.

<표 6-1> 드론 체크리스트

비행일시 : 2021.00.00

드 론 명 : 00드론(고정익)

비행목적 : 포트홀탐지

순번	항목	내용	결과	
			비행 전	비행 후
1	드론 외관 검사	기체 상태확인	YES□ NO□	YES□ NO□
		모터 상태확인	YES□ NO□	YES□ NO□
		프로펠러 및 날개 상태확인	YES□ NO□	YES□ NO□
		케이블 상태확인	YES□ NO□	YES□ NO□
2	배터리 점검	배터리 전압확인	YES□ NO□	YES□ NO□
		배터리커넥터확인	YES□ NO□	YES□ NO□
3	카메라점검	SD카드 상태확인	YES□ NO□	YES□ NO□
		SD카드 포맷확인	YES□ NO□	YES□ NO□
		카메라 배터리 확인	YES□ NO□	YES□ NO□
		카메라 설정 확인	YES□ NO□	YES□ NO□
4	모니터링 장치 점검	모니터링 장치 배터리 확인	YES□ NO□	YES□ NO□
		프로그램 구동 확인	YES□ NO□	YES□ NO□
		사용맵 확인	YES□ NO□	YES□ NO□
		기체정보 수신확인	YES□ NO□	YES□ NO□
5	송수신기 점검	라디오모뎀 연결 상태 확인	YES□ NO□	YES□ NO□
		수동조종컨트롤러 연결 상태 확인	YES□ NO□	YES□ NO□
		연결 케이블 확인	YES□ NO□	YES□ NO□
6	캘리브레이션	캘리브레이션 상태확인	YES□ NO□	YES□ NO□
7	주변안전점검	내 · 외부인 통제	YES□ NO□	YES□ NO□
		비행 중인 항공기 및 드론 확인	YES□ NO□	YES□ NO□
		전파방해요소 재점검	YES□ NO□	YES□ NO□
8	이륙	풍향 · 풍속 확인	YES□ NO□	YES□ NO□
		GPS 수신상태 확인	YES□ NO□	YES□ NO□
		미션 확인	YES□ NO□	YES□ NO□
		이륙 준비 상태 확인	YES□ NO□	YES□ NO□

4.2.2 기상확인

○ 기상상태의 점검방법으로는 다양한 매체를 통한 기상예보 정보 습득, 현장에서 풍속계를 이용한 풍속 확인, 주변 여건을 이용한 기상확인이 있다.

○ 드론이 견디지 못하는 풍속 이상이거나 고온, 눈이나 비가 올 때에는 비행 중 드론의 손상 및 오작동이 발생할 수 있으므로 비행을 하지 않도록 한다. 일반적인 드론이 버틸 수 있는 한계 풍속은 기종에 따라 약간 상이하지만 대략 8~15m/s이다.

가. 매체를 통한 기상확인

○ 기상정보는 다양한 매체를 통하여 확인 할 수 있으며, 대상지의 기상을 검색하여 상황을 확인한다.

나. 풍속계를 이용한 확인

○ 현장에서 풍속계를 사용하여 이·착륙 지역의 풍향, 풍속을 체크할 수 있다. 고정익 비행체는 이·착륙 시 바람의 영향을 많이 받으며 이·착륙 시 맞바람을 받아야 안전하게 비행을 할 수 있고, 고도에 따라 풍속이 다를 수 있으므로 지상에서 풍속계로 계측한 값은 절대적인 것은 아니다.

다. 주변 환경을 이용한 확인

○ 야외에서 구름이 흘러가는 속도나 방향 그리고 높은 나무의 흔들림 등은 풍속 및 풍향을 추정하는데 도움이 되지만 정확한 값은 아니므로 주의하여야 한다.

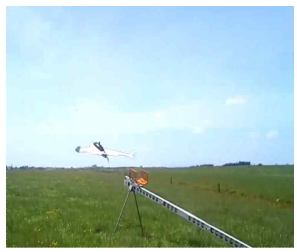
4.2.3 드론의 이륙

○ 대부분의 드론 손상은 이·착륙 시 많이 발생하므로 드론의 이륙 시에는 이륙경로 상의 주변통제를 철저하게 하도록 한다. 이륙 중에 갑작스러운 강풍으로 이륙에 실패할 수 있으므로 주의하여야 한다.

○ 드론의 안전 범위는 기종에 따라 상이하므로 각 기종의 매뉴얼을 참조하여 안전조치를 취하여야 한다. 이륙 시에는 이륙위치 및 안정고도로 올라가기 전의 비행궤도를 안전범위로 보고 이 범위 내에 사람이나 차량, 기타 장애물이 진입하지 않도록 안전요원이 통제하여야 한다.

○ 드론의 이륙방법으로는 고정익의 이륙방법인 핸드 런치, 스틱 이륙, 발사대 이륙과 회전익의 이륙방식인 수직 이륙으로 분류되며, 이륙 방식에 따른 주의사항을 인지해야한다.

<표 6-1> 드론의 이륙방식

			
핸드 런치	**스틱 이륙**	**발사대 이륙**	**수직 이륙**

가. 핸드 런치

○ 소형 고정익 드론의 이륙방식으로 손으로 드론을 던져 이륙시킨다. 스틱 또는 발사대가 필요하지 않아 다른 이륙 방식에 비해 신속하고 간편하게 이륙 가능하다.

○ 프로펠러 회전부와 몸이 가까워 다칠 위험성이 있으며, 사용자 조작 실수에 의해 이륙 실패를 초래할 수 있다.

○ 핸드 런치를 이용한 이륙은 기류를 타고 비행하여야 하며, 갑작스럽게 바람의 방향이 바뀌는 것을 주의하여야 한다.

나. 스틱 이륙

○ 핸드 런치 방식에서 개량된 이륙 방식으로 스틱 상단에 드론을 올리고, 던져서 이륙시킨다.

○ 스틱 이륙 방식은 핸드 런치의 위험성을 완화하였지만, 이륙 정확도가 떨어진다는 단점이 있다.

다. 발사대 이륙

○ 발사대 이륙 방식은 고정익 이륙 방식 중 가장 안전하고 정확하게 이륙할 수 있으며, 무게가 1.5kg 이상인 드론에 적용되는 방식이지만 발사대를 추가적으로 보유해야하며 항상 가지고 다녀야 하는 불편함이 있다.

라. 수직 이륙

○ 일반적인 회전익 드론에 적용되는 이륙 방식으로 이륙의 정확성 및 위치 정확도가 뛰어나며, 좁은 이륙 공간에서도 이륙할 수 있다는 장점이 있다.

4.2.4 드론 비행 중

○ 드론 운용 시 조종자는 드론을 조종하고 부조종자는 드론을 육안으로 감시하여 비행상태, 비행자세, 위험요소 등에 대비해야 한다.

○ 조종자는 비행 중 드론의 위치 및 자세정보, 배터리 잔량 등 상태를 항상 모니터링 해

야 하며, 주의 깊게 관찰하여야 한다.

○ 조종자는 비행 중 계획된 경로비행을 안전하게 수행하고 있는지 주시하여야 한다.

○ 조종자는 비행 중 드론에 이상이 있음을 인지하는 경우에는 즉시 가장 가까운 이·착륙장에 안전하게 착륙하여야 한다.

○ 부조종자는 드론을 육안으로 감시하며 항공기나 타 드론, 맹금류의 접근이 발견되거나 비행이상이 발견되면 즉시 조종자에게 보고하여야 한다.

가. GPS 상태 점검

○ 비행 중 GPS 신호의 수신 상태를 확인하여야 하며, GPS의 수신 개수가 7개 미만이면 드론의 위치 및 고도를 유지할 수 없게 되므로 수동조종 및 자동명령으로 GPS 신호 수신이 양호한 지역으로 드론을 이동하여야 한다.

나. 배터리 상태 점검

○ 비행 중 배터리의 상태를 점검하여야 하며, 배터리의 전압이 낮아질수록 빠르게 방전되므로 안전하게 착륙할 수 있는 배터리 잔량을 확인하여야 한다.

다. 기상상태 점검

○ 비행 중 강풍이 불거나 비 또는 눈이 오는 상황에는 비상착륙 또는 복귀명령을 실행하여 착륙한다.

라. 온도 점검

○ 드론은 고온에서 비행 시 오작동 할 수 있으므로 온도가 상승하면 즉시 착륙하여 열을 식혀준다.

마. 카메라 상태 점검

○ 비행 중 카메라가 정상적으로 작동하는지 확인하여야 하며, 메모리카드의 용량, 사진 매수 등을 점검한다.

바. 비행정보 점검

○ 비행 중 드론의 고도가 일정하게 유지되고 있는지를 확인하며 비행체의 자세정보를 수시로 체크하여 이상 유무를 확인한다.

사. 긴급착륙의 요인

○ 다음의 각 호에 해당하는 경우에는 안전을 위하여 드론을 즉각 착륙시켜야 한다.

① 비행 중 GPS신호가 약하거나 불량하여 위치를 파악할 수 없는 경우
② 배터리 잔량이 충분하지 않아 착륙지로 되돌아 올 수 없는 경우
③ 배터리 소모가 급격하게 발생하는 경우
④ 비행 중 기상의 변경으로 눈, 비가 오거나 풍속이 급격하게 증가하는 경우

⑤ 드론의 온도가 급격하게 상승하여 내구성에 문제가 생기는 경우
⑥ 카메라 메모리 카드나 전원 공급이 불량하여 촬영할 수 없는 경우
⑦ 기체 자체의 결함으로 내구성에 문제가 발생하는 경우
⑧ 기타 외부간섭(항공기, 타 드론, 조류)등에 의해 충돌이 발생할 가능성이 있는 경우

4.2.5 드론의 착륙

○ 드론의 착륙 시에는 착륙 예상 지역의 인원 통제, 착륙방향에 대한 안정성 확보 등이 요구되며 착륙도중 충돌우려가 있을 시에는 즉각 착륙을 취소하고 안전한 곳으로 착륙을 유도하여야 한다.

○ 드론의 착륙 형태는 선형착륙, 선회착륙, 낙하산착륙, 수직착륙 등이 있으며 드론의 기종에 따라 착륙 형태와 안전 범위가 상이함으로 착륙 형태와 드론 매뉴얼을 참조하여 주의사항을 인지하여야 한다.

○ 착륙 전 안전요원은 경광봉, 호루라기 등을 이용하여 착륙궤적 공간에 대해 사람, 차량, 기타 장애물이 진입하지 않도록 통제하여 임무수행을 마친 드론이 안전하게 착륙할 수 있도록 하여야 한다.

가. 선형착륙

○ 선형착륙은 일반적인 고정익 드론의 착륙방법으로 되도록 맞바람을 보고 기류를 타며 안전하게 착륙하도록 한다. 기체의 특성마다 착륙 범위 설정이 상이하며, 통상적으로 착륙 공간은 평평한 지형에서 최소 40m×80m의 공간을 확보하여야 한다.

○ 착륙 시에는 육안으로 드론을 모니터링 하여 착륙도중 충돌이 예상되는 경우 착륙을 중단하고 착륙위치를 조정하여 안전한 착륙을 도모하여야 한다.

나. 선회착륙

○ 선회착륙은 착륙반경 주변의 고도장애물로 인하여 선형착륙이 어려운 지역 또는 착륙하기 협소한 지역에서 사용하는 착륙방식으로 드론이 원을 그리며 선회하면서 고도를 낮추면서 착륙하는 방식이다.

○ 안전한 선회착륙을 위해서는 60m×60m 이상의 착륙공간이 있어야 한다.

다. 낙하산착륙

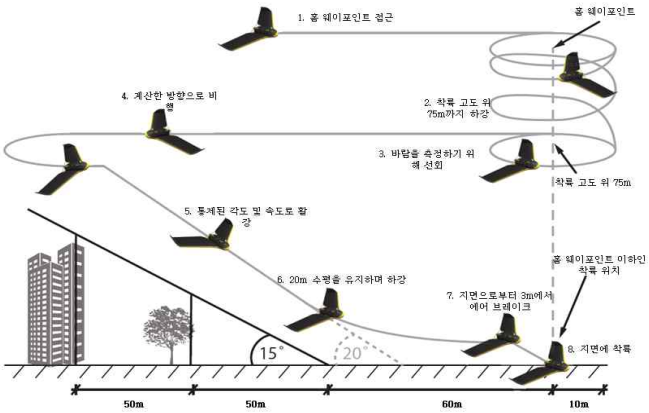
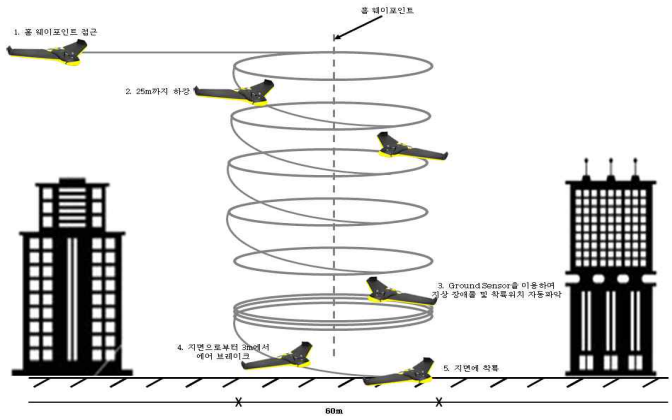
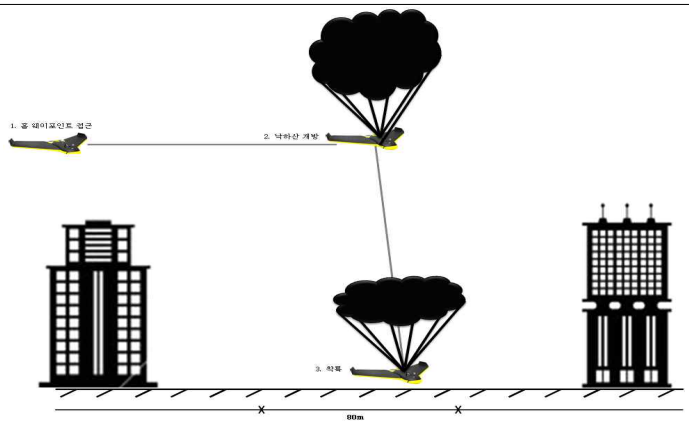
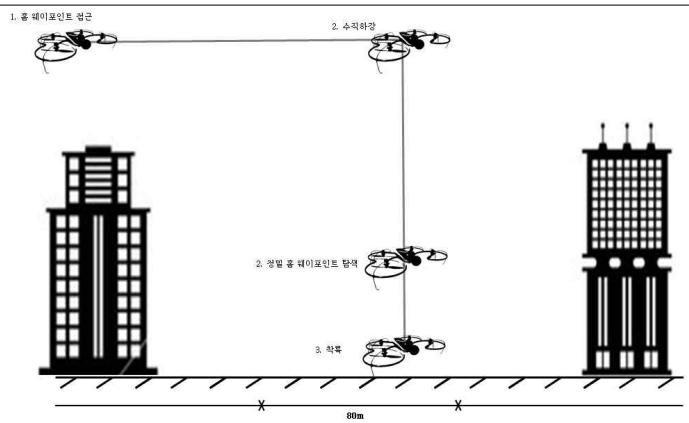
○ 낙하산착륙은 드론을 안전하게 착륙시킬 수 있으며, 비교적 좁은 착륙공간에도 착륙이 가능한 장점이 있지만 비행 중 드론의 착륙버튼을 클릭해야하므로 사용자의 정확한 판단이 요구된다.

