

[붙임5] 2026년 소·중·대 산학협력프로젝트(캡스톤디자인) 결과보고서

2026년 전남대학교 소프트웨어중심대학사업 소·중·대 산학협력프로젝트(캡스톤디자인) 결과보고서

프로젝트명	VLM 기반 설명가능 자율주행 모빌리티 로봇 및 웹 관제 서비스 개발					
Github url 주소	https://github.com/Donok53/Modular_Approach_Autonomous_Driving-					
팀 명	생각하는 바퀴			과제수행기간	2026. 3. 25. ~ 6. 23.	
지도교수	학 과	미래모빌리티융합학과		성 명	김찬수	
프로젝트 수행인원 (※팀장은 첫줄에 기입)	학과(부·복수전공)	학년	학번	성명	연락처	E-Mail
	인공지능학부	4	210736	고병재	010-2794-2686	210736@jnu. ac.kr
	인공지능학부	4	214216	김주완	010-6428-8134	kimjy0700a@na ver.com
	인공지능학부	4	212329	이승원	010-2653-9688	212329@jnu. ac.kr
참여 기업	기업명	멘토명		직위	연락처(HP)	E-Mail
	(주)크리앤비즈	주태경		대표이사	010-5343-8657	jootk2000@na ver.com

위와 같이 2026년 전남대학교 소프트웨어중심대학사업
산학협력프로젝트(캡스톤디자인) 지원 프로그램 결과보고서를 제출합니다.

2026년 6월 17일

신청자명(대표학생) :

고병재 (인)

지도교수 :

김찬수 (인)

전남대학교 소프트웨어중심대학사업단장 귀하

[붙임5] 2026년 소·중·대 산학협력프로젝트(캡스톤디자인) 결과보고서

2026년 전남대학교 소프트웨어중심대학사업 소·중·대 산학협력프로젝트(캡스톤디자인) 결과보고서

프로젝트명	VLM 기반 설명가능 자율주행 모빌리티 로봇 및 웹 관제 서비스 개발					
Github url 주소	https://github.com/Donok53/Modular Approach Autonomous Driving-					
팀 명	생각하는 바퀴			과제수행기간	2026. 3. 25. ~ 6. 23.	
지도교수	학 과	미래모빌리티융합학과		성 명	김찬수	
프로젝트 수행인원 (※팀장은 첫줄에 기입)	학과(부·복수전공)	학년	학번	성명	연락처	E-Mail
	인공지능학부	4	210736	고병재	010-2794-2686	210736@jnu.ac.kr
	인공지능학부	4	214216	김주완	010-6428-8134	kimjy0700a@naver.com
	인공지능학부	4	212329	이승원	010-2653-9688	212329@jnu.ac.kr
참여 기업	기업명	멘토명		직위	연락처(HP)	E-Mail
	(주)엠에이치종합 컨설팅	주태경		대표이사	010-5343-8657	jootk2000@naver.com

위와 같이 2026년 전남대학교 소프트웨어중심대학사업
산학협력프로젝트(캡스톤디자인) 지원 프로그램 결과보고서를 제출합니다.

2026년 6월 17일

신청자명(대표학생) :

고병재 (인)

지도교수 :

김찬수 (인)

전남대학교 소프트웨어중심대학사업단장 귀하

산학협력프로젝트(캡스톤디자인) 결과보고서(요약)

