

1. 표지

(뒷면) (옆면)

(앞면)

보안 과제(), 일반 과제(○) / 공개(○), 비공개()

국토교통기술촉진연구 사업 제2차 연도 최종보고서

3cm

0.5cm

R&D /
18CTAP-C132982-02

2.5cm

9cm

(견고덱31p)

5cm

기반시설 및 플랜트의 증설· 복구 역설계 계획용 라이다 장착 무인비행체 기반 실시간 가상 리트로피팅 시스템 개발 최종보고서

(0.1 cm)

2019. 02. 14.

0.15cm

(견고덱15p)

주관연구기관 / 중앙대학교 산학협력단
협동연구기관 / (주)제이텍솔루션

(견고덱 15.5p)

국토교통부

(전문기관) 국토교통과학기술진흥원

(견고덱 20p)

3 cm

18CTAP-
C132982-
02

4 cm

기반시설
및 플랜트의
증설·복구
역설계
계통
라이다
무인비행체
기반 실
리트로피
시스템 개발

최종보고서

(견고덱
14p)

2018
(견고덱13p)

국토교통과학기술진흥원
국토교통부

Development of the IIAV embedded IIAAR based realtime virtual retrofitting system for railway engineering planning of the infrastructure and plant extension and restoration R&D Report

(견고덱
25p)

(견고덱
17p)

2. 제출문

제 출 문

국토교통과학기술진흥원 귀하

‘기반시설 및 플랜트의 증설·복구 역설계 계획용 라이다장착 무인비행체 기반 실시간 가상 리트로피팅 시스템 개발’(연구개발 기간 : 2017. 06. 30. ~ 2018. 12. 31.) 과제의 최종보고서 12부를 제출합니다.

2019. 02. 14.

주관연구기관명 : 중앙대학교 산학협력단 (대표자) 김 원용 (인)
협동연구기관명 : (주)제이텍솔루션 (대표자) 김 희석 (인)
참여기관명 : (주)제이텍솔루션 (대표자) 김 희석 (인)

주관연구기관책임자: 채 영호
협동연구기관책임자: 김 희석
참여기관책임자: 김 희석

국토교통부소관 연구개발사업 운영규정 제38조에 따라 최종보고서 열람에 동의합니다.

2. 제출문

제 출 문

국토교통과학기술진흥원 귀하

‘기반시설 및 플랜트의 증설·복구 역설계 계획용 라이다장착 무인비행체 기반 실시간 가상 리트로피팅 시스템 개발’(연구개발 기간 : 2017. 06. 30. ~ 2018. 12. 31.) 과제의 최종보고서 12부를 제출합니다.

2019. 02. 14.

주관연구기관명 : 중앙대학교 산학협력단 (대표자) 김 원용 (인)

협동연구기관명 : (주)제이텍솔루션 (대표자) 김 희석 (인)

참여기관명 : (주)제이텍솔루션 (대표자) 김 희석 (인)

주관연구기관책임자: 채 영호

협동연구기관책임자: 김 희석

참여기관책임자: 김 희석

국토교통부소관 연구개발사업 운영규정 제38조에 따라 최종보고서 열람에 동의합니다.

3. 보고서 요약서

보고서 요약서

과제 고유 번호	18CTAP-C132982-02	해당 단계 연구 기간	2017. 06. 30 - 2018. 12. 31 (18개월)		단계구분	(1)/(1)
연구사업명	중사업명					
	세부사업명	국토교통기술촉진연구사업				
연구과제명	대과제명					
	세부과제명	기반시설 및 플랜트의 증설·복구 역설계 계획용 라이다장착 무인비행체 기반 실시간 가상 리트로피팅 시스템 개발				
연구책임자	채 영호	해당단계 참여 연구원 수	총: 13 명 내부: 13 명 외부: 0 명	해당단계 연구개발비	정부: 370,000천원 기업: 130,000천원 정부 외: 천원 계: 500,000천원	
		총 연구기간 참여 연구원 수	총: 13 명 내부: 13 명 외부: 0 명	총 연구개 발비	정부: 370,000천원 기업: 130,000천원 정부 외: 천원 계: 500,000천원	
연구기관명 및 소속 부서명	중앙대학교 산학협력단, 첨단영상대학원				참여기업명 (주) 제이텍솔루션	
국제공동연구	상대국명:				상대국 연구기관명:	
위탁연구	연구기관명:				연구책임자:	
(1) UAV-LiDAR-HMD융합 시스템 개발					보고서 면수: 87쪽	
▪ 융합시스템에 사용된 Pupil Lab. Eye- Tracker를 이용하여 전정안반사 알고리즘을 설계하고 응시점 좌표계산						
▪ 융합시스템에 사용된 Gimbal Camera의 가상좌표계 매핑을 위한 Onboard SDK 제어						
▪ 전정안반사 원리를 포함하는 Eye Simulation을 통해 UAV-HMD 융합시스템의 시뮬레이션						
▪ VoR을 포함한 응시점 좌표를 Gimbal 카메라를 위한 가상좌표계로 매핑하여 Onboard SDK를 통한 제어기를 통해 전시적인 UAV의 Airborne FPV로 구현						
(2) 방향에 따른 형상기술자 활용 및 UAV 스캔 위치 추정기술 개발						
▪ 사전 처리된 3D 포인트 클라우드 데이터를 랜덤 샘플링하고 공통 전역 좌표계로 매핑하여 실시간 스캔되는 부분 3D 포인트 클라우드와 중첩/정합하는 알고리즘 개발						
▪ 3D 포인트 클라우드의 파이프 곡률 추정과 실시간 스캔되는 부분 3D 포인트 클라우드와 중첩/정합으로 3D PCD 파이프의 곡률 정합						
▪ 중첩/정합된 3D 포인트 클라우드 데이터와 배경이 되는 비디오 영상을 정합하여 사실적인 가상 리트로피팅이 될 수 있는 배경을 시각화						
(3) 라이다장착 UAV 기반 실시간 가상 리트로피팅						
▪ UAV-LiDAR와 호스트 스테이션간의 데이터 통신						
▪ 가상 리트로피팅을 위한 시각시스템						
▪ 라이다장착 UAV 기반 KICT 가상 리트로피팅						

4. 국문 요약문

연구의 목적 및 내용	- 리트로피팅 설계계획을 UAV-HMD 융합 시스템으로 Airborne FPV를 구현하여 현장을 보면서, As-built BIM 모델과 증강된 리트로피팅 파트를 레이저 스캐닝 정합을 통해 동시 참조하면서 진행할 수 있는 시스템 개발. - UAV의 자율주행을 보조, 제한하는 인터랙션과 가상 리트로피팅 기반 설계 시뮬레이션을 주도, 참조하는 인터페이스를 통합하여 UAV의 제어, 데이터 취득 및 가상 리트로피팅을 동시에 실현하는 S/W의 개발. 1. UAV-LiDAR-HMD융합 시스템 개발 - UAV-LiDAR 융합시스템의 개발 및 수리 모델 구축 - Airborne FPV를 위한 UAV-HMD 융합 시스템 개발 - UAV-LiDAR 융합시스템의 반자율비행을 위한 통합 제어 SW 개발 2. 형상기술자 활용 및 UAV 스캔 위치 추정기술 개발 - 기초 요소 식별을 위한 방향성 형상 기술자 정의 및 구현 - 방향에 따른 형상기술자 활용 및 UAV 스캔 위치 추정기술 개발 - 실시간 정합 및 요소 식별의 실제 적용을 통한 검증 3. 기반시설, 플랜트의 증설·복구 계획용 가상 리트로피팅 시스템 개발 - AR기반 리트로피팅 효율화 모형 및 인터페이스 구축 - 식별된 요소 기반 비마커 AR을 이용한 가상 리트로피팅 시뮬레이션 - 라이다장착 UAV 기반 실시간 가상 리트로피팅				
연구개발성과	- 실제 의도한 목적을 달성하는 시험을 포함하는 라이다장착 UAV 기반 실시간 가상 리트로피팅 시스템. - IT융합에 의해 기반시설, 플랜트 리트로피팅 비용의 절감, 시설물 증설, 복구 프로젝트 공정기간의 단축, 운영시의 보수 유지비용 절감효과 기대.				
연구개발성과의 활용계획 (기대효과)	- VR/AR을 통한 작업 중에 해당 유지보수나 트레이닝에 필요한, 적절한 자료, 지식을 사용자 앞에 다양한 형태로 보여줄 수 있게 하여 수준 높은 차원의 작업을 진행할 수 있게 해주며, 진행 내용/세션을 저장하면 추 후 재현이나 평가 및 재교육에 효율적으로 사용이 가능. - VR/AR을 이용한 경험 트레이닝은 실시간 피드백과 적극적이고 직접적인 참여유도, 효과적인 기술전수 및 습득기술의 손쉬운 유지로 빠른 학습을 달성할 수 있으며, 위험상황의 사전 반복노출로 교육자는 물론 장비의 안전에도 크게 기여.				
핵심어 (5개 이내)	리트로피팅	무인비행로봇	포인트 클라우드	역설계	가상·증강현실

〈 SUMMARY 〉

Purpose & Contents	▪ Developing a system that enables simultaneous reference of the as-built BIM model and the enhanced retro fitting part by laser scan matching while implementing the airborne FPV with the retrofitting design plan as a UAV-HMD fusion system. ▪ Development of S/W that realizes UAV control, data acquisition and virtual retro fitting simultaneously by integrating interfaces that leads/simulates virtual retrofitting based design simulation and supports/restricts autonomous navigation of UAV at the same time. 1. Development of UAV-LiDAR-HMD fusion system ▪ Development of UAV-LiDAR fusion system with the construction of mathematical system model ▪ Development of the combined UAV-HMD system for airborne FPV ▪ Development of the integrated control S/W for semi-autonomous flight of UAV-LiDAR fusion system 2. Utilization of the directional geometry descriptor and UAV scan position estimation ▪ Define and implement directional geometry descriptors for the basic elements identification ▪ UAV scan position estimation by utilizing the directional geometry descriptor ▪ Validation through the implementation of realtime registration and element identification 3. Development of virtual retrofitting system for expansion and restoration of infrastructure and plant ▪ Construction of AR-based retrofitting efficiency model and interfaces ▪ Simulation of the virtual retrofitting using the identified element based non-marker AR ▪ Development of the UAV embedded LiDAR based realtime virtual retrofitting system
Results	▪ UAV embedded LiDAR based real-time virtual retrofitting system including tests to achieve the intended purpose. ▪ IT convergence is expected to save infrastructure and plant retrofitting costs, reduction in facility expansion, restoration project period, and reduction in operation maintenance costs.
Expected Contribution	▪ Provides the ability to show the appropriate data and knowledge required for maintenance and training in various forms in front of the user while working through the VR/AR to carry out the required works at a higher level, and allows the user to save the contents/session to reuse it efficiently for reappearance, evaluation and retraining. ▪ Experienced training using VR/AR can achieve quick learning with real-time feedback, active and direct participation, effective skill transfer and easy maintenance of the acquired skills, and pre-repeated exposure of dangerous situations contributes to the safety of the equipment as well as the educator.
Keywords	Retrofitting UAV, Unmaned Aerial Vehicle Point Cloud Reverse Engineering Virtual-Augment ed Reality

< Contents >

1. Introduction	9
2. Status of domestic and overseas technology	12
3. Research performance and achievements	16
4. Achievement of goal and contribution to relevant field	76
5. Plan to utilize R&D achievements	79
6. Overseas science and technology information collected during the research process	82
7. Security rating of R & D achievement	83
8. Status of research facilities and equipment registered in the National Science and Technology Comprehensive Information System	83
9. Implementation of safety measures in laboratories due to R & D tasks ..	83
10. Representative research achievements of R & D tasks	85
11. Others	86
12. References	87

7. 본문 목차

< 목 차 >

1. 연구개발과제의 개요	9
2. 국내외 기술 개발 현황	12
3. 연구 수행 내용 및 성과	16
4. 목표 달성도 및 관련 분야 기여도	76
5. 연구개발성과의 활용 계획	79
6. 연구 과정에서 수집한 해외 과학기술 정보	82
7. 연구개발성과의 보안등급	83
8. 국가과학기술종합정보시스템에 등록한 연구시설·장비 현황	83
9. 연구개발과제 수행에 따른 연구실 등의 안전 조치 이행 실적	83
10. 연구개발과제의 대표적 연구 실적	85
11. 기타 사항	86
12. 참고 문헌	87

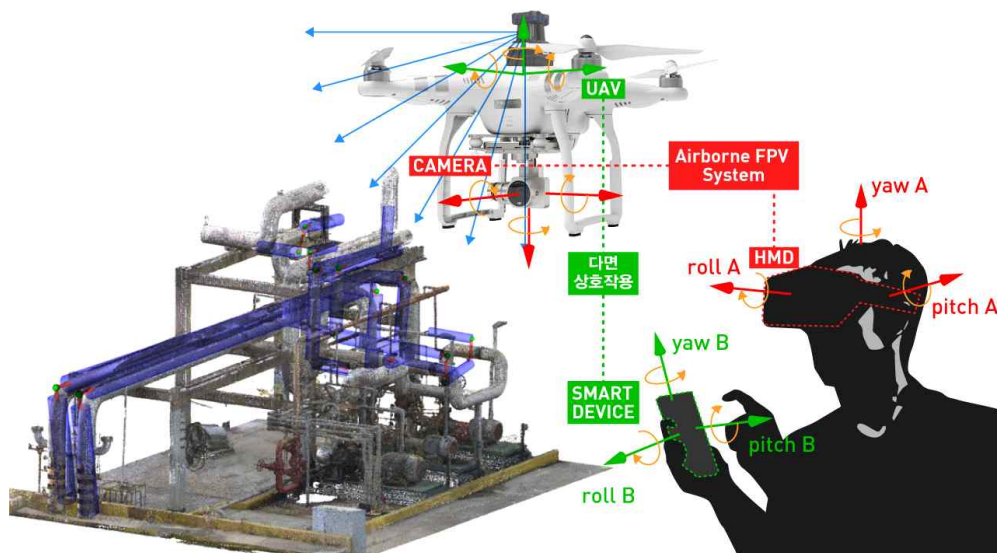
8. 본문

제1장. 연구개발과제의 개요

1. 연구개발 목적

라이다장착 무인비행체 기반 실시간 가상 리트로피팅 시스템 개발

- 리트로피팅을 포함하는 증설, 복구는 복잡하거나 다양한 가능성을 고려해야하는 경우, 특히 위험하거나 접근성이 낮은 지역의 설계계획은 시뮬레이션을 통해 안정성, 신뢰성을 보장하기 위해 가상 리트로피팅이 필요.
- 리트로피팅 설계계획을 UAV-HMD 융합 시스템으로 Airborne FPV를 구현하여 현장을 보면서, As-built BIM 모델과 증강된 리트로피팅 파트를 레이저 스캐닝 정합을 통해 동시 참조하면서 진행할 수 있는 시스템을 개발.



<라이다장착 무인 비행체 기반 실시간 가상 리트로피팅 시스템 개념도>

2. 연구개발의 필요성

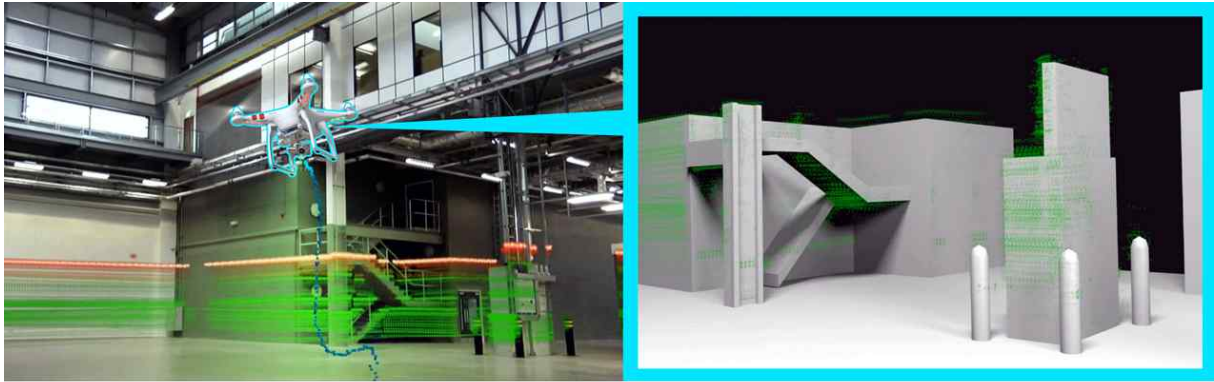
가. 증설·복구용 역설계(Reverse Engineering for the Addition, Restoration)

- 시설물의 유지관리, 리모델링, 재설계, 복원해야 하는 상황에 기존 2D 도면이나 촬영으로 얻은 포인트 클라우드를 이용한 역설계 필요성이 급격하게 커지고 있으나, 포인트 클라우드로부터 형상을 추출할 때 LiDAR 기기 측량으로 발생한 오차, 부분 누락 등이 존재하며, 수많은 수작업으로 이루어져 있어 새로운 인터랙션 환경에서 자동/반자동화하는 기술이 필요.
- 석유화학플랜트의 경우 기존 시스템의 효율화를 위하여 배관, 회전기계, 고정기계, 전기계장설비등의 일부를 교체하는 **Debottlenecking**, 노후 및 유헴 시스템의 재구성을 위한 **Revamping** 및 정기적인 TA(Turn Around), RBI(Risk based Inspection)를 위하여 As-built 모델의 디지털화가 필요.

나. UAV에서 취득되는 포인트 클라우드를 통한 오브젝트 인식 및 모델링

- UAV에서 취득된 포인트 클라우드를 정합(매핑)하여 통합하고 역설계 모델링의

미완성 데이터를 일반설계 모듈로 보완할 수 있게 하면 포인트 클라우드를 입력으로 하는 역설계 프로세스를 자동화할 수 있게 됨.



<UAV embedded LiDAR + Scanned Interior, Point Cloud Data + 3D Model>

- 공간상에 사전 설치된 측위센서 설비 없이도 UAV의 운용을 가능케 하는 자율비행기술이 필요하고 이는 GPS나 유사한 공간측위 센서에 의존하지 않는 위치 추정 기술, 탑재중량을 고려한 다중센서 항법 기술, 짧은 가시거리 환경에서의 영상기반 목표물 인식 기법, UAV 탑재 환경을 고려한 영상처리 알고리즘 실시간화 및 진동에 강인한 영상처리 기법 등을 포함.
- 국토교통부의 **제2차 공간정보 산업진흥 기본계획**을 통해 드론과 자율주행차 등에 활용될 3차원 정밀지도나 고해상도 위성정보와 재난·재해를 예방·대처 하는 데 쓰일 입체공간정보가 생성될 예정이어서 드론에서 취득되는 포인트 클라우드를 통한 오브젝트 인식 및 모델링과제를 진행하는 것은 시의적절.

다. 가상·증강현실을 이용한 가상 플랜트 및 기반시설의 리트로피팅(Retrofitting)

- 최근 건설 분야에서 건설기계가 무인화 되고 UAV를 이용해 수집한 정보를 바탕으로 3D설계안을 완성하거나 인공지능(AI) 컴퓨터로 SOC사업 등의 타당성을 검토하는 등 사업 전반에 ICT, AI 등과의 융·복합이 시도되고 있다. 예를 들어 일본의 건설중장비 업체 KOMATSU는 무인비행체를 통해 정확한 지형분석을 진행한 후 이를 바탕으로 설계도를 작성하고 건설 계획에 대한 사전 VR 시뮬레이션을 실현.

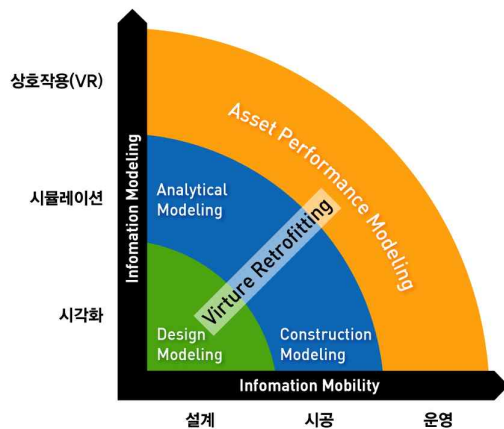


<접근성 낮은 지역 및 위험한 핵심 시설물의 가상 리트로피팅>

- 산업 현장에서 시설 확장 및 개조를 위한 수동 측량은 수개월 혹은 수백 시간이 필요하고, 잠재적으로 위험 지역을 포함하고 As-built 기반시설의 정확도가 낮거나 설계오류의 경우가 있을 때 비용이 증가되는 반면, 고화질 조사(HDS) 기술은 수 시간 내에 1/8인치의 밀도 및 정확도로 수백 군데 조사지점에서 데이터 추출이 가능하여 BIM 툴의 설계 결과를 비교/분석할 수 있는 가상/증강현실 기술의 융합은 효율적 측면에서 반드시 필요.

3. 연구개발 범위

- UAV의 제어와 가상 리트로피팅 기반 설계 시뮬레이션은 각기 독립된 영역에서 진행이 가능하지만, UAV의 자율주행을 보조, 제한하는 인터랙션과 가상 리트로피팅 기반 설계 시뮬레이션을 주도, 참조하는 인터페이스를 **다면 상호작용**(구매예정인 무선 모션추적장치 Xsens MTw Awinda와 스마트장치의 Shake, Slide & Touch 복합 입력) 인터페이스로 실현하여 UAV의 제어, 데이터 취득, 가상 리트로피팅을 동시에 실현.
- UAV-LiDAR-HMD융합 시스템과 LiDAR 정보와 역설계 형상정보의 정합을 위한 **방향성 곡률기반 실시간 요소식별**은 “라이다 장착 무인비행체 기반 실내 환경 역설계를 위한 실시간 포인트 클라우드 취득 프로세스 및 실내 형상 기초 요소 식별 알고리즘의 개발”을 완료한 주관연구기관 담당으로 선행연구를 통하여 관련 기초기술의 개발이 기 완료되었으므로 개발 수월성이 매우 높음.
- 기반시설, 플랜트의 증설·복구 계획용 가상 리트로피팅 시스템 개발은 “격자구조 모델링 방법에 의한 3D 스캔 point cloud 기반 플랜트 재구성 및 설계관리 시스템”의 개발을 공동 진행한 제이텍솔루션이 담당하여 선행연구 경험은 물론 현재 관련 사업의 진행으로, 적절한 인력구성과 제반 관련지식이 준비되어 가상 리트로피팅 시스템의 실현 가능성이 확실.

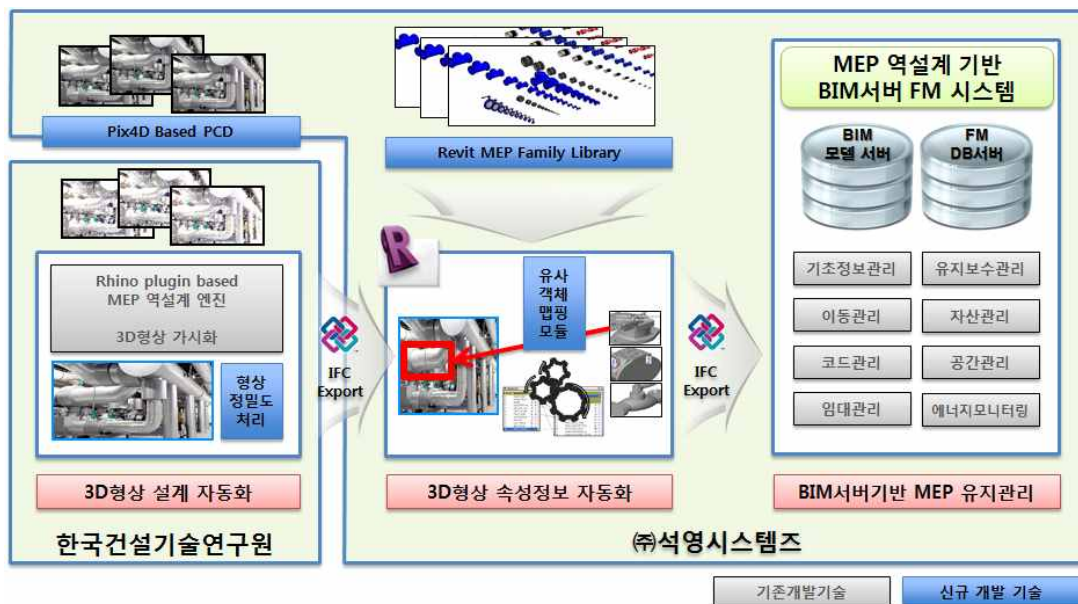


- UAV embedded LiDAR에 의한 역설계 특히 LiDAR 취득과 가상 리트로피팅에 의한 설계 계획을 동일한 인터페이스로 융합하는 개념은 **최초로 제안됨**.
- 무엇보다도, 라이다장착 무인비행체 기반 실시간 가상 리트로피팅 시스템은 정보 모델링 차원에서 시각화와 시뮬레이션을 넘어 상호작용(VR) 포함하고, 정보 유동성 측면에서도 설계, 시공 및 운영 모델링을 가능하게 하는 시스템(가상 리트로피팅) 성격으로 창의적이고 혁신적인 시작품을 개발하여 추후에 **기술이전**을 목표로 함.

제2장. 국내외 기술 개발 현황

1. 국내 현황

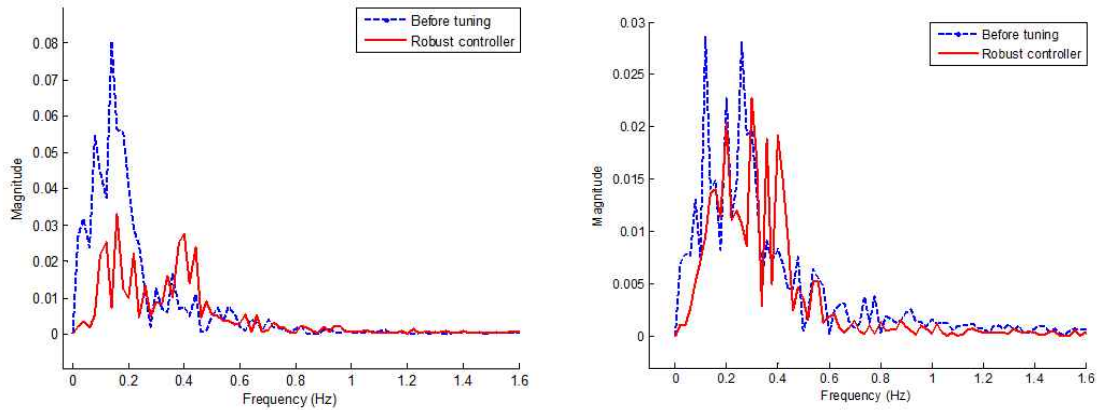
- 한국건설기술연구원: 3D객체 역설계 기반 MEP 설비 유지관리 시스템(2015 ~ 2016), 3차원 스캐닝 Data를 Preprocessing, Extraction 하여, Authoring Tool 기반 자동/반자동 역설계 기술을 통해 3차원 형상을 생성하고, 이것을 BIM 객체에 맵핑한 후, FMS (Facility Management System)에 BIM객체 기반으로 시설물 관리 데이터를 연계하는 기술을 개발함.
- ① 3D 이미지 스캔 데이터를 기반으로 한 MEP 설비 형상 인식, 추출, 시설물 BIM 맵핑을 통해, 설비에 대한 유지 관리 기술 및 시스템을 개발함.
- ② 3D 이미지 스캔 데이터를 이용하여, 건축 객체별로 세그먼테이션을 하고, 세그먼테이션된 3D Point Cloud Data를 이용해, BIM 형상을 모델링하고, 규칙 기반으로 맵핑하여, 시설물 데이터를 효과적으로 관리하는 기술을 개발함.
- ③ 역설계로 생성된 BIM Data를 표준 건축정보통합모델인 BIM(*.ifc), Revit(*.rvt), Archicad(*.pla)를 기반으로 시설물 유지관리 분야, 건축 설계 분야, 설비 엔지니어링 분야 정보를 융합하여, 설비를 유지관리 하는 기술을 플랫폼개념으로 개발함.



<3D객체 역설계 기반 시설물 유지관리 시스템 개념>

- 중앙대 지능메카트로닉스 및 로봇연구실: GPS 수신 불능 환경하에서 쿼드로터 헬리콥터의 자율비행을 위한 강인제어(2017), GPS 불가 환경하에서 쿼드로터 헬리콥터의 강인제어에 대한 연구를 수행하고, 특히, 종래의 연구에서 신중하게 고려되지 않았던 누적오차 문제, 시스템 모델의 타당성 문제 및 불확실성을 고려한 제어기 설계 문제를 모두 고려하여 비행제어 결과에 신뢰성을 입증함.
- ① 종래의 IMU센서 기반의 추정치와 최신 SLAM 알고리즘의 위치 추정치를 융합하여 누적오차 현상을 완벽히 제거하는 동시에 빠른 반응성을 지니는 좋은 품질의 위치 추정치를 획득함.

- ② 넓은 대역에서 균일한 주파수 반응 크기를 지니는 가진 신호를 설계하여 식별을 진행하였으며, 반복 실험을 통해 노이즈와 외란의 영향을 감소시키는 반면 식별된 결과에 대한 불확실성을 제거함.
- ③ 시스템 식별 과정에서 획득한 불확실성 정보와 외부 상황 분석으로 얻은 외란 정보를 모두 반영하는 고정구조 강인제어기를 설계하였으며 이를 시뮬레이션 및 실제 비행실험으로 검증함.



<Spectrum analysis of output(left) & control(right) of forward system at hover>

- ④ 설계된 고정구조 강인제어기가 목표로 했던 제어성능(0.15 Hz 이하의 외란 억제 및 1 Hz 이상의 고주파 제거)을 만족시킴.

2. 국외 현황

- IATEC Plant Solutions Inc.는 장기간 수작업이 위험하고 비효율적이므로 DJI Inspire 1과 Pix4Dcapture를 이용하여 10개 지역에서 총 2,588개의 nadir 및 oblique 이미지를 취득하여 3일 만에 As-built 모델의 사진측량을 하는 시스템을 Petrobras에 제안, 진행하였고, 또한 UAV를 이용하면 NDT(Non-Destructive Testing)를 수행하면, 접근이 불가능하거나 용이하지 않은 지역에서 파손이나 누수를 효과적으로 점검할 수 있고, 열화상 카메라를 이용하여 손상, 부식정도를 고려하여 구체적이고 적용 가능한 수리, 보존일정을 수립할 수 있게 해줌.



<브라질 PETROBRAS 가스 Plant의 2016년 Aerial Photogrammetric Survey와 가상 리트로피팅용 열화상 영상>

- 리트로피팅이 정확하게 진행되기 위해서는 As-built 모델의 3D 레이저 스캐닝이 정확하게 이루어지고 파라메트릭 모델까지 완성하는 Scan-to-BIM, Scan-to-Parametric Model까지 필요하며, 완성 후에는 그림의 MV ANANGEL

TRANSPORTER 선내 선박평형수처리시스템, BWTS(Ballast Water Treatment System) 리트로피팅 시스템과 같이 해당 위치에 As-built 스캔모델과 같이 실행가능성의 예측을 용이하게 해준다. 또한 건설 시공 모니터링에 BIM 데이터와 같이 사용¹⁾되어 정보의 연속적인 흐름의 이해를 통해 제약조건을 손쉽게 파악하고 제한하여 낭비요소를 극소화 시킴.



<독일 Fargo사의 PointSense S/W와 그리스 ARGO Navis Marine의 DWT Bulk carrier 선내 BWTS 리트로피팅>

- 리트로피팅 과정은 물론 리트로피팅이 진행된 후에도, 오일 & 가스 기반시설은 이전 운전자의 know-how와 경험이 최소한의 서비스 중단으로 연속되고 재가동되기 위해서 VR 시뮬레이션을 통하여 이후 운전자에게 교육과 트레이닝이 필요하다. 특히, 2016년 CES에서 발표된 미국의 Daqri사의 스마트 헬멧은 유전, 수처리 시설, 공사현장 등 산업현장에서 사용될 수 있게 카메라와 센서로 주변환경(Valve readings, thermal data)을 취득하여 BIM 데이터와 결합하고, 최소한 움직임과 외부전문정보와의 조합으로 손쉽고 지능적인 조작을 가능하게 함.



<Dassault 시스템의 3D 트레이닝환경, Raken사의 VR 건설현장 어플, Daqri사의 스마트 AR헬멧(2016 CES)>

- 상업적으로 리트로피팅 서비스를 제공하는 미국의 KBR(이전의 Kellogg Brown & Root)는 레이저 스캐닝을 통해 공학 설계, 건설 및 공장 개선 프로젝트의 최적화를 진행하고 있고, UAE 샤르자의 Aries사는 조선관련 리트로피팅을 진행시 3D Scanner를 이용하여 3D 모델을 완성하고 있으며, 런던과 폴란드 Krakow에 거점을 둔 3Deling사는 1. Laser scanning and clash detection in Kozienice power plant 프로젝트에 Scanner Riegl VZ-400, Leica TCRA 1101 plus Total Station을 이용하여 2인이 8일간 800점의 스캐닝을 진행하였고, 2. Aveva PDMS As-built model and design of a midsized chemical plant 프로젝트에 Scanner Riegl VZ-400, Leica HDS7000, Leica TCRA 1101 plus Total Station을 이용하여 2인 3일 350 포인트 스캐닝과 컬러링/정합에 10일을 소요한 바 있음.



<리트로피팅 서비스 제공사: KBR(Kellogg Brown & Root), ARIES Marine & Engineering Services, 3Deling>

- VR/AR을 이용한 트레이닝을 통한 기억유지는 약 70-90%로 전통적 트레이닝의 최대 55%보다 우수하다고 알려져 있다. ORINOX사는 DotProduct 포켓용 레이저스캐너로 3D survey를 시험하였고, 미국 IDEX AR Solutions는 스마트 기기를 활용한 AR을 건설, 제조, 조선 및 에너지 분야의 마케팅, 사전시각화부터 건설, 제조, 점검 및 운전, 유지보수까지 응용하였으며, 전 세계 기반시설 건설/유지용 전문가 S/W개발사인 벤틀리 시스템은 HoloLens를 이용하여 반복적이고 규칙적인 시설 유지보수업무를 단계별로 보조/교육하는 시스템을 실현.



<AR 포인트 클라우드 프로세싱: ORINOX DotProduct Phi.3D, INDEX AR Solutions and Bentley HoloLens>

제3장. 연구 수행 내용 및 성과

1. 1차 연도

가. 개발 목표: 주관연구기관(중앙대학교 산학협력단):

- (1) UAV-LiDAR 융합시스템의 개발 및 수리 모델 구축
- (2) 기초 요소 식별을 위한 방향성 형상 기술자 정의 및 구현

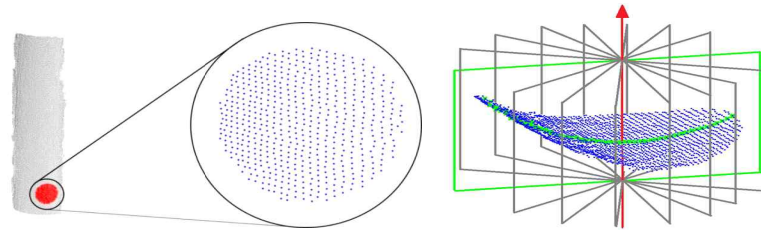
나. 개발 내용 및 범위: 주관연구기관(중앙대학교 산학협력단):

- (1) UAV-LiDAR 융합시스템의 개발 및 수리 모델 구축

- UAV와 지상국 간의 네트워킹과 **원격 관제 시스템** 등을 구축하고 비행체에 전달 될 임무 할당 및 비행 정보와 영상의 송·수신 등의 제어·운용·모니터링을 위한 사용자 친화형 GUI 환경 개발

- (2) 기초 요소 식별을 위한 방향성 형상 기술자 정의 및 구현

- **방향성 형상 기술자**를 적용하여 기초 요소별로 정의된 결과와 유사도에 따라 형상을 추정할 수 있다. 결과는 히스토그램의 형태로 나타나며 그래프의 차이가 작은 패치를 찾아 정의된 기초 요소 또는 기초요소 후보로 규정가능.



<레이저 스캐닝 데이터의 일부분을 이용한 곡률 추출(추정) 방법>

- PCA를 통해 표면 법선을 추정하고 Circle Fitting으로 반경을 추정하여 속성을 추출하는 방법으로 자르는 평면의 수만큼 반경(곡률)이 추정되고, 가장 작은 값을 파이프 Φ 로 예측.