○ 조종자의 능력에 따라 착륙위치가 달라질 수 있으며, 착륙 시 바람의 영향을 받아 예상 착륙지역을 벗어나 착륙할 수 있다.

라. 수직착륙

○ 회전익드론의 착륙방식인 수직착륙은 협소한 착륙공간에서도 착륙이 가능하며, 수직착륙 시 다소 시간이 소요되며, 갑작스런 강풍이나, 조류 등의 접근에 주의하여야 한다.

<표 6-1> 드론착륙의 유형

선형착륙	
선회착륙	
낙하산착륙	
수직착륙	

4.2.6 드론 비행 후

○ 착륙이 완료되면 조종사와 부조종사는 비행 전과 동일하게 체크리스트에 따라 드론을 점검하고 데이터를 취득하여야 한다.

가. 기체 점검

○ 기체에 크랙이 있는지 확인하거나 날개 등 부품이 파손되었는지 확인한다.

○ 기체의 GPS, 모터, 날개 등의 작동을 확인한다.

나. 모터점검

○ 육안으로 체크한 후 모터를 회전시켜 모터를 점검한다.

○ 모터가 부드럽게 돌아가는지와 주변에 이물질이 없는지 확인한다.

○ 모터가 부드럽게 회전하지 않는다면, 모터 안의 이물질을 확인하고 제거한다.

○ 압축 공기를 사용하여 이물질을 빼내면 모터내부에 손상이 생길 수 있으므로 주의하여야 한다.

다. 날개 및 프로펠러 점검

○ 날개는 비행체의 속도와 방향에 지대한 영향을 끼치므로 날개결함이 발견되면 반드시 수리하거나 교체하도록 한다.

○ 육안으로 프로펠러의 휨, 크랙 등을 검토하고, 결합부의 조임 상태를 확인한다.

1.1.1.1.2.1.

라. 라디오 모뎀 및 송수신기 점검

○ USB 라디오모뎀의 안테나의 단자를 확인하고 케이블의 연결상태를 확인한다.

마. 카메라 점검

○ 카메라 렌즈의 파손 및 균열을 확인한다.

○ 카메라 렌즈에 이물질이 있다면 반드시 제거해주어야 한다.

바. GPS 점검

○ GPS의 안테나의 손상유무 및 연결부위를 확인한다.

사. 배터리 점검

○ 배터리 연결케이블 단선 및 부식, 커넥터 손상유무 확인하고, 이상이 발견된 배터리는 즉각 폐기 처분해야 한다.

○ 사용한 배터리는 완충된 상태나 방전된 상태로 장기간 보관하면 배터리의 손상을 초래하므로 비행 후에는 모두 50~70%이상 충전 및 방전하여 보관한다.

4.2.7 데이터 정리

○ 촬영된 이미지 및 동영상과 위치 및 자세정보를 가지고 있는 로그데이터를 따로 분류하여 데이터를 백업한다. 백업 후 원본데이터를 확인하여야 하며 확인이 마무리되면

SD카드의 데이터를 삭제한다.

○ 비행한 날짜, 장소, 이륙·착륙시간, 중복도, 고도, 특이사항, 비행담당자 등 비행에 관련된 사항을 <표 6-7>과 같이 촬영기록부로 작성하여 보관한다.

<표 6-1> 촬영기록부 예시

사업명	*포트홀 탐지*			담당자	*홍길동*	
촬영장소	*20번국도 일원*			촬영일시	*2021년 00월 08일*	
투입장비	*eBeeX*	카메라	*S.O.D.A*	중복도	*종*	*70%*
촬영고도	*73m*	해상도	*2cm*		*횡*	*85%*
기 상	*맑음*	풍향	*서풍*	풍속	*2.0m/s*	
비행횟수	시작시간	종료시간	비행시간	사진 매수	로그파일	
1	*11:25*	*11:47*	*22분*	*244*	*EB-03-20350_0063*	
2	*11:52*	*12:15*	*23분*	*251*	*EB-03-20350_0064*	
3	*12:26*	*12:49*	*23분*	*206*	*EB-03-20350_0065*	
특이사항	*대상지 주변 송전탑으로 인한 일부구간 고도 상승* *비행 중 강풍으로 인한 촬영지연*					

4.3 사고의 보고

○ 항공안전법 제2조 8에 의거 “초경량비행장치사고”란 초경량비행장치를 사용하여 비행을 목적으로 이륙하는 순간부터 착륙하는 순간까지 발생한 다음 각 목의 어느 하나에 해당하는 것으로부터 국토교통부령으로 정하는 것을 말한다.

가. 초경량비행장치에 의한 사람의 사망·중상 또는 행방불명

나. 초경량비행장치의 추락·충돌 또는 화재 발생

다. 초경량비행장치의 위치를 확인할 수 없거나 초경량비행장치에 접근이 불가능한 경우

○ 항공안전법 제129조 3항 및 시행규칙 제312조에 따라 초경량비행장치사고를 일으킨 조종자 또는 그 초경량비행장치의 소유자는 다음 각 호의 사항을 지방항공청장에게 보고하여야 한다.

① 조종자 및 그 초경량비행장치 소유자의 성명 또는 명칭

② 사고가 발생한 일시 및 장소

③ 초경량비행장치의 종류 및 신고번호

④ 사고의 경위

⑤ 사람의 사상 또는 물건의 파손 개요

⑥ 사상자의 성명 등 사상자의 인적사항 파악을 위하여 참고가 될 사항

<표 6-1> 사고 시 통보기관 연락처

분류	담당부서	연락처
관할 지방항공청	서울지방항공청 항공안전과	전화 : 032-740-2146 fax : 032-740-2149
	부산지방항공청 항공안전과	전화 : 051-974-2147~8 fax : 051-971-1219
	김포/제주항공 관리사무소 및 지방항공 출장소	전화 : 02-6096-1032 fax : 02-6096-1031
항공 · 철도사고조사위원회(항공조사팀)		전화 : 044-201-5445 fax : 044-201-5696

* 관할 지방항공청 1곳과 항공철도조사위원회에 각각 통보

제 5 장

포트홀 탐지

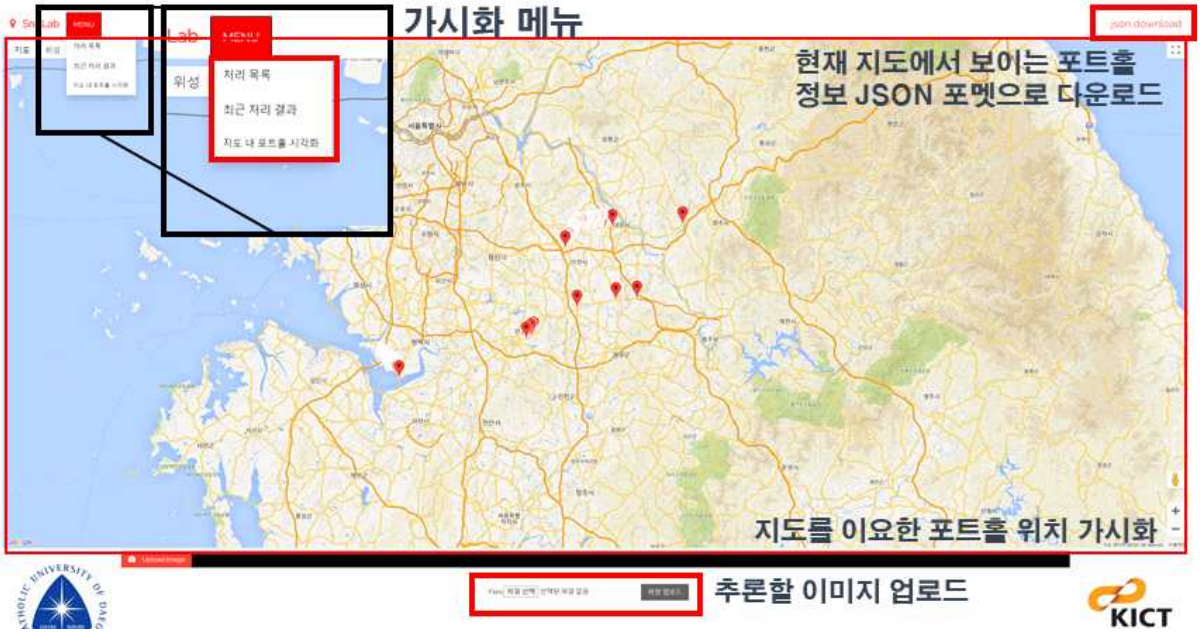
5.1 포트홀 자동 탐지시스템

5.2 포트홀 가시화 시스템

제5장 포트홀 탐지

5.1 포트홀 자동 탐지시스템

○ 포트홀 자동 탐지시스템은 항공 이미지를 업로드하면 항공 영상 내 포트홀을 탐지하여 포트홀의 위치 및 면적, 가로, 세로, 크롭된 포트홀 이미지 등을 반환한다. <그림 5-1>은 항공 영상 업로드 및 위치 가시화를 위한 포트홀 자동 탐지시스템의 메인 페이지이다.



포트홀 자동 탐지시스템 메인 페이지

5.1.1 이미지 업로드

○ 포트홀 자동탐지 시스템은 웹 어플리케이션으로 구현이 되어있으며 사람들이 주로 사용하는 Firefox, Chrome, Opera, Microsoft Edge에서 가용성 테스트를 진행하였다. 결과는 아래의 <표 5-1>과 같으며 4개의 웹브라우저 모두 이미지 업로드 및 포트홀 자동탐지 결과 확인이 가능하다.

○ 촬영한 항공 이미지 업로드 방법은 그림 <그림 5-2>와 같으며, 먼저 파일선택 버튼을 클릭하여 파일선택 창을 화면상에 띄우고 업로드를 원하는 이미지를 선택한다. 그리고 선택된 이미지를 자동탐지시스템에 업로드 하기 위해 파일 업로드 버튼을 클릭한다. 이후 <그림 5-2>의 우측에 이미지 업로드 진행 상황을 볼 수 있는 창을 확인할 수 있다. 이미지 업로드가 끝날 때까지는 해당 페이지에서 대기하여야 한다.

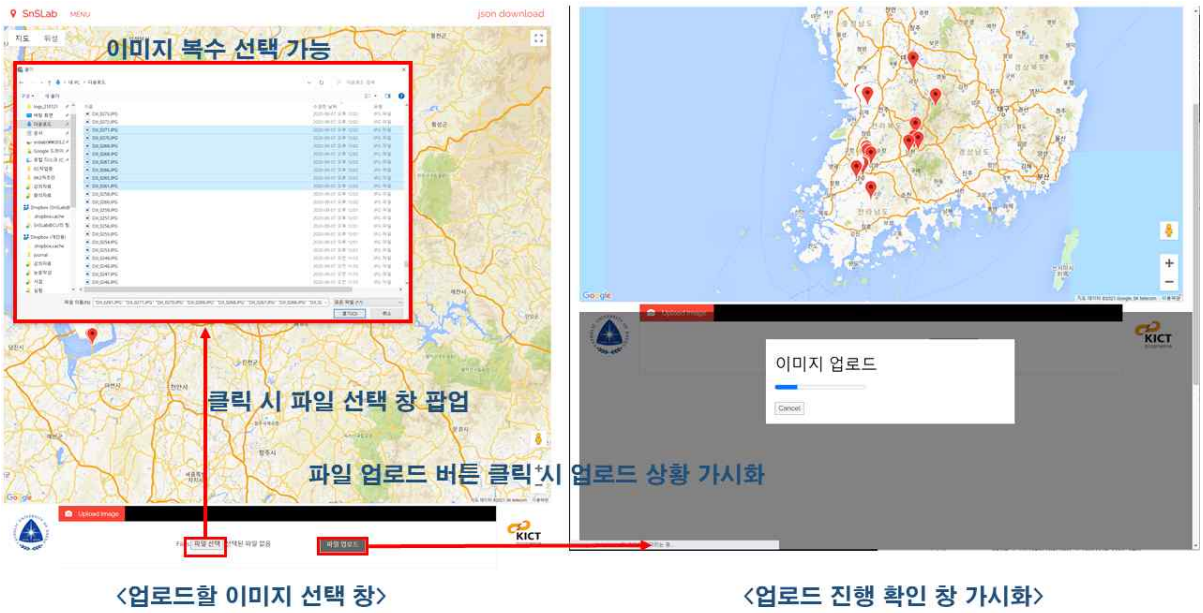
○ 이미지 업로드 시, DJI 회전익 드론, SensorFly 고정익 드론의 영상은 포트홀의 면적

측정을 위해 자동으로 픽셀 당 단위면적을 구할 수 있지만, 이미지 촬영 시 카메라의 CMOS 스펙이 다르거나 메타데이터의 키-값 구성이 다른 이미지가 들어올 시 업데이트가 필요하다.

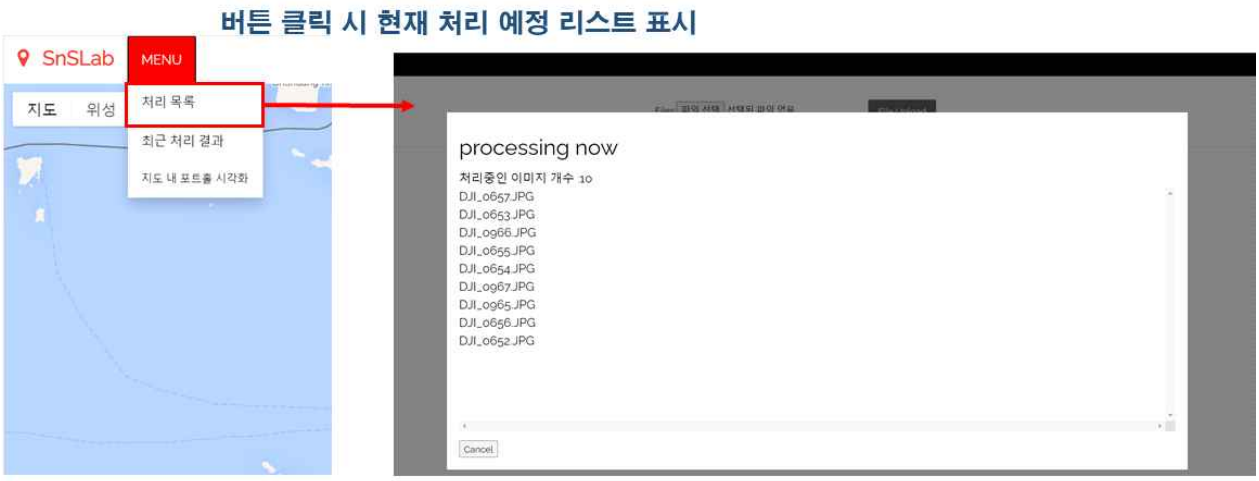
○ 이미지 업로드가 끝나고 처리 현황을 보기 위해서는 좌측 상단의 가시화 메뉴에서 처리목록 버튼을 클릭하여 <그림 5-3>과 같이 현재 처리 대기 중인 이미지들을 확인할 수 있다.