프로젝트명	VLM 기반 설명가능 자율주행 모빌리티 로봇 및 웹 관제 서비스 개발		
수행기간	2026. 3. 25. ~ 6. 23.	소요예산	256,440
소요예산	-회의비 171,200		
세부내역	-재료비 85,240		
참여인원	구분	인원수	성명(모두 기재)
	교수	1	김찬수
	석박사과정	-	
	학부생	3	고병재, 김주완, 이승원
	기업체	1	주태경
	계	5	
추진배경	- 실외 모빌리티 로봇 서비스 확대 - 자율주행 기술이 차량 중심에서 배달, 순찰, 안내 등 실외 모빌리티 로봇 서비스로 확대되고 있다. - 복잡한 실외 환경 대응 필요 - 실외 환경은 보행자, 차량, 임시 장애물 등 변화 요소가 많아 안정적인 위치 추정과 장애물 회피 판단이 필요하다. - 설명가능성과 관제 기반 신뢰성 확보 - 기존 자율주행 시스템은 판단 근거를 사용자가 이해하기 어려우므로, VLM 기반 설명 기능과 웹 관제를 통한 신뢰성 확보가 요구된다.		
목표 및 내용	- 로봇 플랫폼 구성 - Scout mini 로봇 차체에 LiDAR, IMU, 카메라, 연산 PC를 탑재하여 실외 자율주행, VLM 추론, 웹 관제 연동이 가능한 하드웨어 환경을 구성한다. - 실외 자율주행 및 회피 기능 구현 - LiDAR와 IMU 기반 위치 추정, 전역 지도 생성, 경로 추종 기능을 통합하고, 주행 중 장애물이 감지될 경우 회피 경로 생성 또는 정지 판단을 통해 안전한 주행을 수행한다. - VLM 기반 주행 판단 설명 생성 - 카메라 영상과 로봇의 주행 판단 정보를 활용하여 장애물 종류, 위치, 회피 방향 등을 자연어 설명 문장으로 생성한다. - 웹 관제 서비스 구축 - 로봇의 주행 상태, 3D 포인트클라우드, 미니맵, 카메라 영상, VLM 설명 로그를 웹 화면에서 실시간으로 확인할 수 있도록 구현한다.		
기대효과	- 다양한 실외 서비스로 확장 가능 - 개발한 자율주행 로봇 플랫폼은 배달, 순찰, 안내, 물류 운반 등 다양한 실외 모빌리티 서비스에 적용할 수 있다. - 자율주행 판단 신뢰성 향상 - VLM 기반 설명을 통해 로봇이 어떤 객체를 인식했고 왜 회피 또는 정지 판단을 수행했는지 사용자가 쉽게 이해할 수 있다. - 웹 관제를 통한 운영 편의성 향상 - 로봇의 주행 상태, 카메라 영상, 장애물 인식 결과, 설명 로그를 원격에서 확인하여 주행 모니터링과 사후 분석에 활용할 수 있다.		

1. 프로젝트 개요

프로젝트명	VLM 기반 설명가능 자율주행 모빌리티 로봇 및 웹 관제 서비스 개발
주제영역	☐ 생활 ☐ 업무 ☒ 공공/교통 ☐ 금융/핀테크 ☐ 의료 ☐ 교육 ☐ 유통/쇼핑 ☐ 엔터테인먼트
기술분야	☐ IoT ☐ 모바일 ☐ 데스크톱 SW ☒ 인공지능 ☐ 보안 ☐ 가상현실 ☐ 빅데이터 ☒ 자동제어기술 ☐ 블록체인 ☒ 영상처리 ☐ 기타()
성과목표	☒ 논문게재 및 포스터발표 ☐ 앱등록 ☐ 프로그램등록 ☐ 특허 ☐ 기술이전 ☐ 실용화 ☐ 공모전(<i>공모전명</i>) ☐ 기타()