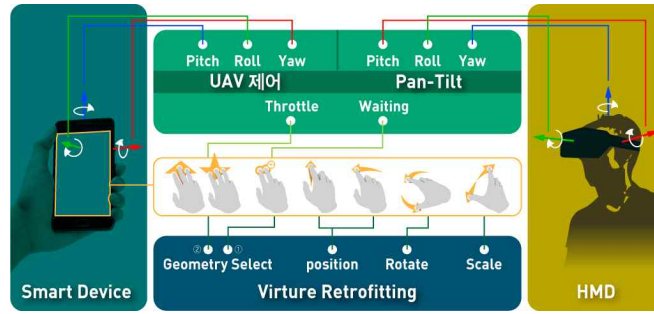
다. 개발 목표: 공동연구기관(주식회사 제이텍솔루션):

- (1) AR기반 리트로피팅 효율화 모형 및 인터페이스 구축

라. 개발 내용 및 범위: 공동연구기관(주식회사 제이텍솔루션):

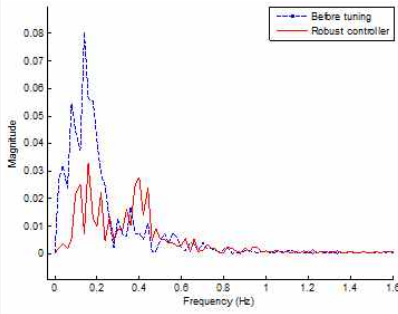
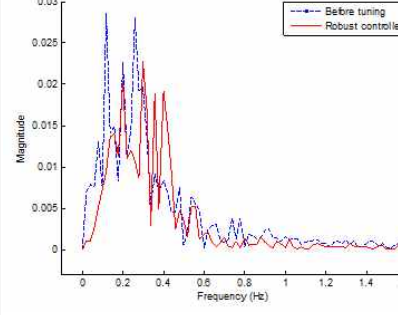
- (1) AR기반 리트로피팅 효율화 모형 및 인터페이스 구축

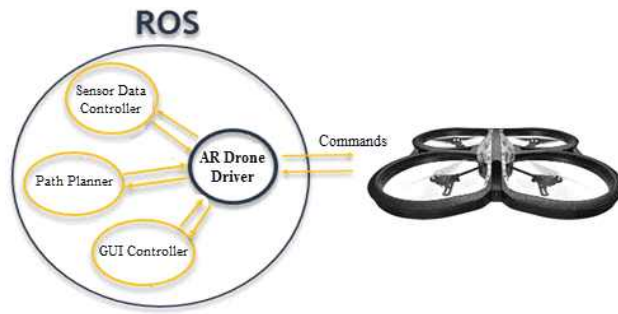
- 증설, 복구를 위한 계획단계에서 잠재적 결함을 발견, 수정할 수 있게 하는 가상 리트로피팅 시스템은 역설계 정보와 자동 요소 식별을 통한 실시간 정합을 포함한 효율화 모형 개발로 현장에서 작업을 하는 것에 비해 시간과 비용을 획기적 절약.
- HMD를 착용한 사용자는 UAV와 장착된 Pan-Tilt 카메라를 직관적으로 제어하면서, 제공되는 Airborne FPV를 통해 가상 리트로피팅 작업을 효율적으로 수행할 수 있게 스마트장치를 이용한 **Shake, Slide & Touch** 복합 입력 인터페이스를 구현.



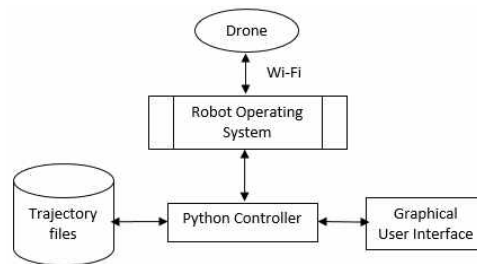
<스마트장치의 Shake, Slide & Touch 복합 입력 인터페이스>

마. 연구 수행 방법 및 구체적인 내용:

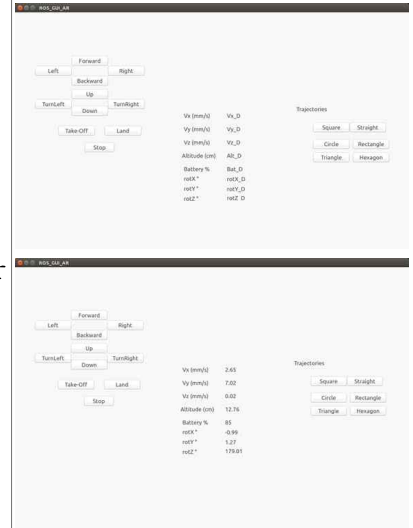
연구 범위	연구 수행 방법 (이론적 · 실험적 접근방법)	구체적인 내용
- GPS 수신 불능 환경하에서 쿼드로터 헬리콥터의 자율비행을 위한 강인제어 - Graphical User Interface Based Quadcopter Control by Using Robot Operating System	- 중래의 IMU센서 기반의 추정치와 최신 SLAM 알고리즘의 위치추정치를 융합하여 누적오차 현상을 제거하는 동시에 빠른 반응성을 지니는 위치추정치를 획득. - 넓은 대역에서 균일한 주파수 반응 크기를 가진 신호를 설계하여 식별을 진행하였으며 실험을 통해 노이즈와 외란의 영향을 감소시키고 식별된 결과에 대한 불확실성 제거. - 시스템 식별 과정에서 획득한 불확실성 정보와 외부 상황 분석으로 얻은 외란 정보를 모두 반영하는 고정구조 강인제어기를 설계하였으며 이를 시뮬레이션 및 비행실험으로 검증. - 시스템식별 및 강인제어기 설계과정에서 이용된 Parrot 社의 AR.Drone 2.0은 다양한 내장센서, 강력한 연산능력의 메인보드, 추가적인 디지털 신호처리장치 및 가볍지만 충격에 강한 소재로 구성되어 있음에도 불구하고, 비교적 저렴한 가격에 이용할 수 있어 본 연구에도 이용. - 드론과 호스트간의 연결은 계획, 내비게이션 및 조작이 용이한 오픈소스 로봇 미들웨어인 ROS(Robot Operating System)를 이용하였고, 후위에서 AR Drone과 호스트 컴퓨터 간의 명령과 상태정보 제공 등의 통신을 담당.	 <AR.Drone의 테스트 비행 환경>  - Spectrum analysis of output of forward system at hover: 고주파제거 > 1Hz  - Spectrum analysis of control input of forward system at hover: 외란억제 < 0.15Hz



- 파이썬 컨트롤러는 다이나믹 시스템의 제어를 담당하고 ROS와 GUI를 연결한다. GUI를 통하여 드론의 고도, 선/각 속도 등을 표시하고 이/착륙, 선행운동, 회전등의 명령입력을 가능하게 하며 자동계획운동을 실현하여 리트로피팅에 응용.

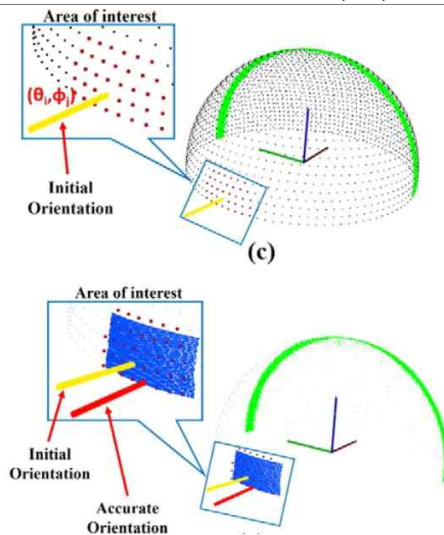


<Software architecture for the proposed system>

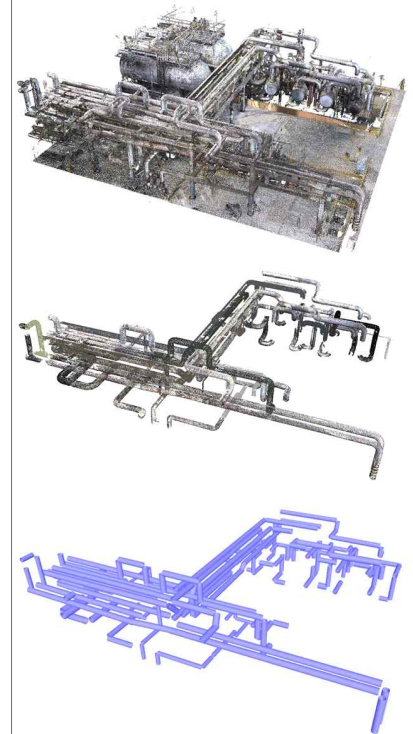


<구현된 GUI를 통한 navdate>

- An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds



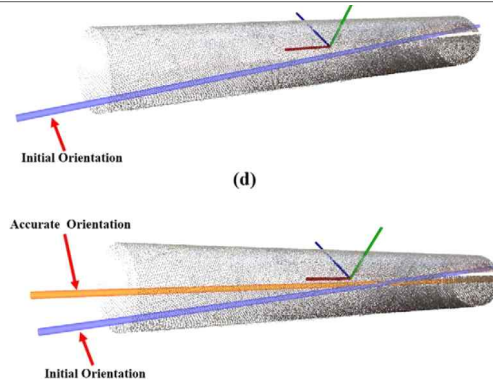
- Rabaani의 Hough 트랜스폼을 수정한 영역기반 Adaptive Hough 트랜스폼은 Gaussian Sphere의 관심영역 표본 밀도를 증가시켜 시간을 줄이고 아래와 같이 실린더 중심축 추정 정확도를 높임.



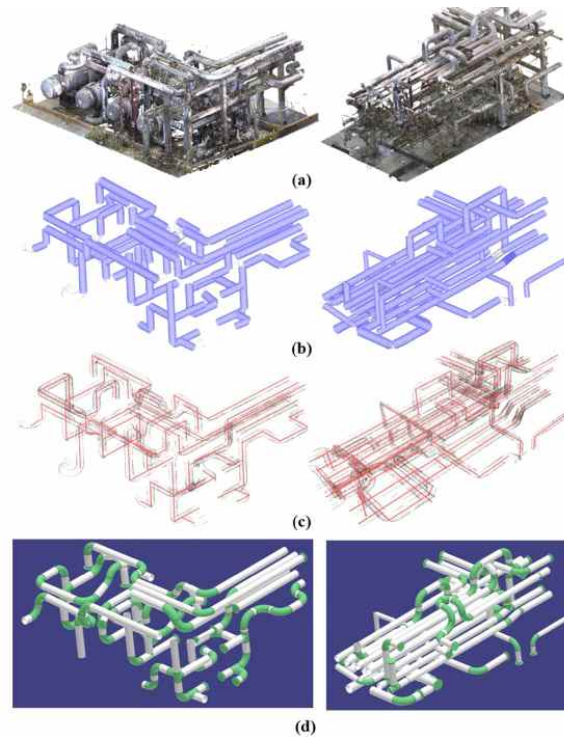
<자동 실린더 생성을 위한 포인트 클라우드와 추출된 실린더>

Input Point Cloud	Cylinder Exist	Hough Transform		Area-based Hough T	
		Straight Cylinders	%	Straight Cylinders	%
Part-A	215	183	85.11	197	91.62

<Statistics for straight cylinder detection>



- 영역기반 Adaptive Hough 트랜스폼을 이용하여 방향추정의 정확도를 높인 실린더 추적으로 포인트 클라우드에서 SP3D 모델링까지 완성.



(a) laser-scanned point cloud (b) cylinder detection
(c) axis network reconstruction (d) SP3D reconstruction results

Input Point Cloud	Cylinder Exist	Hough Transform		Area-based Hough Transform	
		Cylinder with Elbow	%	Cylinder with Elbow	%
Part-A	215	165	76.74	176	81.86

<Statistics for straight cylinders after elbow estimation>

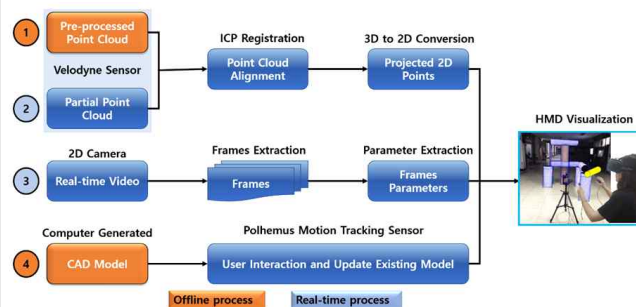
- 다양한 형태의 실린더 존재 상황에서 Universal 방향 취득과 Algebraic Circle Fitting 형상기술자로 실린더 형상을 검출하여 Hough 변환보다 개선된 직선파이프 91.62%, 엘보우 포함파이프 81.86%를 달성.



<Hough 변환과 영역기반 Hough변환의 실린더 추출비교>

- Hybrid approach for alignment of pre-processed 3D point cloud, video and CAD model using partial point cloud in retrofitting application

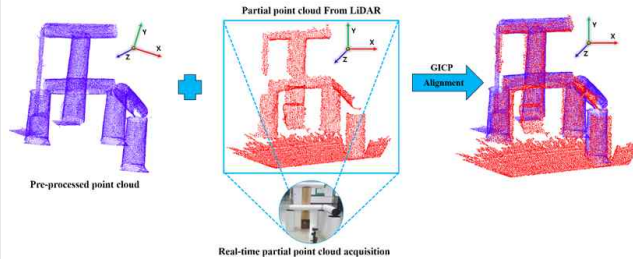
- 4 in 1 정합 프레임워크를 통하여 사전에 취득된 포인트 클라우드와 실제 환경을 실시간으로 취득하는 라이다 포인트 클라우드로 정합한 후에 준비된 CAD모델과 함께 VR장비로 시현하여 가상 리트로피팅이 가능한 프로세스 구현.



<Overview of the proposed retrofitting application>

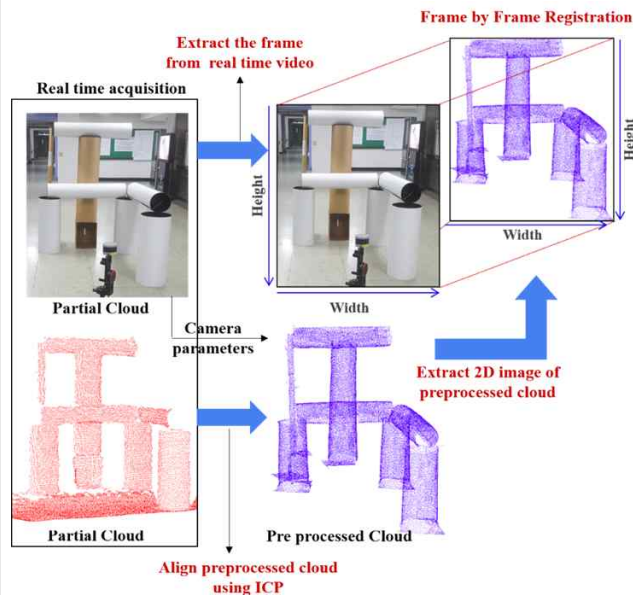
- 4 in 1 정합 프레임워크로 비디오 입력되는 환경에서 각 시점에서 정합과 위치, 회전 오차가 아래와 같이 정적 리트로피팅 오차 만족.

- 사전에 취득되고 정합된 포인트 클라우드 모델은 실시간으로 취득되는 부분적 포인트 클라우드와 ICP(Iterative Closest Point)알고리즘을 통하여 연결되는데, 이는 시시각각 변하는 전경을 부분 스캔으로 모델과 연결하여 리트로피팅을 실현.

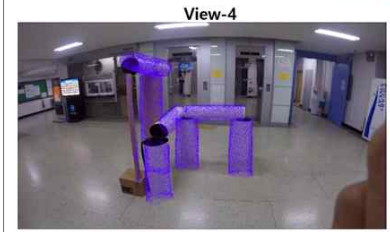
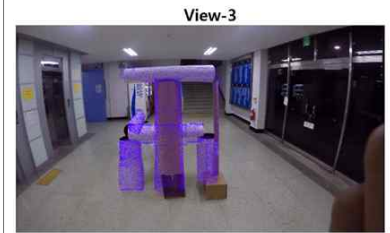
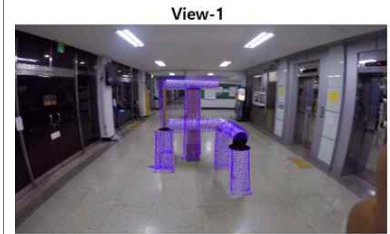


<Alignment of pre-processed & partial point cloud by GICP algorithm>

- LiDAR 스캐너로 취득되는 포인트 클라우드는 정합을 통하여 2D 비디오 프레임과 연결되는데, 이미지 평면 물체의 위치와 오리엔테이션을 추정한 후, Hao DU와 Peter Henry의 Frame to Frame 레지스트레이션 방법을 이용한 연결로 리트로피팅을 위한 모델과 전경의 일치를 완성.

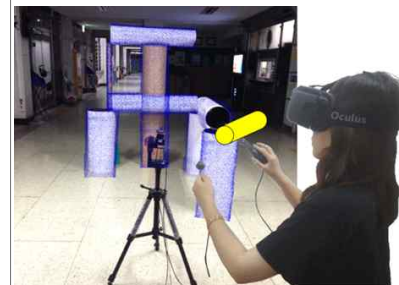


<Overview of pre-processed point cloud in video>



Alignment Views	반복	Alignment					
		Translation			Rotation		
		T_x	T_y	T_z	R_{roll}	R_{pitch}	R_{yaw}
View-1	8	-0.022	0.234	0.138	2.459	2.718	2.315
		-0.101	0.277	0.362	2.299	2.096	2.703
		-0.115	0.279	0.363	1.721	1.958	2.688
View-2	13	-0.091	-0.151	-0.155	-1.513	0.477	0.308
		-0.161	-0.265	-0.470	-1.306	-0.599	1.405
		-0.230	-0.270	-0.504	-1.276	-0.589	2.398
View-3	15	0.174	-0.224	-0.155	-0.535	-2.638	-0.128
		0.300	-0.168	-0.599	-0.220	-2.100	1.341
		0.318	-0.174	-0.612	0.130	-1.062	2.093
View-4	11	0.061	0.105	-0.079	2.483	-0.628	0.115
		0.168	0.241	-0.138	2.411	-0.619	1.102
		0.169	0.242	-0.138	1.919	0.705	2.698

- HMD, 사용자 모션 추적장치 전경카메라로 Interactive한 리트로피팅 환경을 구현.



<Virtual retrofitting with interactions>

2. 2차 연도

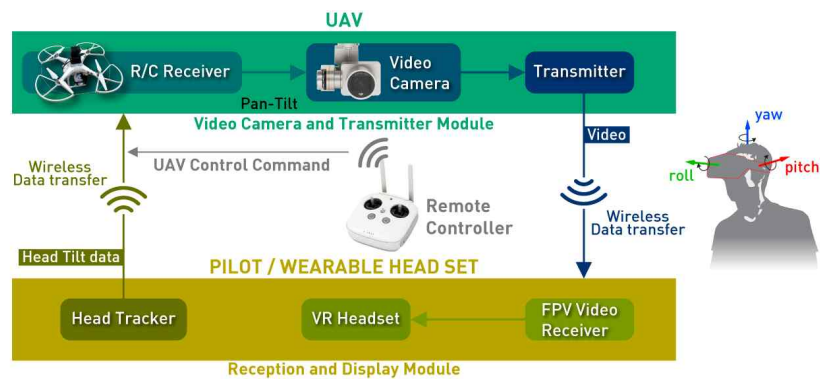
가. 개발 목표: 주관연구기관(중앙대학교 산학협력단):

- (1) Airborne FPV를 위한 UAV-HMD 융합 시스템 개발
- (2) 방향에 따른 형상기술자 활용 및 UAV 스캔 위치 추정기술 개발

나. 개발 내용 및 범위: 주관연구기관(중앙대학교 산학협력단):

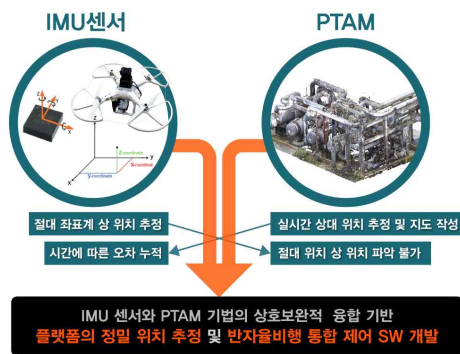
- (1) Airborne FPV를 위한 UAV-HMD 융합 시스템 개발

- **Airborne FPV**(First Person View) 융합 시스템은 UAV에 Pan-Tilt 모듈을 포함한 비디오카메라와 Transmitter가 장착되며, 사용자는 착용형 Head-Tracker(예: Arduino Pro Mini GY-85 9DOF IMU Board)가 결합된 Reception & Display 모듈로 UAV와 통신하면서 제어하여 UAV에서 바라보는 항공시야를 경험.



<Airborne FPV시스템의 전체 구성도>

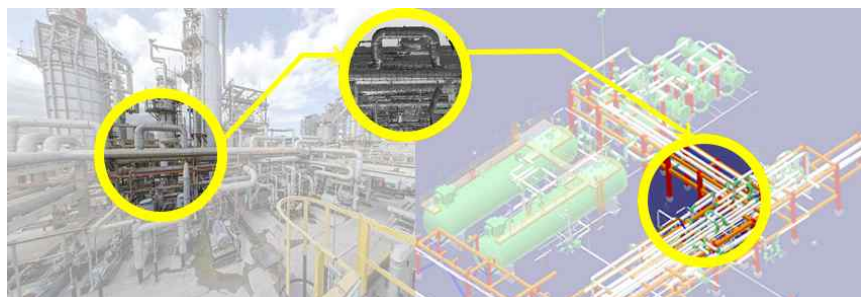
- SLAM(Simultaneous Localization And Mapping) 기반 기술을 무인비행체의 운용에 적용, 특히 **PTAM**(Parallel Tracking and Mapping)은 실시간으로 외부 환경 지도를 자율적으로 구축하거나 비행체의 정밀 위치 측정에 활용가능.



<IMU센서-PTAM 상호보완적 융합기반 제어>

- UAV 비행경로 상에 존재하는 장애물 즉 원활한 비행의 방해 요소들의 위치를 계측, 회피하면서 임무를 완료하는 **IMU센서-PTAM기반 반자율비행용 통합 제어 S/W**의 개발.

- (2) 방향에 따른 형상기술자 활용 및 UAV 스캔 위치 추정기술 개발



<속성 정보와 기존 모델링 비교를 통한 위치 추정(정합)>

- UAV의 좌표 방향을 기준으로 곡률의 최대 방향과 최소 방향을 파악하여 방향별

형상 기술자를 적용하여 표면 법선의 변화를 통해 속성 정보를 추출.

- 추출된 속성 정보와 기존 모델링 데이터의 비교 및 분석을 통해 현재 UAV의 스캔 위치를 파악할 수 있고, 데이터 스캔과 동시에 속성 정보 추출이 가능하여 **실시간으로 스캔된 데이터를 역설계 데이터와 비교하여 정합된 데이터를 확인.**
- 측정 포인트와 스캔 대상물의 거리는 스캐너의 권장 스캐닝 범위 이내에서 설정되어야 하며, 권장 범위 이내의 스캐닝시 제안된 기술을 통해 구현된 3D 정보와 실제 시공현황과의 오차범위가 스캐너의 오차범위 안에 포함될 수 있도록 함.
- 다수의 파이프가 복잡한 형태로 배치될 때 미세한 조립 오차로 인해 파이프사이의 간섭이 생기는 시공 오류를 피하기 위해 현장에서 배관 형태를 변경하고 모델링 데이터에는 시공사례를 반영하지 않았을 경우 스캔 위치의 조정이 필요.

다. 개발 목표: 공동연구기관(주식회사 제이텍솔루션):

(1) 라이다장착 UAV 기반 실시간 가상 리트로피팅

라. 개발 내용 및 범위: 공동연구기관(주식회사 제이텍솔루션):

(1) 라이다장착 UAV 기반 실시간 가상 리트로피팅

- **증강현실 기반 가상 리트로피팅**을 수행하기 위해 먼저 Target 모델과 증강되는 정보를 준비하고, 스캔된 대상의 포인트 클라우드 좌표값을 사용하여 현장의 구조물을 인식한다. 이후, 방향성 곡률기반 히스토그램을 이용하여 점검하고자 하는 객체를 추적하여 해당 구조물의 정보를 증강시키고 증강된 정보를 활용하여 가상 리트로피팅 시뮬레이션을 수행.



(a)



(b)



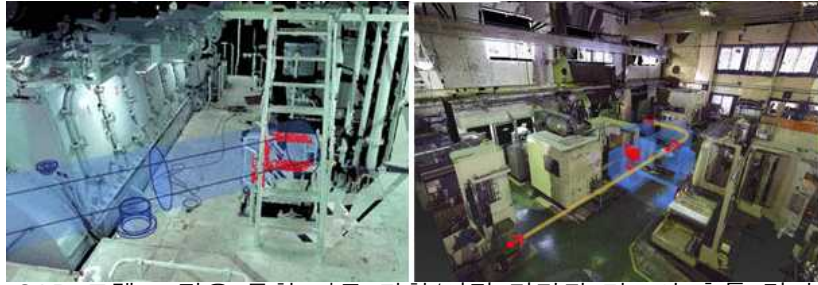
(c)



(d)

<(a) 현장, (b)결점 모델 AR, (c)결점 제거 현장, (d) 리트로피팅 AR>

- 가상 리트로피팅을 통해 사용자가 현장에 가지 않고 증강되는 CAD 데이터와 As-built 포인트 클라우드 데이터 사이의 충돌을 감지할 수 있게 하고, 포인트 클라우드의 특정 그룹을 은폐 또는 강조하여 장비 반출 경로, 레이아웃 변경 등의 시뮬레이션을 수행할 수 있게 함.



<CAD 모델 조작을 통한 빠른 관찰/사전 결정된 경로의 충돌 감지>

- 증강된 CAD 데이터와 포인트 클라우드의 비교는 디자인 개념에서 실제의 차이를 검출할 수 있다. 가상 리트로피팅과 실제 리트로피팅의 차이를 건설기술연구소 테스트플랜트에서 비교하여 일치(적합)도를 포함하여 구현 가능성(Feasibility)을 평가한다. 충돌감지, 장비반출경로, 레이아웃변경 등 다양한 Task에 따라 비교기준이 다를 수 있으므로 주요 Task별 임계변수를 설정하고, 정적범위를 제안함.

3. 2차 연도 연구 수행 방법 및 구체적인 내용

가. Airborne FPV를 위한 UAV-HMD 융합 시스템 개발

(1) 융합시스템에 사용된 Eye-Tracker

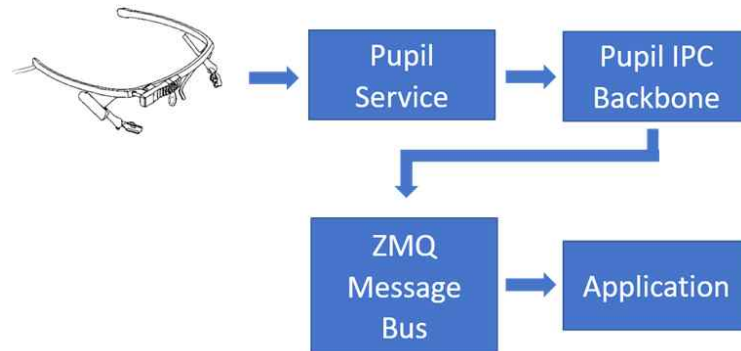
- 증강된 융합 시스템에 사용된 Eye Tracker는 Pupil Lab^[1]의 양안 추적장치로 아래 그림과 같이 좌우 눈동자를 추적하는 2개의 IR 카메라와 시야 전면을 추적하는 1개의 월드 카메라를 포함함.



Pupil Eye tracker	
Sensor	Global Shutter. 200x200 @ 200 fps, 400x400 @ 120 fps.
Illumination	IR camera with IR illumination (dark pupil tracking)
Latency	4.5 ms
FoV	100 degrees diagonal. Lenses are exchangeable: headset ships with 60deg and 100deg FoV lenses
Audio	Record audio via host device microphone

- Python, C, C++를 써여진 Pupil 응용 프로그램은 Eye Tracking의 핵심으로 Eye Process는 다음과 같은 기능을 포함.
 - 아이 카메라 스트림으로부터 아이 카메라 영상 획득
 - 아이 카메라 영상에서 눈동자 위치 식별
 - 식별된 눈동자의 위치 공개
- Pupil 응용 프로그램중 다른 하나인 World Process는 다음과 같은 기능을 포함.
 - 월드 카메라 스트림에서 월드 카메라 영상 획득
 - 아이 프로세스로부터 눈동자 위치 수령
 - 눈동자 위치로부터 응시점 계산을 위한 교정 매핑 수행
 - 응시점 고정, 곡면 추적 등의 다양한 플러그 인 로드
 - 비디오와 데이터 녹화 등 대부분의 제어, 조정 명령은 월드 프로세스가 수행

- Pupil 카메라를 이용한 Gaze획득은 다음과 같은 순서로 이루어짐.
 - ① 아이 트래커를 시리얼 케이블을 이용하여 시스템과 연결.
 - ② Pupil Labs 서비스를 통해 영상을 캡처하여 데이터를 획득.
 - ③ Pupil Labs IPC 백본으로부터 ZMQ 메시징 프로토콜로 데이터를 얻음.
 - ④ Pupil 데이터는 정규화된 응시 데이터로 변환되어 사용.
 - ⑤ 아래는 아이 트래커에서 응시 데이터 및 응용프로그램으로의 데이터 흐름.



< Dataflow diagram representing the acquisition of gaze data into application >

- Pupil 응용 프로그램은 ㉠ Pupil data를 아래와 같은 형식으로 내보낸다. 이 때 'norm_pos'가 적외선 카메라로 얻는 정규화된 눈동자 좌표이고, 응시점 계산을 위해 좌우 눈동자가 카메라 좌표계에서 얻어져야 함.

```

1. { # pupil datum
2.   'topic': 'pupil',
3.   'method': '3d c++',
4.   'norm_pos': [0.5, 0.5], # norm space, [0, 1]
5.   'diameter': 0.0, # 2D image space, unit: pixel
6.   'timestamp': 535741.715303987, # time, unit: seconds
7.   'confidence': 0.0, # [0, 1]
8.
9.   # 2D ellipse of the pupil in image coordinates
10.  'ellipse': { # image space, unit: pixel
11.    'angle': 90.0, # unit: degrees
12.    'center': [320.0, 240.0],
13.    'axes': [0.0, 0.0]},
14.  'id': 0, # eye id, 0 or 1
15.
16.  # 3D model data
17.  'model_birth_timestamp': -1.0,
18.  'model_confidence': 0.0,
19.  'model_id': 1
20.
21.  # pupil polar coordinates on 3D eye model. The model assumes a fixed
22.  # eye ball size. Therefore there is no `radius` key
23.  'theta': 0,
24.  'phi': 0,
25.
26.  # 3D pupil ellipse
27.  'circle_3d': { # 3D space, unit: mm
28.    'normal': [0.0, -0.0, 0.0],
29.    'radius': 0.0,
30.    'center': [0.0, -0.0, 0.0]},
31.  'diameter_3d': 0.0, # 3D space, unit: mm
32.
33.  # 3D eye ball sphere
34.  'sphere': { # 3D space, unit: mm
35.    'radius': 0.0,
36.    'center': [0.0, -0.0, 0.0]},
37.  'projected_sphere': { # image space, unit: pixel
38.    'angle': 90.0,
39.    'center': [0, 0],
    'axes': [0, 0]}
  
```

< Pupil Data exported from the Pupil Apps > [Data: 아이트래커 → Pupil Labs 응용 프로그램]

- Pupil 응용 프로그램은 ⑥ Gaze data를 아래와 같은 형식으로 내보낸다.
정규화된 응시점의 좌표(line 9)와 노말(line 7)을 포함.

```
{ # gaze datum
1.   'topic': 'gaze',
2.   'confidence': 1.0, # [0, 1]
3.   'norm_pos': [0.5238293689178297, 0.5811187961748036], # norm space, [0, 1]
4.   'timestamp': 536522.568094512, # time, unit: seconds
5.
6.   # 3D space, unit: mm
7.   'gaze_normal_3d': [-0.03966349641933964, 0.007685562866422135, 0.9991835362811073],
8.   'eye_center_3d': [20.713998951917564, -22.466222119962115, 11.201474469783548],
9.   'gaze_point_3d': [0.8822507422478054, -18.62344068675104, 510.7932426103372],
10.  'base_data': [<pupil datum>]} # list of pupil data that was used to calculate the
gaze
```

< Gaze Data exported by the pupil apps > [Data: 아이트래커 → Pupil Labs 응용 프로그램]

- Pupil 응용 프로세스간의 통신(IPC Backbone)은 ZeroMQ^[2] 네트워크 라이브러리를 기본으로 하는 소켓(REQ-REP: reliable one-to-one communication, PUB-SUB: one-to-many communication)을 이용하며, IPC Backbone으로 ① pupil ② gaze ③ notify ④ logging와 같은 토픽에 게재하거나 연관 토픽에 관련 데이터를 얻게 되며, 아래 코드는 원격 Pupil 서비스를 통한 Gaze data를 얻은 부분.

```
1. import zmq
2. ctx = zmq.Context()
3. # The requester talks to Pupil remote and receives the session unique IPC SUB
   PORT
4. requester = ctx.socket(zmq.REQ)
5. ip = 'localhost' #If you talk to a different machine use its IP.
6. port = 50020 #The port defaults to 50020 but can be set in the GUI of Pupil
   Capture.
7. requester.connect('tcp://s:s'%(ip,port))
8. requester.send_string('SUB_PORT')
9. sub_port = requester.recv_string()
10. # open a sub port to listen to pupil
11. sub = context.socket(zmq.SUB)
12. sub.connect("tcp://{:}:{:}".format(addr, sub_port))
13. # set subscriptions to topics
14. sub.setsockopt_string(zmq.SUBSCRIBE, 'gaze')
15. # receives the string and print it on the console for C++ std::in to read.
16. while True:
17.     try:
18.         topic = sub.recv_string()
19.         msg = sub.recv()
20.         msg = loads(msg, encoding='utf-8')
21.         print("%s,%s" % (msg.get('norm_pos')[0],
22.                             msg.get('norm_pos')[1]),flush=True)
23.     except KeyboardInterrupt:
24.         break
```

< Usage of IPC Backbone for remotely fetching gaze data > [Python: IPC 백본→ Gaze 응용 프로그램]

- ZeroMQ(ØMQ: zero-em-queue) 라이브러리는 AT&T, Cisco, EA, Los Alamos Labs, NASA, Weta Digital, Zynga, Spotify, Samsung Electronics, Microsoft, CERN에서 사용되는 오픈소스 네트워킹 라이브러리로 다음과 같은 기능을 포함.
 - 어떤 언어/플랫폼으로도 해당 코드를 연결
 - IPC, TCP, TIPC, multicast, inproc 간의 메시지를 전달
 - Pub-sub, push-pull 및 router-dealer 같은 스마트한 패턴

- 아주 작은 라이브러리로 고속, 비동기 I/O 엔진 실현
- 크고 활발한 오픈 소스 커뮤니티의 지원
- 대부분의 최신 언어와 플랫폼을 지원
- 집중, 분산, 소용량, 대용량등 어떤 아키텍처로도 빌드

- HTC Vive^[3]가 포인트 클라우드 데이터의 렌더링에 사용되었고, 아래 그림과 같이 Pupil Eye tracker가 add-on^[4]된 장치를 통해 VR/AR 응용이 가능하지만, 가상 리트로피팅의 경우 SW간의 종속성문제로 융합 시스템에 추 후 적용예정.



< Pupil Eye tracker add-on to HTC Vive >

(2) 융합시스템에 사용된 Gimbal Camera

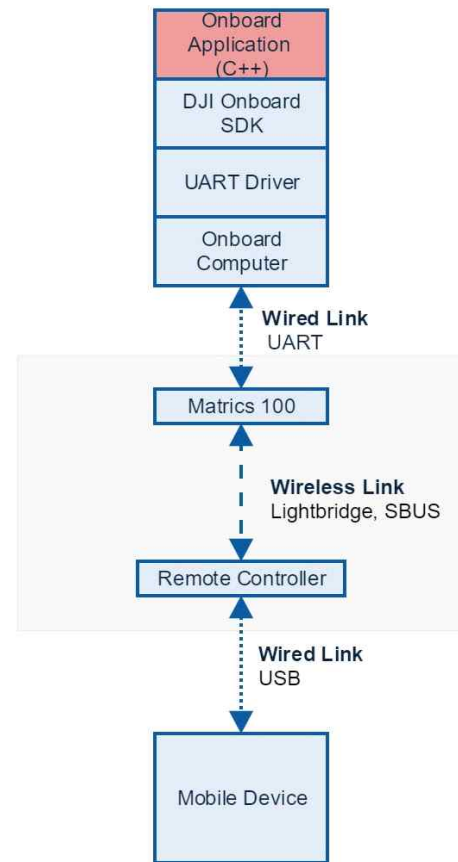
- 융합시스템에 사용된 UAV는 DJI Matrix 100^[5]이고, Zenmuse X3 gimbal 카메라^[6]가 하단에 아래 그림과 같이 장착되어 있고, 기술적 특성은 아래 표와 같음.