<표 5-1> 웹 어플리케이션 가용성 테스트

구분	Opera	Chrome	Firefox	Microsoft Edge
이미지 업로드	O	O	O	O
탐지결과 확인	O	O	O	O



이미지 업로드 진행



<좌측 상단 메뉴 버튼>　　<메인 페이지 지도 하단에 현재 처리 예정 리스트 표시>

처리 예정 이미지 확인 기능

5.1.2 포트홀 자동탐지 결과 확인

○ 포트홀 자동탐지가 끝나면 탐지된 포트홀의 위치가 구글 맵 상에서 마커 형태로 표시된다. <그림 5-4>와 같이 마커를 클릭 시, 항공 영상 내 포트홀의 위치를 확인할 수 있다.

○ 직전에 업로드한 이미지들의 처리결과를 한 번에 보기 위해서는 메인페이지 좌측상단 메뉴 버튼 중 최근 처리결과 버튼을 클릭하면 된다. 최근 처리결과 버튼을 클릭하면 <그림5-5>의 우측 이미지와 같이 여러 장의 이미지 자동탐지 결과를 볼 수 있다.

○ 메인페이지 내 구글 맵에 시각화된 탐지결과를 한 번에 보기 위해서는 메인페이지 좌측 상단 메뉴 버튼 중 지도 내 포트홀 시각화 버튼을 클릭하면 <그림 5-6>의 우측 이미지와 같이 여러 장의 이미지 자동탐지 결과를 볼 수 있다.

○ <그림 5-1>의 우측 상단의 json download 버튼을 클릭할 시, 현재 구글 지도에서 보이는 마커들의 포트홀 탐지결과 메타데이터를 다운 받을 수 있다. 메타데이터에 대한 자세한 내용은 5.2.3의 내용을 참고하길 바란다.



<메인 페이지의 마커 클릭>　　<자동 탐지 결과 시각화 페이지 >

포트홀 자동탐지 결과 확인



<최근 업로드 이미지 처리 결과 확인 버튼>　　최근 업로드 이미지 목록　　<최근 업로드 이미지 처리 결과 시각화>

최근 업로드된 이미지의 포트홀 자동탐지 결과



<메인 페이지 지도 내 포트홀 시각화 버튼 클릭>　　<이미지들의 포트홀 자동 탐지 결과 시각화>

메인 페이지 지도 내 포트홀 자동탐지 결과

5.1.3 포트홀 시각화

○ 포트홀 가시화 시스템은 포트홀 자동 탐지시스템의 결과를 시각화하고 실무자들이 포트홀의 위치와 면적, 가로, 세로 등의 데이터를 받을 수 있도록 JSON(JavaScript Object Notation) 형태의 메타데이터를 제공한다.

가. 포트홀 시각화 페이지

○ 포트홀 시각화 페이지는 <그림 5-7>과 같다. 메인 페이지의 지도상 특정 위치의 포트홀과 포트홀에 대한 정보를 확인하기 위한 페이지이다.

1) 항공 영상 시각화

○ <그림 5-7>의 좌측 하단에 위치한 항공 영상 목록에서 원하는 항공 영상을 선택하면, 해당 항공 영상의 포트홀 추론 결과가 페이지의 우측에 출력된다.

○ <그림 5-8>과 같이, 현재 도로의 하행, 상행을 알기 위해 우측 상단의 4 표식을 이용하여 북쪽을 표시한다.

2) 탐지된 포트홀 시각화

○ 객체분할 인공신경망의 자동탐지 결과로 나온 포트홀을 시각화한다.

○ <그림 5-9> 우측의 포트홀 이미지 클릭 시, 포트홀의 위치를 포트홀 <그림 5-9> 좌측에 있는 항공 영상 이미지에 시각화하도록 구현하여 항공 영상 내 포트홀의 위치를 확인할 수 있다.

○ <그림 5-10>의 우측 그림에 위치한 포트홀 가로, 세로 출력 버튼을 클릭하면 포트홀의 면적과 가로, 세로를 보여주어 실무자들이 포트홀의 위험도를 좀 더 쉽게 측정할 수 있다. 그리고 포트홀 영역 및 좌표 출력 버튼을 클릭하면 포트홀에 대한 탐지 영역과 해당 포트홀의 좌표가 출력된다.

3) 메타데이터 다운로드

○ <그림5-7> 우측 상단의 Download Json 버튼을 클릭하면 현재 포트홀 시각화 페이지에서 보이는 포트홀의 메타데이터 정보를 얻을 수 있다.



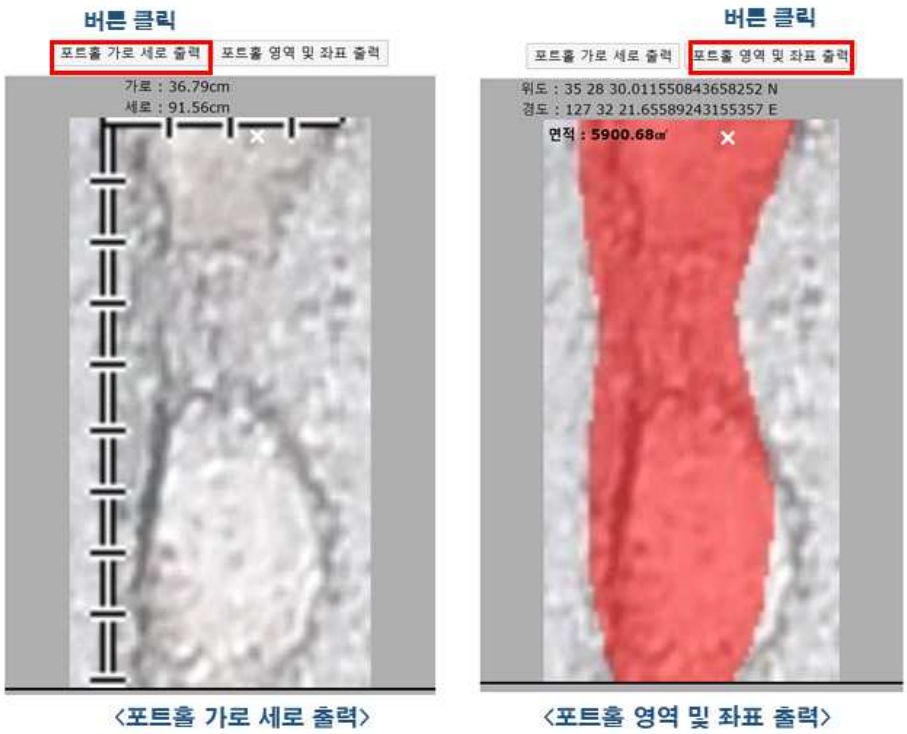
포트홀 시각화 페이지



정북 방향 확인



항공 영상 내 포트홀 위치 가시화 기능



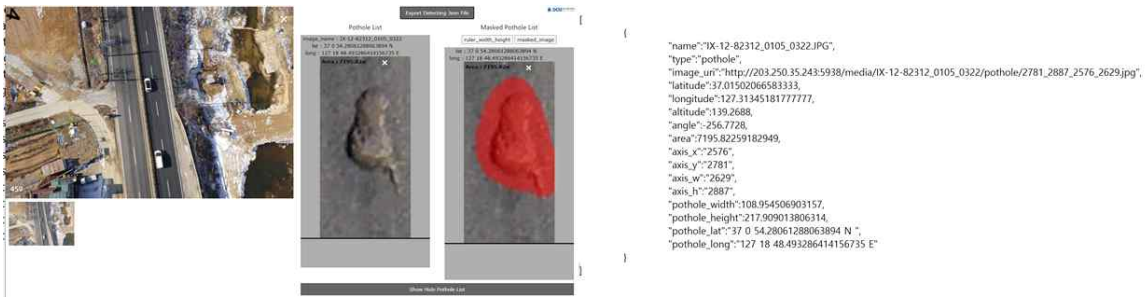
포트홀 자동탐지 결과 및 정보 시각화 기능

5.2.3 메타데이터 정보

○ 포트홀 탐지결과 데이터는 JSON(JavaScript Object Notation) 포맷으로 구성이 되어 있다.

○ 포트홀 탐지결과 데이터는 원본 영상의 이름과 class, 이미지 저장 경로, 위도, 경도, 고도, 포트홀의 면적, 포트홀의 좌표, 포트홀의 가로, 세로 등의 데이터가 저장되어있다.

○ <그림 5-11>은 JSON 포맷으로 구성된 포트홀 탐지결과 데이터의 예시이다.



포트홀 탐지결과 데이터의 예시

5.2 포트홀 가시화 시스템

○ 드론의 촬영 영상의 포트홀 탐지 영상분석 결과데이터를 가시화하여 운용할 수 있는 웹 서비스 시스템이다. 운용자는 포트홀 영상분석 결과데이터를 실시간 업로드하여 Vworld기반 지도 레이어에 가시화 할 수 있으며, 포트홀 영상분석 데이터의 직관적인 조회기능을 제공한다. 포트홀 영상분석 전용 뷰어를 통해 포트홀의 탐지위치 확인과 원본 촬영영상 다운로드 등의 운용편의성 UI를 제공한다.

○ 위험도 기준은 786개의 차량 데이터를 조사하여 DB화 하였으며 DB데이터를 바탕으로 대형, 중형, 소형 차량의 타이어 면적에 따라 위험도 임계치는 각각 상,중,하로 표현된다.

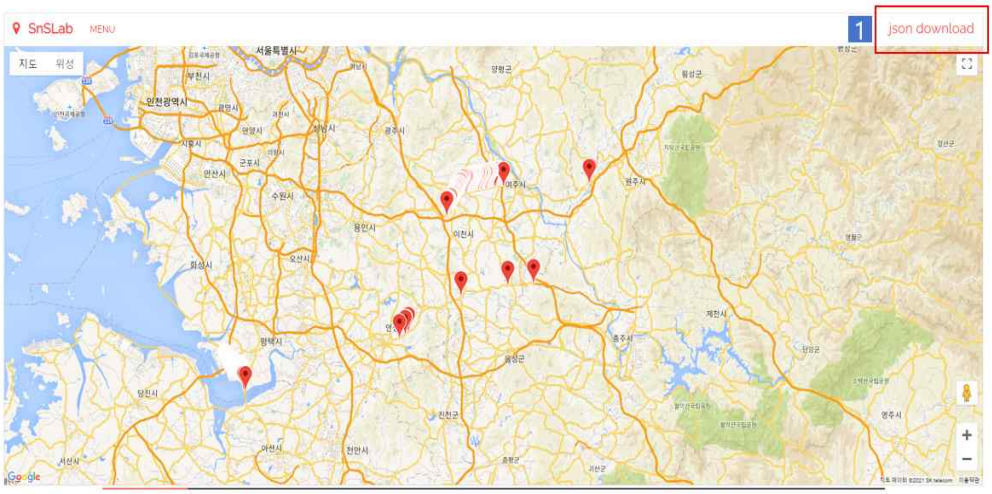
○ 포트홀 영상분석 결과와 탐지위치에 대한 작업 리포트 PDF출력 기능을 제공한다.

○ 포트홀 영상분석 결과에 대한 엑셀 출력 기능을 제공한다.

5.2.1 데이터 연동

○ 포트홀 데이터 업로드 시 시스템 서버는 외부네트워크가 오픈되어 있어야 한다. (KAKAO API 연동 GPS 주소정보 변환, 영상이미지 수집을 위한 분석서버와의 외부 통신 요구.)

○ 포트홀 분석 웹서비스(http://203.250.35.243:5938)에 접속하여 ①번 “json download” 버튼을 클릭하면 시스템에 업로드 가능한 포트홀 영상분석결과가 다운로드 된다.



포트홀 분석 데이터 다운로드

○ 포트홀 업로드 데이터 형식은 다음과 같다.

<표 8-1> 데이터 업로드 형식 (JSON)

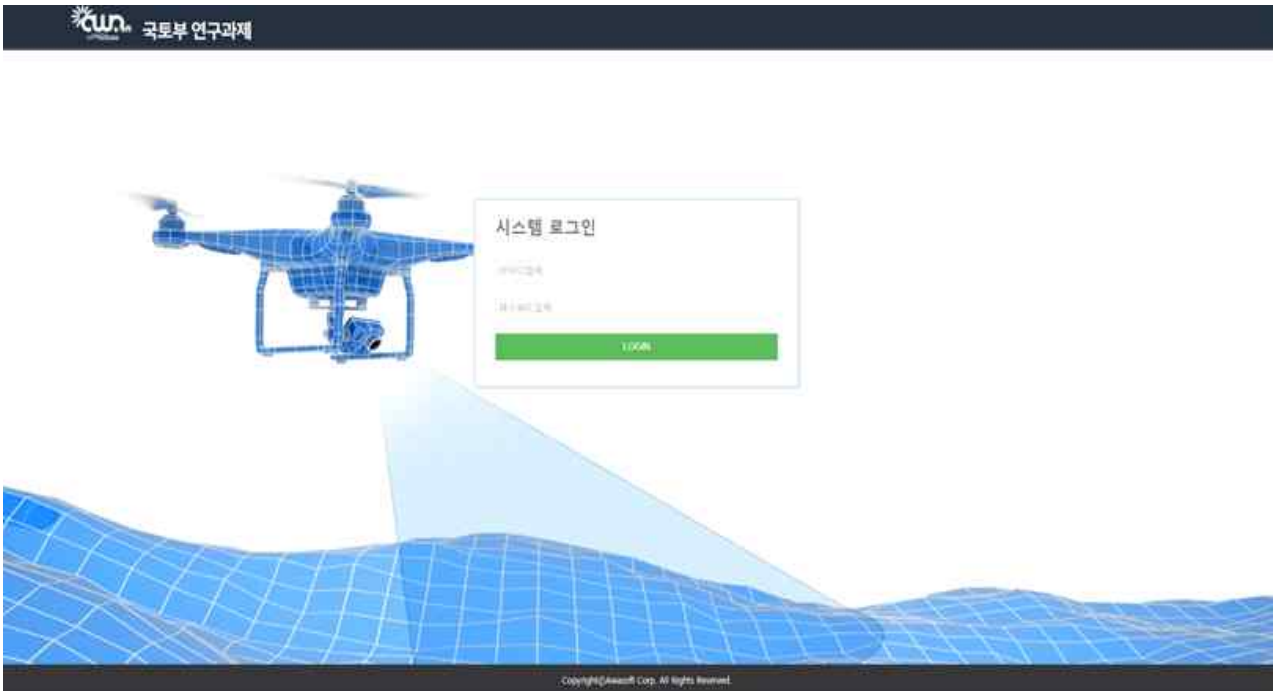
키	값(샘플예시)	설명
name	IX-12-82312_0105_0319.JPG	드론 촬영이미지 파일명
type	pothole	탐지유형
image_uri	http://203.250.35.243:5938/media/IX-12-82312_0105_0319/pothole/286_368_2670_2729.jpg	탐지 이미지 URL
latitude	37.015123742777774	드론 촬영위치 위도
longitude	127.31386238472221	드론 촬영위치 경도
altitude	139.7881	고도
angle	251.28594	정북방향 차이각도
area	6259.829098111819	탐지된 포트홀 면적
axis_x	2670	탐지 된 Tile의 시작 x좌표
axis_y	286	탐지 된 Tile의 시작 y좌표
axis_w	2729	탐지 된 Tile의 종료 x좌표
axis_h	368	탐지 된 Tile의 종료 y좌표
pothole_width	121.943534047535	포트홀 너비
pothole_height	169.480843930472	포트홀 높이
pothole_lat	37 0 54.65381581755275 N	포트홀 위도
pothole_long	127 18 49.06482596814736 E	포트홀 경도