2. 프로젝트 추진배경

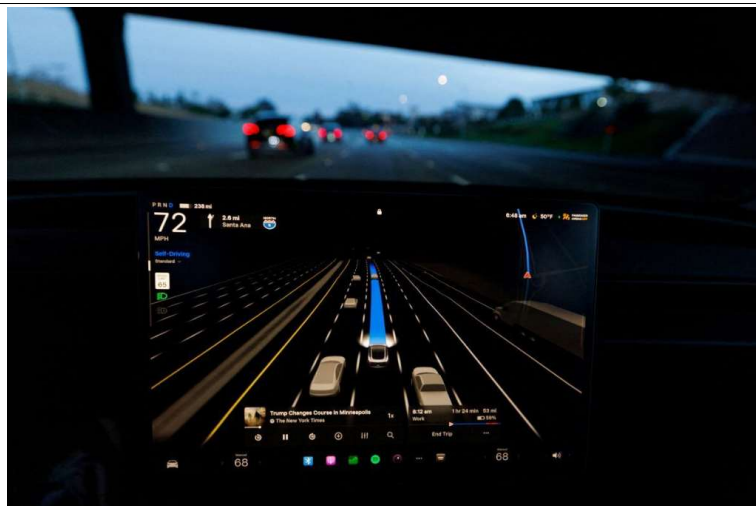
2-1 개발동기

최근 자율주행 기술은 차량 중심의 고속도로 주행이나 주차 보조 기능을 넘어, 물류 로봇, 배달 로봇, 순찰 로봇 등 다양한 모빌리티 서비스 영역으로 확장되고 있다. 특히 실외 환경에서 운용되는 자율주행 로봇은 반복적인 이동 업무를 자동화하고, 사람이 직접 수행하기 어렵거나 비효율적인 이동 작업을 대체할 수 있다는 점에서 높은 활용 가능성을 가진다. 이에 따라 자율주행 기술은 단순히 정해진 경로를 따라 이동하는 수준을 넘어, 복잡한 주변 환경을 이해하고 상황에 맞게 안전하게 판단하는 방향으로 발전하고 있다.

그러나 실제 실외 환경은 실내나 정형화된 주차장 환경에 비해 훨씬 더 복잡하고 예측하기 어렵다. 보행자, 차량, 적재물, 임시 장애물, 주차된 차량 등 다양한 요소가 수시로 변화하며, 로봇은 이러한 환경 속에서 안정적으로 주행하고 필요한 경우 적절한 회피 판단을 수행해야 한다. 따라서 실외 자율주행 로봇에는 위치 추정과 경로 추종뿐만 아니라, 주변 상황을 인식하고 변화하는 환경에 대응할 수 있는 주행 판단 능력이 요구된다.

기존 자율주행 시스템은 주로 경로 생성, 위치 추정, 장애물 회피와 같은 주행 기능 자체에 초점을 맞추어 발전해 왔다. 하지만 이러한 시스템은 로봇이 어떤 대상을 인식했는지, 왜 특정 방향으로 회피했는지, 또는 왜 정지 판단을 내렸는지에 대한 근거를 사용자가 직관적으로 이해하기 어렵다는 한계가 있다. 즉, 자율주행 로봇이 정상적으로 동작하더라도 그 판단 과정이 블랙박스처럼 보인다면, 사용자는 시스템의 안정성과 신뢰성을 충분히 확인하기 어렵다.

그림 1-1. 테슬라 모델3 FSD 자율주행 화면



최근 인공지능 분야에서는 Vision-Language Model(VLM)과 같은 멀티모달 AI 기술이 발전하면서, 이미지나 영상 속 장면을 자연어로 설명하고 상황을 추론하는 기술이 주목받고 있다. 이러한 기술은 자율주행 로봇이 인식한 주변 상황을 사람이 이해할 수 있는 언어 형태로 표현할 수 있다는 점에서 의미가 있다. 특히 로봇이 장애물을 회피하거나 정지하는 과정에서 “무엇을 보고”, “왜 그렇게 판단했는지”를 설명할 수 있다면, 자율주행 시스템의 판단 과정을 보다 투명하게 확인할 수 있다.