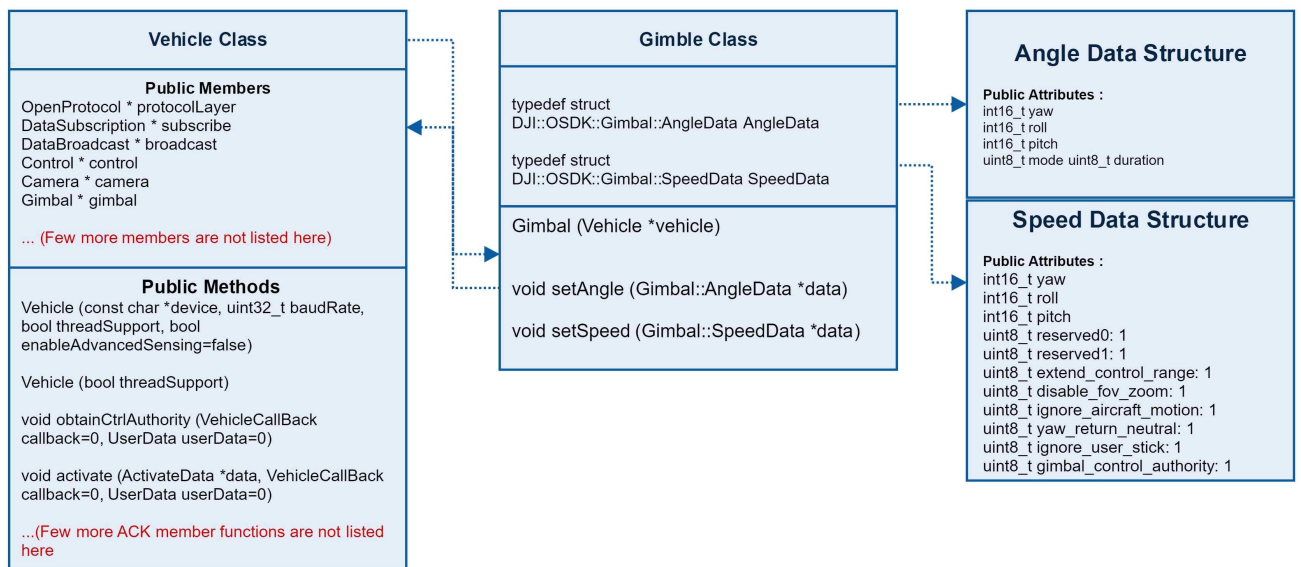
Zenmuse X3	
GENERAL	
Dimensions	107.3×75.5×94.8mm
Weight	221g
CAMERA	
Sensor	CMOS 1/2.3"
Lens	F/2.8 20mm (35mm format equivalent)
FOV	94°
Photo Resolutions	4000×3000
Video Resolutions	UHD: 4096×2160 p24/25
	3840×2160p 24/25/30
	FHD: 1920×1080 p24/25/30/48/50/60
	HD: 1280×720 p24/25/30/48/50/60
Anti-Flicker	Auto, 50Hz, 60Hz
GIMBAL	
Angular Vibration Range	±0.03°
Mount	Detachable
Controllable Range	Pitch: -90° to +30°
	Pan: ±320°
Mechanical Range	Pitch: -125° to +45°
	Pan: ±330°
Max Controllable Speed	Tilt: 120°/s
	Pan: 180°/s

< Matrix 100 DJI developer Done with Zenmuse X3 gimble camera mounted and its technical spec >

- DJI Onboard SDK^[7]는 Gimbal과 카메라를 이용하여 ㉠ Take Photo ㉡ Start Video ㉢ Stop Video ㉣ Control Gimbal Angle ㉤ Control Gimbal Speed를 할 수 있고, 우측 그림은 UAV와 응용프로그램사이의 위치와 관계를 나타냄, 본 과제에서 구현하는 Eye Tracker - Gimbal 카메라 융합시스템은 최상단의 C++ Onboard Application 임.
- 아래는 <dji_gimbal.hpp>의 gimbal class 구조를 보여주고 Constructor로 Vehicle 오브젝트를 기본으로 가지고 있으며, Telemetry 관련 정보를 포함한다. Gimbal의 Angle과 Speed 데이터는 구조체로 정의.



< Placement of Apps, Onboard SDK & UAV >



< Gimble Class Diagram of the DJI Onboard SDK >

- 다음은 공개 프로토콜을 이용하여 카메라 관련 제어를 위한 ㉠ Camera class이고 shootPhoto(), videoStart(), videoStop()함수로 접근.

```

1. DJI::OSDK::Camera Class Reference
2. #include <dji_camera.hpp>
3. Public Member Functions
4.     Camera (Vehicle *vehicle)
5.     void shootPhoto ()
6.     void videoStart ()
7.     void videoStop ()

```

[dji_camera.hpp: camera 클래스]

- 아래는 공개 프로토콜을 이용하여 Gimbal 제어를 위한 ㉞ Gimbal class로 AngleData와 SpeedData구조체가 정의되어 있고, setAngle, setSpeed 함수를 통해 제어.

```

1. DJI::OSDK::Gimbal Class Reference
2. #include <dji_gimbal.hpp>
3. struct AngleData: The Angle Data struct for gimbal control.
   Public Attributes: (int16_t yaw, int16_t roll, int16_t pitch, uint8_t mode, uint8_t
duration)
4. struct SpeedData: The Speed Data struct for gimbal control.
   Public Attributes: (int16_t yaw, int16_t roll, int16_t pitch, uint8_t reserved0: 1,
uint8_t reserved1: 1, uint8_t extend_control_range: 1, uint8_t disable_fov_zoom:
1, uint8_t ignore_aircraft_motion: 1, uint8_t yaw_return_neutral: 1, uint8_t
ignore_user_stick: 1, uint8_t gimbal_control_authority: 1)
5. Public Types
6.     typedef struct DJI::OSDK::Gimbal::AngleData AngleData
7.     typedef struct DJI::OSDK::Gimbal::SpeedData SpeedData
8. Public Member Functions
9.     Gimbal (Vehicle *vehicle)
10.    void setAngle (Gimbal::AngleData *data)
11.    void setSpeed (Gimbal::SpeedData *data)

```

[dji_gimbal.hpp: gimble camera angle & speed]

- 브로드캐스팅, Gimbal Angle 반환 등을 위한 ㉟ Telemetry 구조는 다음과 같고, roll, pitch, yaw, pitchLimit, rollLimit, yawLimit 등의 변수를 포함.

```

1. DJI::OSDK::Telemetry::Gimbal Struct Reference
2. #include <dji_telemetry.hpp>
3. Public Attributes
4.    float32_t roll
5.    float32_t pitch
6.    float32_t yaw
7.    uint8_t pitchLimit: 1
8.    uint8_t rollLimit:
9.    uint8_t yawLimit: 1
10.   uint8_t reserved: 5
11.
12. Member Data Documentation
13.    float32_t DJI::OSDK::Telemetry::Gimbal::pitch degree;
14.    uint8_t DJI::OSDK::Telemetry::Gimbal::pitchLimit
15.        1 - axis reached limit, 0 - no;
16.    float32_t DJI::OSDK::Telemetry::Gimbal::roll degree;
17.    uint8_t DJI::OSDK::Telemetry::Gimbal::rollLimit;
18.        1 - axis reached limit, 0 - no;
19.    float32_t DJI::OSDK::Telemetry::Gimbal::yaw degree;
20.    uint8_t DJI::OSDK::Telemetry::Gimbal::yawLimit
21.        1 - axis reached limit, 0 - no

```

[dji_telemetry.hpp: roll, pitch & yaw of UAV]

- Android와 iOS를 이용하여 짐벌과 UAV의 움직임 궤적을 프로그램 할 수 있으며, 아래는 Gimbal과 Camera를 위한 DJI의 다양한 Mobile SDK를 보여준다. 일반적으로 짐벌의 움직임을 제어하고 사진을 찍고 비디오를 녹화하는 카메라의 기능을 포함.

(3) 융합시스템을 위한 Eye Simulation

- 융합 시스템의 Eye-Camera 매핑을 실현하기 위하여 안구 시뮬레이션을 진행하였다. 복수의 목표물을 다양한 시나리오로 응시하고, Eye Tracker로 눈동자를 추적하여 안구운동의 단속운동과 전정안반사(VoR: Vestibular-ocular Reflex)를 확인.
- VoR의 원리는 다음과 같음.
 - ① The eyeball moves faster towards the target than the head, which follows behind.
 - ② The head follows the target until the target is reaches a comfortable position within the FoV (generally 0.03mm in the normalized coordinate system).
 - ③ During step ②, the eyeball remains fixed on the target, moving in the opposite direction from the head so as to compensate for head movement.
- VoR의 원리를 이용한 시뮬레이션의 절차는 다음과 같음.
 - ① Start
 - ② Populate the scene with user defined scenario target points taken from the scenario file
 - ③ Compute the movement of the head from the initial point to the next set of points
 - ④ Compute the time required for head movement from the current position to the next position
 - ⑤ Start timer
 - ⑥ Change the angle of the viewport, Incrementally in x, y, and z directions until the target is reached, following the equations below (input angle is derived from the positions of scenario points taken from the scenario file)

$$V_x = CA_x \cos(A_x \gamma \frac{\pi}{180}) + CA_z \sin(A_x \gamma \frac{\pi}{180})$$

$$V_y = CA_y \cos(A_y \gamma \frac{\pi}{180}) - CA_z \sin(A_y \gamma \frac{\pi}{180})$$

$$V_z = \pm CA_{(x/y)} \cos(A_z \gamma \frac{\pi}{180}) + CA_z \sin(A_z \gamma \frac{\pi}{180})$$

where, $V_{(x/y/z)} = \text{View Port } x/y/z$

$CA_{(x/y/z)} = \text{Current ViewPort Angle } x/y/z$

$A_{(x/y/z)} = \text{New ViewPort Angle } x/y/z$

$\gamma = \text{OpenGL resolution correction factor (VoR/Fixed Eye)}$

- ⑦ Change the position of the gaze such that

$$G_{VoR}^w = \begin{cases} G_x^P = C_x^P \pm |2a\alpha| \\ G_y^P = C_y^P \pm |2a\alpha| \end{cases}$$

$$G_{VoR} = \begin{cases} G_x^P = C_x^P \pm |a\beta| \\ G_y^P = C_y^P \pm |a\beta| \end{cases}$$

where, $G_{VoR}^w = \text{Gaze with VoR}$

$G_{VoR} = \text{Gaze without VoR}$

$G_{x/y}^P = \text{Gaze Position } x/y \text{ (in pixels)}$

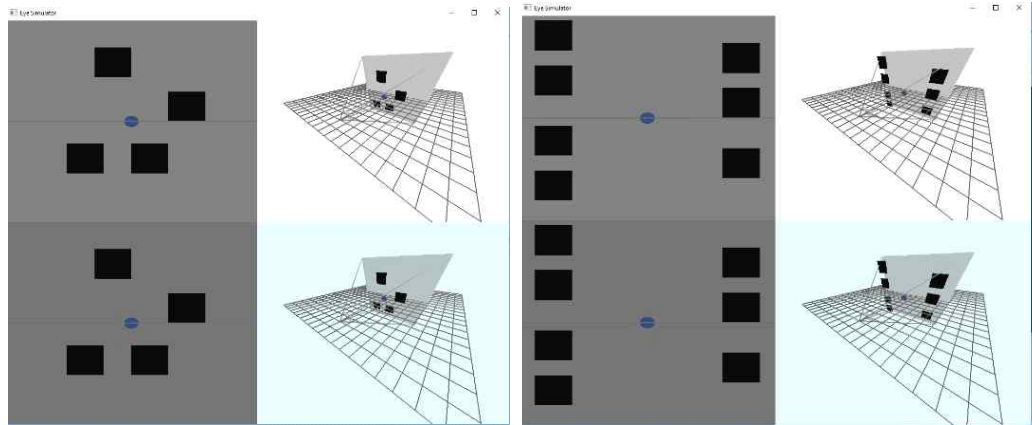
$C_{x/y}^P = \text{Current Position } x/y \text{ (in pixels)}$

$a = \text{New angle of gaze normal}$

$\alpha/\beta = \text{OpenGL resolution correction factors}$

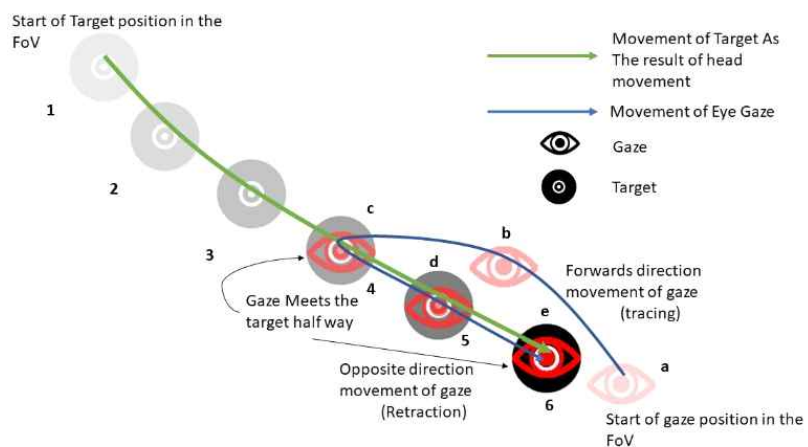
- ⑧ Compute the positions of the gaze and head movement during the simulation
- ⑨ Stop timer
- ⑩ End

- VoR의 원리를 이용한 시뮬레이션 환경은 아래좌측의 7개의 좌우 정렬된 응시점과 4개의 불규칙한 응시점을 대상으로 실험하여 확인하여 융합시스템 설계에 적용.



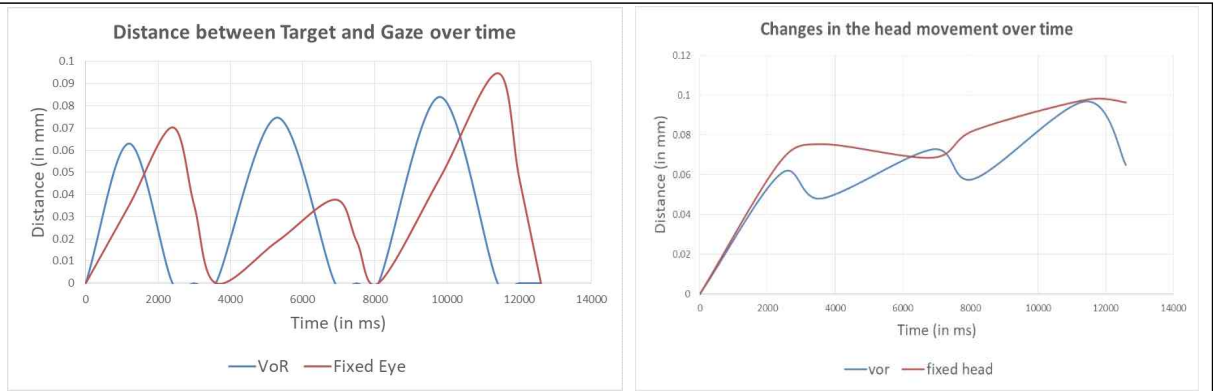
< Scenarios with random and zigzag patterns, (zigzag 7 points / random 4 points) >

- VoR의 원리를 이용한 시뮬레이션 환경에서 목표물을 응시하기 위해 머리 움직임(1~6)과 눈동자의 움직임(a~e)을 분리하여 구현하였고, 단속안구운동과 전정안 반사를 통해 목표물에 신속하게 그리고 머리움직임을 줄여 영상의 흔들림을 최소화할 수 있음.



< Movement of target and gaze during simulation >

- 아래 시뮬레이션 결과 그래프에서 분리된 안구 단독움직임 없이 항상 머리의 움직임으로만 중심시야를 확보해야 하는 단일시야의 경우보다, 중복 시야의 구현을 통해 목표물에 빠르게 접근하여 목표물을 안정적으로 응시할 수 있는 시간을 확보한 것을 알 수 있음, 또한, 우측 머리 움직임의 변화율을 통해 VoR이 구현된 중복시야에서 일정한 패턴의 변화를 감지할 수 있음.



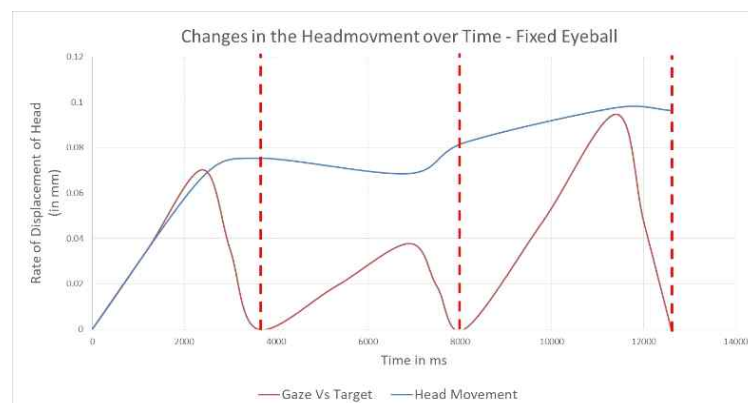
< Changes in gaze for Fixed vs. VoR cases > < Head movement for Fixed vs. VoR cases >

- VoR을 적용한 중복시야의 경우에 목표물에 도달하여 머리의 움직임의 속도가 붉은 점선 안과 같이 줄어들어 안정적으로 목표물을 응시하고 다음 목표로 이동하는 패턴으로 해당 기간에서 상대적으로 머리 움직임을 적게 하여 안정적으로 영상을 유지하게 됨.



< Effect of head movement on achieving the target via gaze in VOR case >

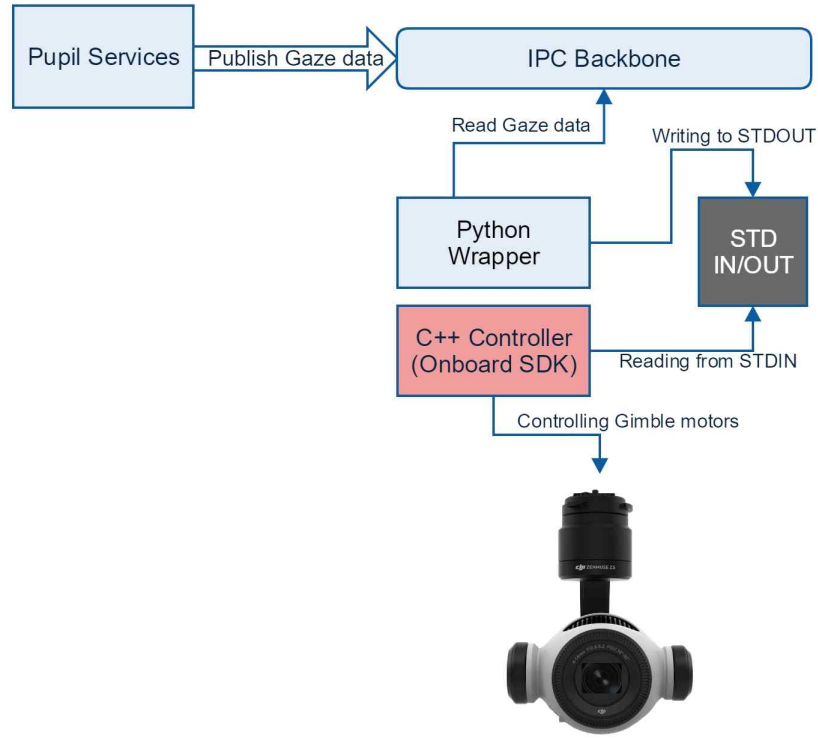
- VoR을 적용하지 않은 단일시야의 경우에 목표물에 도달하는 시간은 중복시야보다 느리고 머리의 움직임의 패턴은 응시 결과 그래프와 관계없이 목표물의 상대적인 위치에 따라 정해지며 영상의 흔들림도 목표물의 상대적인 위치에 따라 불안정하게 정해짐.



< Effect of head movement on achieving the target via gaze in Fixed eyeball case >

(4) Eye-Tracking과 Gimbal Camera를 통한 Airborne FPV

- VoR을 포함하여 계산된 정규화된 응시점 좌표는 먼저 Gimbal 카메라를 위한 가상좌표계로 매핑된 이후에 아래 그림과 같이 Onboard SDK를 통한 제어기를 통해 안정적이고 전시적인 UAV의 Airborne FPV 를 구현하는데 사용됨.



< Data flow diagram for converting gaze data into gimbal directions >

- Eye Tracker에서 얻어지는 응시점 좌표는 2D 정규 좌표계이고, Gimbal camera 모터는 $pitch_{angle}$ 과 yaw_{angle} 을 필요로 하므로, 2D 좌표계는 다음 식에 의해 카메라 각도로 변환됨.

$$S = \frac{(Max_r - Min_r)}{(Max_{gN} - Min_{gN})}$$

$$(pitch_{angle}, yaw_{angle}) = Min_r + S \times (gN - Mid_{gN})$$

where, $S = slope$

Max_r, Min_r : max and min restricted movement of the gimbal camera

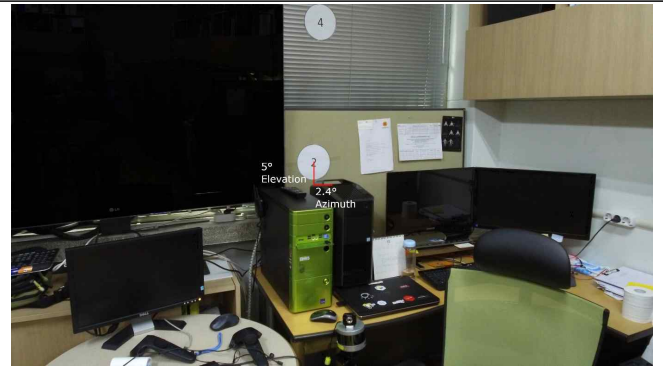
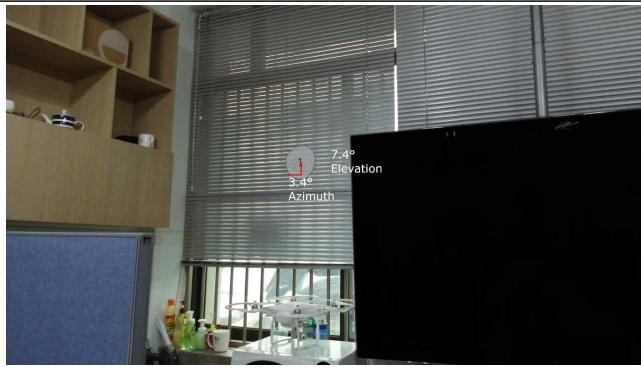
Min_{gN}, Max_{gN} : max and min values of gaze normal

gN : current gaze normal

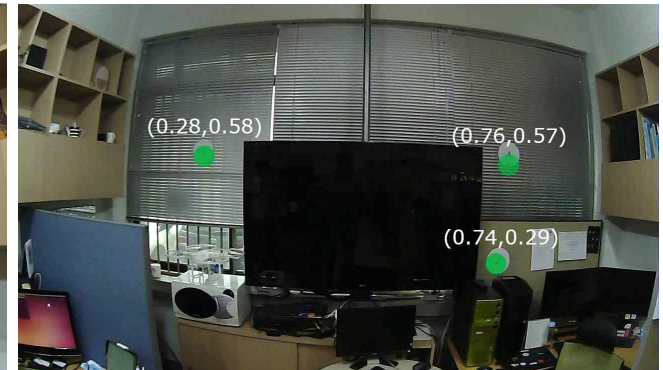
$$Mid_{gN} = \frac{(Max_{gN} + Min_{gN})}{2}$$

- Eye Gaze를 통한 Camera Movement 조종 절차는 다음과 같음.
 - Retrieve **GazeNorms**(x, y) data from the IPC backbone messaging bus of the Pupil service.
 - Map the retrieved **GazeNorms**(x, y) using a mapping function as described in equation 1 and 2 such that

$$(pitch_{angle}, yaw_{angle}) = f(GazeNorms(x, y))$$
 - get the current position of the gimble angle. (yaw and pitch)
 - If $(|current\ gimble\ pitch - pitch_{angle}| < 10^\circ \text{ OR } |current\ gimble\ yaw - yaw_{angle}| < 5^\circ)$ go to step ⑥ else go to step ⑤
 - move the gimble to new angle, $(pitch_{angle}, yaw_{angle})$
 - Go to step 1 until end of session



< Elevation and Azimuth of the gimble camera view of the target at -10° pitch & -15° yaw, 7° pitch and 32° yaw >



< Elevation and Azimuth of the gimble camera view at -13° pitch & 32° yaw and the normalized gaze data >

- Pupil Eye Tracker 응시점 데이터를 기반으로 하는 3개의 Gimbal 카메라의 움직임을 추적하고, 추적이 완료된 상태(목표점이 Gimbal 카메라의 FoV의 중심과 일치)에서 응시점과의 차이를 비교하여 아래 표에 나타냄.

Target Points	1	2	4
Normalized Gaze Data	(0.28, 0.58)	(0.74, 0.29)	(0.76, 0.57)
Pitch & Yaw of Gimbal Camera	$(-10^\circ, -15^\circ)$	$(7^\circ, 32^\circ)$	$(-13^\circ, 32^\circ)$
Elevation Error from the center of FoV	7.4°	5°	5.1°
Azimuth Error from the center of FoV	3.4°	2.4°	1.4°

- Gimbal 카메라의 초점 거리가 35mm이고, Eye tracker의 초점거리는 22mm로 다른 만큼 아래 그림에서 같은 목표점을 응시하고 있을 때 FoV는 비슷하지만 초점 거리의 차이만큼 서로 다른 수평, 수직 시야각을 인지할 수 있음.



< Field of View - Target 1 >



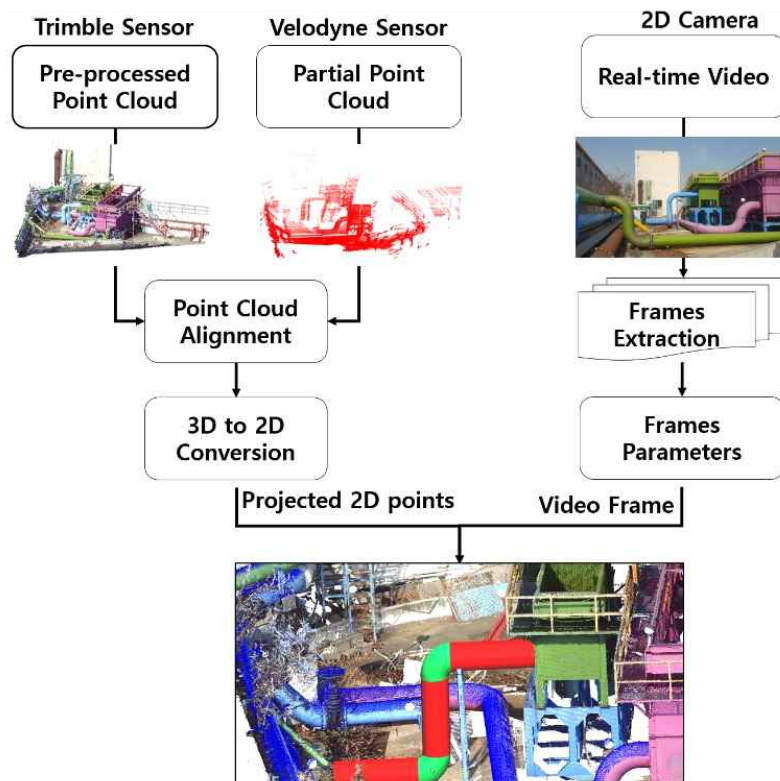
< Field of View - Target 2 >

< Field of View - Target 4 >

나. 방향에 따른 형상기술자 활용 및 UAV 스캔 위치 추정기술 개발

(1) 3D 포인트 클라우드 정합

- 사전 정합된 3D 포인트 클라우드와 실시간 취득되는 부분 포인트 클라우드와 정합이 이루어진 이후에 부분 포인트 클라우드를 취득할 때 얻어진 실시간 비디오 프레임과 정합이 단계별로 이루어진다. 제안하는 방법은 2번의 스캐닝 프로세스와 1회 실시간 스캐닝과 동시에 이루어지는 비디오 프레임을 이용.

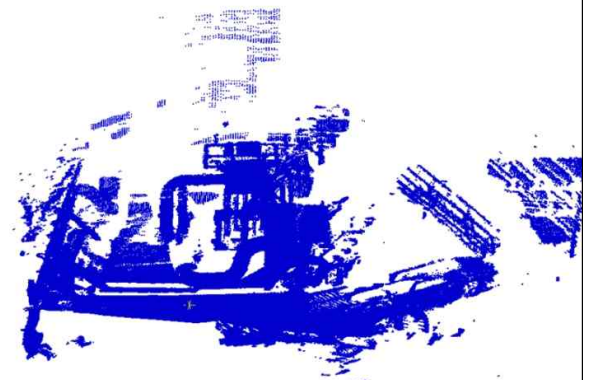


< Overview of the proposed point cloud alignment technique >



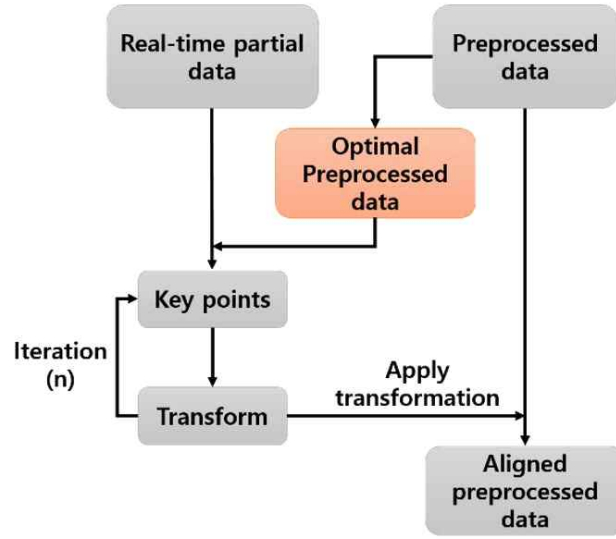
< Two different 3D scanning hardware setups at the KICT water treatment plant >

- 상용 Trimble TX3 3D^[8] 레이저 스캐너가 플랜트 현장의 파이프라인의 3D 포인트 클라우드를 취득하는데 사용되었고(위 좌측 행) 결과는 아래 좌측 그림과 같이 파이프라인과 함께 주변 환경까지 취득된 것을 알 수 있다. 아래 우측 그림은 Velodyne^[9] PUCK VLP-16으로 취득한(위 우측 행) 부분적인 3D 포인트 클라우드로 실시간 취득되어 비디오 영상과 함께 사전 처리된 플랜트 현장과 정합하여 가상 리트로피팅을 진행함.



< Preprocessed 3D point cloud and partial 3D point cloud >

- 전 방향에서 취득한 각 구간의 포인트 클라우드 데이터를 하나의 완전한 3D 포인트 클라우드 데이터로 만드는 것을 정합이라고 하고 이는 따로 취득된 데이터를 하나의 공통적인 전역 좌표계에 배치하는 것을 의미한다. 제안하는 정합은 사전 정합 처리된 포인트 클라우드와 별도로 취득되는 부분 포인트 클라우드를 G-ICP알고리즘^[10]을 이용하여 아래 블록 다이어그램 절차와 같이 중첩(정합)하는 최적화 프로세스.



< Block diagram depicting the optimal alignment process >

- 사전 처리된 정합 포인트 클라우드 데이터는 부분 포인트 클라우드와 정합 전에 센서 좌표계의 차이를 조절하기 위해 해당 지역 좌표계를 변환하고 크기를 조절하여 부분 포인트 클라우드의 전역좌표계로 매핑을 다음 식으로 오프라인에서 진행.

$$P_{global} = \left(\frac{P_{local}}{S} \right) - T$$

where, $T(x,y,z) : global\ shift, S : global\ scale, P : 3D\ point$

- 사전 처리된 포인트 클라우드와 부분 포인트 클라우드를 중첩할 때 G-ICP 알고리즘은 제공되는 모든 점을 대상으로 진행되므로, 아래 식과 같은 랜덤 샘플링^[11]으로 특징점을 찾는 해당 그룹의 포인트 밀도를 줄여 실시간으로 중첩 정합이 이루어 질 수 있도록 하였다. 아래 그림은 P_p 와 P_o 포인트 클라우드 데이터.

$$P_o = randomsample(P_p)$$

where, $P_o : optimal\ sampled\ preprocessed\ points,$

$P_p : original\ preprocessed\ points$



< Preprocessed point cloud data set and optimal subsampled point cloud >

- 사전 랜덤 샘플링된 최적의 포인트 클라우드(P_o)가 얻어진 후, 실시간 부분 포인트 클라우드(P_r)와 중첩하기 위한 변환 행렬 T 는 G-ICP 알고리즘에 의해 다음 식으로 얻어짐.

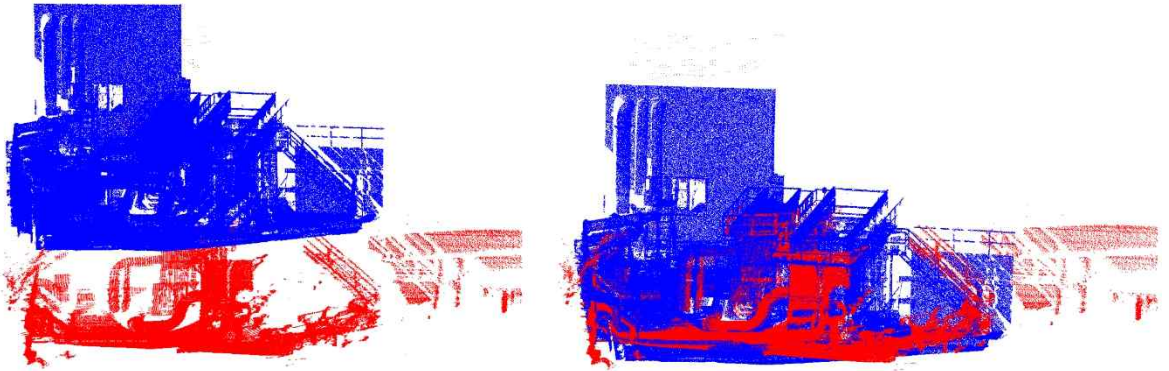
$$T = \arg \min_T \sum_{i=1}^n d_i(P_{r_i} + TP_{o_i} T^T) d_i^T$$

where, $d_i = a_i - Tb_i$ 이고 여기서 a_i, b_i 는 각각 P_o 와 P_r 의 포인트 클라우드

- G-ICP에 의해 부분 포인트 클라우드(P_r)와 랜덤 샘플(P_o)를 이용하여 변환 행렬(T_h)이 얻어지면, 사전 처리된 포인트 클라우드(P_p)에 적용해($P_t = T_h P_p$) 정렬된 포인트 클라우드(P_t)를 얻고, 다음 그림은 중첩 정렬된 포인트 클라우드의 결과를 보여줌.

$$T_h = \begin{bmatrix} c\phi c\theta & c\phi s\psi s\theta - c\psi s\phi & s\phi s\psi - c\phi s\psi s\theta & t_x \\ s\phi c\theta & c\phi c\psi + s\phi s\psi s\theta & c\psi s\phi s\theta - c\phi s\psi & t_y \\ -s\theta & s\psi c\theta & c\psi c\theta & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where, ψ, ϕ and θ 는 roll, pitch and yaw, 벡터 $\vec{t} = (t_x, t_y, t_z)$ 는 translation



< Alignment results >

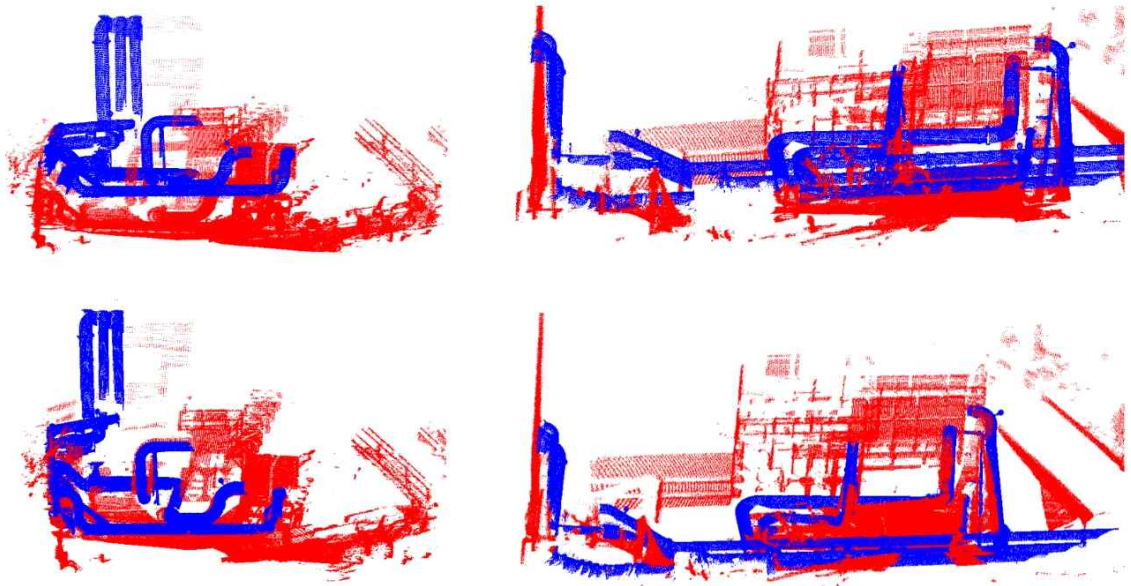
- 사전 정합처리된 포인트 클라우드를 부분 포인트 클라우드와의 Alignment를 위한 반복횟수, 실행 시간 및 Alignment Score를 View 1과 View 2 두 방향에서 G-ICP와 최적화 된 G-ICP를 적용하여 비교하였다. 다음 쪽 테이블에서와 같이 최적화된 G-ICP를 사용하여 실행 시간은 80% 이상 줄었으며 정확도는 10% 이상 증가하였고, 이를 통해 적은 반복으로 사전 정합 처리된 포인트 클라우드를 부분 포인트 클라우드와 정확하게 중첩/정합 할 수 있음.