5.2.2 포트홀 가시화 시스템 매뉴얼

○ 로그인

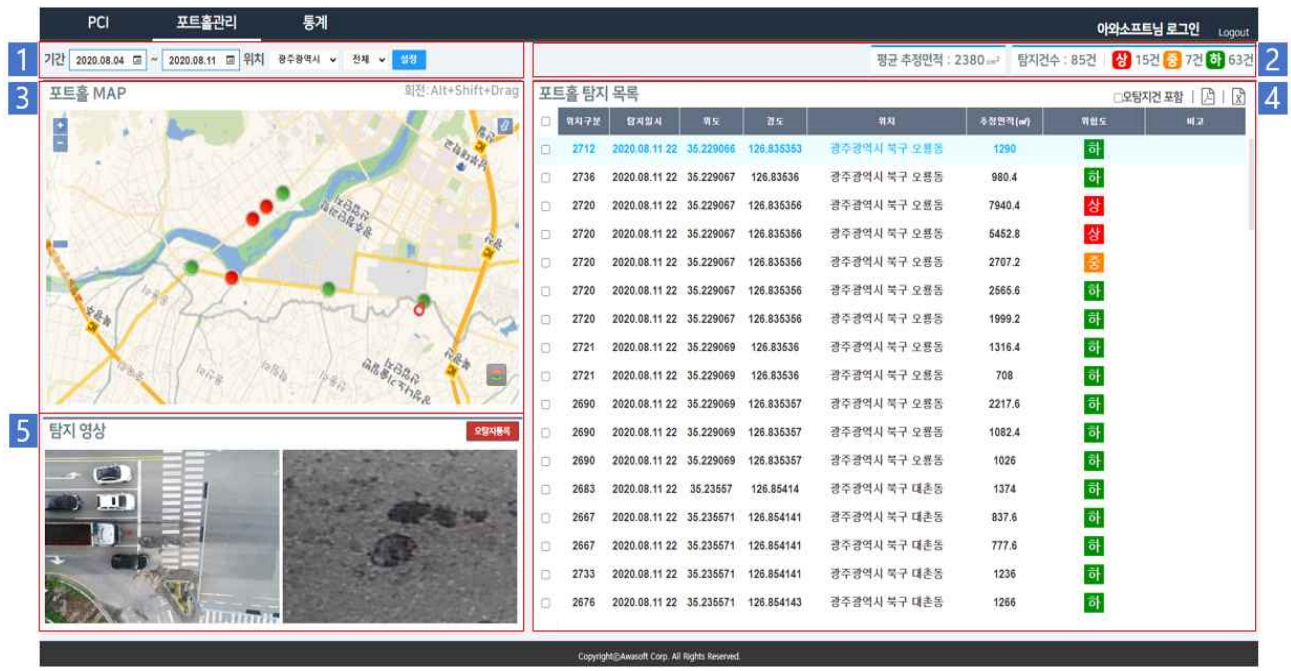
가. 등록된 사용자에 대해서만 시스템 접속이 가능하다.

나. 로그인 성공 시 포트홀 대시보드 페이지로 이동한다.



포트홀 가시화 시스템 로그인

○ 포트홀 대시보드



포트홀 대시보드

① 탐지기간 및 위치 검색조건을 설정할 수 있다. “설정” 버튼은 시스템 관리 권한 유저에게만 노출되며 포트홀데이터 업로드, 오탐지 일괄처리 등 운용관련 기능을 제공한다.

② 조회된 포트홀 영상분석 데이터의 Summary정보를 표출한다.

③ 포트홀의 탐지위치정보가 매핑된 지도 레이어, 탐지 영상의 정북방향 각도가 동기화되어 표출된다. 상,중,하 각 위험도별 아이콘을 표기하며 포인트 클릭 시 해당 위치의 ④탐지목록을 하이라이트 해주며, ⑤탐지영상의 이미지를 갱신한다.

④ 조회된 포트홀 영상분석 데이터를 목록형태로 출력한다. 목록내 행을 클릭 시 해당 데이터를 기준으로 ③,⑤레이아웃이 갱신된다. “엑셀출력”아이콘을 클릭하여 데이터의 엑셀출력

기능을 제공한다. 체크박스를 체크 후 “PDF출력” 버튼을 클릭하여 포트홀 작업 리포트 출력 기능을 제공한다.

⑤ ③,④의 UI를 사용하여 선택된 데이터의 드론촬영영상(좌), 포트홀탐지영상(우)를 출력한다. “오탐지등록”버튼을 통한 오탐지처리 기능을 제공한다. 영상을 클릭 시 포트홀 영상 뷰어 화면이 표출된다.

○ 포트홀 영상 뷰어



포트홀 영상 뷰어

① ③드론 촬영 영상의 확대, 축소

② 드론촬영 원본이미지 다운로드

③ 드론 촬영영상이미지 ⑤번의 선택한 포트홀 위치에 따라 선택=붉은색 마커, 미선택=노란색마커로 표시한다.

④ ⑤번의 선택된 포트홀의 확대된 영상을 출력한다.

⑤ ③번 드론 촬영 위치 내 탐지된 포트홀들의 면적과 함께 출력한다. 탐지 된 포트홀 클릭 시 ③번영상에 마크업되면 ④번 영상이 갱신된다.



포트홀 통계 화면

① 포트홀 탐지현황 통계자료의 조회기간을 설정한다.

② 일별 시스템 내 포트홀 탐지건수에 대한 차트를 출력한다.

③ ②번 차트 데이터를 기준으로 위험도 분포도를 출력한다.

④ ②번 차트를 클릭 시 해당 날짜의 위치별 포트홀 탐지결과를 출력한다.

⑤ ④번 차트 데이터를 기준으로 위험도 분포도를 출력한다.

드론 초분광
아스팔트 포장상태 조사 매뉴얼

1. 총 칙

1.1 적용범위

(1) 이 매뉴얼은 드론과 초분광 센서를 이용한 도로 포장 상태의 조사에 적용한다.

(2) 도로 포장 상태는 PCI(Pavement Condition Index)로 표현되며, 소성변형, 균열율, 평탄성 등을 이용하여 산출된다.

(3) 산출된 도로 포장 상태를 이용하여 도로의 유지보수 방법 및 보수시기 결정 등에 참조할 수 있다.

1.2 드론을 이용한 초분광 영상 개요

(1) 초분광 영상을 이용한 도로의 상태 조사는 드론을 이용한 원격탐사 방법으로써 초분광 센서를 이용하여 가시광선과 근적외선 파장대를 촬영하여 이용한다.

(2) 조사에 사용되는 초분광 센서는 드론에 탑재가 가능하여야 하며, 가시광선과 근적외선 파장인 400~1,000nm을 주로 이용한다.

(3) 이 매뉴얼에서는 초분광 영상을 이용하여 도로 노후화 상태를 진단하고 도로 포장 상태를 평가한다.

✢ 태양광선은 파장의 길이에 따라 인간의 눈에 보이는 적색(Red), 녹색((Green), 청색(Blue) 등을 포함하는 400 ~ 700nm의 가시광선 파장과 이보다 짧은 200 ~ 400nm의 자외선 파장, 가시광선 보다 긴 700nm 이상의 적외선 파장으로 구분한다.

✢ 적외선은 700 ~ 1,100nm의 근적외선, 1,100 ~ 2,500nm의 중적외선 파장으로 구분한다.

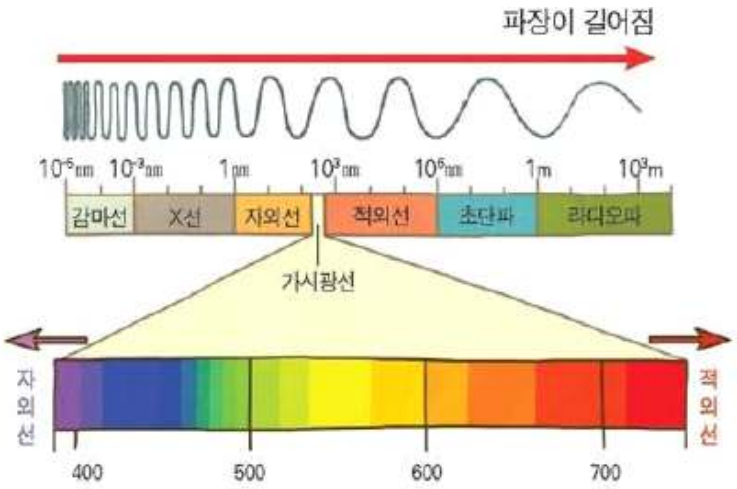


그림 1.1 빛의 파장영역(출처: 네이버 지식백과)

✢ 파장영역을 몇 개로 구분하여 영상을 촬영하느냐에 따라서 다중분광영상(multi-spectral image), 초분광영상(hyperspectral image)으로 구분하며, 이러한 특성을 분광해상도라고 한다.

✢ 일반적으로 다중분광영상은 파장영역 10개 이하의 불연속적인 구간으로 나누어 촬영되는 영상을 의미하고, 초분광영상은 파장영역을 10nm이하의 좁은 폭으로 수백개로 구분하여 연속적으로 촬영하는 영상을 의미한다.

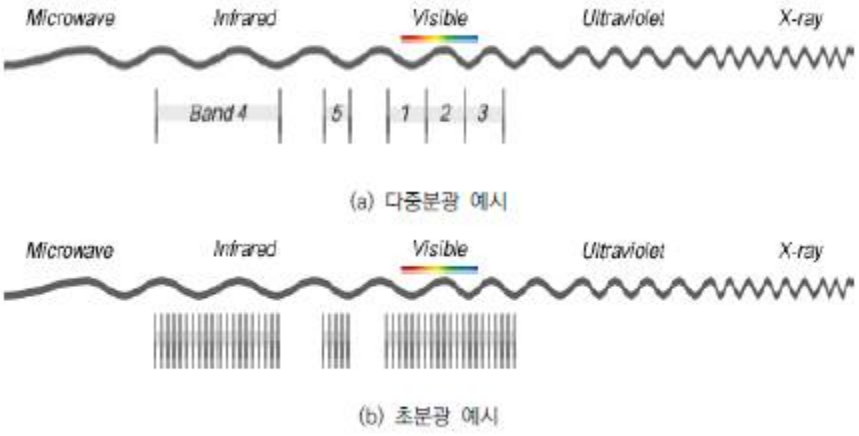


그림 1.2 다준분광과 초분광의 차이(출처: https://gisgeography.com)

✢ 초분광 영상은 빛의 파장 대역을 다수의 밴드로 나누어 대상체가 가지는 고유의 스펙트럼 형상을 추출하여 지표상 물체의 연속적이고 좁은 파장역으로 수백 개의 분광 정보를 취득하므로 대상 물체마다 존재하는 고유의 광학적 성질 및 반사 특징을 분석할 수 있어 토지피복, 식생 등 대상물체 구별에 주로 활용된다.

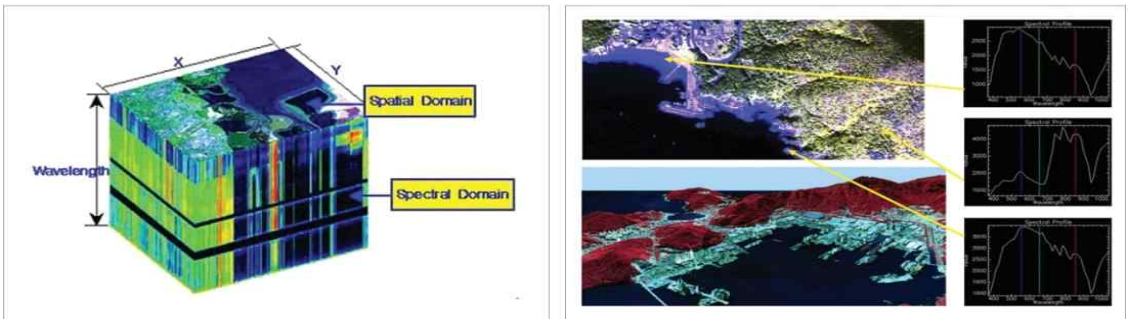


그림 1.3 초분광 이미지 구성 및 분광데이터

2. 사용 장비 및 프로그램 등의 기준

2.1 시스템 구성

(1) 드론 초분광 아스팔트 포장상태 조사 시스템은 초분광센서, 드론, IMU/GPS, 데이터 처리 SW로 구성된다.



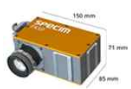
그림 2.1 드론 초분광 아스팔트 포장상태 조사 시스템의 구성

2.2 초분광 카메라

(1) 조사 시스템에 적합한 초분광 카메라는 드론에 탑재가 가능하여야 하며, 400~1,000 nm 파장대를 촬영할 수 있어야 한다.

(2) 이 매뉴얼에서는 드론 탑재가 가능한 초분광 카메라 중 미국 Headwall사의 Nano-HyperSpec을 표준으로 하며, 다양한 제품 개발에 따라 적합한 카메라를 선정하여 적용한다.

표 2.1 드론 탑재용 초분광 카메라 종류

센서	Nano-Hyper spec	Micro-CASI 1920	FX-10	SOC710-GX	SHARK	OCI UAV
제조사	headwall (미국)	ITRES (캐나다)	Specim (핀란드)	Surface Optics (미국)	Corning (미국)	Bay Spec (미국)
크기(mm) /무게(kg)	76×76×120 / 0.52	102×190×178 / 1.50	150×71×85 / 1.40	103×200×200 / 1.25	137×88×71 / 0.680	80×80×30 / 0.272
밴드수	270	288	220	120	180	100
분해능력(nm)	2.2	2.1	-	4.2	3.3	5
파장대(nm)	400-1,000	400-1,000	400-1,000	400-1,000	400-1,000	600-1,000
센서사진						

2.3 드론과 짐벌

(1) 드론은 초분광센서를 탑재하고 안정적인 운영이 가능하여야 한다.

(2) 이 매뉴얼에서는 최대 이륙중량이 15.5kg인 DJI사의 Matrice 600 PRO를 표준으로 한다.

(3) 드론에는 짐벌을 설치하여 초분광센서가 안정적으로 촬영할 수 있어야 한다.

표 2-2 Matrice 600 PRO 드론 스펙

장비	항목	세부 사양
 Matrice 600 PRO	크 기	1668×1518×727 mm (프로펠러, 프레임 암, GPS 마운트)
	무 게	9.5 kg (배터리 6개 포함)
	최대이륙중량	15.5 kg
	최대속도	65km/h
	최대 실용 상승 한도 (해발 고도)	2500m

2.4 GPS/IMU 시스템

(1) 드론에는 GPS/IMU 시스템을 장착하여 초분광센서에 장착되어 탑재 비행체의 위치, 시간, 속도, 방향 등의 종합적인 자세정보를 제공하여야 한다.