또한 실외 자율주행 로봇의 실제 운용을 고려하면, 로봇의 주행 상태와 주변 상황을 현장에서 직접 확인하는 것만으로는 한계가 있다. 주행 영상, 로봇 상태, 인식 결과, 판단 설명 로그 등을 원격에서 확인할 수 있는 관제 환경은 주행 과정의 모니터링과 사후 분석에 중요한 역할을 한다. 특히 설명 로그가 함께 제공될 경우, 단순히 로봇이 어디를 주행했는지를 확인하는 것을 넘어, 특정 상황에서 어떤 판단이 이루어졌는지를 함께 파악할 수 있다.

따라서 현재 자율주행 기술의 발전 방향은 단순한 주행 성능 향상에 그치지 않고, 복잡한 환경에 대한 이해, 판단 과정의 투명성, 사용자 신뢰성 확보로 확장되고 있다. 이러한 흐름 속에서 VLM 기반 설명 기능과 웹 관제 기술을 결합한 자율주행 모빌리티 로봇은 실외 주행 환경에서의 안전성, 신뢰성, 활용 가능성을 높이기 위한 중요한 접근 방식이 될 수 있다.

3. 프로젝트(주제) 목표 및 내용

3-1. 목적

본 프로젝트의 목적은 실외 환경에서 정해진 구역 내 자율주행이 가능한 모빌리티 로봇을 개발하고, Vision-Language Model(VLM)을 활용하여 로봇의 주행 판단 과정을 설명할 수 있는 시스템을 구현하는 것이다.

기존 프로젝트에서 수행했던 주차 환경 기반 자율주행 기술을 확장하여, 본 프로젝트에서는 실외 주행 환경에서 로봇이 주어진 경로를 따라 이동하면서 주변 장애물을 인식하고 회피할 수 있도록 하는 것을 목표로 한다. 이를 위해 로봇의 위치 추정, 경로 추종, 장애물 인식 및 회피 기능을 통합하여 실제 주행 환경에서 동작 가능한 자율주행 시스템을 구성하고자 한다.

또한 자율주행 로봇이 단순히 장애물을 회피하는 것에 그치지 않고, 어떤 객체를 인식했으며 어떤 이유로 회피 또는 정지 판단을 수행했는지를 자연어 형태로 설명할 수 있도록 VLM 기반 설명 생성 기능을 적용하고자 한다. 이를 통해 사용자는 로봇의 주행 결과뿐만 아니라 판단 근거까지 함께 확인할 수 있으며, 자율주행 시스템의 신뢰성과 이해 가능성을 높일 수 있다.

마지막으로 로봇의 주행 영상, 현재 상태, 장애물 인식 결과, VLM 기반 설명 로그를 웹 관제 서비스와 연동하여 사용자가 원격으로 주행 상황을 확인할 수 있는 환경을 구축하고자 한다. 이를 통해 실외 자율주행 로봇의 주행 과정과 판단 내용을 실시간 또는 기록 형태로 관리할 수 있는 설명가능 자율주행 모빌리티 로봇 및 웹 관제 서비스 개발을 최종 목표로 한다.

4. 시스템 구성 및 내용

4-1 실험환경 및 하드웨어

실험환경: 전남대학교 예술대학 2호관 ~ AI 융합대학

AI 융합대학

그림 2-1. 실험 환경



센서 및 플랫폼

1) Ouster LiDAR OS1 (128 channel)

- 라이다: 주변환경의 구조를 인식하여, 로봇의 위치를 파악하며, 장애물의 위치와 거리를 파악

2) MicroStrain 3DM-GX5 IMU

- IMU: 가속도/각속도 측정을 하여 LiDAR 사이 구간 차량 자세 보정

3) Scout mini

- 로봇 차체: 자율주행을 수행하는 개체

4) Realsense D435

- 카메라: 영상 정보를 통해 물체의 종류, 형태, 색상등 의미적인 정보를 파악

5) Intel NUC 11 Pro Kit

- pc1: 로봇의 위치 추정 및 경로 계산을 수행하는 pc

6) Nvidia AGX Xavier 16GB

- pc2: VLM 추론 및 웹 관제 서비스 호스팅을 수행하는 pc

그림 2-2. 센서 및 플랫폼		
Ouster LiDAR OS1 (128 channel)	MicroStrain 3DM-GX5 IMU	Intel NUC 11 Pro Kit
		