View-1



View-2



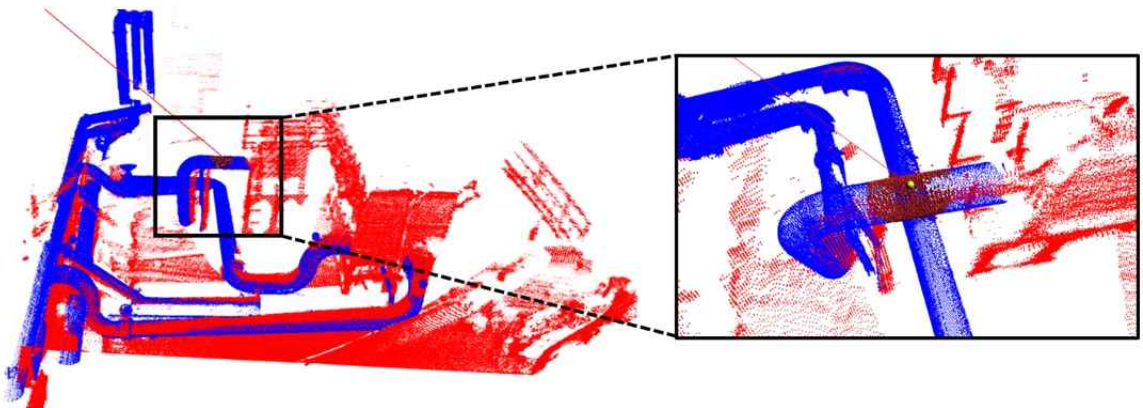


< Preprocessed point cloud alignment results of View-1 & View-2 >

Method	Iterations		Run Time		Alignment Score	
	View-1	View-2	View-1	View-2	View-1	View-2
GICP	6	5	1.04	0.630	0.023	0.037
O-GICP	8	8	0.18	0.097	0.021	0.032

(2) 3D 포인트 클라우드 곡률 정합

- 곡률을 이용하여 포인트 클라우드 파이프 곡면의 오목-볼록한 형상을 인식할 수 있다. 기반시설, 플랜트의 실린더 형상은 상대적으로 큰 곡률로 다음의 단계별 절차를 통해 대표각의 추정으로 포인트 클라우드 정합을 용이하게 진행함.
 - Seed point selection and its neighbor point estimation
 - Normal and tangential vector estimation
 - Point pairing and typical angle calculation

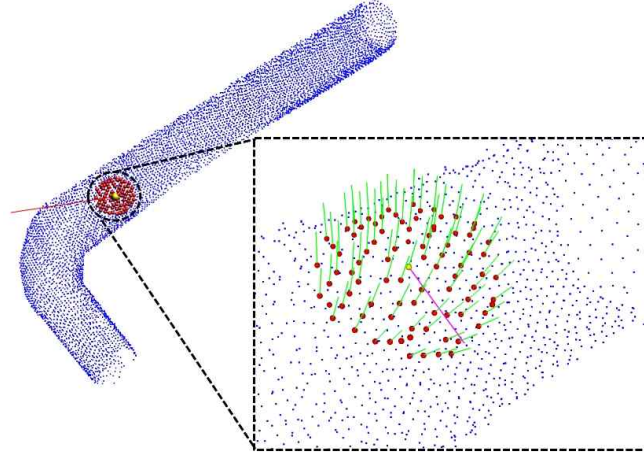


< Seed point selection and its neighbor points >

- 정합대상 각 포인트 클라우드 데이터에서 먼저 Seed 점을 선정하고, Kd-tree의 검색으로 가까운 인접 점들을 찾아 이웃 점집합을 선정하고, 시드점과 이웃점들의 법선을 추정하는데, 아래의 covariance 매트릭스를 이용하는 곡면 노말 추정방법^[12]을 사용하고, 추정된 법선은 다음 그림과 같음.

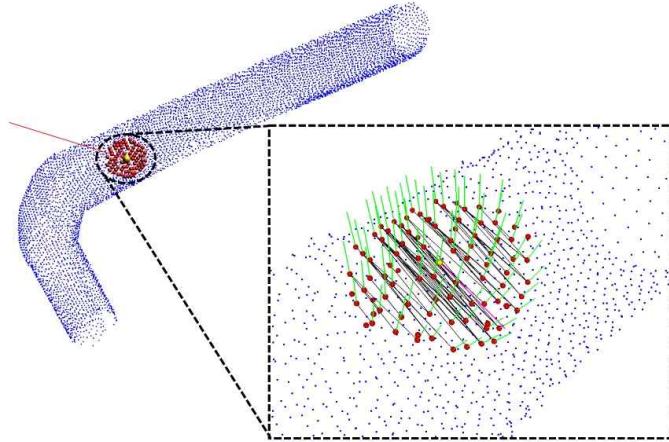
$$c = \frac{1}{k} \sum_{i=1}^k (p_i - p) \cdot (p_i - p)^T, \quad c \cdot \nu_j = \lambda_j \cdot \nu_j, \quad j \in 0, 1, 2$$

where, k is the defined number of nearest neighbors, p represents the centroid of the nearest neighbors, λ_j is the j^{th} eigenvalue, and ν_j is the j^{th} eigenvector.



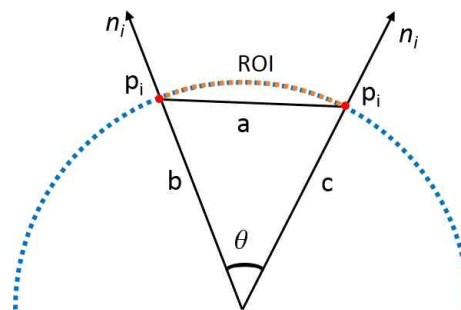
< Result of normal and tangential vector estimation >

- 추정된 이웃 점들의 법선을 기반으로, 시드점의 접선(위 그림의 자홍색 선)을 계산한다. 이 방향은 이웃 점군의 곡률변화가 최대인 방향이다. 해당 방향으로 일정 기준치의 짝 점을 찾아 연결한 것을 아래의 검정라인으로 보여줌.



< Result of point pairing using tangential vector >

- 각 짝 점에서 아래 코사인 법칙을 이용해서 대표각을 계산한다. 아래 그림의 각 변의 길이로 각도 $\cos\theta = (b^2 + c^2 - a^2)/2bc$ 를 계산 할 수 있는데, 해당 ROI에서의 선택된 포인트 클라우드 p_i 에 따라 다른 값을 얻을 수 있으므로, 짝 점 법선의 내적을 이용하여 최대 각을 구해 대표각으로 이용함.



< Typical angle calibration >

- 현재 영역에서 짝 점의 거리부분과 대표각의 관계를 저장된 bin의 결과로 Point Feature Histogram (PFH)^[13]을 적용하여 교정한다. 이 때 교정된 결과가 대표 각이고 대표각과 ROI(Region of Interest)분류에 따라 분류하고자 하는 실린더의 반경을 찾는다. 아래는 각기 다른 파이프의 ROI별 대표 각을 분류한 테이블임.

ROI(cm)	Typical angle range			
	30 cm pipe	25 cm pipe	18 cm pipe	15 cm pipe
10	7°~10°	11°~13°	16°~28°	18°~25°
20	16°~19°	20°~25°	30°~38°	40°~54°
30	31°~37°	37°~41°	40°~59°	-

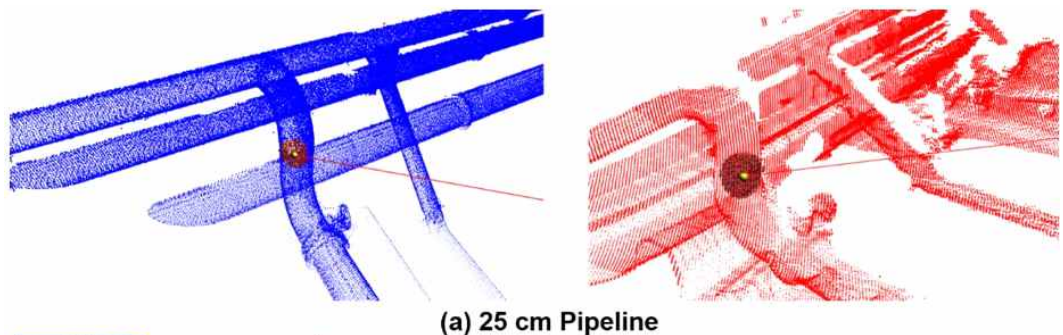
< Typical angle range for pipeline classification with different ROIs >

- 위 테이블과 같이 대표각의 범위는 ROI에 따라 변한다. 30cm, 25cm, 18cm의 파이프의 경우 ROI가 30cm를 넘으면 법선의 예측이나 짝 점의 계산이 불확실해진다. 반지름의 크기보다 큰 ROI의 경우에 해당 실린더 위의 점은 물론 인접 오브젝트, 파이프의 포인트 클라우드를 이웃점으로 포함할 수 있게 되어 PFH의 적용이 무의미해진다. 15cm 실린더의 경우 대표각은 10cm나 15cm ROI에서 적절하게 계산될 수 있음.

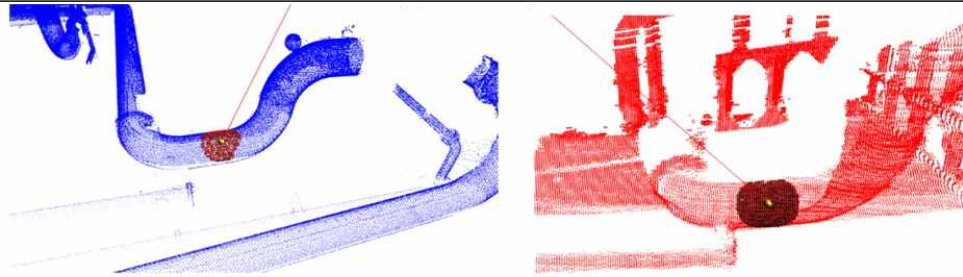
Pipeline	Actual (cm)	Preprocessed (cm)	Partial (cm)	Matching (%)
그림 (a)	25	25	25	100
그림 (b)	30	30	30	100
그림 (c)	15	15	15	100

< Result analysis for pipeline matching >

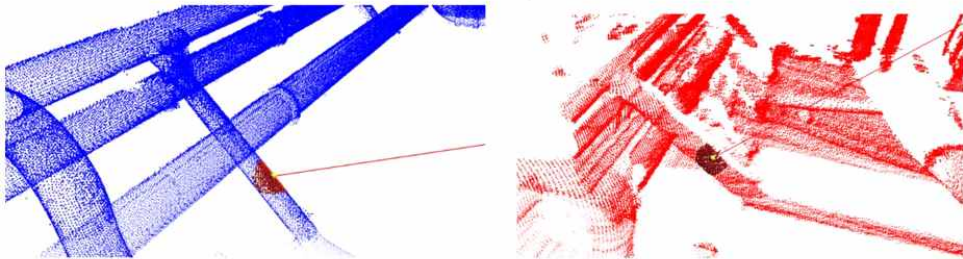
- 건설기술연구원의 수처리시설 파이프의 포인트 클라우드 데이터를 이용하여 실험을 진행하였다. 아래 그림의 좌측 행은 25cm, 30cm, 15cm 파이프의 사전 정합한 포인트 클라우드 데이터이며, 우측 행은 동일한 파이프의 시점별 부분 포인트 클라우드 데이터이다. 파이프 라인 매칭은 위 표와 같이 정확하게 같은 결과를 보여줌.



- 파이프 라인 매칭은 위 25cm 파이프와 다음의 30cm, 15cm 파이프에서 같은 대표 각의 결과에 따른 동일한 반경을 정확하게 추정하였다. 사전 정합 및 부분 포인트 클라우드 그리고 배경 비디오의 정합에 의한 주변과의 사이즈 매칭이 이루어지면 대상 기반시설 및 플랜트의 곡률 매칭을 달성.



(b) 30 cm Pipeline

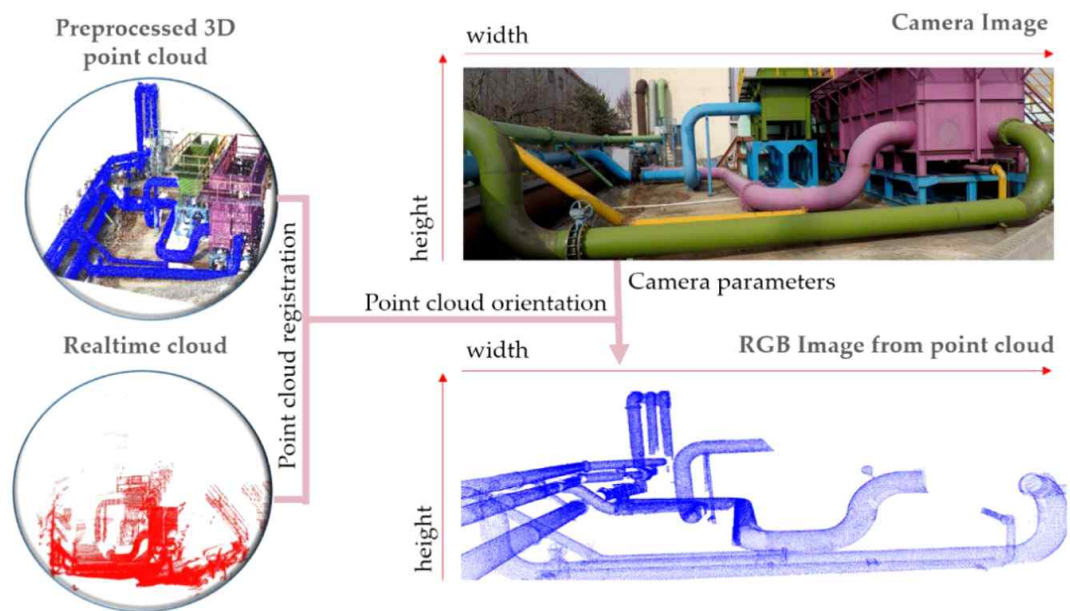


(c) 15 cm Pipeline

< Pipeline geometry (dimension) matching using curvature feature >

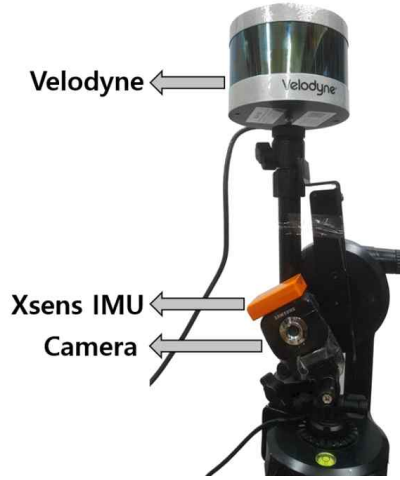
(3) 3D 포인트 클라우드 비디오 정합

- 사전에 정합처리된 포인트 클라우드는 2D 비디오와 Frame-to-frame 정합방법에 의해 자동 중첩된다. Frame-to-frame 정합 전에 이미지 평면에서 포인트 클라우드의 위치와 회전값이 계산된 후에 3D 포인트 클라우드에서 2D 포인트 클라우드 RGB 이미지가 추출되고 2D 비디오 프레임과 중첩되는 일련의 과정은 아래 그림과 같음.



< RGB image extraction from the preprocessed point cloud >

- 카메라로 얻은 영상과 LiDAR로 스캐닝하는 포인트 클라우드는 서로 다른 특성의 데이터로 픽셀단위의 관련유사성을 설정하기 위해 LiDAR 센서와 카메라의 캘리브레이션이 진행되어야 한다. 일반적으로 각 영상에 마커를 사용하여 형상차이를 검출하여^[14] 적용하는데, 본 과제에서는 Velodyne 3D LiDAR 센서와 카메라가 아래 그림과 같이 MTw Xsens IMU^[15]와 함께 연결되어 상대적인 회전값을 추출함.

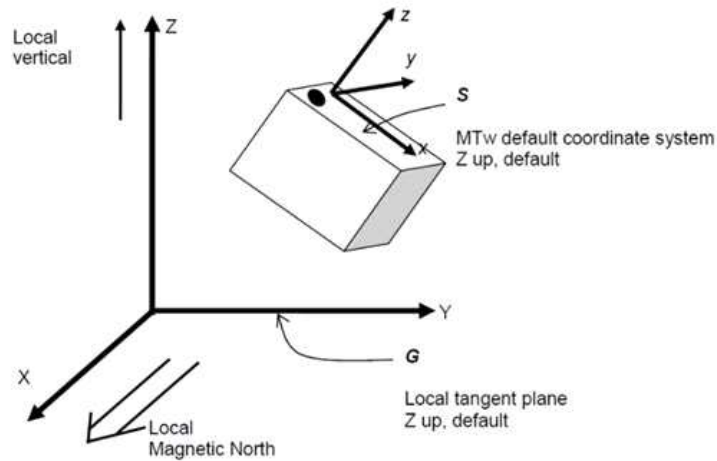


< Velodyne LiDAR, IMU and Camera setup >

- 스캐닝 플랫폼의 회전값인 아래 그림의 Xsens IMU 센서의 단위 쿼터니언 Q_{mat} 을 이용하여 아래 식으로 스캐너 플랫폼에 대한 사전정합 포인트 클라우드의 최초 정렬 C_{P_align} 를 구할 수 있다. 여기서 P_t 는 정렬 전 포인트 클라우드, P_{mat} 은 초점 길이 f 의 카메라 투영 매트릭스, C_{mat} 은 캘리브레이션 매트릭스 임.

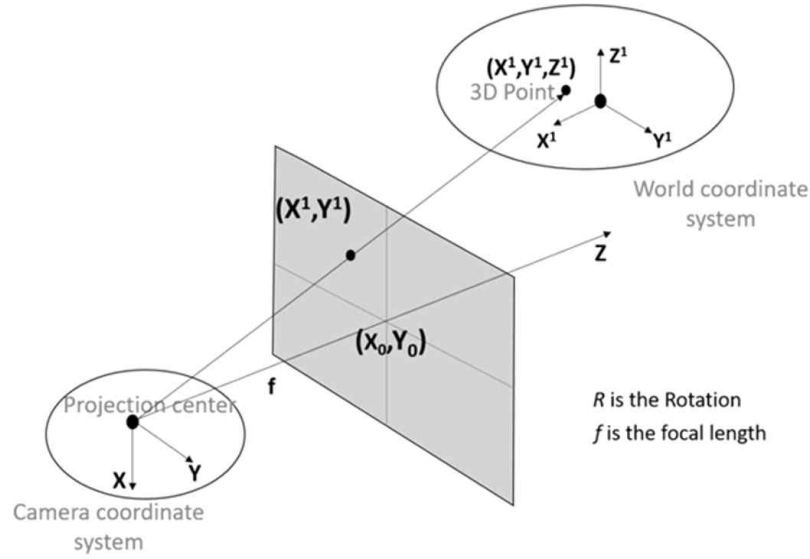
$$C_{P_align} = P_{mat} C_{mat} Q_{mat} P_t$$

$$\text{where, } P_{mat} = \begin{bmatrix} f & 0 & o_x & 0 \\ 0 & f & o_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, C_{mat} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



< Coordinate frame of IMU sensor >

- 사전정합 3D 포인트 클라우드는 아래 그림과 같이 월드 좌표계에서 정의되어 있고, 해당 RGB 이미지를 얻기 위해서는 해당 3D 포인트를 Collinearity 식^[16]을 이용하여 카메라 파라미터로 정의되는 이미지 평면으로 투영해야 한다. 카메라렌즈의 중심이 카메라 좌표계의 중심이고 Z 방향이 투영면 XY평면과 수직이며 LiDAR 센서와 카메라 센서를 관심물체로부터 동일한 거리에 위치시키는데 이는 물체로부터 LiDAR의 거리가 카메라 좌표계의 중심으로부터 초점거리와 같다는 것을 의미.



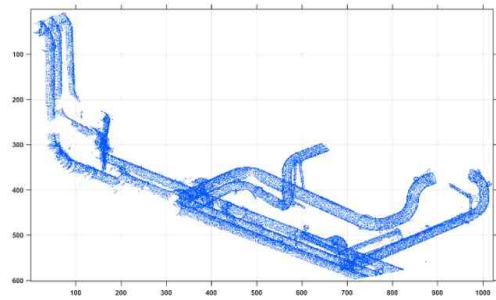
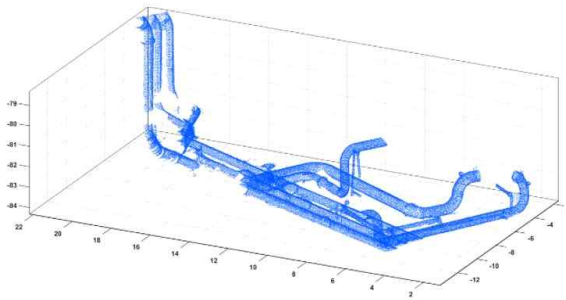
< 3D point cloud projection on the image plane defined by the camera's coordinate system >

- 다음 좌측 그림의 LiDAR 센서로부터 취득되고 사전 정합된 3D 포인트 클라우드는 아래 식으로 카메라 파라미터로 정의되는 이미지 평면에 다음 우측 그림과 같이 투영됨.

$$x = -f \frac{a_1(X - X_0) + a_2(Y - Y_0) + a_3(Z - Z_0)}{a_7(X - X_0) + a_8(Y - Y_0) + a_9(Z - Z_0)}$$

$$y = -f \frac{a_4(X - X_0) + a_5(Y - Y_0) + a_6(Z - Z_0)}{a_7(X - X_0) + a_8(Y - Y_0) + a_9(Z - Z_0)}$$

$$\text{where, } \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



< 3D point cloud to 2D image plane projection: (left) 3D point cloud (right) extracted 2D image representation >

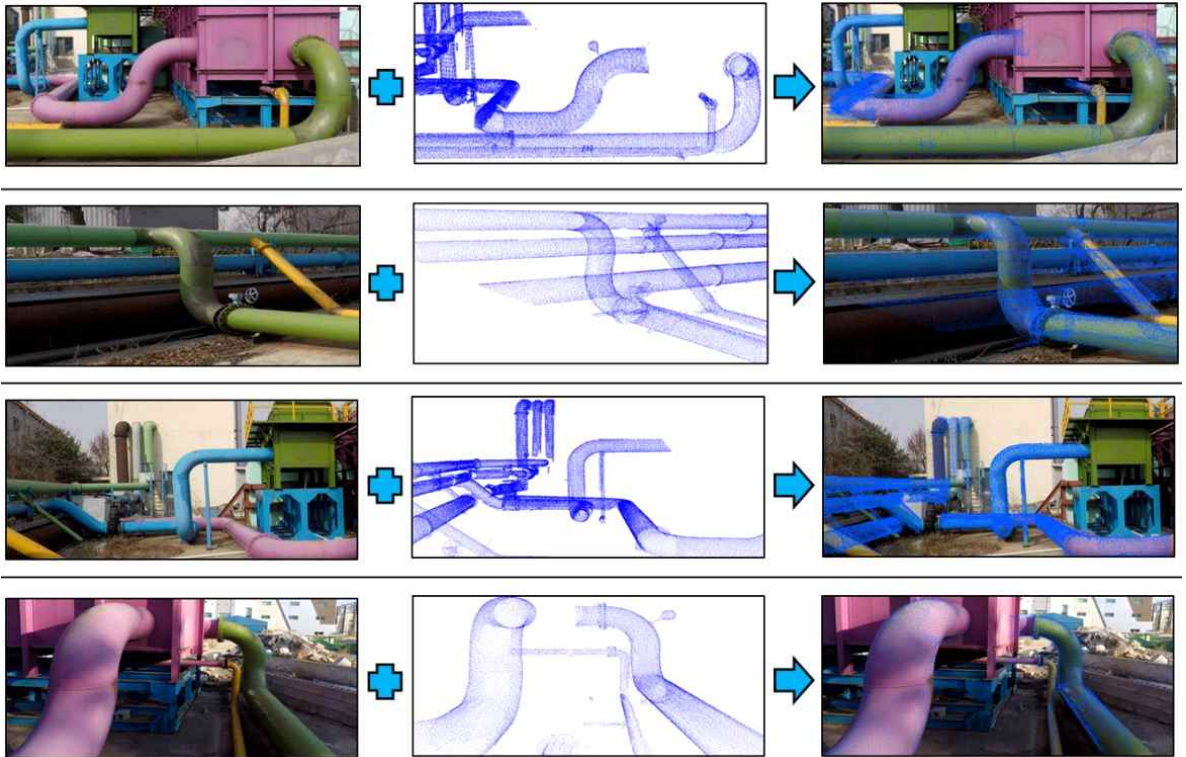
- 포인트 클라우드 영상의 색상 강도(Intensity) 값은 아래 식으로 카메라 프레임에 맞게 수정된다. 여기서 w, h 는 각각 영상의 가로, 세로 길이 임.

$$\text{if } \sum_{i=0}^w \sum_{j=0}^h C_I(i, j)[r, g, b] = I[r : 0, g : 0, b : 255]$$

- 다음에 카메라 비디오 프레임의 픽셀 Intensity $F_I(i, j)$ 는 위 식에서 구한 해당 대응점의 포인트 클라우드 영상 픽셀 Intensity $C_I(i, j)$ 로 아래 식으로 교체.

$$F_I[r, g, b] = C_I[r, g, b]$$

- 아래 그림의 첫 번째 행은 캡처한 비디오 프레임이고, 두 번째 행은 사전정합 3D 포인트 클라우드로부터 추출해 낸 2D RGB 이미지, 그리고 카메라 영상에 포인트 클라우드 이미지를 중첩한 것이 마지막 행의 그림.

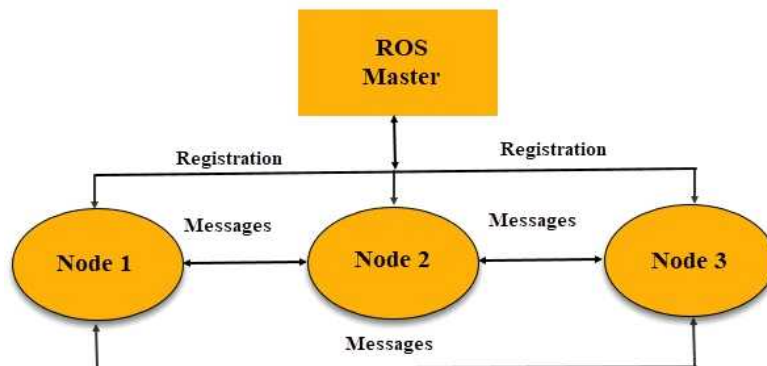


< Point cloud image alignment on the camera image >

다. 라이다장착 UAV 기반 실시간 가상 리트로피팅

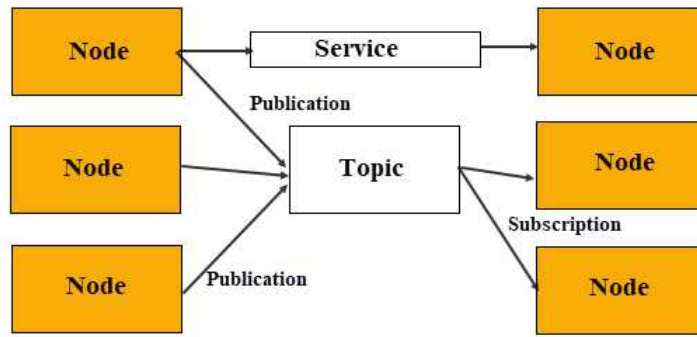
(1) UAV-LiDAR와 호스트 스테이션간의 데이터 통신

- ROS(Robot Operating System)^[17]은 BSD-licensed의 PC에서 로봇 요소를 제어할 수 있는 OS로 hardware abstraction, device drivers, inter-process communications, distributed package management등 로봇 응용을 개발하는데 필요한 유용한 라이브러리를 제공한다. 아래 그림과 같이 ROS 마스터가 허용한 Node로 불리는 여러 개의 프로세스가 peer-to-peer network로 연결됨.



< Architecture of the ROS >

- ROS는 topic과 service라는 두 종류의 messaging mechanism을 제공하는데, topic은 publisher/subscriber간 비동기적 메시지이고 동일한 topic을 여러 publisher/subscriber가 공유할 수 있는 반면, service는 request/response에 기반을 둔 동기화된 하나의 provider가 하나의 request를 담당하는 message 임.



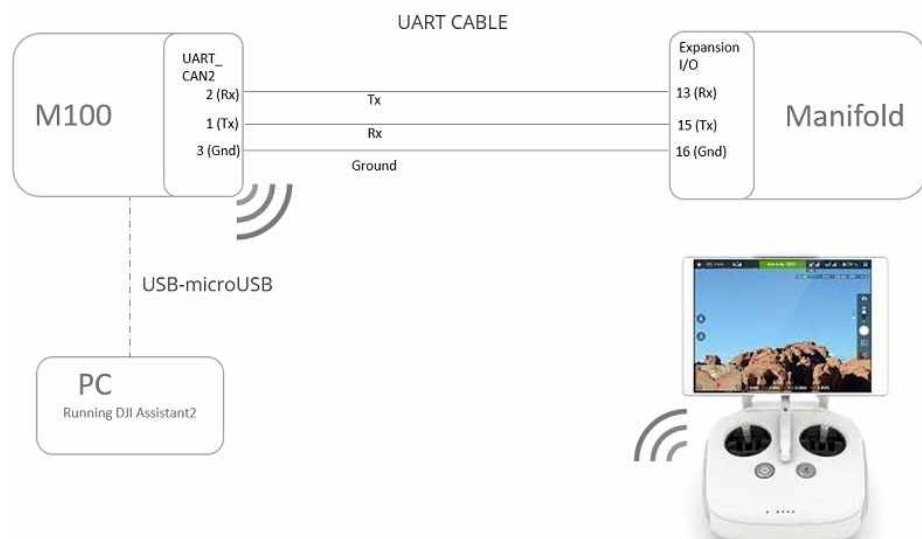
< Two way of node communication of ROS >

- Velodyne LiDAR의 Puck LITE^[18]는 VLP-16의 UAV를 위한 저전력 경량(590g) 레이저 스캐너로 20Hz의 속도로 100m 이내의 3D 환경을 360° 수평 FOV와 ±15° 수직 FOV, 16채널 초당 300,000포인트로 생성하는 multi-beam 3D LiDAR임.



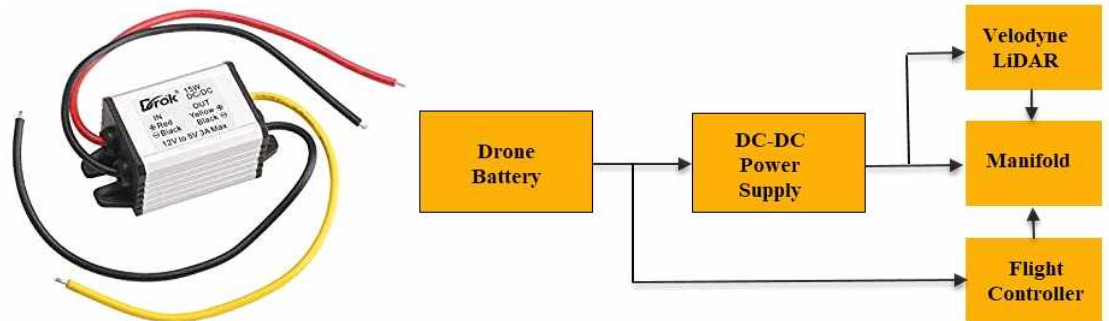
< DJI Manifold와 Velodyne LiDAR Puck Lite >

- DJI Manifold는 DirectX 11, OpenGL 4.4, NVIDIA CUDA 지원 그래픽기능을 포함하는 미니 PC로 컴퓨터 비전, 딥러닝 등의 인공지능 응용과 함께 사용할 수 있어 주변 환경을 촬영, 녹화하는 기본 기능 외에 물체를 인식하고 동시에 반응할 수 있으며 본 과제의 UAV, Sensor, Camera, Host Computer, Visualization System과 통합 사용됨.
- Velodyne Puck LITE와 On board 컴퓨터인 DJI Manifold는 DJI Matrix M100 UAV에 통합되며 Manifold와 M100은 데이터 형식과 전송속도가 설정 가능한 비동기 시리얼 UART 인터페이스로 연결됨.



< Hardware connection between M100 and Manifold >

- 아래 특성의 DROK DC converter regulator가 UAV battery(TB47D)로부터 전원공급을 받아 Manifold와 LiDAR에 공급함.
 - ✓ Properties: non-isolated step-down power supply module
 - ✓ Input voltage: 8-35V
 - ✓ Output voltage: 1.5-24V adjustable
 - ✓ Output current: 5A MAX, recommended: below 5V, 4A long-term; 6-9V, 3A; 10V-15V, 2.5A



< DROK DC Volt convertor regulator, UAV power supply diagram >

- 다음에 M100과 함께 비행할 요소로 LiDAR Puck Lite(590g), Manifold(200g), DROK regulator(9g), Wifi Dongle(80g), Propeller Guard(200g)의 분해도와 총 1,079g Payload를 탑재한 비행체를 보여준다. Firmware v.1.3.1.10의 M100의 경우 1000g Payload에서 약 13분간의 안정적인 hovering으로 비디오를 포함 리트로피팅을 위한 포인트 클라우드를 취득할 수 있음.
- KICT 테스트 리트로피팅 (2019.01.18.) 풍속 2 m/s, UAV 최대 rise speed 5 m/s, 최대 falling speed 4 m/s, 최대 wind resistance 10 m/s, 고도 4~5m flight time, 4~5min.



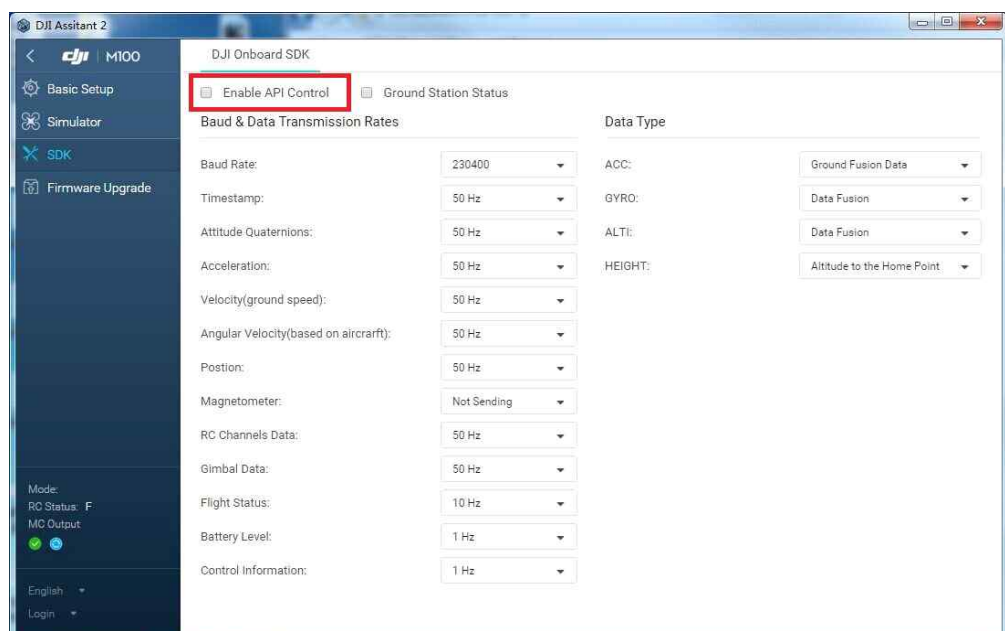
< Fully equipped Matrix 100 UAV with 1,079g payload >

- DJI Manifold에 Ubuntu 14.04 LTS ARM과 함께 ROS indigo distribution를 설치하고^[19], On-board 컴퓨터에서 취득한 PCD 파일을 원격 호스트 컴퓨터에서 동시에 사용하기 위해 Intel 7260 HMW 무선 아답터를 설치하고^[20], 아래의 절차로 DJI onboard SDK^[21]를 설치하여 ROS message와 service를 이용할 수 있게 함.

1. Install DJI SDK Core Library
2. Before **using** the DJI SDK ROS package, we first need to install the DJI SDK core library.
- 3.
4. Download the necessary files from git repo and access the directory:
5. `$ cd ~`
6. `$ git clone https://github.com/dji-sdk/Onboard-SDK`
7. `$ cd (dji_sdk_directory)`
8. Create a build directory and cd into it:
9. `$ mkdir build`
10. `$ cd build`
11. Compile and install DJI SDK core library
12. `$ cmake ..`
13. `$ make djirosdk-core`
14. `$ sudo make install`
15. Setup the DJI SDK ROS package
16. Now that we have the core sdk library installed and ready to go, we can start setting up the DJI SDK ROS package.
- 17.
18. Download the Package
19. First we need to download and install the dji_sdk ROS package.
- 20.
21. Go to your catkin workspace.
22. `$ cd ~/your_catkin_ws/src/`
23. Download the DJI SDK ROS package from our git repository.
- 24.
25. `$ git clone https://github.com/dji-sdk/Onboard-SDK-ROS.git`
26. Run '**catkin_make**' and compile.
27. `$ cd ~/your_catkin_ws/`
28. `$ catkin_make`
29. We almost have all the software setup **for** DJI SDK ROS.