- GPS(Global Positioning System)는 인공위성으로부터 수신된 전파를 해석하여 위치와 속도, 시간을 알 수 있는 범 세계위치결정 시스템이다.
- IMU(Inertial Measurement Unit)는 중력, 대상의 기울기 및 회전력을 감지하는 센서들로 구성되어있는 관성측정장치이다.



그림 2.2 GPS Antenna 및 IMU 센서

2.5 드론 초분광시스템 운용 소프트웨어

(1) 도로포장상태 평가를 위한 도로 분광라이브러리 구축을 위해서는 촬영 대상 도로에 적합한 드론 운용 궤도 및 일정한 고도, 속도를 유지하여야 한다.

(2) 드론 운용을 위해 소프트웨어는 UgCS를 표준으로 한다.

(3) UgCS를 이용하여 최신 디지털고도모델(DEM)과 KML 파일 불러오기와 대상지역의 정확한 촬영을 수행한다.



그림 2.3 드론 운용 SW(Ugcs-Universal Ground Control Software)

2.6 드론 초분광영상 전처리 소프트웨어

(1) 초분광센서의 데이터는 기하보정, 방사보정, 정사보정 등의 전처리를 하여야 한다.

(2) 전처리 과정은 일반적으로 제작사에서 제공하는 소프트웨어를 사용한다.

(3) 이 매뉴얼에서는 Headwall사의 SpectralView SW를 표준으로 한다.

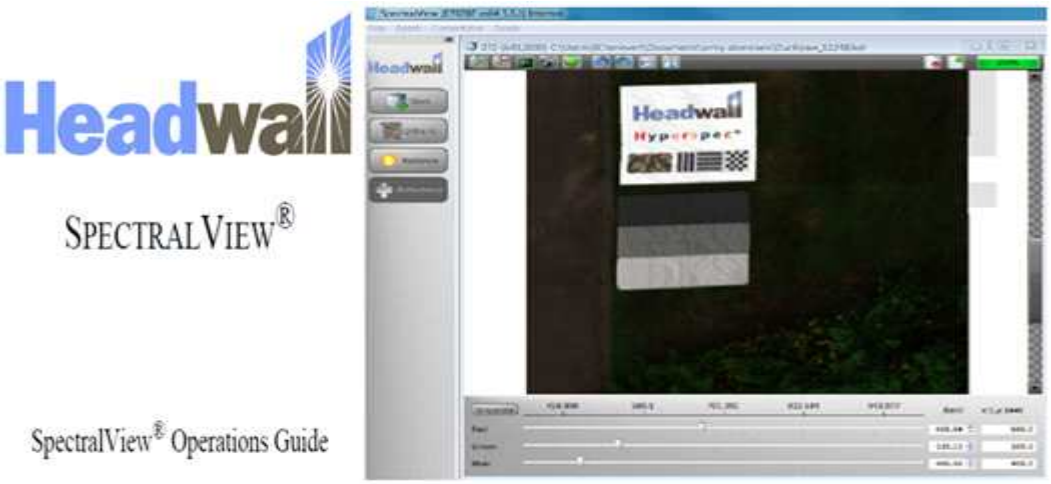


그림 2.4 초분광영상 처리 SW (SpectralView)

2.7 드론 초분광영상 데이터 변환 소프트웨어

(1) 영상데이터의 시각화, 데이터 분석을 위해 오픈소스 기반의 QGIS를 표준으로 사용한다.

(2) QGIS를 이용하여 데이터 보기, 생성, 관리, 내보내기 등을 수행한다.

2.8 드론 초분광영상을 이용한 포장상태분석 소프트웨어

(1) 초분광 포장상태분석 프로그램(HSPCA 1.0)을 이용하여 텍스트(CSV)로 변환된 자료를 분석한다.

(2) HSPCA (Hyper-spectral Pavement Condition Analysis)는 이 시스템을 위하여 개발되었으며, 딥러닝을 이용한 도로인식과 차량, 차선 등의 데이터 제거, 포장상태 분석 등을 수행한다.

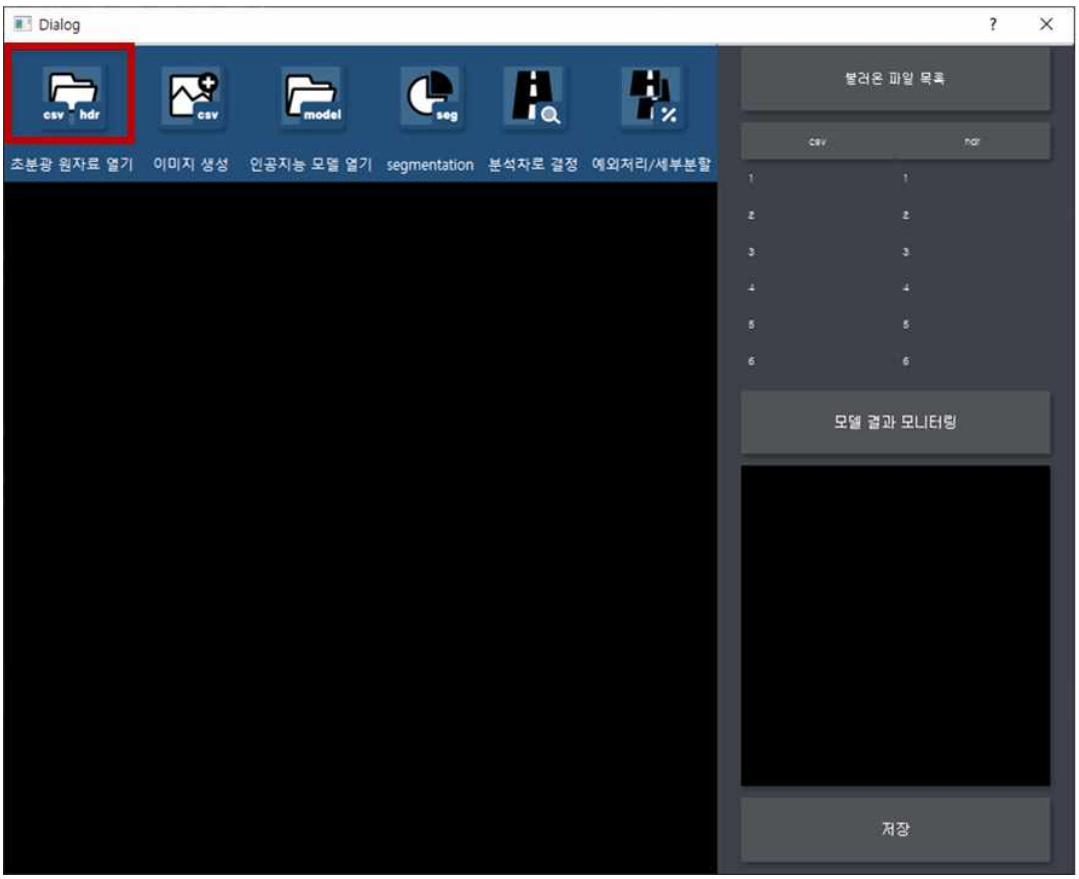


그림 2.5 포장상태분석 소프트웨어 (HSPCA)

2.9 드론 포장상태분석 가시화 시스템

(1) 드론 포장상태분석 가시화 시스템은 분석 결과를 기반으로 PCI 데이터 산출 및 시각화하는 서비스를 제공하는 시스템이다.

(2) 운용자는 초분광 영상 분석 자료를 실시간 업로드하여 Vworld 기반 지도 레이어에 위치정보를 가시화하고, 10m 구간별 포장상태 분석자료를 조회할 수 있다.

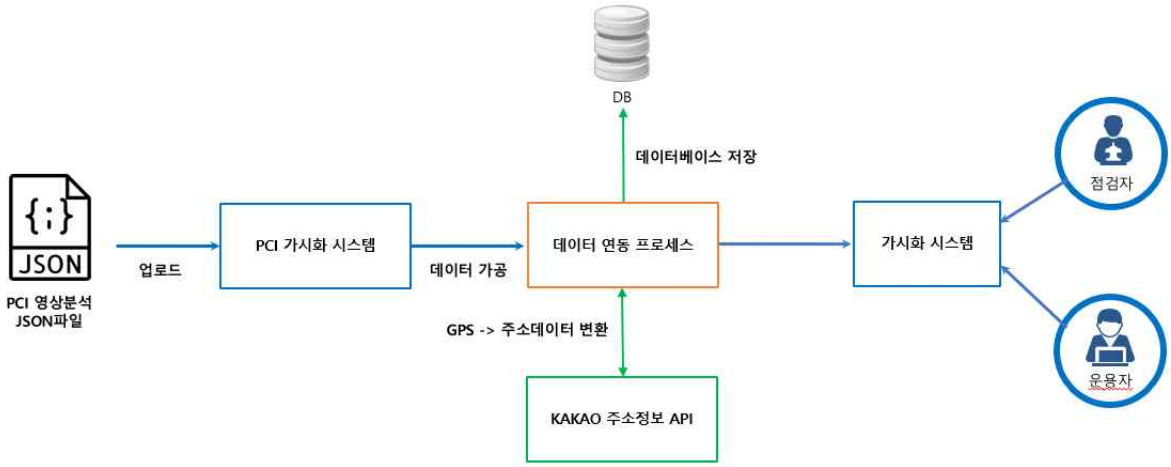


그림 2.6 PCI가시화 시스템

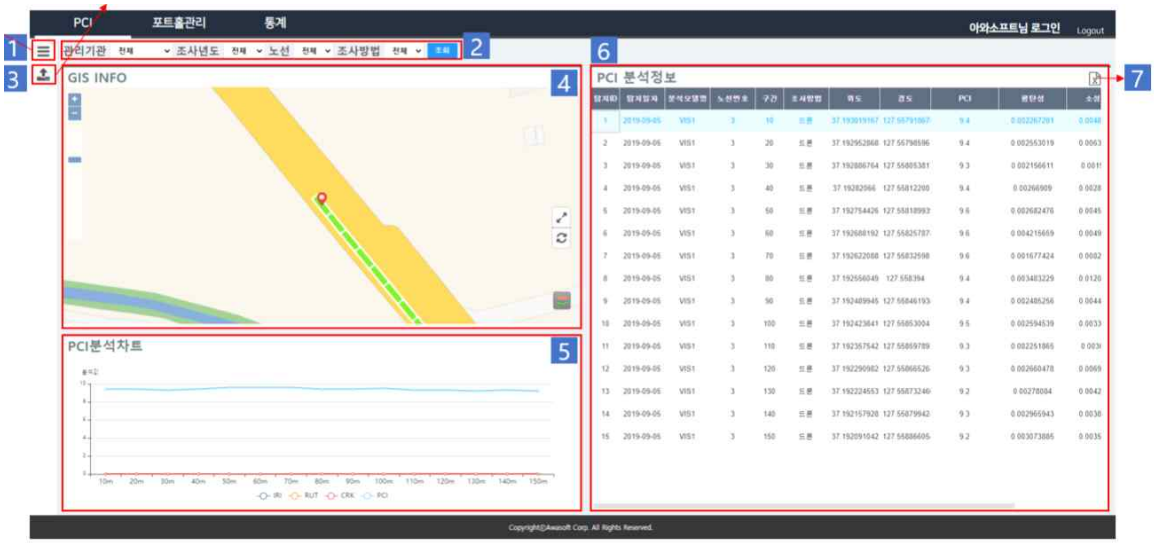


그림 2.7 PCI 데이터 조회

2.10 드론 보험 가입 및 신고

(1) 드론을 사업용으로 운용하기 위해 항공사업법에 따라 보험에 가입하여야 하며, 대인사고 발생시 1.5억원 이상, 대물 피해 2천만원 이상이 보장되어야 한다.

✢ 보험가입 여부는 보험가입증서와 드론 기체 시리얼 번호를 확인한다.

(2) 드론은 사업용으로 사용하기 위해 국토교통부에 항공안전법에 따라 초경량비행장치에 대한 신고를 하여야 한다.

✢ 드론 신고시에는 드론 원스톱 민원서비스(https://drone.onestop.go.kr) 등을 이용하여 항공안전법 시행규칙 별지 제116호 서식에 따른 문서와 관련 첨부서류를 제출한다.

✢ 드론 신규신고 사유일부터 30일 이내 신고를 하여야 하며, 변경·이전 사유발생시에도 30일 이내에 신고하여야 한다. 드론 말소시에는 해당일부터 15일 이내에 신고한다.

✢ 드론 신고 및 변경신고를 위반할 경우 항공안전법에 따라 6개월 이하의 징역이나 500만원 이하의 벌금을 받을 수 있다.

2.11 드론 조종자

(1) 드론을 운용하기 위해 항공안전법에 따라 초경량비행장치의 조종을 위해 발급하는 증명을 받아야 한다.

✢ DJI사의 Matrice 600 PRO 기체는 최대이륙중량이 15.5kg이므로 2종 무인동력비행장치 이상의 조종자 자격이 있어야 한다.

3. 촬영 계획 수립

3.1 대상지 선정 및 공역 확인

(1) 대상지는 활용성, 시급성, 안전성, 지속성 등을 고려해 우선순위를 선정하고, 노선길이, 비행조건, 이·착륙장소, 기상 등을 고려하여 선정하여야 한다.

(2) 대상지 선정 시 노선정보와 위치 등을 확인을 통하여 드론 비행 가능여부 판단해야하고 대상지의 정확한 위치를 인지하여야 한다.

(3) 공역을 확인하기 위하여 일반적으로 한국 드론 협회에서 제작한 스마트폰 앱인 Ready to Fly이나 구글의 와우드로 비행지도, 브이월드 비행지도 등을 이용하여 확인하여야 한다.

✢ Ready To Fly 앱은 안드로이드, 아이폰 모두 설치가 가능하며, 공역정보 조회(전국에 설정된 공역 현황), 지역정보(비행승인 기관 연락처 등), FAQ(드론 관련 필요정보, 질의응답 등), 조종자 준수사항, 기상정보(일출·일몰시각, 풍속, 지구자기장 정보 등), 비행승인신청서 등 비행허가에 필요한 자료를 제공한다.



그림 3.1 Ready to Fly 앱 설치 및 기능 (출처 : 국토교통부)

✢ 와우드로 비행지도는 구글에서 검색하여 접속할 수 있다. 왼쪽의 항목에서 비행금지구역, 비행제한구역, 공항(관제권), 비행장(관제권) 등과 같은 항목을 켜거나 끌 수 있으며, KML 파일로 다운받을 수 있다.

3.2 현장답사

(1) 대상지 최종 선정과 안전한 촬영을 위하여 현장답사를 한다.

(2) 현장답사시 드론의 이·착륙 및 관제위치를 선정하며, 위급상황시 착륙할 수 있는 비상 착륙지 등을 파악한다.

(3) 이·착륙지와 관제위치는 개활지, 공원, 대형 주차장, 운동장과 같이 넓고, 드론의 비행을 조망하기 좋으며, 비행과 전파 송수신이 용이하도록 주변에 높은 장애물이 없는 곳을 선정하는 것이 좋다.

(4) 이·착륙지는 회전익 드론의 경우 최소 10m×10m 범위 내에 고층건물이나 바위, 전력선, 언덕, 나무 등의 장애물이 없는 곳을 선정하여 추락 시 안전에 대비하여야 한다. 또한, 표면이 평평하고 먼지가 발생하지 않아야 한다.