Scout mini	Realsense D435 (Camera)	Nvidia AGX Xavier 16GB
		

5. 프로젝트 결과물에 대한 기술

5-1. 고성능 자율주행 시스템 모델

● 고성능 자율주행 시스템 모듈 구조 및 수행 과정

본 프로젝트의 자율주행 시스템은 실외 환경에서 안정적인 위치 추정, 전역 경로 생성, 장애물 회피 주행, 주행 상태 기록 및 설명가능 정보 제공을 수행하기 위해 모듈화된 구조로 설계되었다. 전체 시스템은 LiDAR와 IMU 기반 위치 추정 모듈, 주행 가능 영역 지도, 전역 경로 계획 모듈, 지역 회피 주행 모듈, 경로 추종 제어 모듈, 그리고 VLM 추론 및 웹 관제 서비스 연동 모듈로 구성된다.

특히 기존 주차 자율주행 시스템에서 사용하던 경로 추종 중심의 구조를 확장하여, 실외 환경에서의 지도 생성 및 위치 추정, 장애물 감지, 회피 경로 생성, 주행 상태 모니터링이 가능하도록 구성하였다. 또한 각 기능을 ROS 토픽 기반으로 분리하여 향후 VLM 기반 설명 생성, 웹 관제 서비스, 추가 센서 또는 서비스 기능과 쉽게 연동될 수 있도록 설계하였다.

● 주요 기능

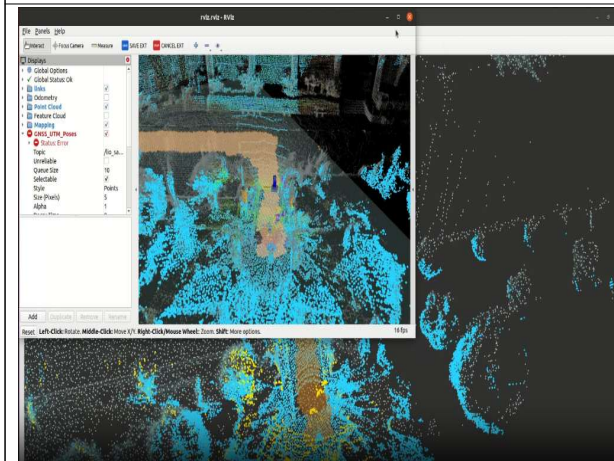
1) 실외 주행을 위한 지도 생성 및 위치 추정

실외 환경에서는 GPS만으로 정밀한 로봇 위치를 얻기 어렵고, 주변 구조물이나 장애물에 따라 위치 오차가 발생할 수 있다. 이를 보완하기 위해 본 시스템은 LiDAR와 IMU 데이터를 활용하는 LIO-SAM 기반 구조를 사용하여 주변 환경의 3차원 지도와 로봇의 위치 정보를 생성한다.

지도 생성 과정에서는 LiDAR로 수집한 포인트 클라우드와 IMU 정보를 결합하여 로봇 주변의 공

간 정보를 누적하고, 이를 기반으로 실외 주행에 필요한 전역 지도를 구성한다. 생성된 지도는 이후 로봇이 같은 환경에서 주행할 때 기준 지도로 활용되며, 로봇은 현재 센서 데이터와 저장된 지도 정보를 비교하여 자신의 위치를 추정한다. 이때 위치 정보는 주행 계획과 제어 모듈에서 공통적으로 사용하는 기준 좌표가 된다.