< DJI SDK Installation process > [Shell Script: DJI S/W installation on Ubuntu O/S]

- DJI SDK를 설치 후에 DJI PC Assistant 2, DJI GO App을 호스트 컴퓨터와 Tablet에 각각 설치하고 온보드 컴퓨터와 UAV, flight controller와 통신을 위해 아래와 같이 OSDK API Control이 On되어 있어야하고, RC status는 'F'로 유지.

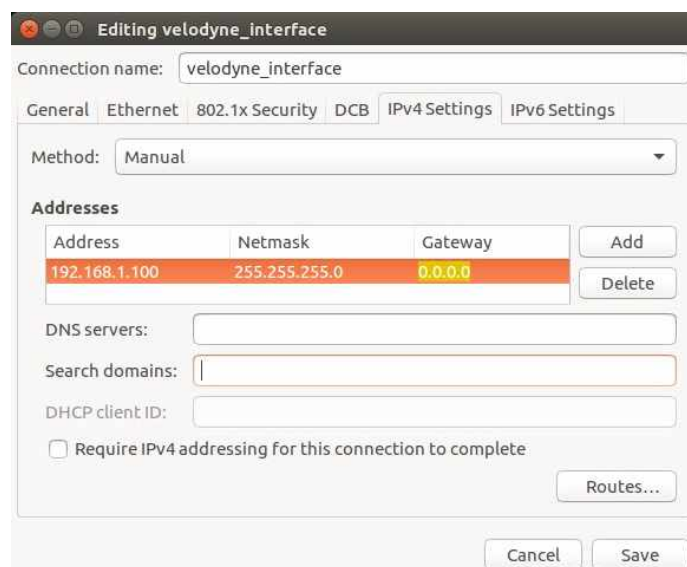


< DJI onboard SDK page to enable API Control >

- UAV와 flight controller로 H/W와 ROS가 통신을 시작할 때 App ID(1055437)와 key pair(47ef00ea3dc8f981363d8484b42cba2377b2c3f594da977b274b049c26e1fb95)가 필요하고, Manifold의 Ethernet 포트와 연결된 Velodyne LiDAR VLP 16을 동작시키기 위해 아래의 configuration 절차가 필요.

1. Configure your computer's IP address through the Gnome interface
2. Access the Gnome Menu (Super key), type "Networks Connections" then run it. Select the connection's name and click on "edit". Choose the IPV4 Settings tab and change the "Method" field to "Manual".
3. Click on "add" and set the IP address field to 192.168.1.100 ("100" can be any number except in a range between 1 and 254, except 201).
4. Set the "Netmask" to 255.255.255 and the "Gateway" to 0.0.0.0.
5. To finish it click on "save".
6. Connecting your computer to LIDAR through terminal
7. Power the LIDAR via the included adapter Connect the LIDAR to an Ethernet port on your computer. Statically assign an IP to **this** port in the 192.168.3.x range.
- 8.
9. `sudo ifconfig eth0 192.168.3.100`
10. Add a **static** route to the LIDAR's IP address. The IP address can be found on the CD **case** which was included with the LIDAR.
- 11.
12. `sudo route add 192.168.XX.YY eth0`
13. Checking the configurations
14. To check the connection open your web browser and access the following sensor's network address: 192.168.XX.YY. The following page should appear:
15. Installing ROS dependencies
- 16.
17. `sudo apt-get install ros-VERSION-velodyne`
18. Installing the VLP16 driver
19. 1. On terminal, inside your ROS workspace, find the "src" folder and execute the following command:
- 20.
21. `$ cd ~/catkin_ws/src/ && git clone https://github.com/ros-drivers/velodyne.git`
22. 2. After that, on terminal, inside your workspace, update all dependencies:
- 23.
24. `$rosdep install --from-paths src --ignore-src --rosdistro YOURDISTRO -y`
25. 3. Then build your workspace:
- 26.
27. `$ cd ~/catkin_ws/ && catkin_make`
28. 4. Now, your Velodyne package is ready to run.

< Velodyne LiDAR VLP 16 configuration > [Shell Script: Velodyne LiDAR configuration on Ubuntu O/S]



< Velodyne LiDAR VLP 16 configuration >

192.168.1.201

Apps QDoubleSpinBo Competitive Pro Mathematical fu Basic Operations Using Regular E

Velodyne® LiDAR

Sensor Model: **VLP-16 Lite** S/N: **AF32714558** MAC: **60-76-88-10-38-de**

VLP-16 USER INTERFACE

Configuration System Info Diagnostics

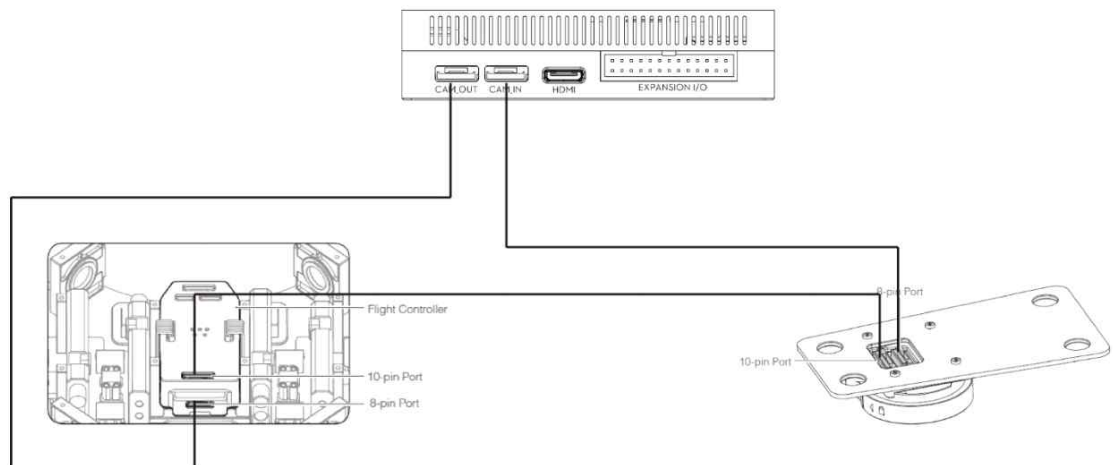
Laser: ☒ On ☐ Off
 Return Type: **Strongest**
 Motor RPM: **600**
 FOV Start: **0** End: **359**
 Phase Lock ☐ On ☒ Off Offset: **0**
 Host (Destination) IP: **255.255.255.255** Data Port: **2368** Telemetry Port: **8308**
 Network (Sensor) IP: **192.168.1.201** Mask: **255.255.255.0** Gateway: **192.168.1.1** DHCP: ☐ On ☒ Off

GPS Position: ☐ PPS: **Absent**
 Motor State: ☒ On RPM: **600** Lock: ☐ Off Phase: **3080**
 Laser State: ☒ On

Velodyne® LiDAR

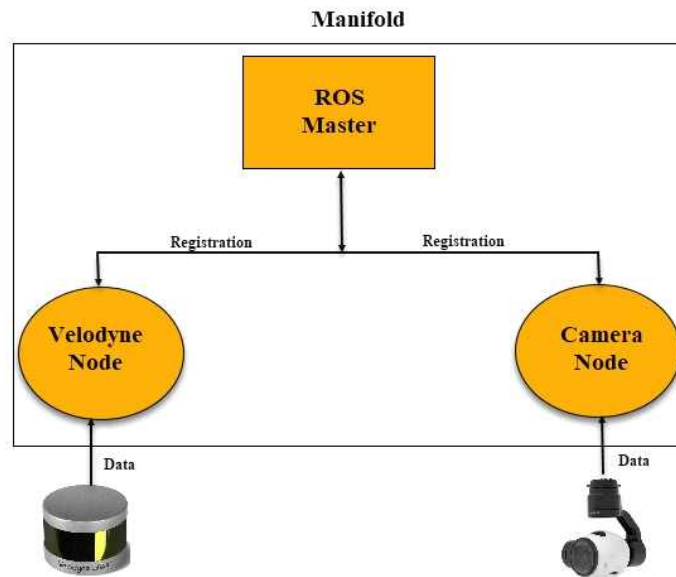
< Velodyne LiDAR VLP 16 configuration >

- DJI Zenmuse X3 gimbal camera를 사용하여 비디오 영상을 공급받기 위해서 아래 그림과 같이 카메라 8-pins 포트는 Manifold CAM_IN 포트와 연결하고, CAM_OUT은 N1 flight controller 8-pins 로 연결하여 다시 카메라의 10-pins 포트에 연결한다. 관련 라이브러리는 GStreamer, FFmpeg, libusb, libdcam.so 임.



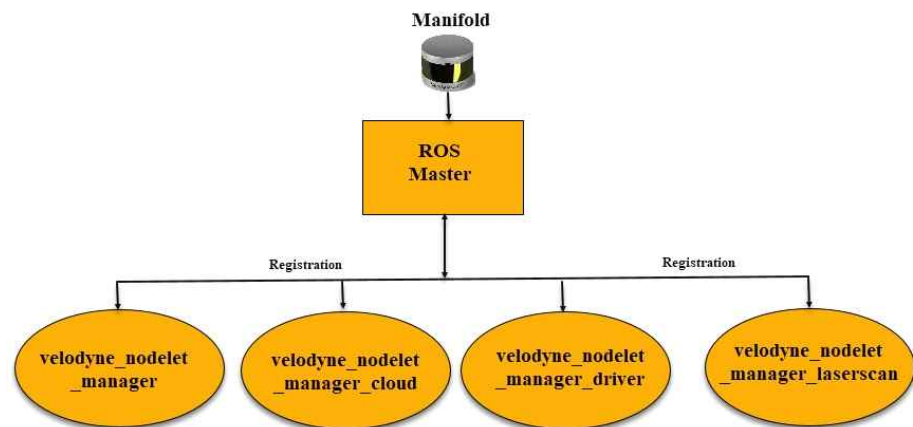
< Zenmuse X3 gimbal Camera connection setup with Manifold >

- ROS master는 다른 ROS Node를 등록하고 통신하게 하는데, 본 과제의 VeloCamData 취득 Node는 아래 그림과 같이 Velodyne LiDAR를 관장하는 Velodyne Node와 Zenmuse X3 gimbal 카메라를 관장하는 Camera Node로 나뉘고, 각 노드는 독립적으로 원격 호스트 컴퓨터의 SSH 명령으로 Manifold에서 생성.



< Architecture of VeloCamData capture >

- Velodyne PCD capture를 위한 Node는 VLP 16 3D LiDAR 데이터를 취득해 공개하는데 드라이버는 장치에 독립적인 velodyne_msgs/VelodyneScan 데이터를 만들고, velodyne_pointcloud 패키지가 이 데이터를 좀 더 편리한 sensor_msgs/PointCloud2 messages로 변환하는 nodes와 nodelets을 제공한다. Roslaunch가 XML 형태의 스크립트를 통해 아래와 같은 4개의 Node를 등록.



< ROS 4 Nodes list for Velodyne PCD capture >

```

1. <!-- -*- mode: XML -*- -->
2. <!-- run velodyne_pointcloud/CloudNodelet in a nodelet manager for a
   VLP-16 -->
3. <launch>
4. <!-- declare arguments with default values -->
5. <arg name="calibration" default="$(find
   velodyne_pointcloud)/params/VLP16db.yaml"/>
6. <arg name="device_ip" default="" />
7. <arg name="frame_id" default="velodyne" />
8. <arg name="manager" default="$(arg frame_id)_nodelet_manager" />
9. <!--<arg name="max_range" default="130.0" />-->
10. <arg name="max_range" default="5.0" />
11. <arg name="min_range" default="0.4" />
12. <arg name="pcap" default="" />
13. <arg name="port" default="2368" />
14. <arg name="read_fast" default="false" />
15. <arg name="read_once" default="false" />
16. <arg name="repeat_delay" default="0.0" />
17. <arg name="rpm" default="300.0" />
18. <arg name="cut_angle" default="-0.01" />
19. <arg name="laserscan_ring" default="-1" />
20. <arg name="laserscan_resolution" default="0.007" />

```

```

21. <!-- start nodelet manager and driver nodelets -->
22. <include file="$(find
    velodyne_driver)/launch/nodelet_manager.launch">
23.   <arg name="device_ip" value="$(arg device_ip)"/>
24.   <arg name="frame_id" value="$(arg frame_id)"/>
25.   <arg name="manager" value="$(arg manager)" />
26.   <arg name="model" value="VLP16"/>
27.   <arg name="pcap" value="$(arg pcap)"/>
28.   <arg name="port" value="$(arg port)"/>
29.   <arg name="read_fast" value="$(arg read_fast)"/>
30.   <arg name="read_once" value="$(arg read_once)"/>
31.   <arg name="repeat_delay" value="$(arg repeat_delay)"/>
32.   <arg name="rpm" value="$(arg rpm)"/>
33.   <arg name="cut_angle" value="$(arg cut_angle)"/>
34. </include>
35. <!-- start cloud nodelet -->
36. <include file="$(find
    velodyne_pointcloud)/launch/cloud_nodelet.launch">
37.   <arg name="calibration" value="$(arg calibration)"/>
38.   <arg name="manager" value="$(arg manager)" />
39.   <arg name="max_range" value="$(arg max_range)"/>
40.   <arg name="min_range" value="$(arg min_range)"/>
41. </include>
42. <!-- start laserscan nodelet -->
43. <include file="$(find
    velodyne_pointcloud)/launch/laserscan_nodelet.launch">
44.   <arg name="manager" value="$(arg manager)" />
45.   <arg name="ring" value="$(arg laserscan_ring)"/>
46.   <arg name="resolution" value="$(arg laserscan_resolution)"/>
47. </include>
48. </launch>

```

[XML Script: Initial parameters for Velodyne LiDAR]

- 본 과제를 위한 Node 환경 중 min_range는 스캐닝을 시작하는 범위이고, max_range는 스캐닝의 최대 한계이다. 스캐너 회전은 최소 300 rpm이 중밀도 포인트 클라우드를 얻을 수 있으며 아래는 환경변수의 값을 나타낸 화면.

```

/home/ubuntu/pavan/catkin_ws/src/velodyne-master/velodyne_pointcloud/launch/VLP16_points.launch http://localhost:11
File Edit View Search Terminal Help

SUMMARY
=====

PARAMETERS
* /roscdistro: indigo
* /rosversion: 1.11.21
* /velodyne_nodelet_manager_cloud/calibration: /home/ubuntu/pava...
* /velodyne_nodelet_manager_cloud/max_range: 5.0
* /velodyne_nodelet_manager_cloud/min_range: 0.4
* /velodyne_nodelet_manager_driver/cut_angle: -0.01
* /velodyne_nodelet_manager_driver/device_ip:
* /velodyne_nodelet_manager_driver/frame_id: velodyne
* /velodyne_nodelet_manager_driver/model: VLP16
* /velodyne_nodelet_manager_driver/pcap:
* /velodyne_nodelet_manager_driver/port: 2368
* /velodyne_nodelet_manager_driver/read_fast: False
* /velodyne_nodelet_manager_driver/read_once: False
* /velodyne_nodelet_manager_driver/repeat_delay: 0.0
* /velodyne_nodelet_manager_driver/rpm: 300.0
* /velodyne_nodelet_manager_laserscan/resolution: 0.007
* /velodyne_nodelet_manager_laserscan/ring: -1

NODES
/
  velodyne_nodelet_manager (nodelet/nodelet)
  velodyne_nodelet_manager_cloud (nodelet/nodelet)
  velodyne_nodelet_manager_driver (nodelet/nodelet)
  velodyne_nodelet_manager_laserscan (nodelet/nodelet)

auto-starting new master
process[master]: started with pid [2487]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to ee351604-cc2e-11e8-a18c-7c5cf8e95d36
process[rosout-1]: started with pid [2500]
started core service [/rosout]
process[velodyne_nodelet_manager-2]: started with pid [2503]
process[velodyne_nodelet_manager_driver-3]: started with pid [2504]
process[velodyne_nodelet_manager_cloud-4]: started with pid [2505]
process[velodyne_nodelet_manager_laserscan-5]: started with pid [2506]

```

< Parameters for the Velodyne node in roslaunch program >

- ROS의 topic은 Node간 message를 publish/subscribe하기 위한 익명의 데이터 버스로 Node는 topic이 누구와 통신하는 데에는 관심이 없고, 어떤 topic이 관련 message를 publish/subscribe 하는 것만 인지가 필요하다. topic을 통해 정보의 생산과 소비를 분리할 수 있고 Velodyne LiDAR의 topic 리스트는 다음과 같음.

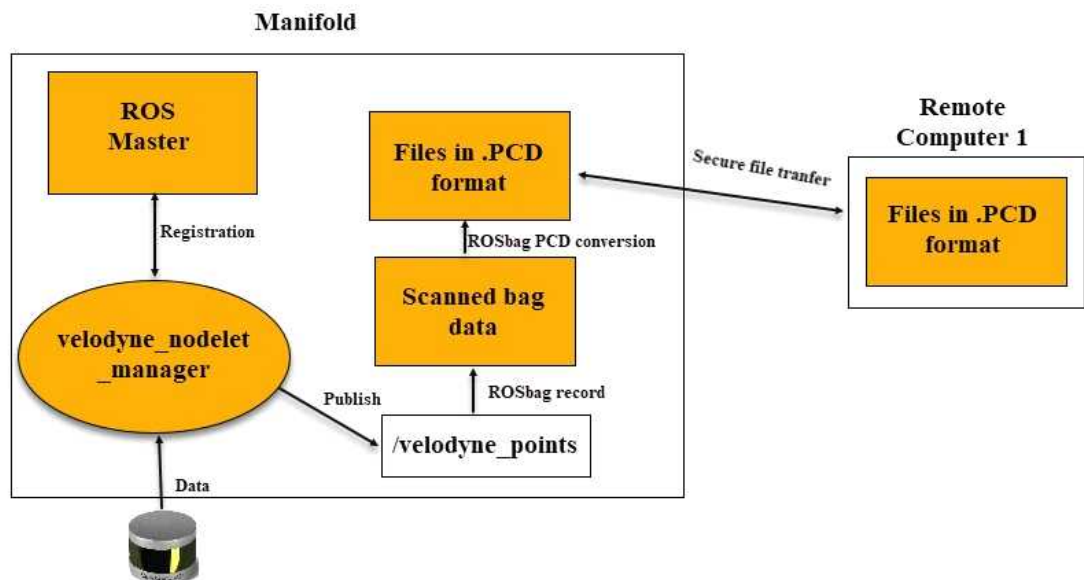
```

root@tegra-ubuntu: ~
File Edit View Search Terminal Help
root@tegra-ubuntu:~# rostopic list
/diagnostics
/rosout
/rosout_agg
/scan
/velodyne_nodelet_manager/bond
/velodyne_nodelet_manager_cloud/parameter_descriptions
/velodyne_nodelet_manager_cloud/parameter_updates
/velodyne_nodelet_manager_driver/parameter_descriptions
/velodyne_nodelet_manager_driver/parameter_updates
/velodyne_nodelet_manager_laserscan/parameter_descriptions
/velodyne_nodelet_manager_laserscan/parameter_updates
/velodyne_packets
/velodyne_points
root@tegra-ubuntu:~#

```

< ROS topics list of Velodyne LiDAR >

- Velodyne LiDAR 스캐닝 데이터를 /velodyne_points topic에 publish/subscribe 하게 되는데, 한 바퀴 데이터를 위한 topic은 /velodyne_packet이다. Node 간 통신을 위한 /Velodyne_data message는 3D 스캐닝 데이터로 ROSbag(.bag) 파일이며 시각화를 위한 형태가 아니기 때문에 ROSbagPCD 변환을 통해 .PCD 파일로 변환되고, 각 프레임별 .PCD 파일은 합쳐져서 한 방향의 .PCD 스캐닝 데이터로 저장되어 원격 호스트 컴퓨터로 옮겨져 정합절차에 사용됨.



< Architecture of Manifold and Remote Computer 1 remote communication >

- M100의 Zenmuse X3 짐벌 카메라를 이용하여 Velodyne LiDAR 스캐닝과 동시에 전경을 녹화하기 위해서는 Velodyne 노드 등록 전·후에 카메라 캡처 노드 역시 등록되어야 한다. gray_or_rgb 변수는 컬러 비디오 녹화여부 이고, to-mobile이 켜지면 DJI Go와 연결되거나 Mobile SDK 응용을 사용하기 위한 것이며 아래는 ROSlanch로 카메라 관련 Node를 등록한 후의 파라미터 화면.

```

/home/ubuntu/pavan/catkin_dji/src/Onboard-SDK-ROS-3.2/dji_sdk_read_cam/launch/manifold_c
File Edit View Search Terminal Help

started roslaunch server http://tegra-ubuntu:51952/

SUMMARY
=====

PARAMETERS
* /dji_sdk_read_cam/gray_or_rgb: 1
* /dji_sdk_read_cam/to_mobile: 1
* /rostdistro: indigo
* /rosversion: 1.11.21

NODES
/
  dji_sdk_read_cam (dji_sdk_read_cam/dji_sdk_read_cam)

ROS_MASTER_URI=http://localhost:11311

core service [/rosout] found
process[dji_sdk_read_cam-1]: started with pid [3381]
1
System init.....
create ivafilter success
Inside NvxLiteH264DecoderLowLatencyInitNvxLiteH264DecoderLowLatencyInit set DPB
and Mjstreamingdisable host
enable device
mount gadgetfs
system init ok!

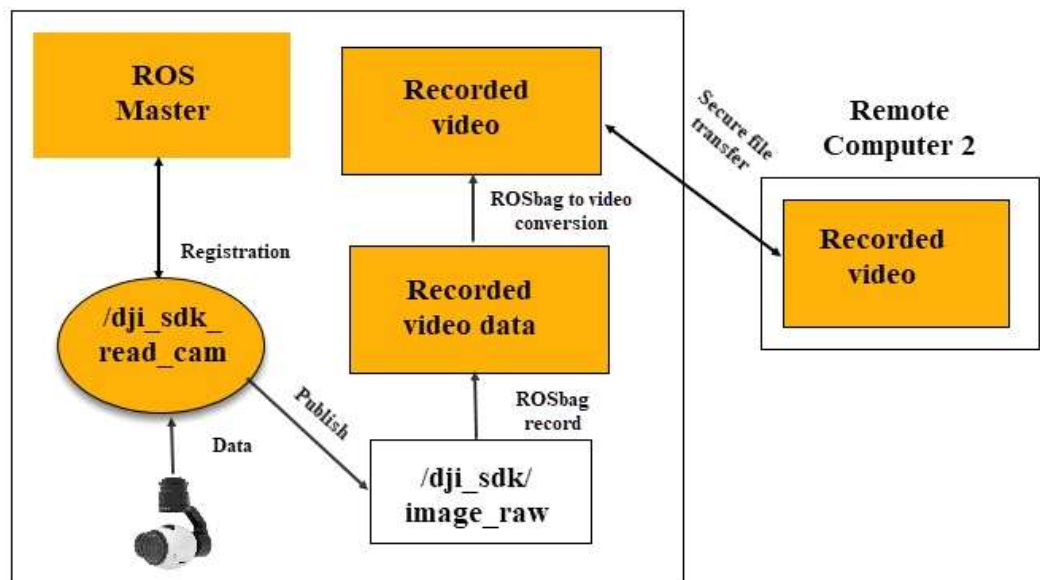
udc soft connect

```

< Parameters for the camera node in Roslaunch program >

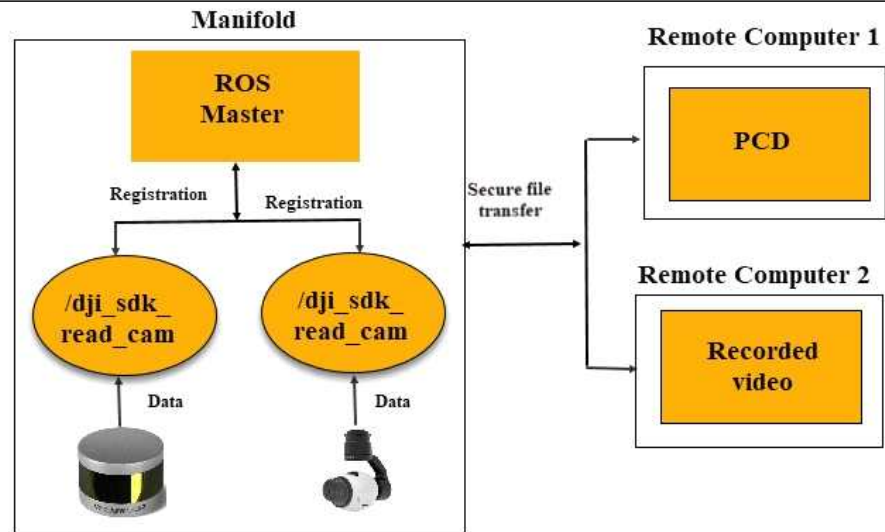
- Camera 노트 /dji_sdk_read_cam을 등록하면 /dji_sdk/image_raw topic이
게시되고 M100으로부터 영상이 공급된다. 해당 비디오는 먼저 .bag 형식으로
녹화되고 ROSbag to video 변환을 통해 원격 호스트 2에서 정합된 포인트
클라우드 데이터와 함께 실시간 비디오 정합을 위한 데이터로 사용.

Manifold



< Architecture of Manifold and Remote Computer 2 remote communication >

- 다음 그림 (a)는 VeloCamData를 취득하기 위한 전체 구성을 보여준다. 먼저
Manifold와 원격 호스트 1이 연결되고, 시작하면 (b) 1번 옵션으로 Velodyne
LiDAR를 위한 Node가 등록된 후, 4번 옵션으로 카메라가 게시된 후 스캔이
종료된 직 후, 2번 옵션으로 스캔 데이터를 .PCD로 저장한다. 원격 호스트 1에서
PCD 데이터의 정합이 완료되면 2번 컴퓨터에서 비디오 정합을 연결하여 완성.



(a) Architecture of Manifold and Remote Computer 1, 2 remote communication

```

root@tegra-ubuntu: ~
File Edit View Search Terminal Help
root@tegra-ubuntu:~# ./a.out
-----
1. Launch LiDAR to scan
2. Record Scan data
3. Process the scanned data format
4. Launch video recorder
5. Process the recorded bag to video
6. Stop the whole scanning system
7. Eliminate previous scanned and recorded video data
-----

```

(b) Prepared necessary function menus of the Manifold computer

```

1. case 1: system("roslaunch velodyne_pointcloud VLP16_points.launch");
2. break;
3. case 2: system("roslaunch velodyne_pointcloud VLP16_points.launch");
4. cout<<" Scanned data is getting converted to PCD format....!!"<<endl;
5. system("rosrun pcl_ros bag_to_pcd ./share/ScannedData.bag /velodyne_points
./share/ScannedData");
6. break;
7. case 3: cout<<" Scanned data is getting converted to PCD format....!! \n"<<endl;
8. system("rosrun pcl_ros bag_to_pcd ./share/ScannedData.bag /velodyne_points
./share/ScannedData");
9. break;
10. case 4: system("roslaunch dji_sdk_read_cam manifold_cam.launch");
11. break;
12. case 5: cout<<"video is getting recorded!!"<<endl;
13. system("rosrun rosbag record --duration=5
./share/bag_video/recordFile/record.bag /dji_sdk/image_raw");
14. cout<<"video conversion is taking place"<<endl;
15. system("python /home/ubuntu/share/bag_video/bag2video.py -o
/home/ubuntu/share/bag_video/recordFile/record.avi -v --encoding bgr8
/dji_sdk/image_raw /home/ubuntu/share/bag_video/recordFile/record.bag");
16. break;
17. case 6: system(" rosnode kill -a");
18. break;
19. case 8: system("sudo chmod -R 777 /home/ubuntu/share/");
20. break;
21. case 7: cout <<"Data is being Eliminated ..... \n"<< endl;
22. system("rm /home/ubuntu/share/ScannedData/*.*");
23. system("rm /home/ubuntu/share/bag_video/recordFile/*.*");
24. break;

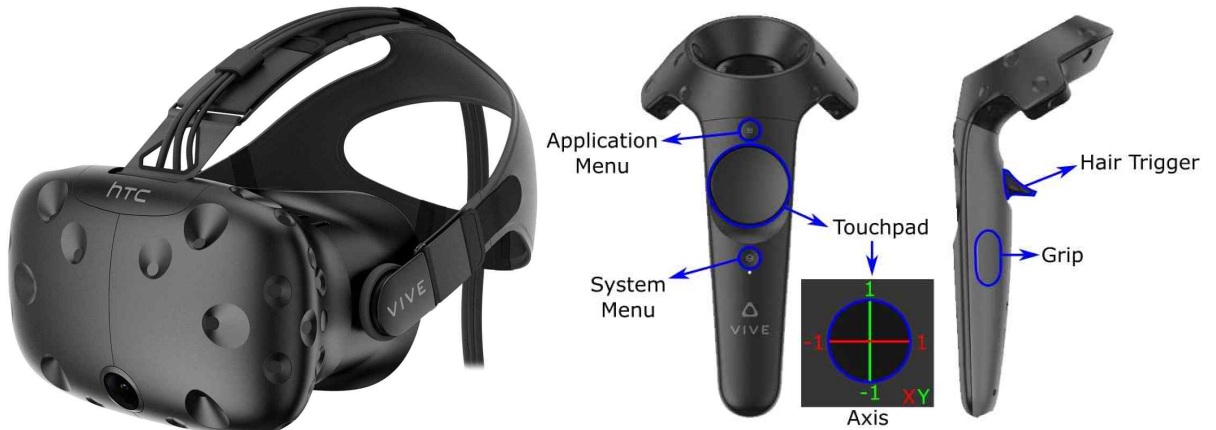
```

(c) Prepared code script of the menu functions [C++: Triggering scanning and recording through RC1]

< Overall architecture of VeloCamData capture and code snippet >

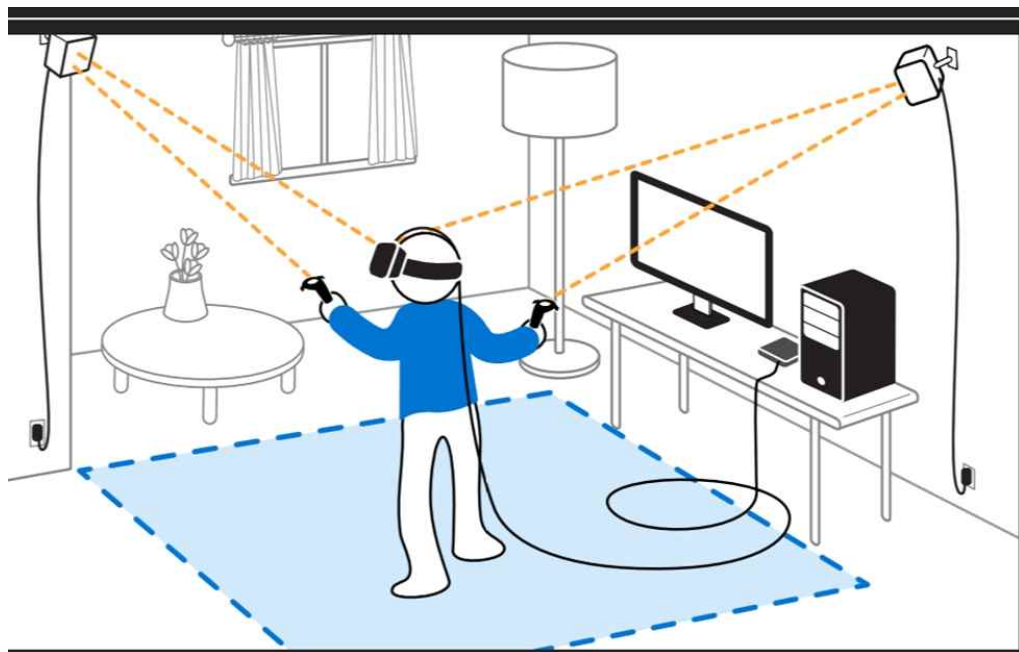
(2) 가상 리트로피팅을 위한 시각시스템

- 기반시설 및 플랜트의 증설, 복구 역설계를 위한 리트로피팅의 사전 가상 리트로피팅은 실 시공에 앞서 노력, 시간 및 비용을 줄일 수 있는 효과적인 방법이다. 본 과제의 가상 리트로피팅을 위한 시각시스템은 아래 그림과 같은 HTC VIVE와 사용자 입력을 위한 Controller가 사용됨.



< HTC VIVE headset and Controllers >

- HTC VIVE를 사용하는 시각시스템 및 360° 인터랙션 실내 기본환경은 아래 그림과 같이 최소 2m x 1.5m의 자유로운 활동영역을 필요로 하고, 베이스 스테이션으로부터 최대 5m 이내, 30~45° 하방에서 사용자 인터랙션이 이루어져야한다. 사용자 입력을 위한 Controller를 사용하여 전 방향으로 자연스럽게 걷고 선택하는 모든 움직임을 가상 리트로피팅 시각시스템과 연결시킬 수 있음.



- 시각시스템은 VTK(Visualization Tool Kit)을 사용하였다. VTK는 오픈소스 객체지향 C++ 클래스 라이브러리로 Python, JAVA등의 번역기 인터페이스 제공과 함께 제공되어 컴퓨터 그래픽스 및 영상처리 분야에서 시각화 알고리즘, Implicit 모델링, 폴리곤 축소, 이미징 절차 등에 다양하게 이용됨.

- HTC VIVE의 Stream VR을 위하여 OpenVR API를 사용한 렌더링을 포함하였고, 사용자 인터랙션을 위한 내장형 버튼 GUI를 위하여 QT 라이브러리를 이용하였으며, OpenGL과 Direct 3D를 통해 키보드/마우스에 접근할 수 있는 SDL 개발용 라이브러리를 사용함.
- 아래와 같이 SDL_DIR, OpenVR_DIR을 지정하고 DModule_vtkGUISupportQt와 DModule_vtkRenderingOpenVR을 ON, DVTK_OPENVR_OBJECT_FACTORY은 OFF로 하는 것을 포함 VTK를 Qt, OpenVR 그리고 SDL과 함께 wrapping함.

```

1. -DCMAKE_INSTALL_PREFIX:PATH=${SDL_DIR} \
2. -DBUILD_TESTING:BOOL=OFF \
3. -DModule_vtkGUISupportQt:BOOL=ON \
4. -DModule_vtkRenderingOpenVR:BOOL=ON \
5. -DVTK_OPENVR_OBJECT_FACTORY:BOOL=OFF \
6. -DVTK_QT_VERSION:STRING=5 \
7. -DQt5_DIR:PATH=${Qt5_DIR} \
8. -DOPENVR_INCLUDE_DIR:PATH=${OPENVR_DIR}/headers/ \
9. -DOPENVR_LIBRARY:FILEPATH=${OPENVR_DIR}/bin/linux64/libopenvr_api.so \
10. -DSDL2_LIBRARY:FILEPATH=${SDL_DIR}/lib/libSDL2-2.0.so.0.4.1 \
11. -DSDL2_INCLUDE_DIR:PATH=${SDL_DIR}/include/SDL2 \

```

< Setting up VTK rendering in Qt application > [System Parameter: QT with VTK in RC2]

- Qt 응용프로그램에서 vtk를 렌더링하기 위해 먼저 현재 renderwindow에서 vtk renderer를 qvtkWidget에 포함하고(line 12), button 인터랙션을 추가하고(line 24), QtVTK 응용에서 포인트 클라우드 데이터를 읽는 등(line 27) 아래 그림과 같은 절차로 Interactive 시각 시스템을 실현.

```

1. // Constructor
2. RenderWindowUISingleInheritance::RenderWindowUISingleInheritance()
3.
4. {
5.
6.     this->ui = new Ui_RenderWindowUISingleInheritance;
7.     this->ui->setupUi(this);
8.     this->ui->qvtkWidget->SetRenderWindow(renderWindow);
9.     render->SetBackground(1, .5, 1);
10.
11.     // VTK/Qt wedded
12.     this->ui->qvtkWidget->GetRenderWindow()->AddRenderer(renderer);
13.     QWidget::showMaximized();
14.     // Set up action signals and slots
15.     connect(this->ui->actionExit, SIGNAL(triggered()), this, SLOT(slotExit()));
16.     connect(this->ui->pushButton, SIGNAL(clicked()), this, SLOT(Test()));
17.     connect(this->ui->pushButton_2, SIGNAL(clicked()), this, SLOT(Model1()));
18.     connect(this->ui->pushButton_3, SIGNAL(clicked()), this, SLOT(Model2()));
19.     connect(this->ui->pushButton_4, SIGNAL(clicked()), this, SLOT(Model3()));
20.     connect(this->ui->pushButton_5, SIGNAL(clicked()), this, SLOT(interaction()));
21.     connect(this->ui->pushButton_6, SIGNAL(clicked()), this, SLOT(camera()));
22.     ui->qvtkWidget->update();
23.
24. void RenderWindowUISingleInheritance::Test()
25. {
26.     vtkSmartPointer<vtkUnstructuredGridReader> reader1 =
27.         vtkSmartPointer<vtkUnstructuredGridReader>::New();
28.     reader1->SetFileName("FARO_Scan_PlantPart_ninehundred_2.vtk");
29.     reader1->Update();
30. }

```

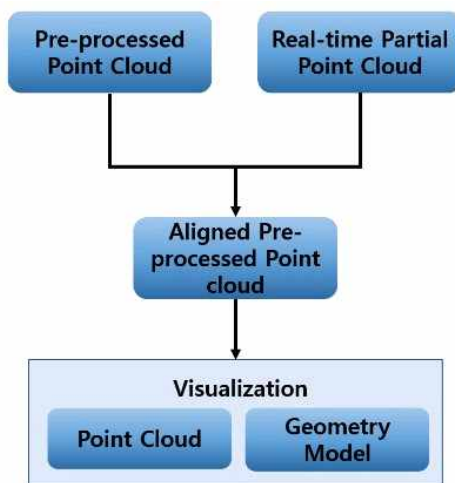
< Creating interactive GUI > [C++: Creating interactive GUI for retrofitting in RC2]

- 가상 리트로피팅의 시각환경은 한국건설기술연구원(KICT: Korea Institute of Construction Technology)의 수처리시설을 대상으로 하였다. 아래 그림은 KICT의 수처리 시설과 이를 라이다 스캐닝하여 포인트 클라우드 데이터로 재구성, 변환한 그림으로 As-built 환경의 시각화를 배경으로 알맞은 변형이나 증설을 위한 환경 확인을 위한 가상 리트로피팅의 기본 환경임.