✢ 이·착륙지 선정시 불가피할 경우 차량통행이 없는 도로나 단층건물의 옥상 등 하늘이 보이는 지역을 활용할 수 있다. 다만, 조종자의 보다 높은 주의가 필요하다.

(5) 드론이 비행 중 고장이나 예기치 못한 사고로 복귀하지 못하는 경우를 대비하여 이·착륙지 기준을 고려하여 노선 주변에 비상 착륙지를 파악하여야 한다.



운동장



공원



개활지

그림 3.2 이 · 착륙 장소(예)

3.3 촬영계획 수립

(1) 현장답사를 통해 선정한 이착륙 장소와 관제위치를 확인하고, 주변의 아파트, 송전탑, 나무 등의 고도장애물을 고려하여 비행고도와 비행노선을 결정한다.

(2) 비행고도는 목표하는 촬영 해상도에 적합하여야 하며, 고도 장애물 보다 20m 이상 높아야 한다.

(3) 표준 초분광 카메라로 선정한 미국 Headwall사의 Nano-HyperSpec는 비행고도는 약 70m일 경우 20m의 촬영 폭으로 영상을 취득할 수 있으며, 비행 표준 속도는 5m/s이다.

✢ Headwall사의 Nano-HyperSpec 초분광 카메라는 라인스캔(Push-Broom Linescan) 방식이므로 1회 촬영시 전체적인 스냅샷을 얻는 것이 아니라 비행경로의 횡방향 1라인 데이터만을 취득하므로, 횡방향 공간해상도와 종방향 공간해상도가 일치하지 않는다.

✢ 초분광 카메라의 횡방향 공간 해상도는 기준면으로부터 비행 고도, 초점 거리, 인접 광전 소자사이 이격거리를

기준으로 계산하며, 종방향 공간해상도는 수평 속도와 초당 촬영 수에 따라 결정된다.

(4) 비행노선은 도로 갓길의 상공을 비행하여 추락시에도 주변 피해를 최소화 할 수 있도록 안전을 확보하여야 한다.

(5) 비행노선을 설정하기 전 비행고도에 따른 카메라의 촬영면적을 확인하며, 도로의 폭과 곡선반경 등을 고려하여 경로를 결정한다.

(6) 비행노선은 촬영하는 도로보다 30% 이상 추가 유효폭을 확보할 수 있어야하며 다중경로 촬영시 20~30%이상 중첩되어야 한다.

(7) 비행노선은 직선으로 계획하여 곡선주행시에 발생하는 오차를 최소화하고, 곡선으로 주행하는 구간은 분석에서 제외하는 것이 좋다.

(8) 촬영시각은 태양 반사광이 카메라 센서(CCD)에 들어와서 빛 번짐 현상이 발생하지 않도록 태양고도가 45°~65°가 되는 오전 9시 ~ 오전 11시, 오후 2시 ~ 오후 4시가 유리하다.

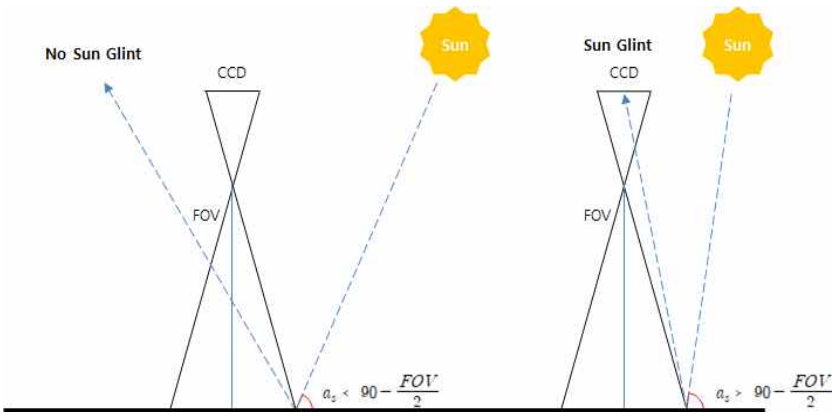


그림 3.3 태양 고도각에 따른 Sun Glint 발생 유무

(9) 촬영 계획이 수립되면 다음 사항을 포함한 '촬영 및 비행계획서'를 작성한다.

① 비행개요: 목적, 일시, 장소

② 비행계획: 비행고도, 비행속도, 소요시간

③ 장비제원: 기종, 명칭, 촬영장비

④ 조정사 정보: 성명, 소속, 연락처

⑤ 안전대책

⑥ 대상지: 지역명, 주소, 이착륙장소, 노선길이

⑦ 담당자

⑧ 촬영일정

⑨ 기타: 대상지 지도 등

(10) 비행 전 촬영계획에 따라 비행시뮬레이션을 실시하여 대상지의 비행고도, 해상도, 중복도, 비행시간 등에 대한 비행조건을 산출하여 비행 안전성을 높이는 것이 좋다.

3.4 항공촬영허가 및 비행승인신청

(1) 드론 촬영 전 항공사진 촬영 허가신청 및 비행승인신청을 하여야 한다.

✢ 항공사진 촬영허가와 비행승인은 드론 원스톱 민원 포털 서비스 (https://www.onestop.go.kr/drone)를 통해 신청한다.

(2) 항공사진 촬영은 우리나라의 전 지역이 대상지역으로써 촬영 4일 전 국방부의 허가를 받아야 한다.

✢ 항공사진 촬영 허가 신청시 기간, 촬영대상, 용도, 주소, 좌표, 반경/고도, 항속 등을 입력한다.

(3) 비행승인은 공역에 따라 지방항공청, 군공항 등 관련승인 기관에서 처리한다.

✢ 비행 공역은 드론 원스톱 민원 포털 서비스의 비행 계획/비행가능 지역 검색 기능으로 확인할 수 있다.

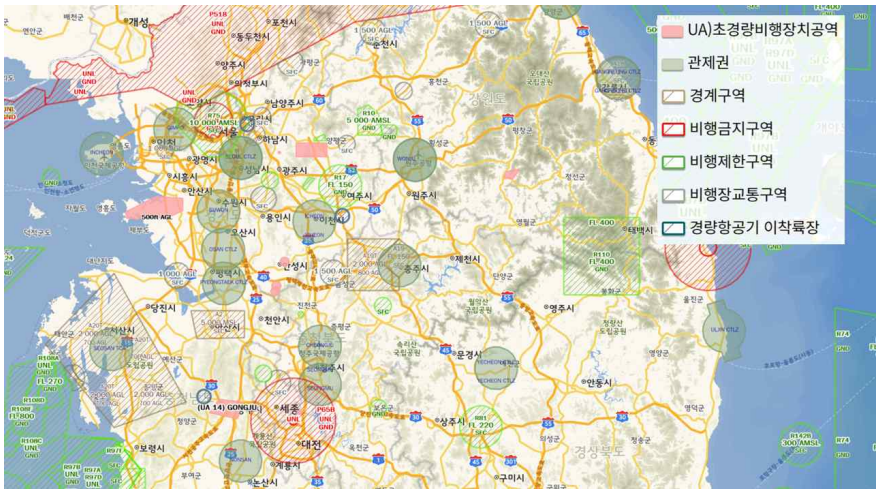


그림 3.4 비행 가능지역 및 공역 (예)

(4) 공항과 국가 중요 시설인 관제권과 비행금지구역의 비행을 위해서는 촬영계획서 및 관련 기관 협조공문 등을 첨부하여 신청하고, 사전승인을 받아야 한다. 다만, 승인을 받았어도 관제권은 고도 150m 이상으로 비행할 수 없다.

4. 드론 운용

4.1 기본 사항

(1) 조종자는 비행장치로 인하여 인명이나 재산에 피해가 발생하지 않도록 하여야 하며, 다음의 사항을 하지 않아야 한다.

① 인명이나 재산에 위험을 초래할 우려가 있는 낙하물을 투하하는 행위

② 인구가 밀집된 지역이나 그 밖에 사람이 많이 모인 장소의 상공에서 인명 또는 재산에 위험을 초래할 우려가 있는 방법으로 비행하는 행위

③ 허가받지 않고 관제공역, 통제공역, 주의공역에서 비행하는 행위

④ 안개 등으로 인하여 지상목표물을 육안으로 식별할 수 없는 상태에서 비행하는 행위

⑤ 비행시정 및 구름으로부터의 거리기준을 위반하여 비행하는 행위

⑥ 일몰시부터 일출시까지의 야간에 비행하는 행위 (허가를 받아 비행하는 행위 제외)

⑦ 주류, 마약류, 환각물질 등의 영향으로 조종업무를 정상적으로 수행할 수 없는 상태에서 조종하는 행위 또는 비행 중 주류 등을 섭취하거나 사용하는 행위

⑧ 그 밖에 비정상적인 방법으로 비행하는 행위

(2) 무인비행장치 조종자는 해당 무인비행장치를 육안으로 확인할 수 있는 범위 내에서 조종하여야 한다. 다만, 허가를 받아 비행하는 경우는 제외한다.

(3) 초경량비행장치 조종자는 항공기 또는 경량항공기를 육안으로 식별하여 미리 피할 수 있도록 주의하여 비행하여야 한다.

(4) 초경량비행장치의 통행우선권은 다음과 같다.

① 조종자는 비행 중 다른 드론의 발견과 회피를 위해 외부 경계를 지속적으로 유지하여야 한다. 규칙에 따른 통행우선권을 준수하여야 하며 통행우선권을 가진 드론이 있을 때 그의 위, 아래 또는 앞을 근접하여 통과해서는 안 된다.

② 드론이 정면으로 서로 접근할 경우 각 조종자는 운항경로를 오른쪽으로 변경한다.

③ 추월되는 드론은 통행 우선권을 가지게 되고 추월하는 드론은 오른쪽으로 운항경로를 변경하여 추월하여야 한다.

④ 착륙 중이거나 착륙을 위한 최종 접근로 상에 있는 드론은 비행 중인 다른 드론 또는 지상에서 이동 중인 드론 보다 통행우선권을 가진다.

⑤ 동일한 종류의 드론이 거의 동일고도에서 집중되는 경우에는 우측 드론이 우선권을 가진다.

⑥ 집중되는 드론들의 종류가 상이하다면 동력을 이용하는 초경량비행장치 조종자는 모든 항공기, 경량항공기 및 동력을 이용하지 아니하는 초경량비행장치에 대하여 진로를 양보하여야 한다.

- 기구는 다른 종류의 드론들에 대하여 통행우선권을 가진다.
- 글라이더는 비행선, 동력비행장치 또는 회전익 드론에 대하여 통행 우선권을 가진다.
- 비행선(Airship)은 동력비행장치 또는 회전익 드론에 대하여 우선권을 가진다.

⑦ 2대 이상의 드론이 착륙을 위하여 비행장으로 접근중일 때에는 더 낮은 고도에 있는 드론이 통행 우선권을 갖는다.

4.2 드론 비행 전 준비 사항

(1) 드론 비행 전 드론 기체 및 초분광 카메라의 오작동과 사고를 방지하고, 높은 품질의 측정을 위해 드론 기체 점검 및 확인을 하고, 초분광 카메라의 현장 보정을 하여야 한다.

(2) 드론 기체점검은 모터, 프로펠러, GPS, 송수신기, 배터리 등에 대하여 다음과 같이 점검한다.

① 모터 주변에 이물질이 있는지 육안으로 체크하고, 회전시켜서 부드럽게 회전하는지 확인한다. 압축 공기를 사용하여 이물질을 제거시 모터내부에 손상이 생길 수 있으므로 주의하여야 한다.

② 프로펠러의 휨, 크랙 여부를 육안으로 확인하고, 조임부를 확인한다. 결합이 약할 경우 진동이 발생할 수 있다. 여분의 프로펠러를 보관하고 있어야 하며, 프로펠러에 크랙이 확인될 경우 교체한다.

③ 송수신기 안테나의 단자는 수신감도에 영향이 크므로 파손 여부를 점검하여야 한다.

④ GPS의 신호가 7개 이상 수신되는지 확인하여야 하며, 위성의 수가 7개 이상이어도 위성 데이터가 Fixed 되지 않아 깜박일 경우 비행하지 않는다.

⑤ 드론 기체, 짐벌 등의 모든 배터리는 완충된 상태에서 사용하여야 하며, 충전상태를 확인한다. 배터리를 육안으로 확인하여 손상이 있을 경우 비행에 사용하지 말고 폐기하여야 한다.

⑥ 드론 기체를 처음 운영하는 지역이거나 오작동이 발생시 드론 짐벌, IMU, 콤파스(Compass) 캘리브레이션을 평평한 지면에서 수행한다.

⑦ 자동으로 캘리브레이션을 실시하는 기체는 수평을 유지하며, 뒤집거나 이동시키지 않는다.

⑧ 드론에 탑재된 GPS 등은 자기(자석 및 전자파)간섭에 민감함으로 모든 전자 장치로부터 멀리 떨어져 있어야 한다. 캘리브레이션 중에는 전자장비(휴대폰, PDA, 스마트기기 등)를 소지하지 않아야 한다.

(2) 초분광 카메라는 일반 RGB 카메라 보다 보정작업에 따라 촬영되는 결과의 품질이 영향을 크게 받으므로 다음 항목에 대하여 현장에서 보정한다.

표 4.1 초분광 카메라 현장 캘리브레이션 항목

항목	점검사항	비고
하이퍼 전용 GPS 캘리브레이션	전용 SW(Spectral Ⅲ)를 이용한 GPS 연결상태 확인	Spectral Ⅲ 이용
하이퍼 전용 IMU 캘리브레이션	수평축, 수직축 캘리브레이션 실시	
Black 밸런스 점검	카메라 렌즈 덮개를 막고 Black 밸런스 점검	
White 밸런스 점검	카메라 렌즈를 풀고, 반사도 95% 순백지를 이용하여 점검 실시	

(3) 촬영 범위의 평지에 반사율이 11%, 32%, 56% 등인 표준 타프(Tarf)를 설치하여 드론으로 촬영시 함께 촬영되도록 한다.

4.3 드론 비행 및 촬영

(1) 드론 운용 중 눈이나 비가 내릴 경우에는 드론의 손상 및 오작동이 발생할 수 있으므로 드론 비행을 중단하여야 하며, 이를 위해 비상착륙하거나 복귀명령을 실행하여야 한다.

(2) 드론의 한계 풍속 이상으로 강풍이 있을 경우에는 비행은 물론 이·착륙시 안전에 문제가 발생할 수 있으므로 비행하지 않아야 한다. 일반적인 한계 풍속은 8~15m/s이다.

(3) 회전익 드론의 수직 이·착륙은 좁은 공간에서도 가능하지만, 이·착륙시 드론 손상 및 안전사고가 많이 발생하므로 주변통제를 철저히 하여야 하며, 갑작스런 강풍 발생 여부를 확인하여야 한다.

(4) 드론 비행 중 조종자는 드론이 계획된 경로비행을 안전하게 수행하고 있는지 주시하여야 한다. 일반적으로 조종자는 드론을 조정하고, 부조종자는 드론을 육안으로 감시하며 비행상태, 비행자세, 위험요소 등에 대비하여야 한다.