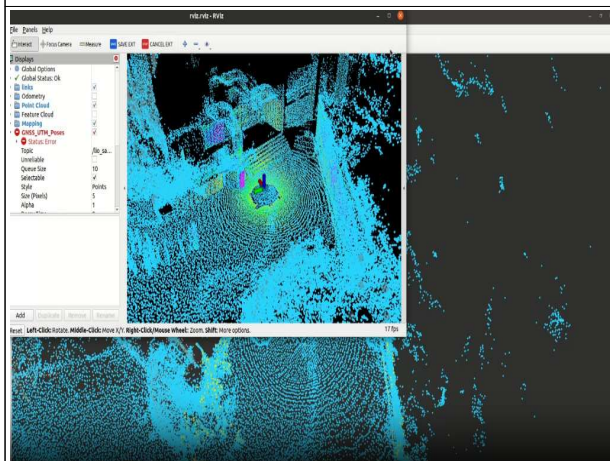
그림 2-3. LIO-SAM으로 제작한 전역 지도



2) 전역지도 확장 기능

또한 본 시스템은 저장된 지도만 사용하는 것이 아니라, 필요에 따라 기존 지도에 새로운 구간을 추가할 수 있는 확장 구조를 포함한다. 지도 확장 기능은 기존 맵을 기반으로 새로운 주행 구간을 추가하거나, 실외 환경 변화에 맞춰 주행 가능 영역과 경로 정보를 갱신하는 데 활용될 수 있다. 이를 통해 정해진 주행 구역이 변경되거나 새로운 서비스 공간으로 확장될 경우에도 시스템 전체를 새로 구성하지 않고 지도와 경로 정보를 갱신하여 사용할 수 있다.

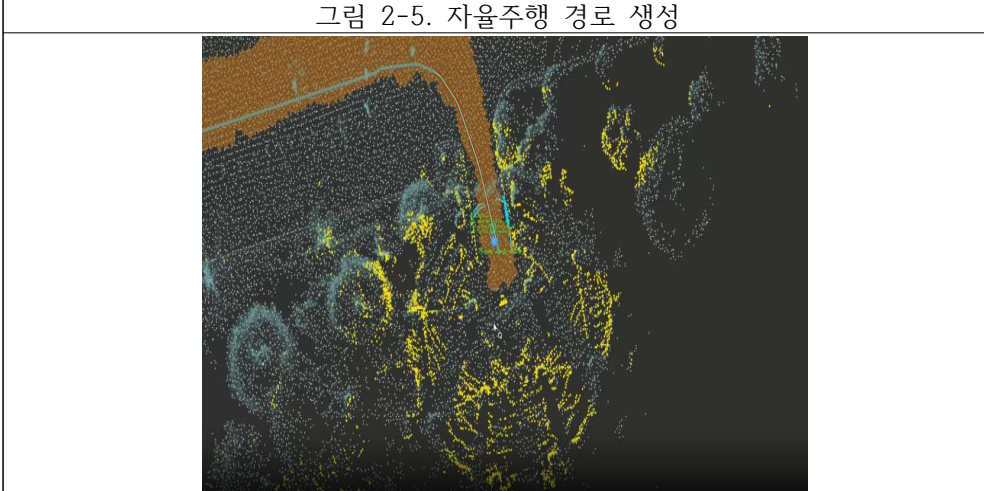
그림 2-4. 전역지도 확장 기능



3) 전역 경로 생성

전역 경로 생성은 OSM 형식의 경로 데이터와 기준 좌표 정보를 활용하여 수행된다. 시스템은 지도 파일과 기준 좌표 파일을 읽어 로봇이 이동해야 할 노드와 엣지를 구성하고, A* 알고리즘을 통해 시작 지점에서 목표 지점까지의 전역 경로를 생성한다. 실외 환경에서는 좌표계 처리가 중요하기 때문에, 기준 좌표 정보에 따라 실내/실외 모드를 구분하고 ENU 또는 UTM 기반 좌표 변환을 적용할 수 있도록 구성되어 있다.