< KICT water treatment facility and its LiDAR scanned pre-processed point cloud data >

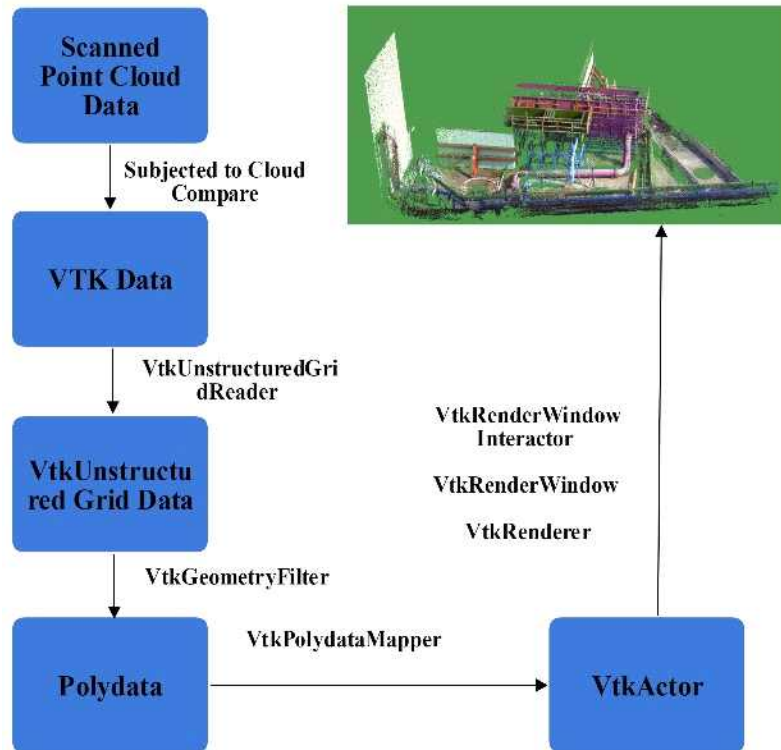
- 가상 리트로피팅을 위해서 As-built 환경과 추가, 변형되는 환경의 정합이 필수적이다. 이를 위해 본 과제에서는 가상 리트로피팅 시행 시 해당 지역의 부분 포인트 클라우드를 실시간으로 취득하여 G-ICP(Generalized Iterative Closest Point) 알고리즘을 이용하여 아래 왼쪽 그림과 같은 절차로 정합을 실시하고, 정합 결과에 수정, 변형을 원하는 형상모델을 아래 우측의 VTK 포인트 클라우드 모델링 데이터에 중첩시킴으로써 가상 리트로피팅을 완성함.



< Block diagram for point cloud alignment and LiDAR scanned point cloud output using VTK >

- PCD 형식으로 정합된 수처리시설 포인트 클라우드 데이터는 Cloud Compare S/W를 이용하여 VTK의 Unstructured Grid 형식으로 변환하고 vtkUnstructuredGridReader로 VTK에서 읽힌다. 이 Reader의 출력은

vtkUnstructuredGrid 데이터 오브젝트이고 vtkGeometryFilter의 입력으로 사용된다. 이 필터는 어떤 데이터에서도 형상과 관련된 데이터를 추출한다. 형상은 0D, 1D, 2D 셀로 추출되고 폴리곤의 형태로 출력되어 vtkPolyDataMapper의 입력으로 사용된다. 이후 그래픽 프리미티브와 매핑되고 vtkActor, vtkRenderer, vtkRenderWindow를 거쳐 위 우측그림과 같이 그려지고, vtkRenderWindowInteractor를 통해 사용자 인터랙션이 구현된다. 아래 그림은 PCD 파일이 VTK 파이프라인을 통해 시각화 되는 과정을 보여줌.

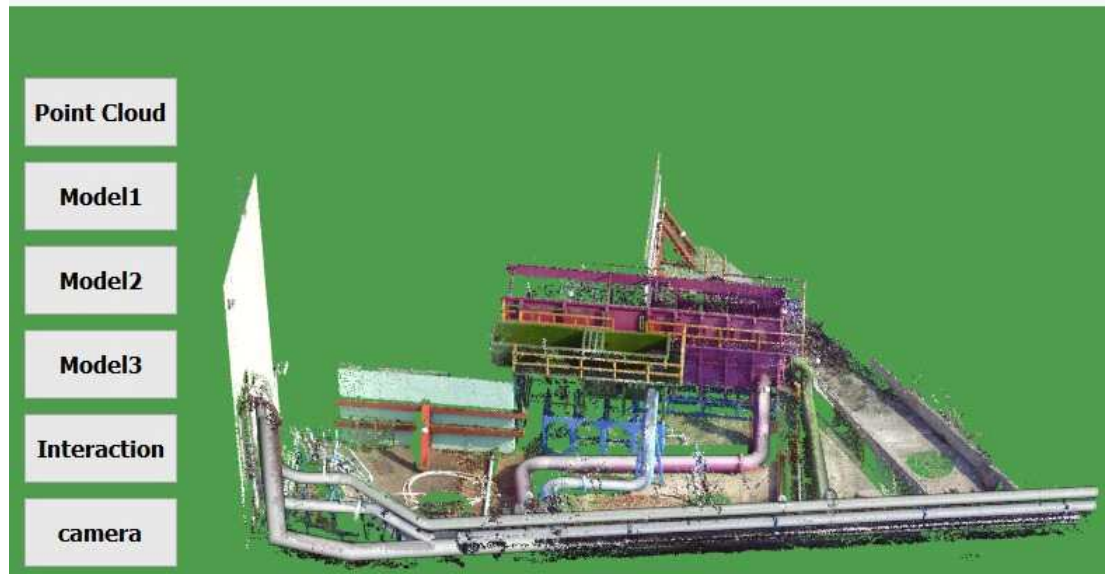


< Architecture of the PCD Visualization process >



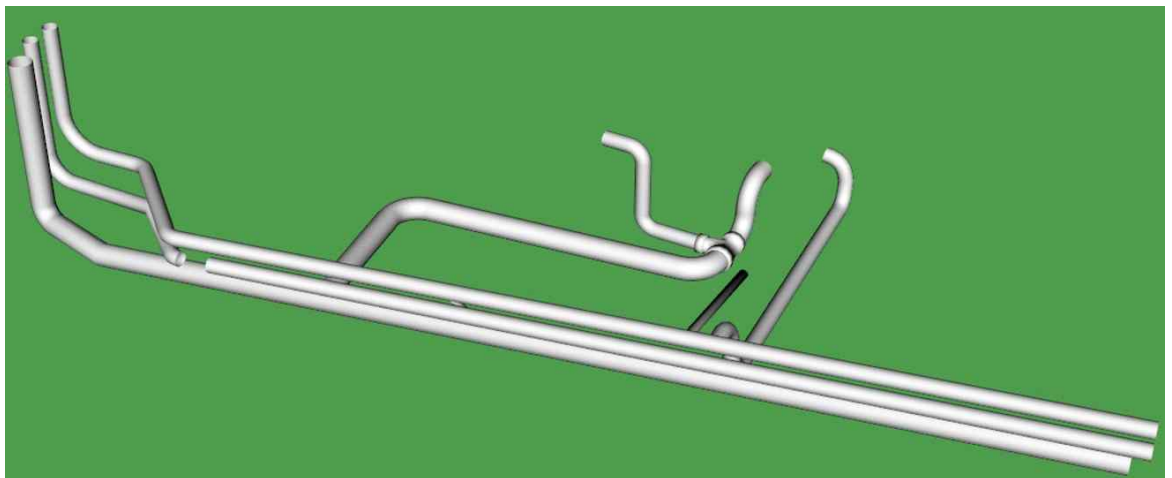
< Generated CAD model of pipes in the KICT water treatment facility >

- 위 그림은 KICT 수처리시설의 리트로피팅을 위한 파이프 .obj 포맷의 CAD 모델을 추출한 것으로 아래 그림에서 포인트 클라우드 모델과 중첩되어 있는 As-built 펌프시설과 함께 일정 유속의 흐름을 만들 수 있는 시설로 유량 공급을 확대할 수 있게 확장하는 리트로피팅 프로세스를 구현하고자 함, 다음 그림 좌측의 리트로피팅 프로세스 상호작용을 위한 메뉴는 VIVE controller를 통해 선택과 해당 기능의 구현과 연결됨.



< Generated CAD pipes along with Point Cloud data in the KICT water treatment facility >

- 기존 모델을 기준으로 파이프를 흐르는 유속을 변화시켜 효율을 향상시키기 위한 다양한 시도를 아래 그림의 T-junction과 같은 추가 형상으로 모델링하고 해석 패키지를 통해 검증한 CAD 모델을 vtkOBJReader를 통해 읽어서 중첩하여 확인하는 과정으로 리트로피팅 프로세스를 진행함.

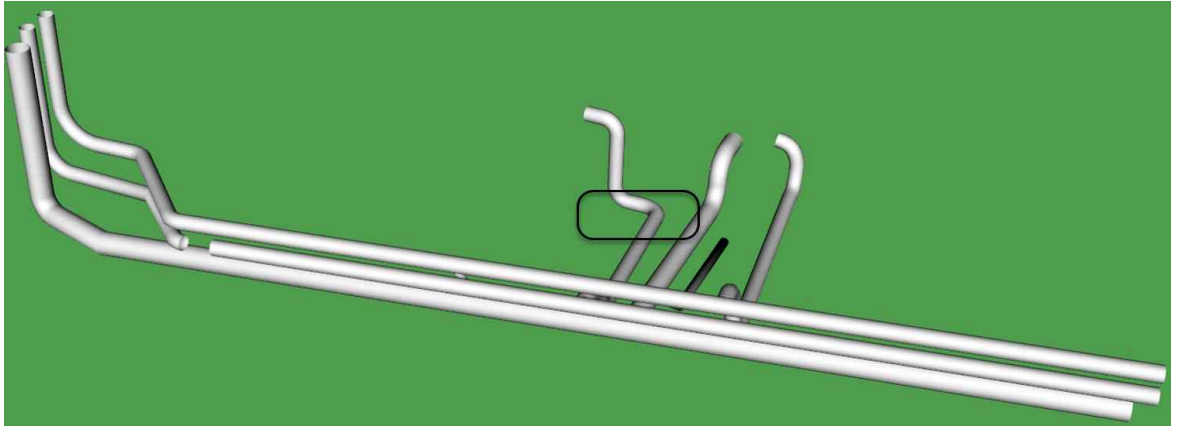


< Retrofitting pipes CAD model 1 for retrofitting in the KICT water treatment facility >



< Retrofitted CAD pipes model 1 along with Point Cloud data >

- 기존 모델을 기준으로 파이프 네트워크의 복잡도를 낮추고 단순화시켜 유량을 변화시키는 다양한 시도를 아래 그림과 같이 수정 변경한 CAD 모델로 읽어서 중첩하여 확인하는 과정으로 또 다른 리트로피팅 프로세스를 진행함.

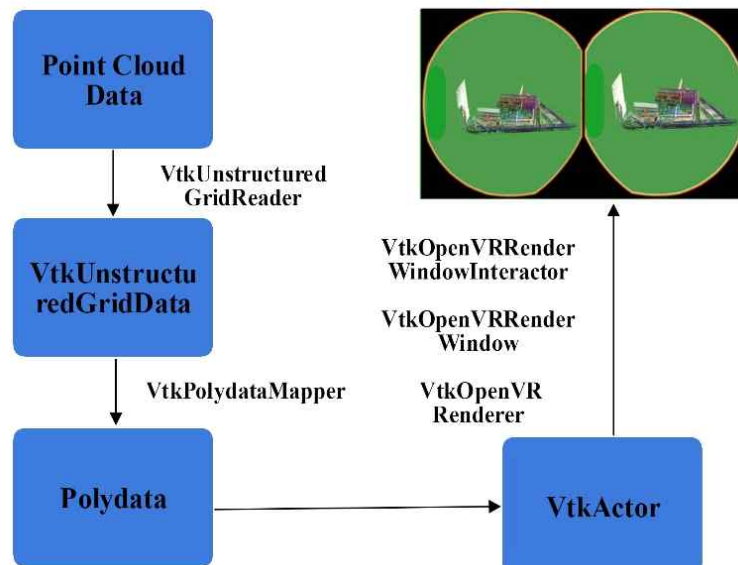


< Retrofitting pipes CAD model 2 for retrofitting in the KICT water treatment facility >



< Retrofitted CAD pipes model 2 along with Point Cloud data >

- 위와 같은 일련의 리트로피팅 프로세스는 OpenVR을 이용하여 StreamVR로 HTC VIVE를 통해 시각화 되었다. 아래 그림과 같은 환경에서 Vive Controller에 의한 메뉴조작으로 실감나는 가상 리트로피팅 환경을 구현.



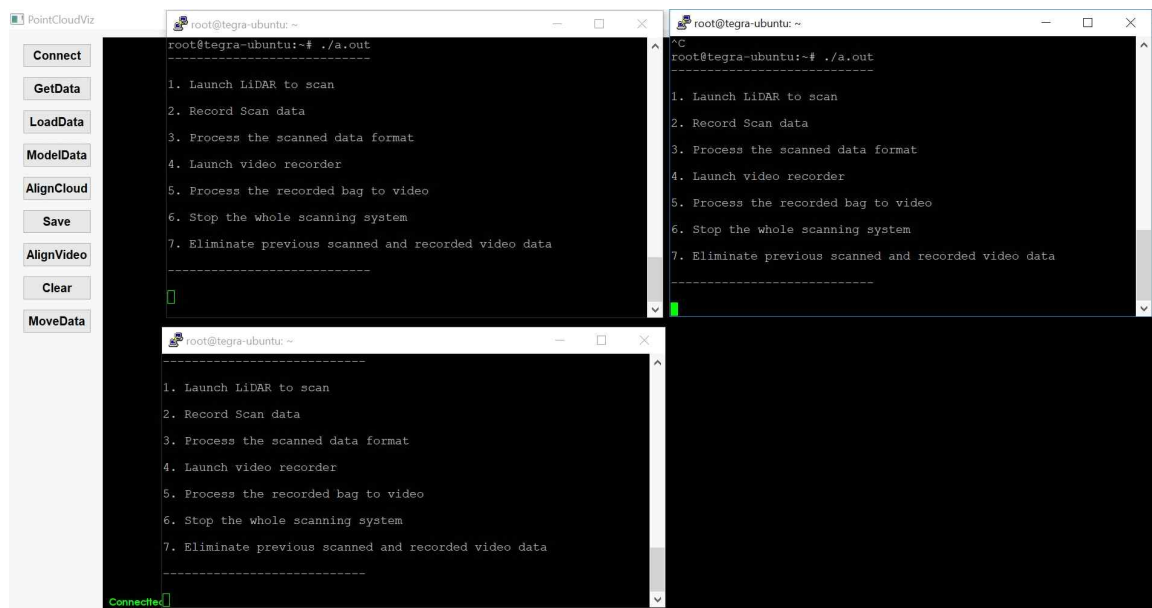
< Architecture of the PCD Visualization process in HMD >

(3) 라이다장착 UAV 기반 KICT 가상 리트로피팅: 실내 테스트

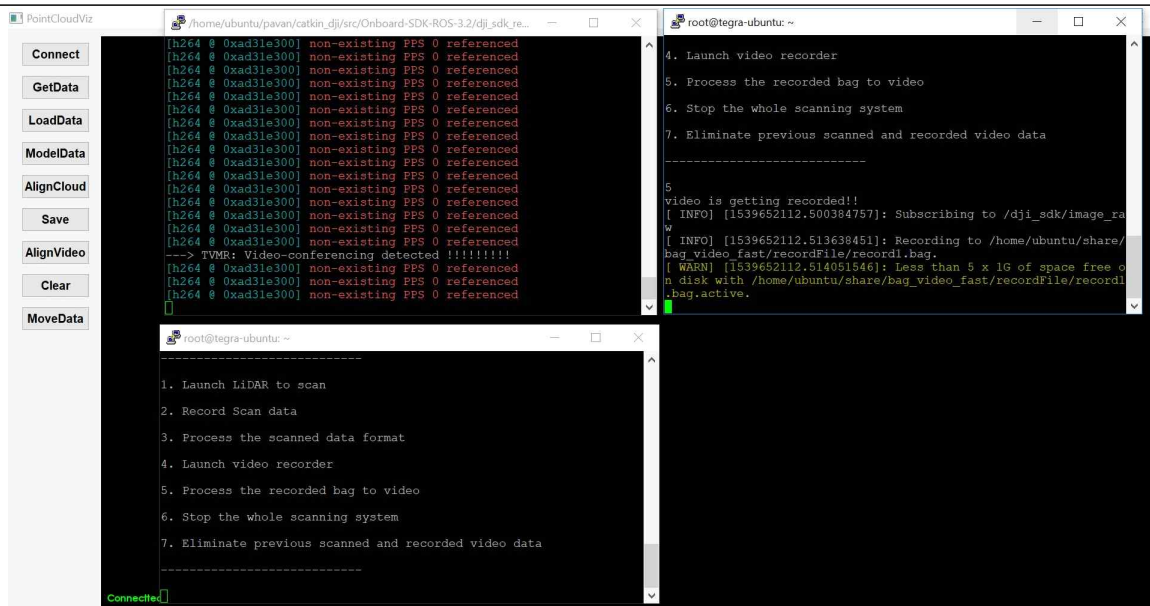
- 라이다 장착 UAV 기반 가상 리트로피팅의 건설기술연구원 현장 검증에 앞서 중앙대학교 첨단영상대학원 제 19 회 첨단영상제[2018. 10. 15 ~ 10. 18]에 “실내 라이다 기반 가상 리트로피팅”을 출품하여 전체 프로세스 점검 및 검증 준비.



- 위 그림은 2대의 원격 호스트 컴퓨터와 라이다, Manifold 및 짐벌 카메라를 장착한 시스템, 테스트 용 샘플형상이며, 아래 좌에서 우로 launching Camera node, launching Velodyne node, recording Velodyne & Camera node 화면.

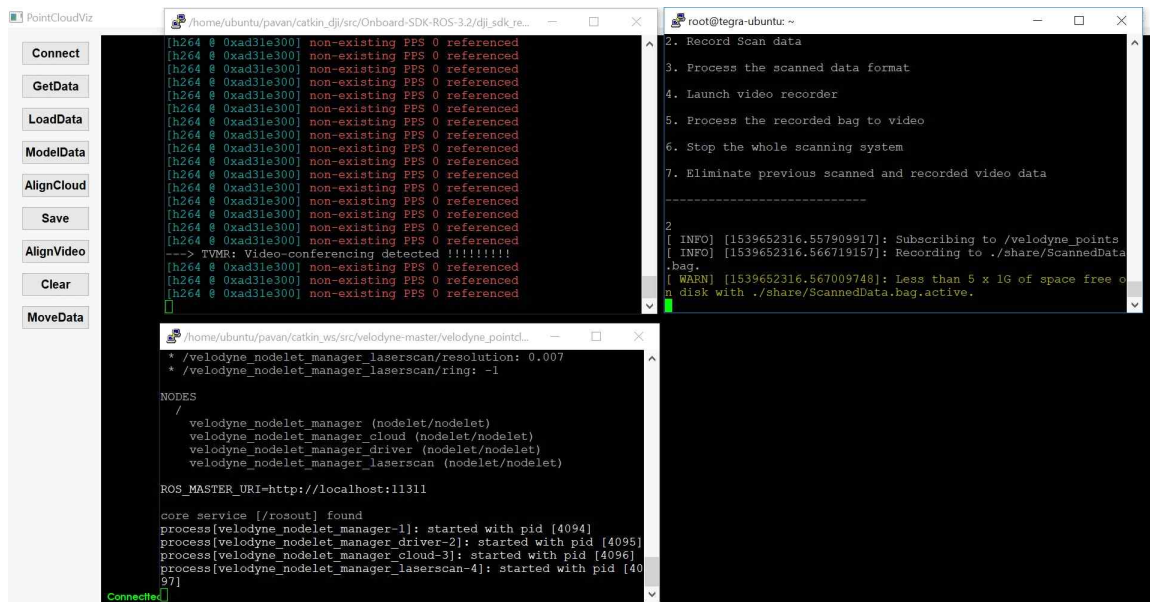


< Initial setup for VeloCamData capture >

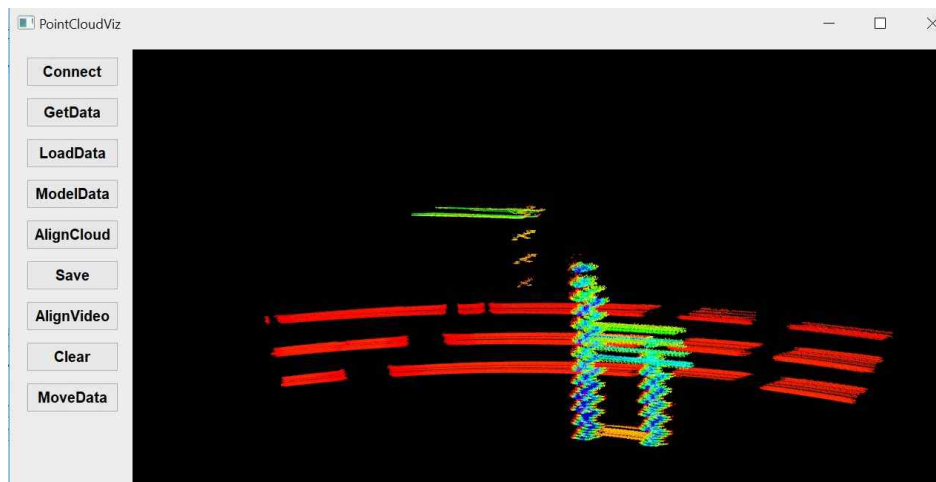


< Video recording process after the camera node launch >

- 위 화면에서 Camera node launch후의 일련의 비디오 캡처 프로세스를 보여주고, 아래 화면에서는 Velodyne node launch후에 스캐닝 프로세스를 보여준다. 두 데이터 모두 .bag 파일로 저장된 후에 각기 Video와 PCD로 전환.

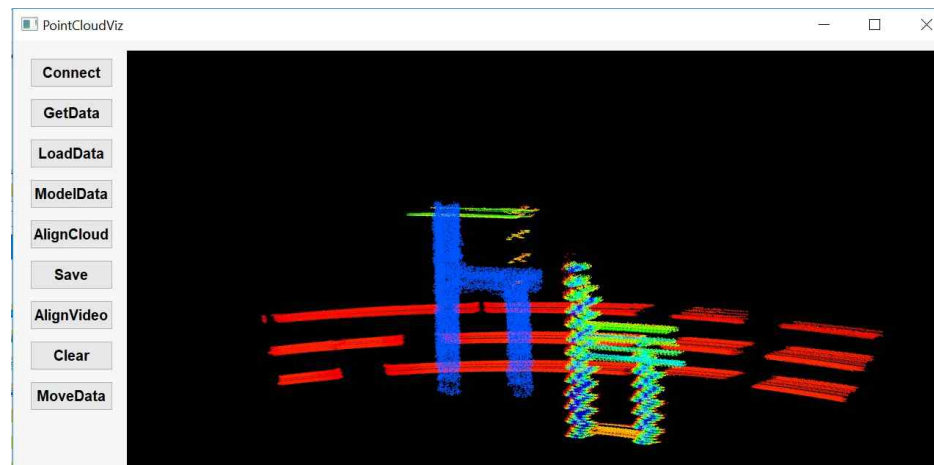


< Data recording process after the Velodyne node launch >

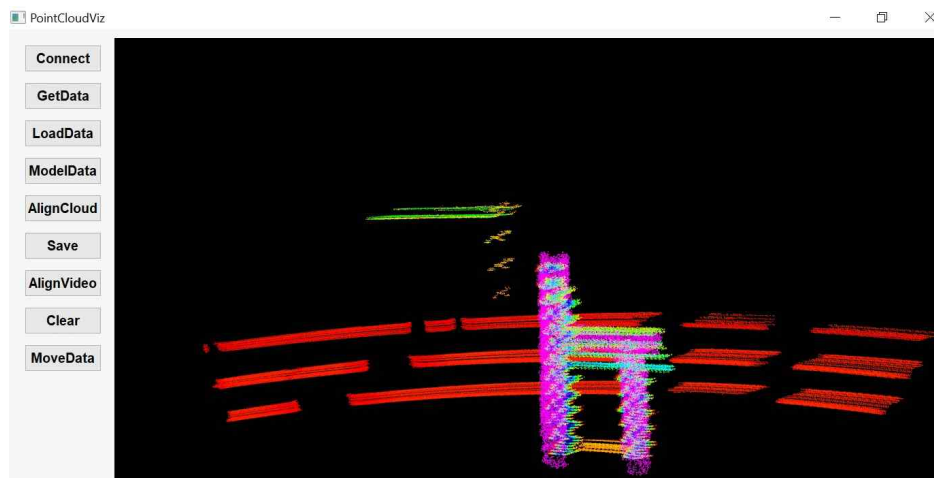


< Scanned partial point cloud data from Velodyne LiDAR >

- 성공적인 VeloCamData의 확보가 이루어지면 원격 호스트로 비디오와 PCD가 전송되고, 위 그림의 스캔한 부분 PCD 데이터는 아래 그림의 청색 부분인 사전에 정합한 As-Built 모델과 정합을 개시함.

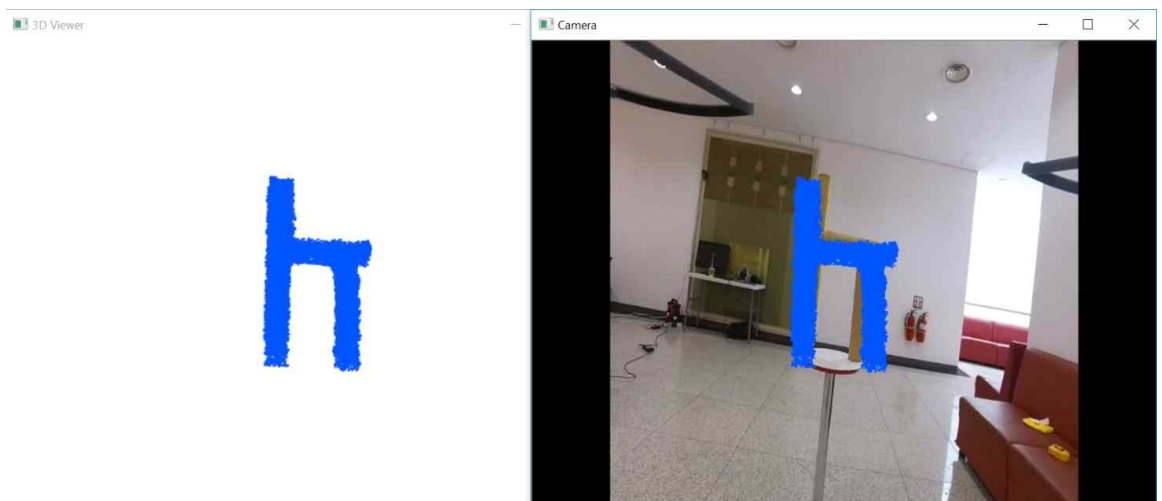


< Scanned partial point cloud data with model data >



< Alignment of scanned partial point cloud data with model data >

- 정합절차에 따라 부분 포인트 클라우드와 As-built 모델이 위 그림과 같이 정합되면, 부분 포인트 클라우드는 제거하고, 실시간 부분 포인트 클라우드와 함께 취득된 비디오 영상 면으로 As-built 모델을 투영하여, 아래 영상과 같이 주변 환경과 중첩되는 포인트 클라우드 모델의 리트로피팅 환경을 준비.



< Point cloud data alignment with physical model in real time video >

(4) 라이다장착 UAV 기반 KICT 가상 리트로피팅: 성능 점검표

Performance Assessment of the Virtual UAV-Lidar Retrofitting Process at KICT

1. **Gaze emulation performance of the airborne camera:** An average of 7~8%, i.e. an average of 8.26° reduction in the changes of viewpoint was achieved. As indicated in Table 1. The targets were found well within the expected foveal range (**Elevation < 10°, Azimuth < 5°**) as indicated in table 2.

Scenario	Base line target for (A&B)	Target X → Target Y	With VOR (Start)		With VOR (End)		Without VOR (Start)		Without VOR (End)		A	B	Difference in viewpoint change (in degree)
			X	Y	X	Y	X	Y	X	Y	Scene displacement (W/ VOR)	Scene displacement (W/O VOR)	
1	2	1 → 2	3500	1648	2036	1664	3500	1648	1656	1628	1464.09	1844.11	8.93
		2 → 3	2036	1664	412	1576	2036	1664	408	1636	1626.38	1628.24	0.04
2	1 & 4	1 → 2	1956	1324	840	1400	1956	1324	184	1352	1118.58	1772.22	15.36
		2 → 3	840	1400	3056	1648	840	1400	3592	1536	2229.83	2755.36	12.35
		3 → 4	3056	1648	2968	1544	3056	1648	2736	1560	136.24	331.88	4.60

Table 1. Displacement of scene and changes in viewpoint of the UAV mounted gimbal camera

Scenario	Target	Position(in pixels)		Elevation within foveal range (in degrees)	Azimuth within foveal range (in degrees)
		X	Y		
1	1	2000	1392	2.54	0.00
	2	2032	1672	4.04	0.75
	3	1832	1464	0.85	3.95
2	1	1960	1320	4.23	0.94
	2	1824	1476	0.56	4.14
	3	1952	1580	1.88	1.13
	4	1804	1560	1.41	4.61

Table 2. Position of targets within foveal range (Prepared by Adithya B. )

2. **Curvature matching of the model geometry:** Table 3 shows the analysis of curvature matching with different radius of pipe and elbow shapes and **92.30% and 87.5% of curvature matching** of the model geometry of straight and elbow pipes respectively.

Pipeline	Curvature(cm)	No. of shapes	No. of shapes matching			
Straight pipes	15	3	26	2	24	92.3%
	25	16		15		
	30	7		7		
Elbow shapes	15	0	7	0	6	85.7%
	25	5		4		
	30	2		2		

Table 3. Analysis of curvature matching of the model geometry (Prepared by Ashok Kumar Patil )

3. **Dynamic performance of the UAV-AR retrofitting:** Table 4 shows the dynamic performance of the UAV-AR retrofitting with less than 20 quantitative repetitive adjustment and **Roll/Pitch displacement less than 2° (RMS) and Yaw displacement less than 5° (RMS)**.

Alignment Views	Iterations		Alignment			
			Distance(mm)	Roll(°)	Pitch(°)	Yaw(°)
View-1	16	Initial	15.48	-3.5	-1.9	8.7
		16 th	3.55	-1.3	-1.3	3.8
View-2	13	Initial	12.89	2.25	3.04	3.2
		13 th	7.29	1.04	2.33	3.8

Table 4. Dynamic performance of the UAV-AR retrofitting (Prepared by Pavan Kumar B. N. )

January 25, 2019 Project Leader

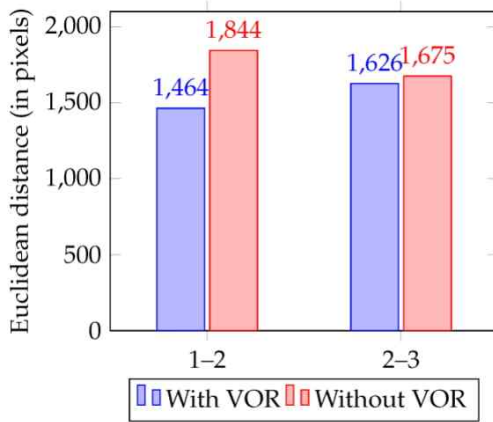
Young Ho Chai, Ph.D.

Korea Institute of Civil Engineering and Building Technology

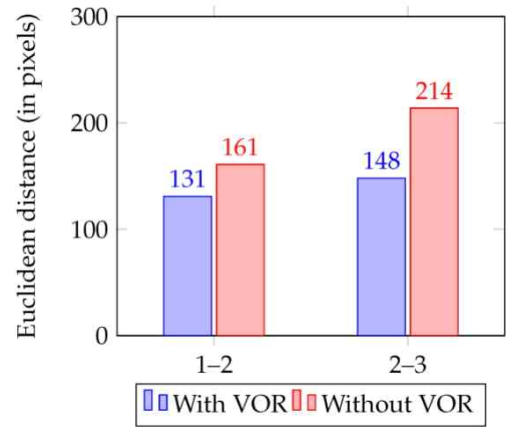




Appendix A, Gaze emulation performance of the airborne camera

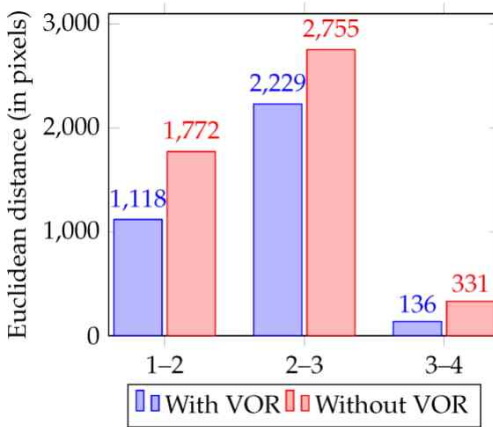


(a)

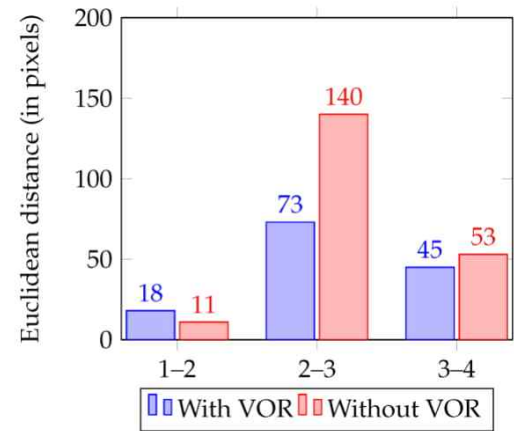


(b)

< Scene changes with and without vestibular-ocular reflex (VOR) for Scenario 1: (a) gimbal camera and (b) simulation >



(a)



(b)

< Scene changes with and without vestibular-ocular reflex (VOR) for Scenario 2: (a) gimbal camera and (b) simulation >

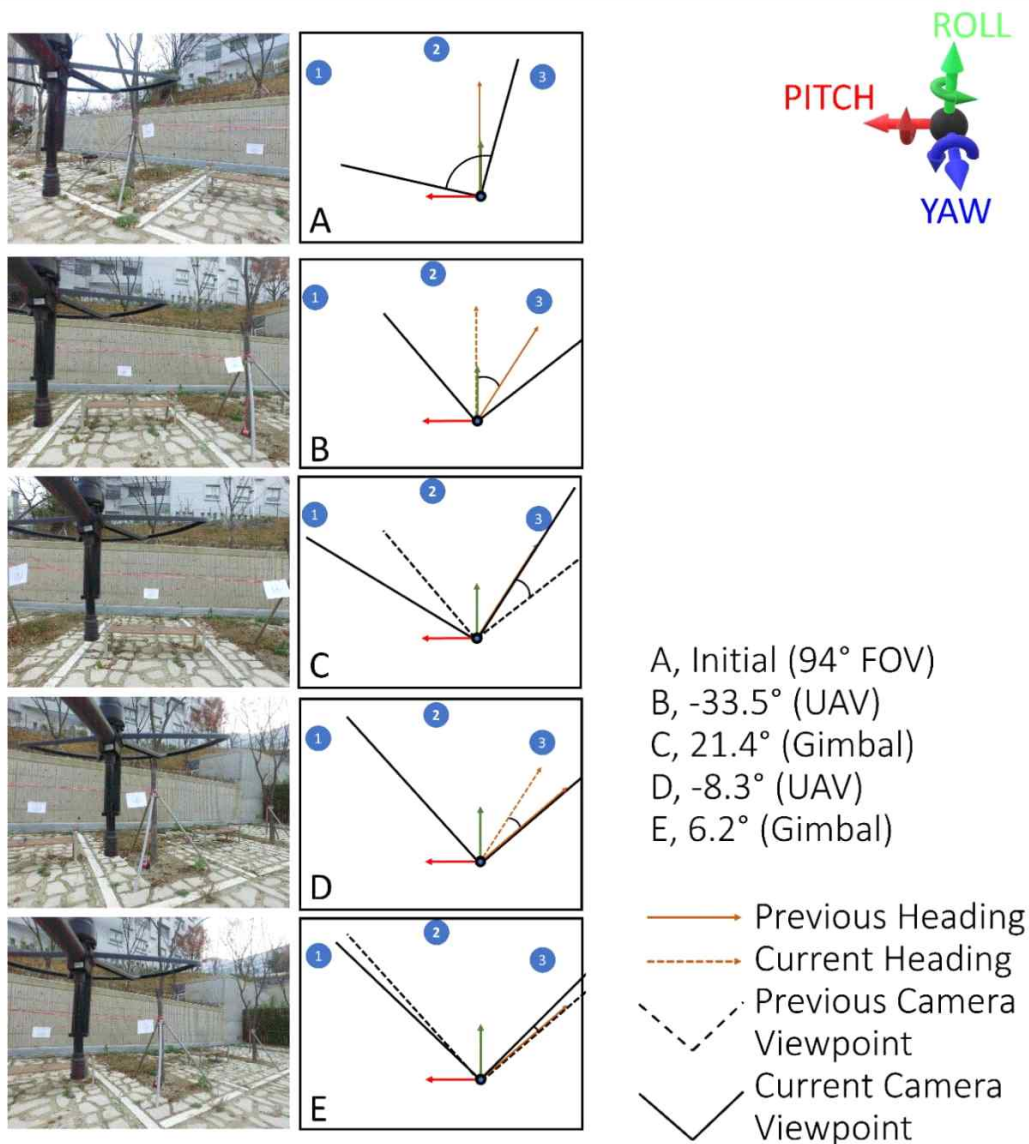
■ [기존 기술대비 진보성 및 차별성] 1. Gaze emulation performance of the airborne camera

✓ Performance Assessment의 Table 1, 2

1. Table 1의 Column A와 B는 VOR 적용유무에 따른 시작시점과 종료시점의 픽셀거리이다.
2. 마지막 컬럼은 VOR에 따른 이동픽셀거리를 42.55 픽셀(1°)로 나눈 시점의 차이를 보여준다.
3. Table 2의 Elevation/Azimuth within foveal range는 시점이 화면의 중심으로 몇 픽셀 떨어져 있는지를 확인하여 42.55(4,000/94° FOV)로 나누어 응시점 Elevation, Azimuth 각을 계산한다.