(5) 조종자는 비행 중 드론의 위치, GPS 신호 수신상태, 자세정보, 배터리 잔량 등의 상태를 항상 주의 깊게 모니터링 해야 한다. 특히, 배터리는 전압이 낮아질수록 빠르게 방전되므로 이를 고려하여 안전하게 착륙할 수 있는 배터리 잔량을 확인하여야 한다.

(6) 조종자는 비행 중 드론의 이상을 인지할 경우 즉시 가장 가까운 이·착륙장에 안전하게 착륙하여야 한다. 특히 GPS의 수신 개수가 7개 미만이면 드론의 위치 및 고도를 유지할 수 없게 되므로 수동조종 및 자동명령으로 GPS 신호 수신이 양호한 지역으로 드론을 이동하여야 한다.

(7) 부조종자는 드론을 육안으로 감시하며 항공기나 타 드론, 맹금류의 접근이 발견되거나 비행이상이 발견되면 즉시 조종자에게 보고하여야 한다.



그림 4.1 드론 탑재형 초분광시스템 촬영 (예)

(8) 드론의 비행 후 비행 전과 동일하게 드론 기체 점검을 하여야 한다.

(9) 촬영된 초분광 원시데이터는 HyperspecⅢ SW를 활용하여 카메라 자체 스토리지에 있는 영상을 유선네트워크를 연결하여 영상처리가 가능한 노트북 등으로 데이터를 복사한다.

(10) 촬영 데이터의 품질을 육안으로 검토하여 촬영계획과 동일하게 데이터가 취득되었는지를 확인한다.

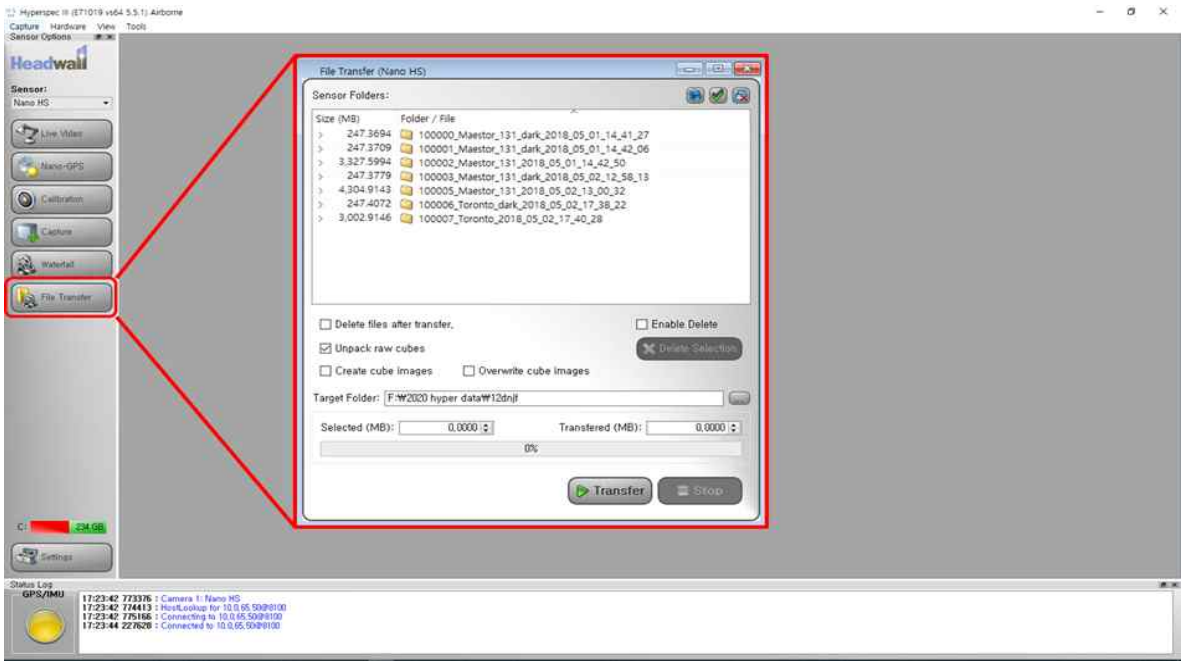


그림 4.2 초분광 원시데이터 취득

5. 드론 초분광 영상 분석 및 가시화

5.1 전처리

(1) 초분광 영상 원자료는 DN(Digital Number)로 되어있으며, 이 값을 방사보정, 기하보정 등의 전처리 과정에서 보정하여 반사율(Reflectance)로 변환한다.

- ✢ 방사보정은 태양의 고도각, 탐측기의 응답 특성, 대기의 상태, 지형적 특성에 의한 영상의 밝기값 변화로 나타나는 왜곡을 보정하기 위해 실시하는 과정이다.
- ✢ 대기보정은 태양 빛이 지표면 물체에 반사된 후 촬영시 대기의 산란, 흡수, 반사, 투과 등의 영향으로 자료의 밝기에 영향을 주어 발생하는 왜곡을 보정하는 것이다.
- ✢ 기하보정은 초분광 데이터를 취득 할 때 지구와 드론 기체 및 센서의 특징으로 인한 기기적 한계로 인해 발생하는 초분광데이터 영상의 공간적 왜곡(기하오차)를 보정하는 것이다. 위성 영상의 픽셀좌표와 지상의 지리좌표를 대응하여 초분광 데이터영상을 실제 위치로 보정한다.



그림 5.1 초분광영상 전처리 과정

(2) 방사보정을 위해 Headwall사의 Spectral View를 이용한다. 현장에서 카메라 센서 보정시 수행한 Black 밸런스와 White 밸런스로 얻어진 Dark Reference와 White Reference 결과를 적용하여 방사보정된 Radiance 값으로 변환한다.

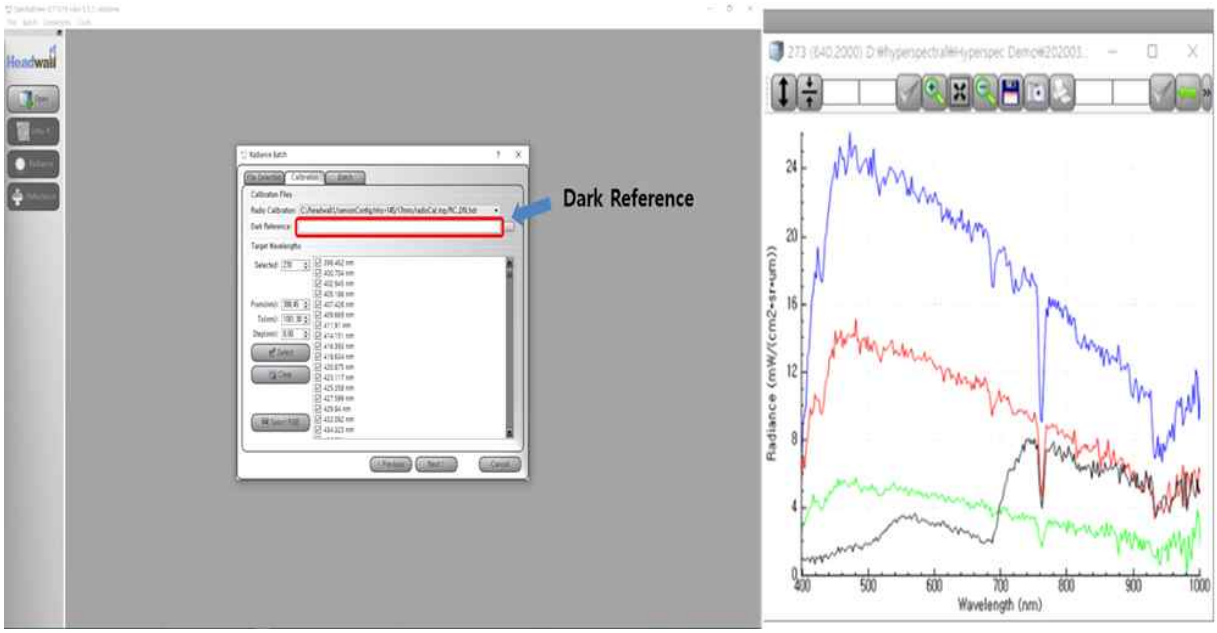


그림 5.2 방사보정을 통한 Radiance값 취득

(3) 대기보정은 현장에 설치된 타프의 반사율 기준값을 기준으로 다음 식으로 보정한다.

$$\text{대기보정} = \frac{(Raw\,data - Dark\,Reference)}{(Tarf_{(\text{반사율}\,11,32,56\%)}\,Reference - Dark\,Reference)}$$

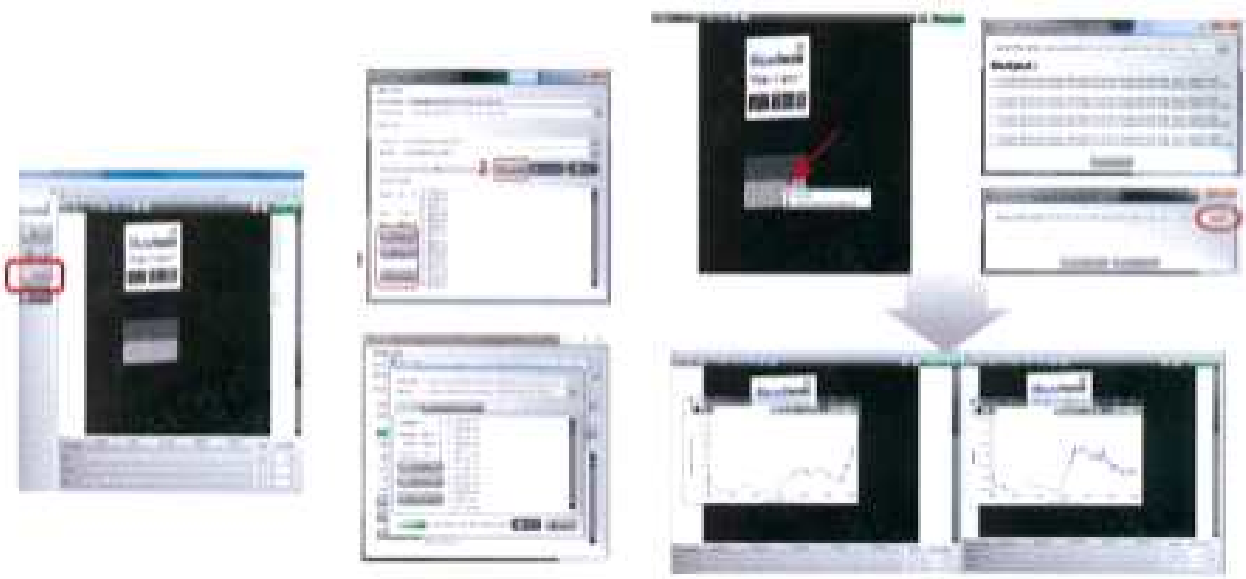


그림 5.3 초분광 영상 대기보정 과정

(4) 기하보정은 고정밀의 DEM을 적용하여 기복변위 및 기복으로 인한 반사광 변화를 고려한 보정한다. DEM의 정밀도가 높을수록 초분광영상의 품질에 높으므로 DEM은 5m 해상도를 표준으로 한다. 기하보정시 GPS/IMU 관련 매개변수에 대상지역에 맞는 값을 입력하여 영상 품질을 향상시킬 수 있다.



그림 5.4 초분광 영상 기하보정

5.2 포장상태 분석

(1) QGIS를 이용하여 전처리된 초분광 영상자료를 텍스트(CSV) 형식의 자료와 데이터 속성 정보 파일(hdr)로 변환한다.

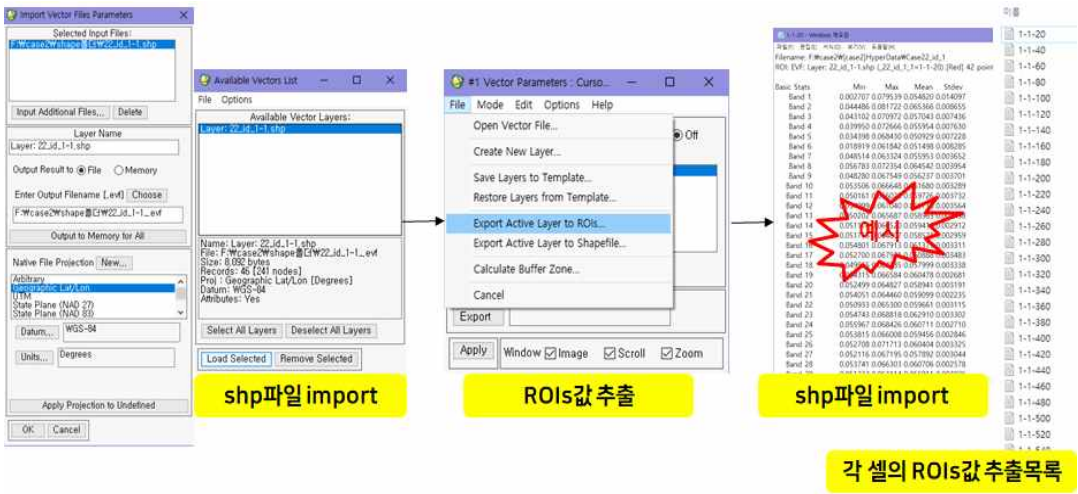


그림 5.5 초분광 자료의 CSV로 변환 과정

(2) 초분광 포장상태분석 프로그램(HSPCA 1.0)을 이용하여 다음과 같은 방법으로 포장상태를 분석한다.

① '초분광 원자료 열기' 버튼을 클릭하여 초분광 영상(CSV) 파일과 데이터속성(hdr) 파일을 업로드한다.



그림 5.6 HSPCA의 분석용 버튼

② 창 오른 쪽의 불러온 파일 목록에 파일 정보가 표시되는지 확인한다.



그림 5.7 HSPCA의 초분광 원자료 정보 표시

③ ‘이미지 생성’ 버튼을 클릭하여 시각화된 이미지를 생성한다.