✓ Appendix A에서의 With VOR과 Without VOR과의 차이

1. 비행 중 응시시나리오 1, 2에서 VOR을 적용하지 않은 Gimbal움직임에 비해서, VOR을 적용하여 약 7~8%를 덜 움직이고 해당 응시를 완성할 수 있었다.
2. 시뮬레이션 비행에서는 VOR 안정화를 통해 3~4%의 픽셀 움직임의 향상이 있었다.



< Gimbal and UAV yaw angle changes for Scenario 1: **(left)** photographs from various camera and UAV positions and **(right)** third-person view showing the angles >

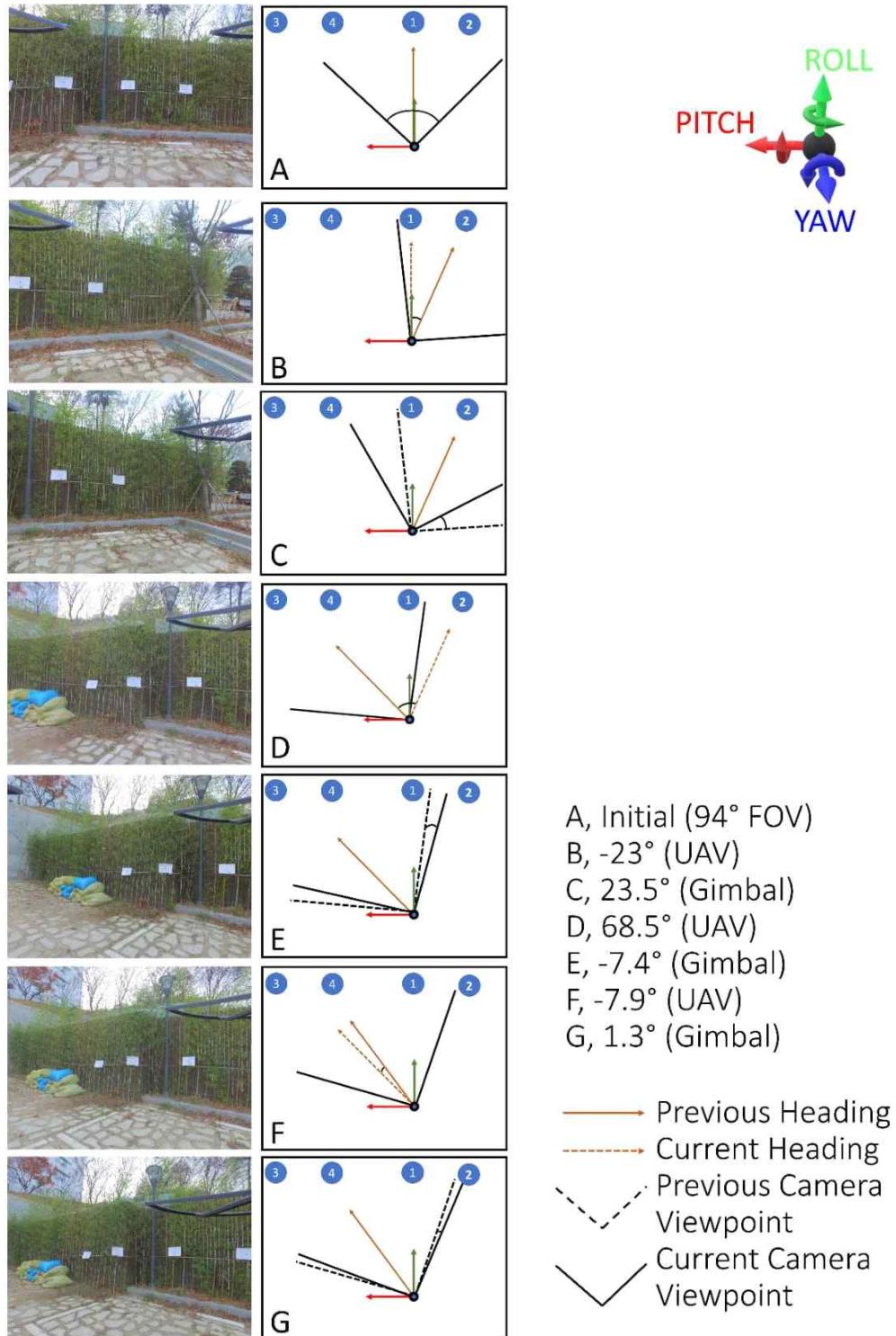
▪ [Performance Assesment] 1. Gaze emulation performance of the airborne camera 실험절차

✓ VOR 기능을 포함하는 Gimbal의 움직임 절차

3. 사전에 정한 거리와 높이에 목표물을 거치한다.
4. 비행 전 UAV 캘리브레이션을 실시한다.
5. UAV가 이륙하여 첫 번째 지점으로 이동하여 카메라가 1번 목표물을 향하게 한다.
6. UAV의 방향을 수정하여 순차적으로 다음 목표물을 향하게 한다.
7. UAV의 방향 수정에 따라 Gimbal의 보상 움직임이 사전 프로그램 된 것에 따라 진행된다.
8. 전체 비행의 카메라 영상을 4,000x4,000 픽셀의 비디오로 녹화한다.

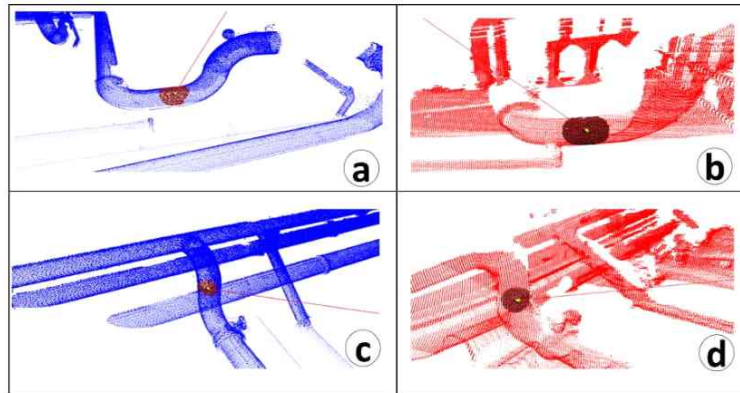
✓ VOR 기능을 포함하지 않는 Gimbal의 움직임 절차

1. UAV가 이륙하여 첫 번째 지점으로 이동하여 카메라가 1번 목표물을 향하게 한다.
2. UAV의 방향을 수정하여 순차적으로 다음 목표물을 향하게 한다.
3. 전체 비행의 카메라 영상을 4,000x4,000 픽셀의 비디오로 녹화한다.

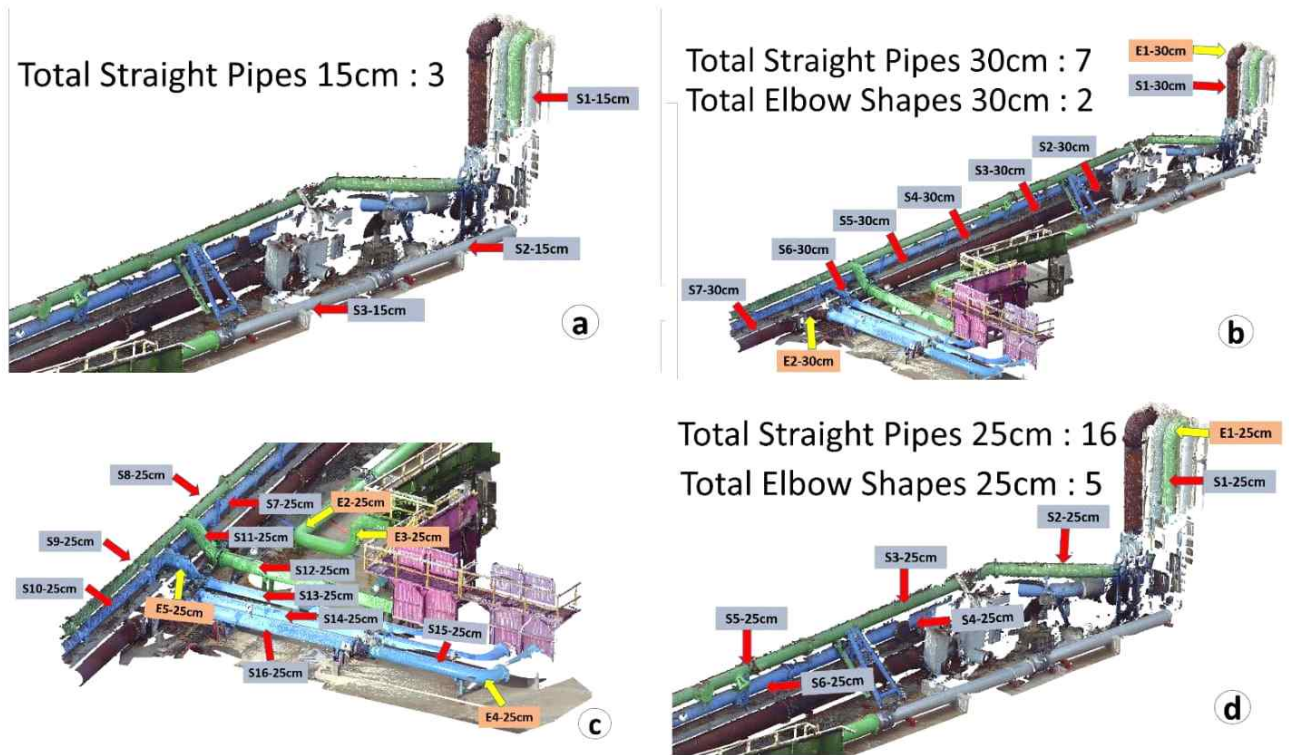


< Gimbal and UAV yaw angle changes for Scenario 2: **(left)** photographs from various camera and UAV positions and **(right)** third-person view showing the angles >

Appendix B, Curvature matching of the model geometry

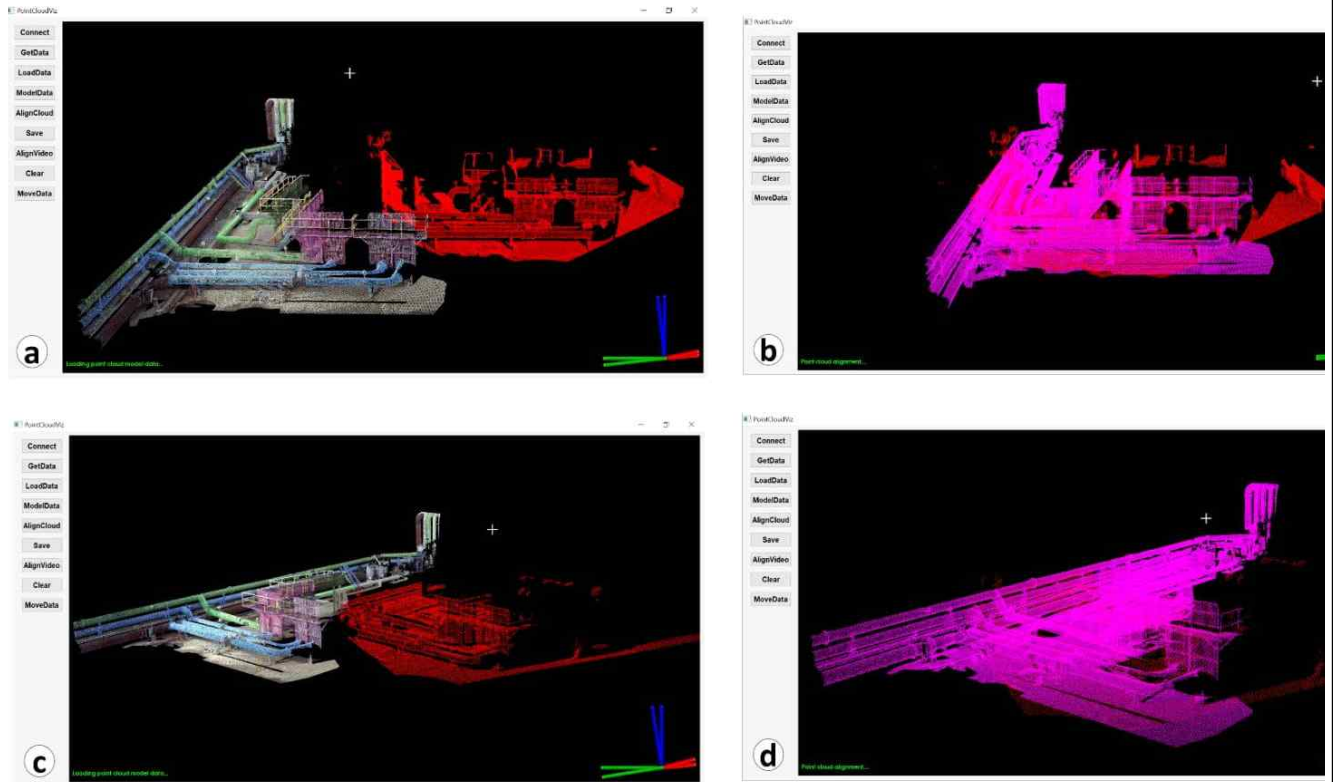


< Curvature matching of the model geometry (a) KICT Model 25cm (b) Partial Model 25cm (c) KICT Model 30cm (d) Partial Model 30cm>

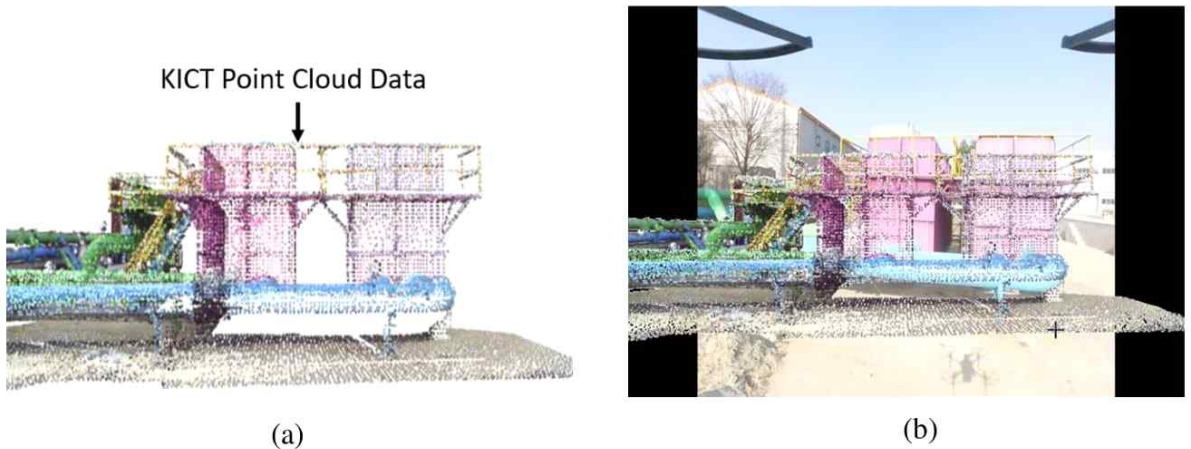


< (a) 15cm Model geometry, (b) 30cm Model geometry, (c) & (d) 25cm Model geometry>

Appendix C, Dynamic Performance of the UAV-AR Retrofitting



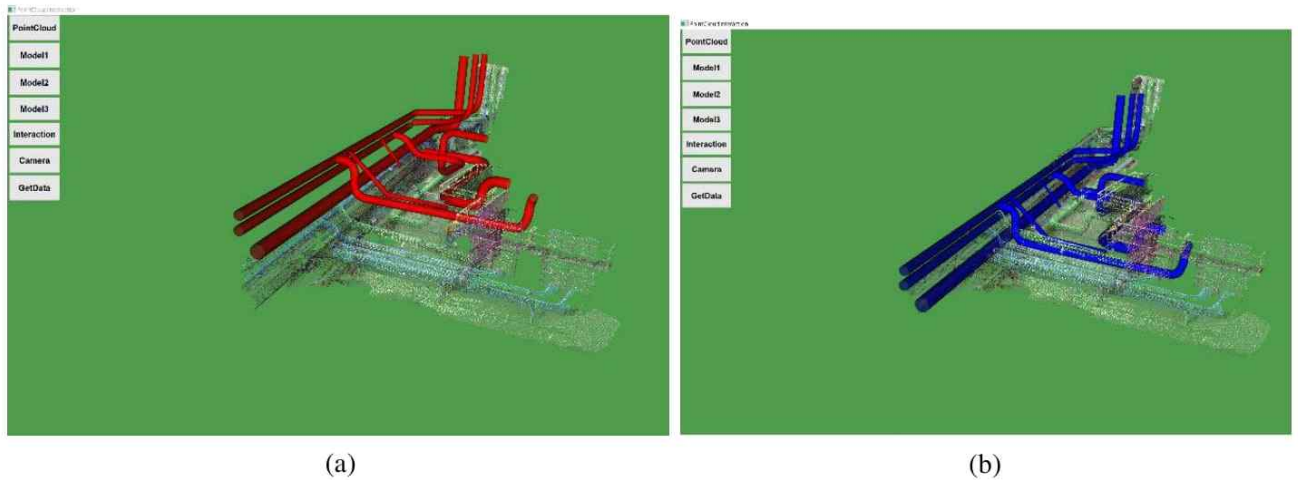
< (Row 1) View 1 : (a) before alignment, (b) after alignment (Row 2) View 2 : (c) before alignment, (d) after alignment >



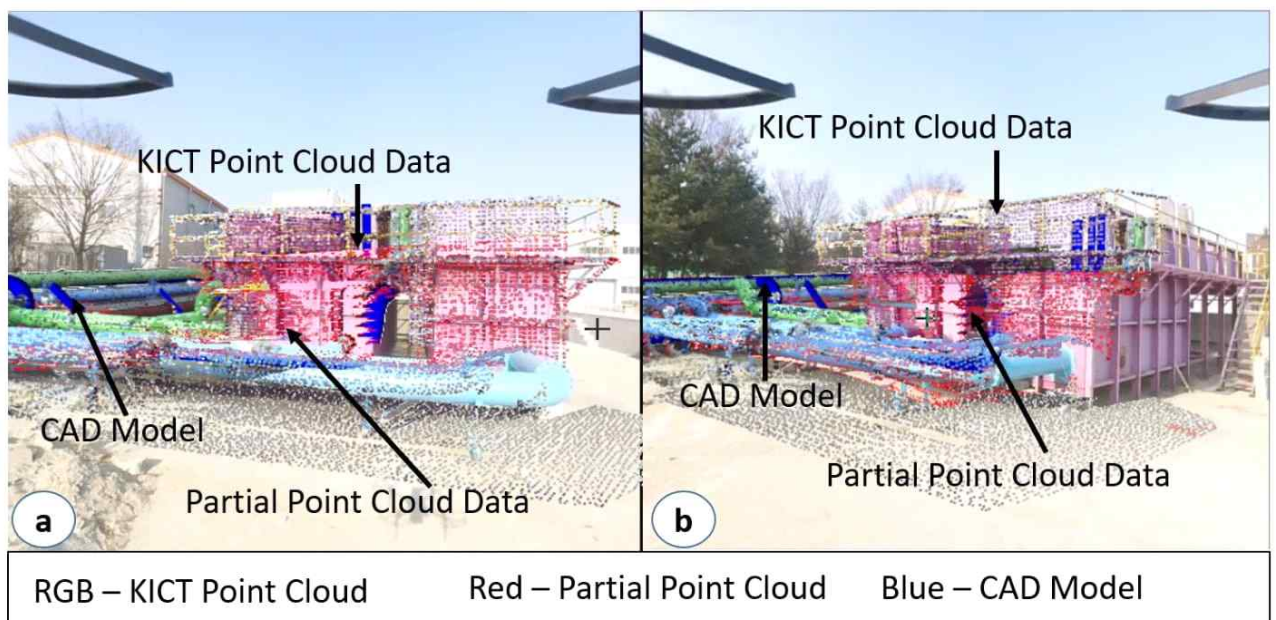
< Point Cloud Video alignment : (a) KICT Point Cloud Data (b) KICT Point Cloud Data alignment with in video>

■ [KICT 가상 리트로피팅] 3. Dynamic performance of the UAV-AR retrofitting

1. 본 과제를 통해 증설, 복구 역설계 환경의 기하학적 사양을 확인, 수정 및 평가하기 위한 가상 가설 후 평가 또는 가상 시설 개조를 현장에서 수행하기 위한 프레임 워크를 제안한다.
2. 전처리 된 3D 포인트 클라우드, LiDAR 센서에서 생성 된 실시간 포인트 클라우드와 컴퓨터 생성 CAD 모델 및 UAV 탑재 카메라에 의해 포착된 실제 배경을 이용한다.
3. 사용자 관점에서 전처리 된 포인트 클라우드 데이터를 정렬하고, 3D 포인트 클라우드 및 CAD 모델에서 추출한 RGB 이미지를 실시간으로 비디오 프레임과 통합한다.
4. 두 가지 시점에 대한 포인트 클라우드의 정렬 결과를 포함하는 KICT 현장의 가상 리트로피팅의 성능 평가를 통해 제안하는 프레임 워크가 기반 시설 및 플랜트의 기존 설비를 시각화, 수정 및 업그레이드하는 데 효율적으로 사용될 수 있다는 것을 보여준다.



< (a) Visualization of Point Cloud Data with Predefined CAD Model (b) Point Cloud data after retrofitting with CAD Model >



< 4 in 1 Alignment : (a) View 1, (b) View 2 >

■ [Performance Assesment] 3. Dynamic performance of the UAV-AR retrofitting 실험절차

✓ Onsite 데모의 실험 절차

1. 비행 전 UAV 캘리브레이션을 실시한다.
2. 원격 컴퓨터_1(데이터 통신)와 원격 컴퓨터_2(리트로피팅)를 준비한다.
3. On-board 컴퓨터(Manifold)에서 Velodyne LiDAR의 포인트 클라우드 취득을 시작한다.
4. On-board 컴퓨터(Manifold)에서 Gimbal 카메라의 영상 취득을 시작한다.
5. Manifold에서 취득된 포인트 클라우드와 카메라 영상을 원격 컴퓨터_1으로 전송한다
6. 기 취득된 전체 포인트 클라우드와 이송된 부분 포인트 클라우드를 O-GICP로 정합한다.
7. 정합된 포인트 클라우드와 취득된 비디오를 프레임별로 중첩하여 원격 컴퓨터_2로 전송한다.
8. 사전에 준비한 다양한 CAD모델을 프레임별로 중첩된 PCD-VIDEO 프레임에 연결한다.
9. 원격 컴퓨터_2에서 4-in-1 정합을 통한 리트로피팅이 이루어지는 동안, 원격 컴퓨터_1에서 다른 각도의 포인트 클라우드와 비디오 프레임을 취득하여 연속적인 리트로피팅 절차를 순차적으로 진행할 수 있게 한다.

Appendix D, UAV-AR Virtual Retrofitting on January 18, 2019 at KICT



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

< KICT : (a) & (b) UAV Calibration, (c) Water facility, (d),(e) & (f) Data acquisition using UAV LiDAR, (g) & (h) Point Cloud Data Registration and Retrofitting in Remote Computer -1 and 2>

Appendix E, Brief description of the virtual UAV-LiDAR retrofitting system S/W

Overview of application

1. Sensory UAV setup design for data acquisition (Partial point cloud and Video).
2. Communication setup to transfer acquired data from UAV's On-board Computer to Remote Computers.
3. Alignment of pre-processed point cloud data with acquired partial point cloud data.
4. Video alignment with pre-processed point cloud data.
5. Visualization of pre-processed point cloud data and intuitive retrofitting using CAD model with pre-processed point cloud data.

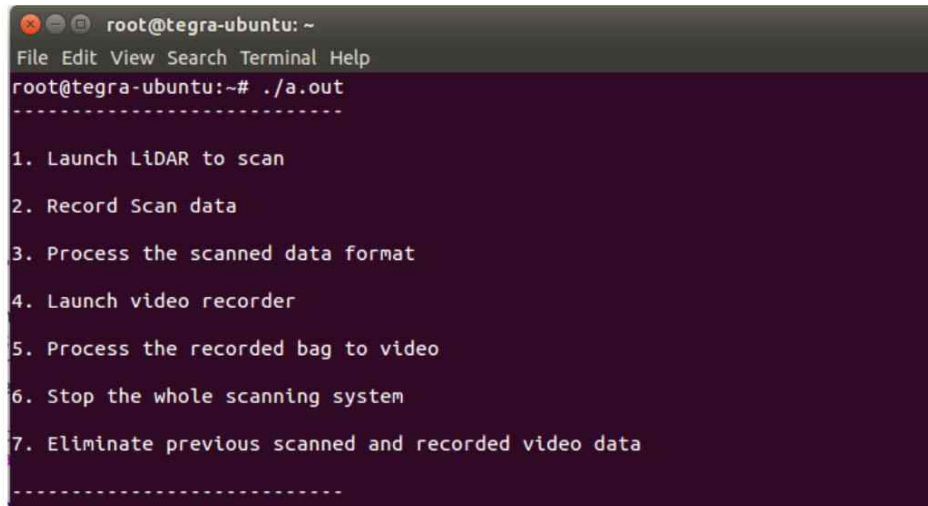
Major source file and functions used in application development

- **UAV on-board computer setup:** Communication setup to transfer acquired partial point cloud data and recorded video from UAV's On-board Computer to Remote Computers.
 - **VeloCamCapture.c**
 - Roslaunch – A Robot Operating System (ROS) functionality to launch multiple programs.
 - Rosrun rosbag record – To acquire partial point cloud data from LiDAR.
 - Rosrun pcl_ros bag_to_pcd – To convert acquired LiDAR partial point cloud data in .bag to .pcd file format.
 - Rosnode kill – To kill all the applications.
- **Remote Computer-1:** GUI interface application for point cloud alignment and communication setup between UAV's on-board computer with Remote computer-1.
 - **pclviewer.cpp**
 - loadFileButtonPressed - Load partial point cloud data acquired from LiDAR sensor
 - saveFileButtonPressed - Save aligned point cloud data to folder and from this folder Remote computer-2 gets aligned data.
 - ConnectButtonPressed- Create communication between UAV on-board Computer and Remote Computer-1
 - GetDataButtonPressed- Get partial scanned data and video from UAV On-board computer to Remote computer-1.
 - AlignModelData- Point cloud alignment function.
 - **OGICP.cpp**
 - alignPointCloud- Optimized Generalized-ICP Point cloud alignment function.
 - GetalignedCloud- Final output data of aligned preprocessed point cloud data.
 - **PointCloudVideoAlignment.cpp**
 - PCLVisualizer- Aligned point cloud visualization function
 - OpenCVRenderer- Video and point cloud alignment visualization function
- **Remote Computer-2:** Get aligned pre-processed point cloud from Remote Computer-1 for visualization and retrofitting using CAD models.
 - **Retrofitting_Application.cpp**
 - GetData - To get aligned pre-processed point cloud from Remote Computer-1.
 - pcdViz - To visualize aligned pre-processed point cloud.
 - Model1 - To retrofit CAD Model1 with aligned pre-processed point cloud.
 - Model2 - To retrofit CAD Model2 to increase water flow in existing water facility.

- Model3 - To retrofit CAD Model3 to reduce complexity and increase efficiency.
- OnMiddleButtonDown - When mouse middle button is pressed for picking of CAD models to retrofit.

User interface

- UAV On-board computer setup



```

root@tegra-ubuntu: ~
File Edit View Search Terminal Help
root@tegra-ubuntu:~# ./a.out
-----
1. Launch LiDAR to scan
2. Record Scan data
3. Process the scanned data format
4. Launch video recorder
5. Process the recorded bag to video
6. Stop the whole scanning system
7. Eliminate previous scanned and recorded video data
-----

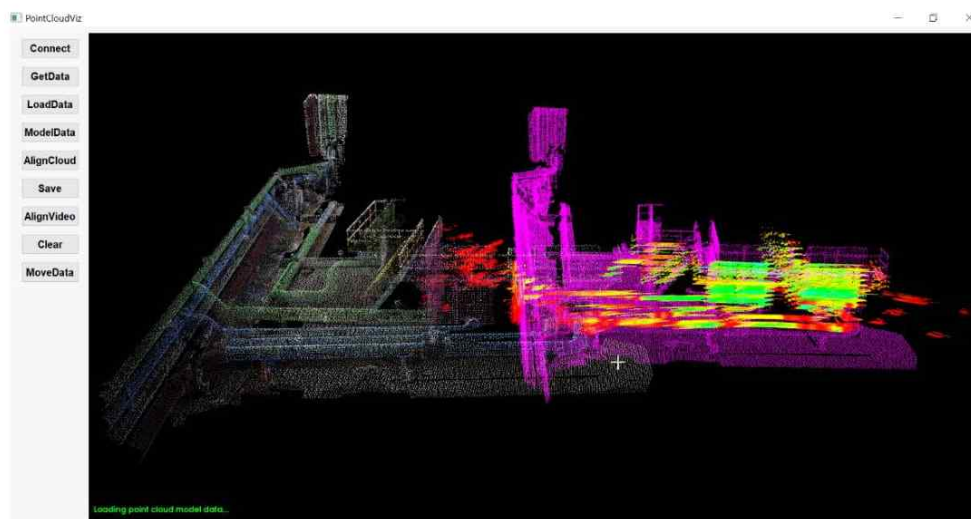
```

<Various inputs On-board computer for data acquisition>

Keyboard Input

1. To Launch LiDAR.
2. To record the partial point cloud data and convert into .pcd format.
4. To record Video from UAV camera.
5. To process recorded .bag into .mp4 format.
6. To stop whole scanning process.
7. To eliminate previous data.

- Remote Computer-1

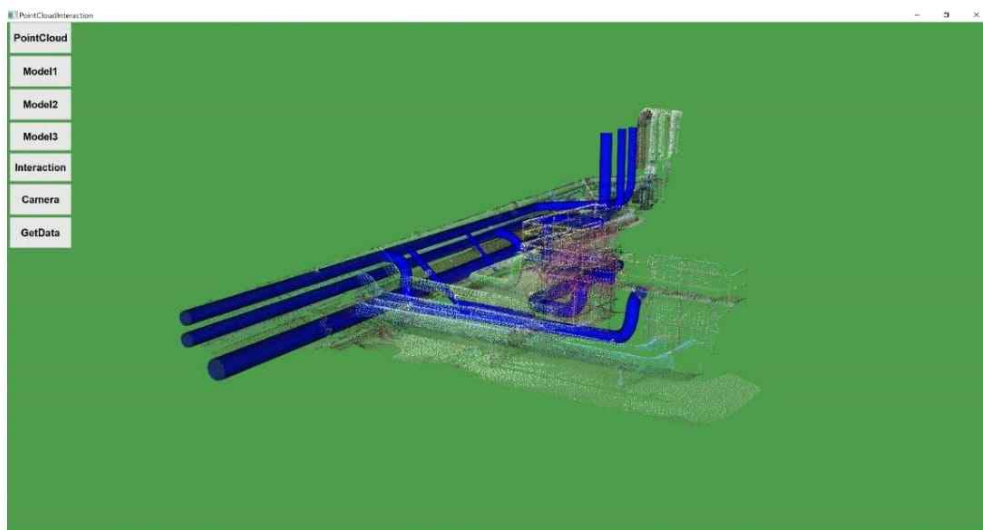


<Remote Computer-1 GUI>

- **Major Remote Computer-1 GUI buttons**

- Connect- Establish communication between UAV on-board computer and Remote computer-1
- GetData- Get LiDAR acquired partial point cloud data and UAV camera recorded video file.
- LoadData- Load partial point cloud data.
- ModelData- Load pre-processed point cloud data
- AlignCloud- Point cloud alignment algorithm using optimized GICP algorithm
- Save- Save aligned point cloud data for further process.
- AlignVideo- Video and point cloud alignment function

- **Remote Computer-2**



<Remote Computer-2 GUI>

- **Major Remote Computer-2 GUI buttons**

- PointCloud - To visualize aligned pre-processed point cloud.
- Model1 - To retrofit CAD Model1 with aligned pre-processed point cloud.
- Model2 - To retrofit CAD Model2 to increase water flow in existing water facility.
- Model3 - To retrofit CAD Model3 to reduce complexity and increase efficiency.
- Interaction - To perform interaction with CAD model by Rotating, Scaling and Translating to perform retrofitting.
- Camera - To perform interaction with camera to change and set the view according to requirement.
- GetData - To get Point Cloud Data from Remote Computer1.

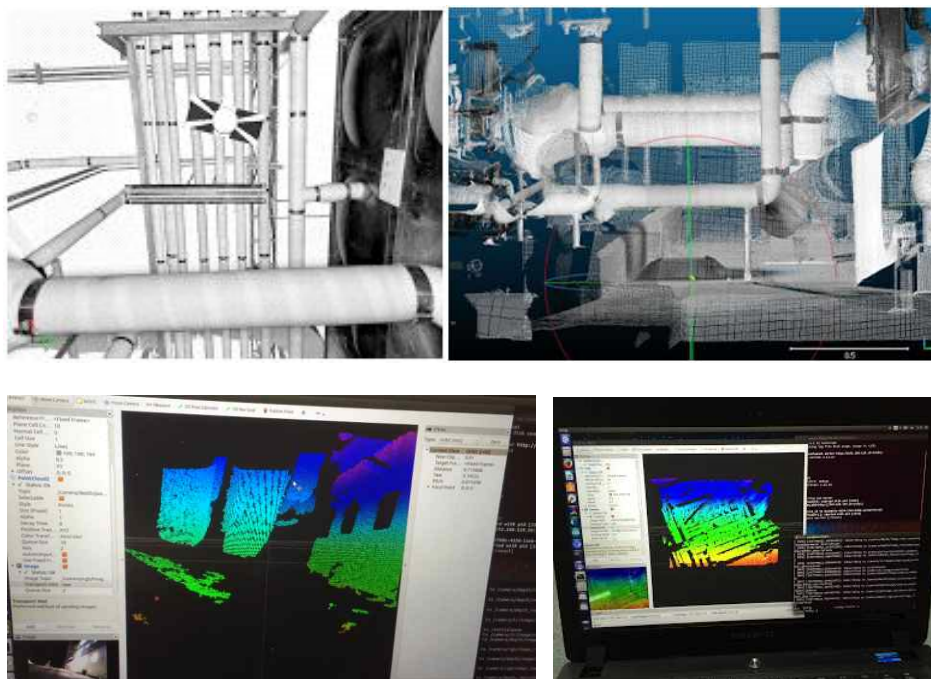
(5) 스캐닝 할 때 발생하는 사각지대에 대한 보완 방안: 로버 스캔 기반 이미지 스캔

- 3차원 스캔 시 대상의 크기나 위치가 너무 낮아 작업자가 진입하기 어려운 공간은 별도의 방법이 없으면, 스캔 데이터를 획득할 수 없음, 이 상황에서 역설계를 진행할 경우 올바른 산출물을 얻기가 매우 어려워 짐.
- 사람이 진입하기 어려운 부분도 LiDAR를 마운팅 할 수 있는 로버를 개발하여 적용한 한국건설기술연구원 ICT융합 연구실의 결과를 참조하여 해결할 수 있다. 개발한 로버 유형은 실내용 메카넘휠 기반, 실내용 4WD기반, 실외용 캐터필러 기반 로버 3종으로 LiDAR와 IR센서를 이용한 스캔 테스트를 완료한 상황임.



< 작업자 진입 불가 영역의 로버 기반 3차원 이미지 스캔, 메카넘휠, 캐터필러, 4WD 로버 기반 스캔 >

- 개발된 로버는 무선으로 위치나 방향을 제어할 수 있도록 되어 있으며, 로버의 스캔 시점 및 데이터도 원격으로 확인할 수 있도록 되어 있음. 이를 이용해, 스캔 테스트를 수행한 결과 다음 그림과 같이, 설비가 너무 복잡하고 낮아, 작업자 진입이 어려운 영역에 대해서 손쉽게 데이터 취득하는 것을 알 수 있음.



< 로버 기반 3차원 이미지 스캔의 포인트 클라우드

(상단=LiDAR 스캔, 하단 좌측=캐터필러 로버 기반 IR스캔, 하단 우측=4WD 로버 기반 IR스캔) >

제4장. 목표 달성도 및 관련 분야 기여도

1. 목표 달성도

가. UAV-LiDAR-HMD융합 시스템 개발: 100%

- (1) 융합시스템에 사용된 Pupil Lab. Eye-Tracker를 이용하여 전정안반사 알고리즘을 설계하고 응시점 좌표계산
- (2) 융합시스템에 사용된 Gimbal Camera의 가상좌표계 매핑을 위한 Onboard SDK 제어
- (3) 전정안반사 원리를 포함하는 Eye Simulation을 통해 UAV-HMD 융합시스템의 시뮬레이션
- (4) VoR을 포함한 응시점 좌표를 Gimbal 카메라를 위한 가상좌표계로 매핑하여 Onboard SDK를 통한 제어기를 통해 전시적인 UAV의 Airborne FPV로 구현

나. 방향에 따른 형상기술자 활용 및 UAV 스캔 위치 추정기술 개발: 100%

- (1) 사전 처리된 3D 포인트 클라우드 데이터를 랜덤 샘플링하고 공통 전역 좌표계로 매핑하여 실시간 스캔되는 부분 3D 포인트 클라우드와 중첩/정합하는 알고리즘 개발
- (2) 3D 포인트 클라우드의 파이프 곡률 추정과 실시간 스캔되는 부분 3D 포인트 클라우드와 중첩/정합으로 3D PCD 파이프의 곡률 정합
- (3) 중첩/정합된 3D 포인트 클라우드 데이터와 배경이 되는 비디오 영상을 정합하여 사실적인 가상 리트로피팅이 될 수 있는 배경을 시각화

다. 라이다장착 UAV 기반 실시간 가상 리트로피팅: 100%

- (1) UAV Manifold와 호스트 스테이션간의 데이터 통신을 통해 Velodyne RiDAR 스캐닝과 전경 비디오 프레임을 통합하는 VeloCamData 취득
- (2) PCD와 모델링 데이터를 VTK, OpenVR로 시각화 하는 가상 리트로피팅용 시각시스템
- (3) KICT 수처리 시설 가상 리트로피팅: 2018년 11/1: 1차 스캐닝 테스트, 11/23: 전체 시설 스캐닝, 2019년 1/18: 라이다장착 UAV 기반 스캐닝 포함 가상 리트로 피팅

2. 관련 분야 기여도

가. 기술의 시장성(현재 또는 미래 시장확대 전망)

- 객체 역설계 기술은 현재 시설물 유지관리, 모듈러 시공, 3D 프린팅, 현장 검측 등 관련 시장의 핵심적인 기술임, 예를 들어 시설물 유지관리와 관련된 시설물 관리 시스템 시장 규모만을 고려해 볼 때 관련 시장의 최소 5%를 차지할 경우 **2020년 3,400억원의 매출**이 예상됨.
- 국내 무인비행체 시장은 향후 15년간 약 1.62조원 규모로 확대될 것으로 전망되고 있으며, 항공우주연구원은 2022년까지 약 1천여 대의 수요가 발생될 것으로 예상하고 있으며, 국제무인시스템협회는 무인비행체 이용 규제가 완화되어 상업용 시장이 활성화 될 경우 향후 3년간 약 7만여 개의 새로운 일자리가 창출될 예상.

- 국토교통부는 **제2차 공간정보 산업진흥 기본계획**을 마련했다고 밝혔다.
공간정보산업은 수치지도, 지적도, 3차원(3D) 지도 등 공간 정보를 생산·유통하거나 다른 정보와 융복합해 제공하는 산업을 말한다. 특히 3차원 정밀지도에는 드론을 포함한 항공기가 안전하게 하늘을 날 수 있도록 송전탑 등 장애물 정보가 담겨 전면 개방될 예정.
- 수준 높은 ICT기술과 무인비행체기술, 객체 역설계 및 공간정보기술의 융합을 통하여 무인비행체 기반 **다양한 잠재적 사업화 영역이 창출**되고, 이를 통해 4차 산업혁명의 실현에 크게 이바지 하게 될 것으로 전망.

나. 경제적 기대성과(수입대체, 수출가능성)

- 플랜트 IT융합에 의해 플랜트 신축 비용의 8% 절감, 플랜트 프로젝트 공정기간의 14% 단축, **운영시의 보수 유지비용 5~15% 절감**효과가 기대되며, 플랜트산업에서 IT 기술의 접합 부족으로 생기는 비용손실은 미국 내에서 연간 158억 달러라는 보고가 있음.
- 가상현실 기술은 자원 및 에너지의 사용을 절감하는 **대표적인 녹색 기술** 가운데 하나로 현실 상황에서 불가능한 체험을 사용자에게 제공하기도 하지만, 현실 상황에서 충분히 가능한 체험을 적은 비용으로 구현한다는 점에서 녹색성장위원회로부터 녹색기술로서의 가치를 인정받음.

다. 국민복지, 안전 등 사회적 파급효과

- 장치산업의 특성상 설비 건설에 장기간이 소요되고, 장시간 연속적으로 가동해야 하는 특징을 갖고 있는 석유화학산업은 각별한 안전관리가 요구되며, 안전을 경영의 최우선 목표로 삼아 안전관리 전문 인력의 양성 및 확보, 노후화된 플랜트의 교체, 가동 중인 설비의 철저한 점검과 빈틈없는 유지보수, 정기점검 등 안전 확보에 나서고 있음.
- 그러나 울산석유화학공단에 업체 입주가 시작 된지 이미 34년이 경과하고 있으며, 여수공단도 26년이 지나 플랜트와 배관의 노후화가 가속화되고 있다. 플랜트는 1~2년에 1회 정도 가동을 중단하고 공장을 정비(Shut down)해 설비의 상당부분과 부속품에 개체가 이루어지고 있지만, 배관은 이렇다 할 보수 없이 노후화돼 사고 우려가 증대되는 상황.
- 울산석유화학단지 20년 이상 된 노후배관이 70.2km에 달하고 여러 가지 배관이 좁은 공간의 공용도로상에 엉켜 매설돼 사고발생시 인접배관에 영향을 미칠 가능성이 높다. 따라서 안전제고를 위해서는 정밀한 관리를 위한 지리정보시스템 구축, 각 시설물간 통합관리 방안 마련 및 가상 유지보수 시스템 등 과제들이 산적해 법과 제도의 정비가 시급한 실정.
- 본 과제를 통해 위험한 장치산업에서 **국민의 안전을 저해하는 요인을 사전에 제거할 수 있는 기술개발**이 이루어지며, 아울러 플랜트 운영 및 유지보수 기술의 국제 경쟁력 확보를 바탕으로 플랜트 기술 수출 증대 및 요소기술 수입절감 효과를 가져 올수 있다. 또한, 가상현실 기술의 산업화를 통하여 플랜트 IT 산업의 활성화, 수출증대 및 수입절감 효과가 있으며, 플랜트 유지보수의 효율화를 통한 비용절감 효과를 기반으로 **새로운 지식기반 플랜트 서비스를 창출**.

라. 플랜트 유지보수를 위한 VR/AR 트레이닝의 기대효과

 Safety • Promote safety, lower risk of mistakes and eliminate disruption of work with virtual training.	 Cost • Significantly reduced initial and ongoing costs of learning and development.
 Knowledge • Knowledge transfer & retention (captures enterprise knowledge).	 Value • High value for dangerous and remote working environments. (e.g. Oil Platform, FPSO, FLNG).
 Motivation • Higher levels of engagement and motivation, especially for today's learners, Interactive & Dynamic.	 Operation • Training can be started before the real plant is taken into operation.
 Proficiency • Reducing training time to proficiency by 50%.	 Performance • Increase in overall employee performance 40%.

<Key Benefits of Augmented Reality for Plant Maintenance>²⁾

- VR/AR을 이용한 경험 트레이닝은 실시간 피드백과 적극적이고 직접적인 참여유도, 효과적인 기술전수 및 습득기술의 손쉬운 유지로 빠른 학습을 달성할 수 있으며, 위험상황의 사전 반복노출로 교육자는 물론 장비의 안전에도 크게 기여.
- VR/AR을 통한 작업 중에 해당 유지보수나 트레이닝에 필요한, 적절한 자료, 지식을 사용자 앞에 다양한 형태로 보여줄 수 있게 하여 수준 높은 차원의 작업을 진행할 수 있게 해주며, 진행 내용/세션을 저장하면 추 후 재현이나 평가 및 재교육에 효율적으로 사용이 가능.

제5장. 연구개발성과의 활용계획

1. 접근이 용이하지 않은 시설물/구조물/플랜트의 증설, 복구 역설계 계획

- 시설물/에너지 유지관리/운영, 모듈러(Modular) 시공, Prefabrication, 시공 현장 검증, 시설물 개축 등에 필수적인 핵심 기술로 플랜트 시설물, 교량, 터널과 같은 구조물의 증설, 복구, 재건을 위한 역설계 계획수립에 활용.
- UAV에 열화상 검사 장비를 탑재하면 수작업 또는 접근성이 낮은 지역 파이프라인의 크랙, 누출 및 노후화를 탐지할 수 있어 원격으로 문제점을 확인 가능하고, UAV로 파악한 문제에 대한 해결책 제시를 위해 요소별 리트로피팅 절차 DB를 구축하고 작업 시뮬레이션 결과를 제시.

2. 기반시설/플랜트의 점검과 보수

- 건물, 댐, 도로, 교량 등 대형 구조물의 경우, 안전을 위해 정기적인 현장 점검이 필요하며 UAV를 통한 레이저 스캐닝을 이용하여 현재 구조물의 형태에 대한 정보를 수집한 후 설계치 또는 과거 정보와 비교함으로써 구조물의 변형, 균열 등 안전에 영향을 줄 수 있는 문제를 파악할 수 있고 보수 계획에 대한 시뮬레이션이 가능.



<CAD 데이터와 포인트 클라우드의 비교>

- 건물의 구조 및 실내의 결합여부조사뿐만 아니라 내부 공간의 효율적인 이용, 설비의 주기적인 점검과 교체를 포함한다. 레이저스캔 기술을 이용해 시설물 내부 이용실태에 대한 현황을 기록함으로써 공간 활용을 체계화할 수 있으며 시설물 교체 현황을 파악함으로써 유지 관리와 관련된 작업 일정을 효과적으로 관리.

3. 고위험 기반시설/플랜트의 유지보수용 재구성

- 고위험 시설물/플랜트 관리, 건물 노화로 인한 리노베이션 및 구조적 복구를 위해 역설계/재구성을 위해 As-built 모델의 가상, 입체 출력으로 모델을 확인하며 작업하여 정밀하고 신속한 수정, 업그레이드 작업이 가능.

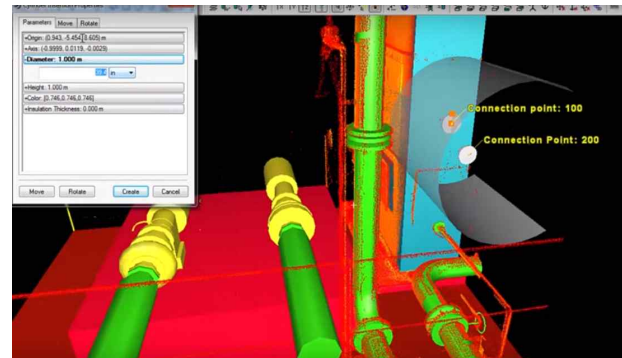


<고위험 기반시설/플랜트의 유지보수용 시뮬레이션을 위한 간략 시각화>

- 오일/가스/광업 산업의 압축기/펌프/터빈/열교환기/컨베이어 등의 유지보수를 위해서는 무엇보다도 그 정지시간을 최소화 하는 것이 절실하다. 미래에는 유지보수작업을 위한 인터페이스가 시각적이고 분석적이며 동시에 상호작용이 가능하고 모바일 환경으로도 제공될 것이다. 유지보수용 3D 시각화는 보수시간을 줄여 정지시간을 최소화하며, 유효적절한 절차를 통해 생산성을 높이고, 전체 절차를 용이하게 숙지할 수 있게 하여 작업자의 안전을 도모함.

4. 시뮬레이션을 통한 신규 운전/정비 인력 교육훈련

- 플랜트 운영 및 유지보수 시장에서 고령화된 운전원의 대체 인력인 미숙련 신규 운전원 교육 훈련에 저비용 고효율 교육 효과를 얻을 수 있음. 이 기술은 해외 선진 업체에서도 개발 단계에 있으므로 국내의 요소 기술을 통합하여 실증을 거쳐 패키지화하면 해외 시장 진출도 충분히 가능함. 예를 들어, 도시가스 정압 설비 및 석유화학 플랜트에 VR 안전훈련시스템을 적용 후 성능 검증하여 시스템 실용화에 활용이 가능.



<비마커 증강현실을 적용한 간섭 시뮬레이션>

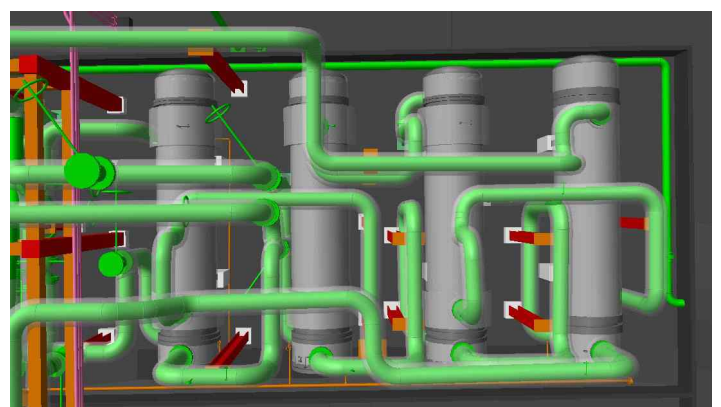
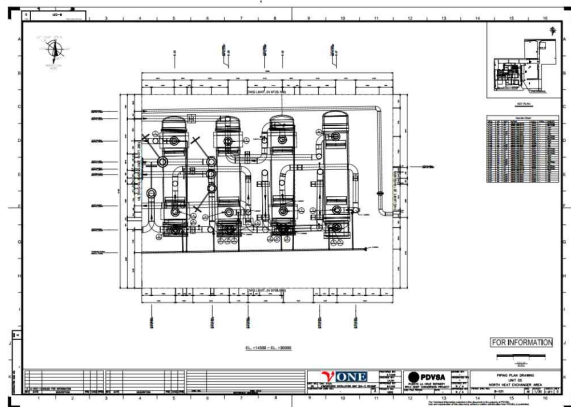
5. (주)제이텍솔루션의 개발성과 활용계획 및 마케팅전략

- 최종평가 사전점검 후 11월 19일(월) 참여기업인 (주)제이텍솔루션에 방문하여 개발성과 활용 계획 논의, 참여업체는 진행되는 스캔 및 역설계 절차에 개발결과를 순차적으로 적용시킬 예정이며 필드 테스트를 진행한 후에 수정사항을 보완하여 신규과제 수주에 개발된 리트로피팅 절차를 직접 이용할 뿐 아니라, H/W + S/W 솔루션 형태로 관련 기업 마케팅을 진행할 예정임.
- 참여업체는 ① 포스코철 No. 3 Duct 스캔 예정 ② YNCC는 입찰중이며 2019년 상반기 작업 예정 ③ 대림여수 화학공장은 사업 타당성 조사 준비 중이며 2019년 상반기 작업 예정 ④ 포스코건설 포항 3고로 개보수 작업 스캔 입찰중이며 2019년 상반기 작업예정

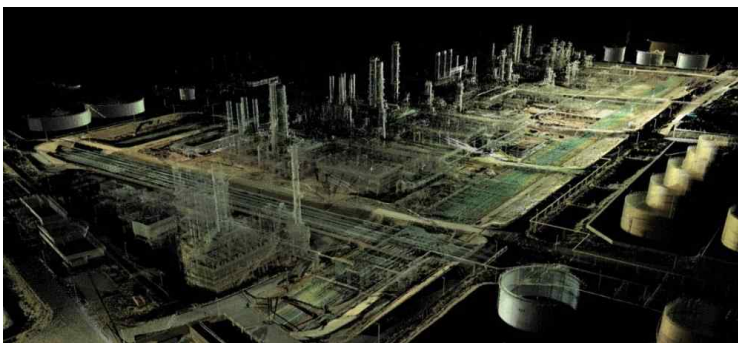
- YNCC Naphtha Cracking Center의 Ethylene Plant와 BTX Plant에는 수 개의 리트로피팅 공정을 포함하고 있으며, 아래 그림의 (주)제이텍솔루션 기존 리트로피팅 수행 작업의 사례를 통해 다양한 형태의 리트로피팅 요소의 필요성을 보여줌.



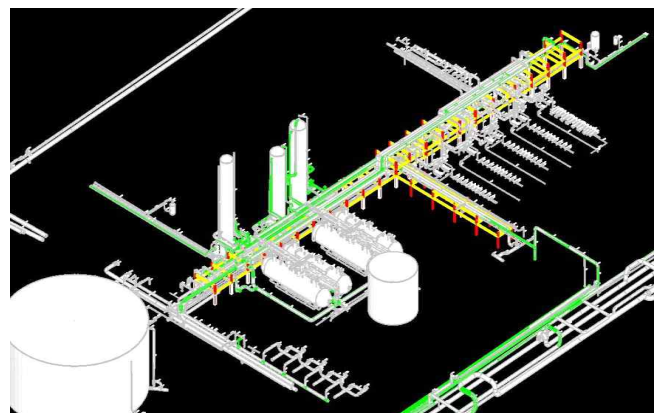
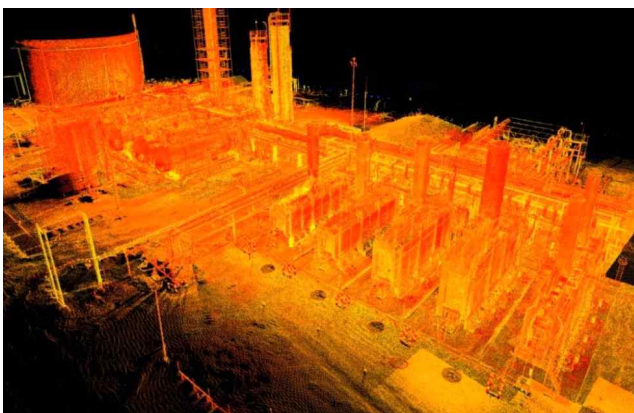
<CloudWorx for S3D를 Pipe/Equipment Modeling: Insulation의 설계 및 적용>



<S3D를 이용한 Drawing Extraction>



<RPLC Deep Conversion Project: 배관, 기계, 구조, Support 모델링, ISO, GA Drawing 생성, 4MM/4달>



<MCG Revamp Project: 배관, 기계, 구조 모델링, GA, 구조 Drawing 생성, 8MM/3달>

제6장. 연구 과정에서 수집한 해외 과학기술 정보

1. 3D 스캐닝 기반 플랜트 3D 모델 생성 분야 특성

- 세계적으로 '70~80년대에 건설된 노후 플랜트 및 가동 플랜트의 개선사업에 플랜트 3D CAD 모델 구축이 필수적으로 요구되고 있어 이를 지원하는 기술에 대한 수요가 높음
 - 3D 스캐닝 기술은 플랜트의 시공, 운영, 유지보수 과정에서 3D 모델 생성에 필요한 플랜트 시설물들의 치수 실측작업을 광학적 방법으로 대신하여 업무 생산성 및 작업성 향상
 - 플랜트의 설계, 시공, 운영관리, 확장 및 교체 과정에서 3D모델 생성, 도면비교, 조립성 검사, 정도 검사, 간섭 검사 등의 다양한 작업에 널리 활용
- 일반 3D 스캐닝 장비 및 관련 SW는 상업화가 많이 진척됐으나 이를 3D CAD 시스템과 연계하여 실제 3D 모델의 생성 및 활용 작업은 참고 데이터 및 체계적인 지원 시스템의 부재로 실제 작업에 많은 인력·시간이 소요되고 있음
- 3D 스캐닝 기반 플랜트 3D 모델 생성 및 활용 서비스는 인력의 숙련도가 높은 우리나라에 적합한 고부가가치 산업으로 성장할 가능성이 높으므로 관련 국내 중소·중견 기업의 기술력 및 경쟁력 향상을 위한 지원 시스템 개발을 정책적으로 지원할 필요가 있음

2. 개발 단계의 경쟁사 제품

- EdgeWise 5.0은 ClearEdge3D(<http://www.clearedge3d.com/>)사의 개발 프로그램으로 파이프, 구조물, 덕트, 실내 공간 모델링을 할 수 있는 소프트웨어이다. 포인트 클라우드 데이터를 사용하여 파이프 및 특징을 자동으로 추출하여 모델링하며 이외의 실내 공간 모델링을 위한 벽, 창문 자동 추출 등의 기능을 제공하고 있다. 파이프 자동화 모델링의 경우 90%이상의 정확도를 가지고 있으며 자동화를 통해 모델링 시간을 70%이상 줄일 수 있다고 한다. 이 소프트웨어는 포인트 클라우드 데이터를 사용하여 형상 속성 정보를 추출하거나 입력된 속성 정보를 통해 모델링을 하기에 실제 건축 시공에 사용하는 객체의 정보를 가지고 있지 않다는 단점이 있다. 최근 BIM을 사용하여 3차원 가상공간에서 실제로 건물을 짓는 것과 같은 방식으로 3차원 모델링을 하여 실제 설계에 발생할 수 있는 여러 문제점을 사전에 검토하는 사례가 많아지고 있다. EdgeWise 5.0은 BIM을 사용한 모델링이 아닌 형상 속성 정보를 통한 모델링을 제공하기에 시설물 유지관리의 편의성이 다소 낮다는 단점이 있음



<Pipe modeling in EdgeWise 5.0>

제7장. 연구개발성과의 보안등급

- 일반과제

제8장. 국가과학기술종합정보시스템에 등록된 연구시설·장비 현황

구입 기관	연구시설/ 연구장비명	규격 (모델명)	수량	구입 연월일	구입 가격 (천원)	구입처 (전화번호)	비고 (설치 장소)	NTIS장비 등록 번호

제9장. 연구개발과제 수행에 따른 연구실 등의 안전 조치 이행 실적

1. 연구실 안전관리 개요

: 교내 연구실 안전사고를 예방하고 연구실의 안전성을 확보하고자 연구실안전관리규정에 의거 연구실 정기점검 및 정밀안전진단실시, 연구활동종사자의 교육 및 검진을 통한 안전성 확보, 사고 발생시 초동대응을 통한 인적, 물적 피해를 최소화 하고자함.

2. 연구실 안전관리 조직 및 담당업무

구 분	담당 업무
총장 및 부총장	연구실 안전관리 총괄
연구환경안전관리자 (시설관리처 안전관리팀)	- 연구(실험)실 안전관리 총괄 관리 - 연구활동종사자 안전교육 운영 관리 - 상시 연구활동종사자 건강검진 진행 - 연구(실험)실 안전점검 진행 (정기점검, 정밀안전진단) - 관련법, 안전수칙 교육 및 홍보 - 법 이행사항 보고 외부 점검 수행
대학 연구환경담당자 /학과담당자	- 계열 연구(실험)실 안전관리 관리 - 연구활동종사자 안전교육 홍보 및 이수현황 관리 - 상시 연구활동종사자 건강검진 홍보 및 현황 관리 - 연구(실험)실 안전점검 관리(정기점검, 정밀안전진단) - 관련법, 안전수칙 교육 및 지도 - 각종 폐기물 처리 지침 준수 및 교육

연구실안전관리자 (정,부)	• 연구(실험)실 실별 안전관리 총괄 관리 • 연구활동종사자 안전교육 홍보 및 이수현황 관리 • 상시 연구활동종사자 건강검진 홍보 및 현황 관리 • 연구(실험)실 안전점검 대비 일지 작성 • 관련법, 안전수칙 교육 및 지도 • 각종 폐기물 처리 지침 준수 및 교육 • 위험물 및 유해인자별 MSDS 비치 및 교육 • 각종 폐기물 처리 지침 준수 및 교육
(상시)연구활동종사자	• 연구실 안전교육 및 건강검진 참여 • 연구(실험)실 안전점검 실시 • 각종 폐기물 처리 지침 준수 폐기 • 위험물 및 유해인자별 MSDS 확인 후 실험

3. 2018년 연구실안전관리 이행 결과

- 1) 연구실 위험등급별 조정 및 조정에 따른 안전 개선 조치
- 2) 연구실 등급별 정밀안전진단 및 정기안전점검 실시
- 3) 단과대학별 안전용품 수요조사를 실시하여 위험요소에 따른 안전용품 지급
- 4) 연구실 사전유해인자 위험분석 실시 및 연구개발활동별 R&DSA 실시
- 5) 2018년 연구실 안전 업무 진행 상세일정

일자	내용	세부사항	비고
매월 시행중	Clean & Safety day 실시	매월 자체 안전점검 실시 안전점검 시 필요 사항 즉시 개선	
매 학기 시행중	사전유해인자 위험분석 실시	신규 과제를 수주할때마다 R&DSA를 포함하여 실시	
2018년 4월	연구실 안전공제보험 갱신가입	과학기술분야 재학생 및 상시연구활동 종사자 대상	
2018년 4월	연구실 정밀안전진단 실시	외부 전문 업체 위탁 진단 실시	
2018년 5월	연구실 정밀안전진단 결과 개선	각 연구실 지적사항 개선 실시	
2018년 4월	연구실 LMO 자체안전점검 실시	서울/안성 동시 실시, 하반기 11월 점검	학기별 실시
2018년 7월	연구실 건강검진(특수검진) 실시	상반기 특수검진 실시, 하반기 12월 실시	학기별 실시
2018년 3월	1학기 온라인 안전 교육 실시	과학기술분야 학부 재학생(주전공자), 대학원생, 수료생 및 연구원 대상	학기별 실시
2018년 6월	연구실 안전관련 집기설치 및 시설 개선	위험물 보관 캐비닛 및 안전관련 집기 수요조사후 설치 및 개선	
2018년 6월	연구실(생물)안전관리 위원회 개최	연구실 안전관리 현안에 대한 검토, 하반기 위원회 12월 실시	학기별 실시
2018년 10월	연구실 정기 안전진단 실시	2018년 10월 말 실시	
2018년 12월	연구실 건강검진(특수검진)실시	하반기 특수검진 12월 실시	
2018년 9월	2학기 연구실 안전교육 실시	과학기술분야 학부 재학생(주전공자), 대학원생, 수료생 및 연구원 대상	학기별 실시

제10장. 연구개발과제의 대표적 연구 실적

번 호	구분 (논문/ 특허/ 기타)	논문명/특허명/기타	소속 기관명	역할	논문 게재지/ 특허 등록 국가	영향력 지수	논문 게재일 /특허 등록일	사사 여부 (단독 사사 또는 중복 사사)	특기 사항 (SCI 여부/인용 횟수 등)
1	논문	Hybrid approach for alignment of pre-processed 3D point cloud, video and CAD model using partial point cloud in retrofitting application	중앙대 첨단영상 대학원	주저자, 교신 저자	International Journal of Distributed Sensor Networks	1.787	2018. 03.22	중복사사	SCIE
2	논문	Gaze-Controlled Virtual Retrofitting of UAV-Scanned Point Cloud Data	중앙대 첨단영상 대학원	주저자, 교신 저자	Symmetry	1.256	2018. 11.29	중복사사	SCIE
3	논문	Inspired by Human Eye: Vestibular Ocular Reflex Based Gimbal Camera Movement to Minimize Viewpoint Changes	중앙대 첨단영상 대학원	주저자, 교신 저자	Symmetry	1.256	2019. 01.16	중복사사	SCIE
4	논문	파이프 라인 리트로피팅 어플리케이션을 위한 3D 포인트 클라우드, CAD 모델, 실시간 카메라 뷰 및 부분 포인트 클라우드의 4대 1 정렬	중앙대 첨단영상 대학원	주저자, 교신 저자	대한 기계학회 논문집 A		2018. 08.01	단독사사	SCOPUS
5	논문	안구 운동을 적용한 FPS게임의 이중 응시	중앙대 첨단영상 대학원	주저자, 교신 저자	정보과학회 논문지: 소프트웨어 및 응용		편집 위원 평가중	중복사사	KCI등재지
6	특허	객체의 가상 리트로피팅을 위한 정렬 데이터 생성 방법 및 이를 수행하는 단말 장치	중앙대 첨단영상 대학원	발명자	대한민국		2018. 08.10	단독기여	10-2018-0093937
7	특허	비디오를 이용하여 객체의 가상 리트로피팅을 수행하기 위한 정렬 데이터 생성 방법 및 이를 수행하는 단말 장치	중앙대 첨단영상 대학원	발명자	대한민국		2018. 08.10	중복기여	10-2018-0093938
8	소프트 웨어 등록	역설계 계획용 가상 리트로피팅 시스템	중앙대 첨단영상 대학원	저작자	대한민국		2019. 01.28	중복기여	C-2019-002725

제 11장. 기타 사항

Abbreviations:

- **UAV(Unmanned Aerial Vehicle)**: 인간 조종사 탑승 없이 운행하는 비행체로 UAV는 지상 조정장치와 이들 사이의 통신을 포함한다.
- **FoV(Field of View)**: 어느 순간에 관측 가능한 시야의 범위를 말하며, 인간 비전의 경우 보조시각장치에 의해 제한될 수 있다.
- **G-ICP(Generalized Iterative Closest Point)**: 반복근접포인트 알고리즘과 point-to-plane ICP 알고리즘을 조합한 방법이다.
- **O-GICP(Optimized-Generalized Iterative Closest Point)**: 정합시간을 최소화하기 위해서 사전 최적화를 포함한 G-ICP 알고리즘이다.
- **ROI(Region of Interest)**: 특별한 목적을 위해 정의(샘플링)되는 전체 영역중의 일부 관심영역이다.
- **PFH(Point Feature Histogram)**: 다차원 히스토그램 값으로 점 주위 평균곡률을 일반화하여 k-이웃의 형상속성을 정의하기 위해 사용한다.
- **FPV(First-Person View)**: 무선조종장치의 조종을 위한 조종자의 1인칭 시점으로 사용된다.
- **LiDAR(Light Detection and Ranging)**: 레이저 펄스를 쏘고 반사되어 돌아오는 시간을 측정하여 반사체의 위치좌표를 측정하는 레이더 시스템이다.
- **ROS(Robot Operating System)**: 로봇 운영체제는 로봇 응용 프로그램을 개발할 때 필요한 하드웨어 추상화, 하위 디바이스 제어, 일반적으로 사용되는 기능의 구현, 프로세스간의 메시지 패싱, 패키지 관리, 개발환경에 필요한 라이브러리와 다양한 개발 및 디버깅 도구를 제공한다.
- **VoR(Vestibulo-Ocular Reflex)**: 전정안반사는 머리 운동의 반대 방향으로 그에 상응하는 안구 운동을 일으켜 개체가 움직이는 중에도 물체의 상이 망막에 안정되게 머물러있게 한다.
- **UART(Universal Asynchronous Receiver Transmitter)**: 범용 비동기화 송수신기는 병렬 데이터의 형태를 직렬 방식으로 전환하여 데이터를 전송하는 컴퓨터 하드웨어의 일종이다.
- **PCD(Point Cloud Data)**: 점구름은 어떤 좌표계에 속한 점들의 집합이다. 3차원 좌표계에서 점은 보통 X, Y, Z 좌표로 정의되며 종종 사물의 표면을 나타내기 위해 사용되기도 한다. 점구름은 3D 스캐너로 얻을 수 있다.
- **SDK(Software Development Kit)**: 소프트웨어 개발 키트는 일반적으로 소프트웨어 기술자가 사용하여 특정한 소프트웨어 꾸러미, 소프트웨어 프레임워크, 하드웨어 플랫폼, 컴퓨터 시스템, 게임기, 운영 체제 등을 위한 응용 프로그램 등을 만들 수 있게 해주는 개발 도구의 집합이다.
- **API(Application Programming Interface)**: API는 응용 프로그램에서 사용할 수 있도록, 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 뜻한다. 주로 파일 제어, 창 제어, 화상 처리, 문자 제어 등을 위한 인터페이스를 제공한다.
- **SSH(Secure Shell)**: 시큐어 셸(Secure Shell, SSH)은 네트워크 상의 다른 컴퓨터에 로그인하거나 원격 시스템에서 명령을 실행하고 다른 시스템으로 파일을 복사할 수 있도록 해 주는 응용 프로그램 또는 그 프로토콜을 가리킨다.
- **VTK(Visualization ToolKit)**: 컴퓨터그래픽스, 영상처리 및 시각화 등에 사용되는 오픈소스 S/W 시스템이다.
- **SDL(Simple DirectMedia Layer)**: SDL는 C 프로그래밍 언어로 짜여진 크로스플랫폼 멀티미디어 라이브러리이다. 이 라이브러리는 비디오, 오디오, 사용자 입력 등의 계층을 추상화하여, 리눅스, 마이크로소프트 윈도우, 맥 OS X 등 여러 운영 체제에서 실행이 가능하도록 한다.

제12장. 참고 문헌

1. Kassner, Moritz, William Patera and Andreas Bulling, "Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction," In Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication, pp. 1151-1160, ACM, 2014.
 2. Hintjens, Pieter, ZeroMQ: messaging for many applications, " O'Reilly Media, Inc.", 2013.
 3. Dempsey, Paul, "The teardown: HTC Vive VR headset." Engineering & Technology 11, no. 7-8: 80-81, 2016.
 4. Smith, Mason and Ann McNamara, "Gaze Direction in a Virtual Environment Via a Dynamic Full-Image Color Effect," In 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 1-2. IEEE, 2018.
 5. Matrice, D. J. I. "100, 2016." URL <http://www.dji.com/product/matrice100>, 2017.
 6. Svoboda, Bc Adam, "Advanced payload control on DJI platforms using embedded on-board computer," 2018.
 7. DJI, DJI, "Mobile SDK Documentation," 2015-05-26[2016-09-10], <http://developer.dji.com/mobile-sdk/documentation/introduction/index.html>.
 8. Trimble, http://trl.trimble.com/docushare/dsweb/Get/Document-628869/022504-122_Trimble_TX5_DS_1012_LR.pdf (accessed 06 June 2018).
 9. Velodyne LiDAR, <http://velodynelidar.com/vlp-16.html> (accessed 06 June 2018).
 10. Segal A, Haehnel D, Thrun S. Generalized-ICP. In Robotics: Science and systems, Vol. 2, No. 4, p. 435, 2009.
 11. Aldoma, A., Marton, Z.C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R.B., Gedikli, S. and Vincze, M., Point cloud library, IEEE Robotics & Automation Magazine, 1070(9932/12), 2012.
 12. Rusu, R. B. (2010). Semantic 3D object maps for everyday manipulation in human living environments. KI-Künstliche Intelligenz, 24(4), 345-348.
 13. Rusu, R. B., Blodow, N., & Beetz, M., Fast point feature histograms (FPFH) for 3D registration, In Robotics and Automation, ICRA'09. IEEE International Conference on, pp. 3212-3217, 2009.
 14. Martin, V., M. Zdenk Michal and H. Adam, "Calibration of rgb camera with velodyne lidar," In International Conference on Computer Graphics, Visualization and Computer Vision. 2014.
 15. MTw Awinda, <https://www.xsens.com/products/mtw-awinda/> (accessed 06 June 2018)
 16. Collinearity equation, https://en.wikipedia.org/wiki/Collinearity_equation.
 17. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A.Y., ROS: an open-source Robot Operating System, In ICRA workshop on open source software, Vol. 3, No. 3.2, p. 5, 2009.
 18. LiDAR, V., VLP-16 In VLP-16 Manual: User's Manual and Programming Guide; Velodyne LiDAR, Inc.: San Jose, CA, USA, 2016.
 19. Gray J. DJI Manifold.
 20. BRIEF, P., Intel® Dual Band Wireless-N 7260, 2013.
 21. http://wiki.ros.org/dji_sdk.
-

9. 뒷면지

주 의

1. 이 최종보고서는 국토교통부에서 시행한 국토교통기술촉진연구사업의 연구보고서입니다.
2. 이 최종보고서 내용을 발표하는 때에는 반드시 국토교통부에서 시행한 사업의 연구개발성과임을 밝혀야 합니다.
3. 국가과학기술 기밀 유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 안 됩니다.

210mm×297mm[백상지(80g/㎡) 또는 중질지(80g/㎡)]