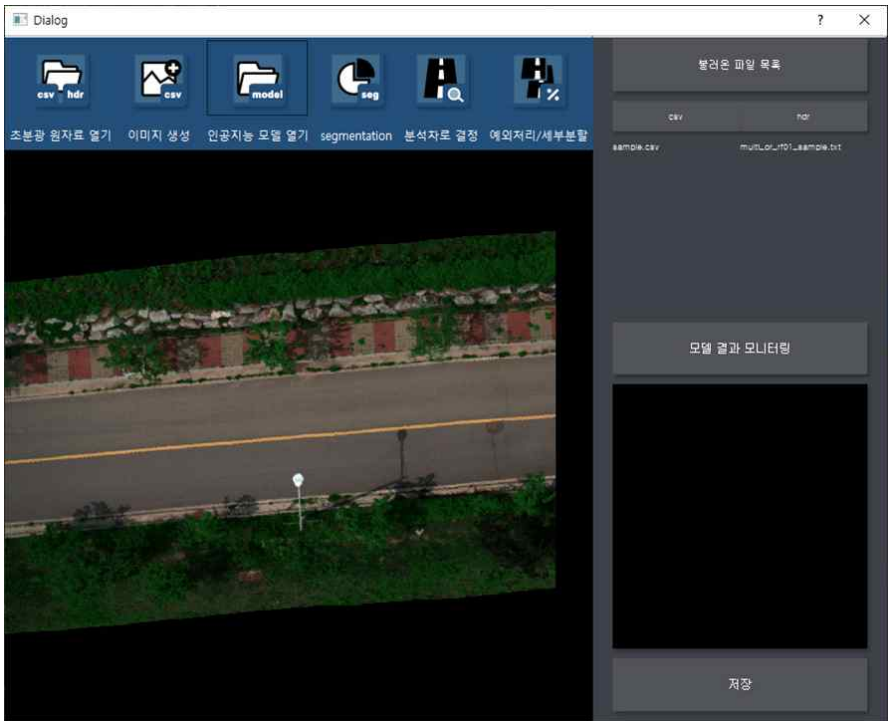
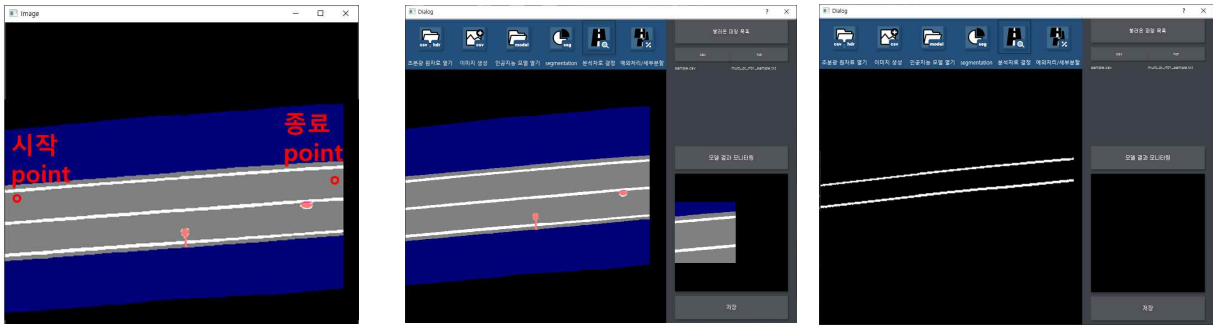


그림 5.8 HSPCA의 초분광 원자료를 이용한 시각화된 이미지 표시

④ ‘인공지능 모델 열기’ 버튼을 클릭하여 이미지분석 모델 파일을 업로드한다.

⑤ ‘segmentation’ 버튼을 클릭하여 인공지능 모델을 이용하여 차선을 분할을 시작한다.

⑥ ‘분석차로 결정’ 버튼을 클릭하여 분석된 이미지에서 분석할 차선의 두 지점을 선택하여 분석 차선을 검출한다.



(a) 분석 차선 내부 2점 선택 (b) 차선 검출 인공지능 모델 (c) 검출된 차선 결과 이미지

그림 5.9 HSPCA의 인공지능 모델을 이용한 차선 검출

⑦ '예외처리/세부분할' 버튼을 클릭하여 검출된 차선 영역에서 20cm 단위로 중앙부/휠패스부로 영역 분할 진행한다. 또한, 노면표시, 횡단보도, 차량 등 데이터 품질 저하 변수 요소 영역을 제외한다.

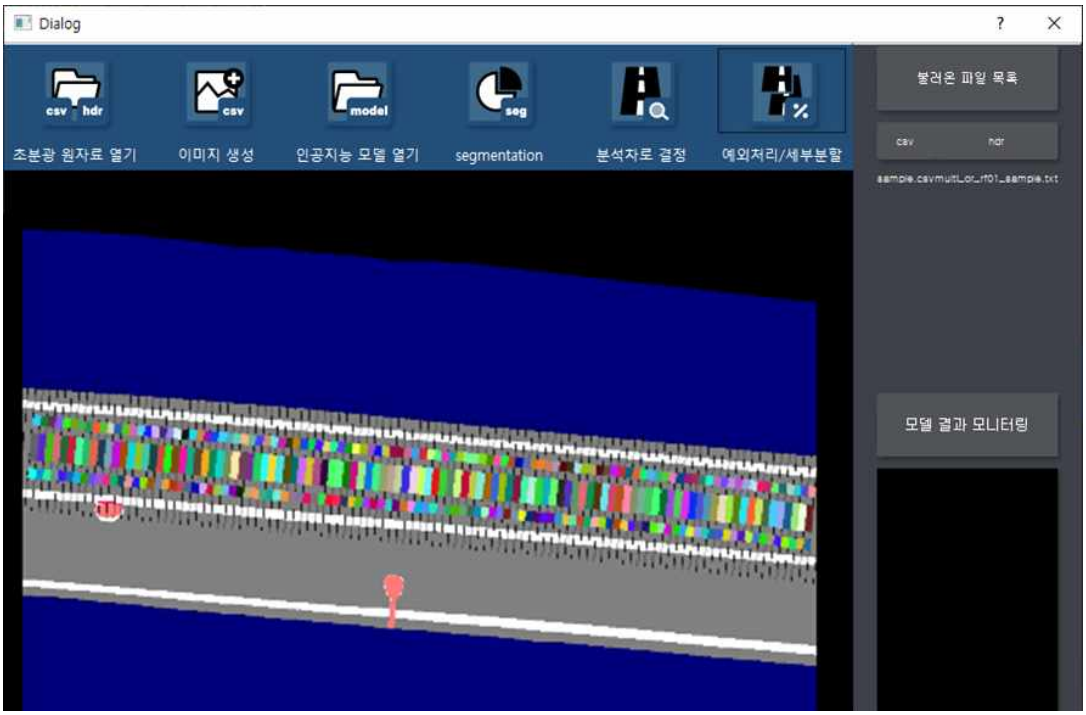


그림 5.10 HSPCA의 검출 차선의 분할

⑧ '저장' 버튼을 클릭하여 분할된 영역으로 산출한 소성변형, 균열율, 평탄성 등을 데이터 파일(CSV) 과 데이터 속성 파일(json)로 저장한다.



	A	B	C	D	
1		0	1	2	
2	prediction_crack	0.068574	0.446197	-0.00378	0.
3	prediction_iri	0.131422	0.224215	0.102562	0.
4	prediction_pci	0.487694	0.525665	0.686994	0.
5	prediction_rutting	1.117223	1.018642	0.512186	0.

{"meta": "ENVI₩ndescription = {HEADWALL Hyperspec III[HEADWA
= nm₩nmap info = {Geographic Lat/Lon,1,1,127.3667560289648,3
(₩n415.119₩n,417.361₩n,419.602₩n,421.844₩n,424.086₩n,426.32
96₩n,572.037₩n,574.279₩n,576.521₩n,592.212₩n,594.454₩n,596.
922₩n,740.164₩n,742.405₩n,744.647₩n,746.889₩n,749.131₩n,751
6.049₩n,939.674₩n,941.916₩n,946.399₩n,966.574₩n,971.058₩n,9
= 0₩n;Ortho DEM File = C:₩₩headwall1₩₩dem₩₩case12_dem.tif
(ms) = 0₩n;Ortho GPS Offset Yaw = 0₩n;Ortho Lems EFL (mm) =
Running Avg. of Raw = 0₩n;Ortho Running Avg. of Roll = 0₩n;Or
Number = nHS-145₩n;Units = mW/(cm2*sr*um)₩n;averagePixelDi
86], [10109, 359], [10036, 322]], "prediction_crack": 11.709249019
0.6011898070573807, "prediction_pci": 0.458267405629158, "pred

(a) json, csv 파일 (b) 소성변형, 균열율, 평탄성 등 데이터 파일 (c) 데이터 속성 파일

그림 5.11 HSPCA의 최종 결과물 데이터 파일 형식

5.3 드론 포장상태 분석 결과 가시화

(1) 드론 포장상태 분석 결과를 시스템에 저장 방법은 다음과 같다.

① 가시화 시스템에 로그인한다. 등록된 사용자만 시스템에 접속할 수 있으며, 계정이 없을 경우 신규생성을 요청하여야 한다.



그림 5.12 시스템 로그인

② 로그인시 PCI 데이터 조회 페이지가 표시된다.

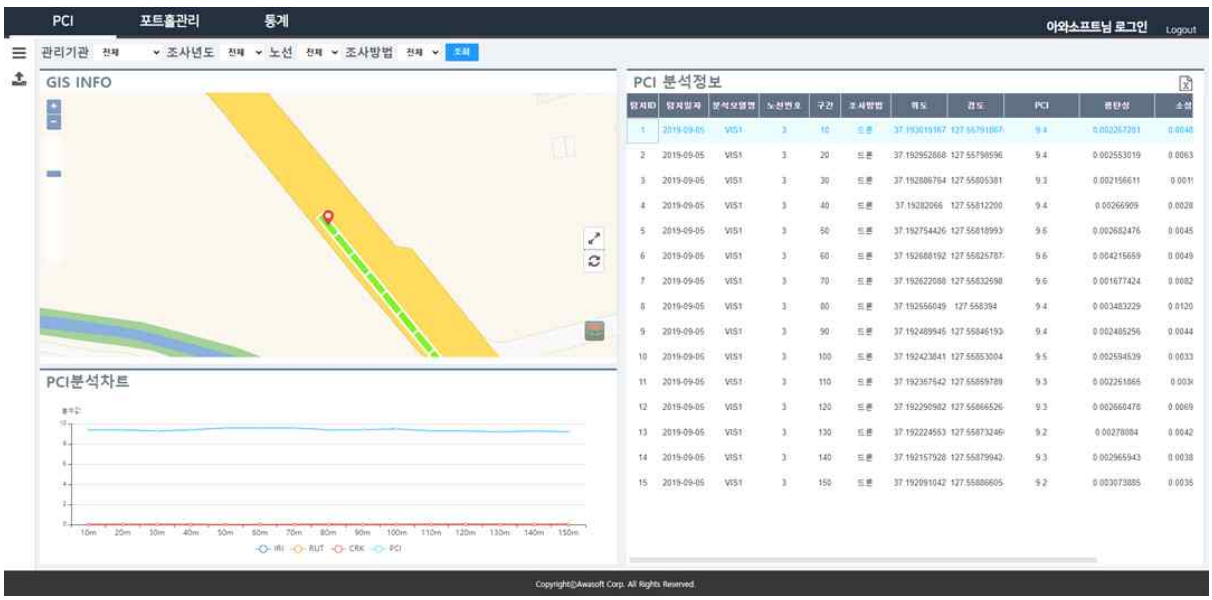


그림 5.13 PCI 데이터 조회

③ 화면의 버튼을 눌러 'PCI데이터 업로드'를 위한 팝업을 표시한다.



그림 5.14 PCI 데이터 업로드 팝업

④ '업로드 파일'의 '파일선택' 버튼을 눌러 PCI분석 데이터 파일을 '업로드 파일'로 선택한다.

⑤ 시스템에 등록된 관리기관을 선택한다.

⑥ 드론 조사일시(포장상태 탐지일자)를 입력한다.

⑦ 조사구간에 대한 노선번호를 입력한다.

⑧ '등록' 버튼을 클릭하여 데이터를 업로드한다.

(2) 분석한 도로의 GPS 좌표가 UI의 GIS 좌표와 오차가 있을 경우 시스템에서 다음과 같은 방법으로 보정한다.

① GIS 지도에서 구간을 선택 후 지도상에서 버튼을 클릭한다.

② 블록으로 표시된 이동할 데이터 시작 위치를 클릭하면 노드가 생성되며, 마우스를 이동하여 지도의 정확한 위치로 조정한 후 마우스를 더블클릭하여 이동을 종료한다. 마우스 커서를 이동하면 구간 이동에 대한 거리가 측정되어 표시된다.



그림 5.15 분석 자료 좌표 오차 보정 (1)

③ “구간 : X에 대해 좌표를 수정하시겠습니까?” 문구가 출력되면 ‘확인’ 버튼을 클릭하여 오차 구간 보정작업을 수행한다.



그림 5.16 분석 자료 좌표 오차 보정 (2)

④ 보정이 완료되면 GIS 지도 정보가 갱신되어 보정된 구간으로 화면에 표시된다.



그림 5.17 분석 자료 좌표 오차 보정 결과

(3) 시스템에 등록된 드론 포장상태 분석 결과를 조회하는 방법은 다음과 같다.

① 인증된 사용자로 가시화 시스템에 로그인한다.

② 관리기관, 조사년도, 노선, 조사방법 등의 조회조건을 선택한 후 ‘조회’ 버튼을 클릭하여 원하는 포장상태 조사결과를 조회한다.

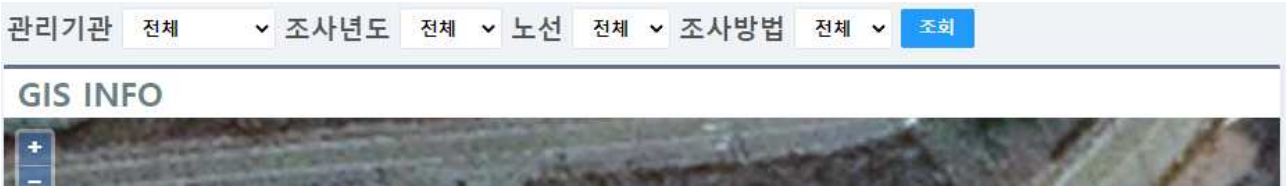


그림 5.18 PCI 데이터 조회

③ 검색조건으로 조회된 PCI 분석정보는 UI 우측에 리스트 형태로 다음 정보가 표시된다.

- 탐지 ID : 포장상태 탐지 10m구간별 고유 식별값
- 탐지일자 : 드론 탐지일자
- 분석모델명 : 영상분석 모델명
- 노선번호 : 분석구간의 노선번호
- 구간 : 10m단위 구간
- 조사방법 : 포장상태 탐지방법 (드론, 차량 등)
- 위도 : 구간 GPS 위도
- 경도 : 구간 GPS 경도
- PCI : 산출된 PCI수치
- 평탄성 : 분석된 평탄성수치
- 소성변형 : 분석된 소성변형 수치
- 균열 : 분석된 균열수치

④ PCI 분석 정보는 UI 좌측 GIS 지도에 10m 구간별로 표시된다. 그리고 GIS 지도의 우측하단 설정을 변경시 배경지도를 수치지도, 항공사진으로 선택할 수 있다.

⑤ PCI 산출 결과는 UI 좌측의 ≡ 버튼을 클릭하여 서울시(SPI), 일반국도(NHPCI) 등의 방법을 선택하여 변경할 수 있다. 표시되는 분석결과는 PCI산출 공식의 임계값을 지정하여 적용할 수 있다.

- 일반국도 : 전체, 양호, 보수필요
- 서울시 : 전체, 매우 양호, 양호, 보통, 불량, 매우 불량

⑥ 검색 리스트는 Excel 파일로 출력하여 저장할 수 있다.

비매품/무료
95530
9 791187 770299
ISBN 979-11-87770-29-9 (PDF)

주 의

1. 이 보고서는 국토교통부에서 시행한 도로기술연구개발사업 ‘드론을 활용한 비탈면 및 도로포장 관리 효율화 기술 개발’ 연구개발과제의 최종보고서이다.

2. 이 연구개발내용을 대외적으로 발표할 때에는 반드시 국토교통부(국토교통과학기술진흥원)에서 시행한 도로기술연구개발사업의 결과임을 밝혀야 한다.

3. 국가과학기술 기밀 유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 안 된다.