그림 2-5. 자율주행 경로 생성



전역 경로는 단순히 목표 지점까지의 이동하기 위한 경로를 제공하는 역할에 그치지 않고, 지역 회피 주행 모듈과 경로 추종 제어 모듈이 참고하는 기준 경로로 사용된다. 로봇은 전역 경로를 따라 이동하다가 장애물이 감지되면 지역 회피 경로로 일시적으로 전환하고, 장애물 회피가 완료되면 다시 전역 경로로 복귀하는 방식으로 주행한다. 이러한 구조는 정해진 구역 내에서 안정적인 주행을 수행하면서도, 실시간 장애물 상황에 대응할 수 있도록 한다.

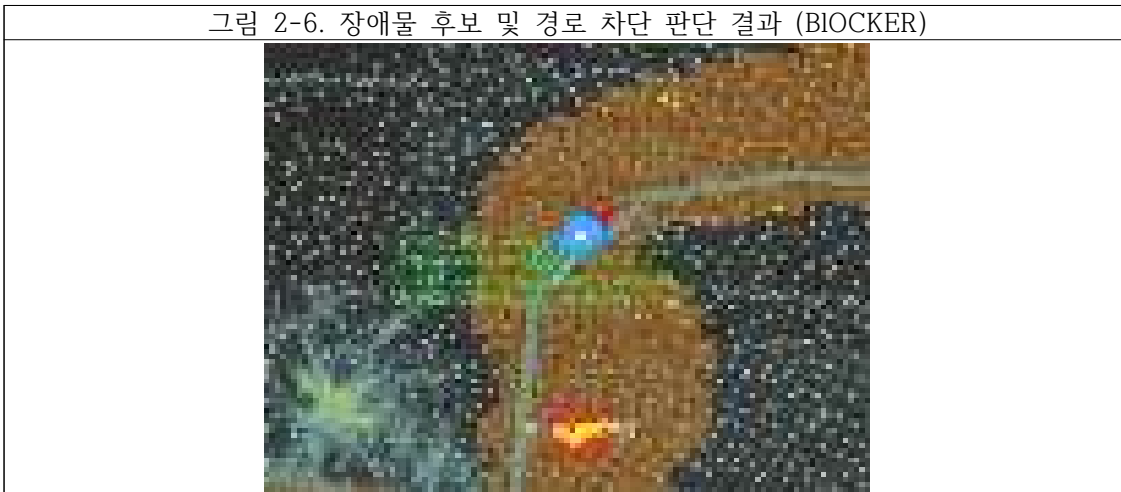
● 회피 자율주행 수행 과정 및 안전 장치

1) 장애물 인식 및 주행 경로 차단 판단

회피 자율주행은 로봇이 전역 경로를 따라 주행하는 과정에서 전방 경로상에 장애물이 존재하는지를 판단하는 것에서 시작된다. 본 시스템은 LiDAR로부터 수집한 포인트 클라우드 데이터를 활용하여 로봇 주변의 장애물 정보를 인식한다. 이때 지면에 해당하는 포인트와 로봇 자신의 차체 영역에 해당하는 포인트는 제거하고, 실제 주행에 영향을 줄 수 있는 비지면 포인트를 장애물 후보로 사용한다.

장애물 후보로 분류된 포인트는 로봇의 현재 위치와 지도 좌표계를 기준으로 변환되며, 주행 가능 영역 지도 위에 반영된다. 이후 시스템은 현재 로봇이 따라가고 있는 전역 경로의 일정 거리 앞을 확인하고, 해당 경로 주변에 장애물이 존재하는지 판단한다. 장애물이 전역 경로와 겹치거나 로봇의 진행 방향에 일정 거리 이상 가까이 존재할 경우, 시스템은 현재 경로가 차단되었다고 판단하고 회피 주행을 준비한다.

그림 2-6. 장애물 후보 및 경로 차단 판단 결과 (BLOCKER)



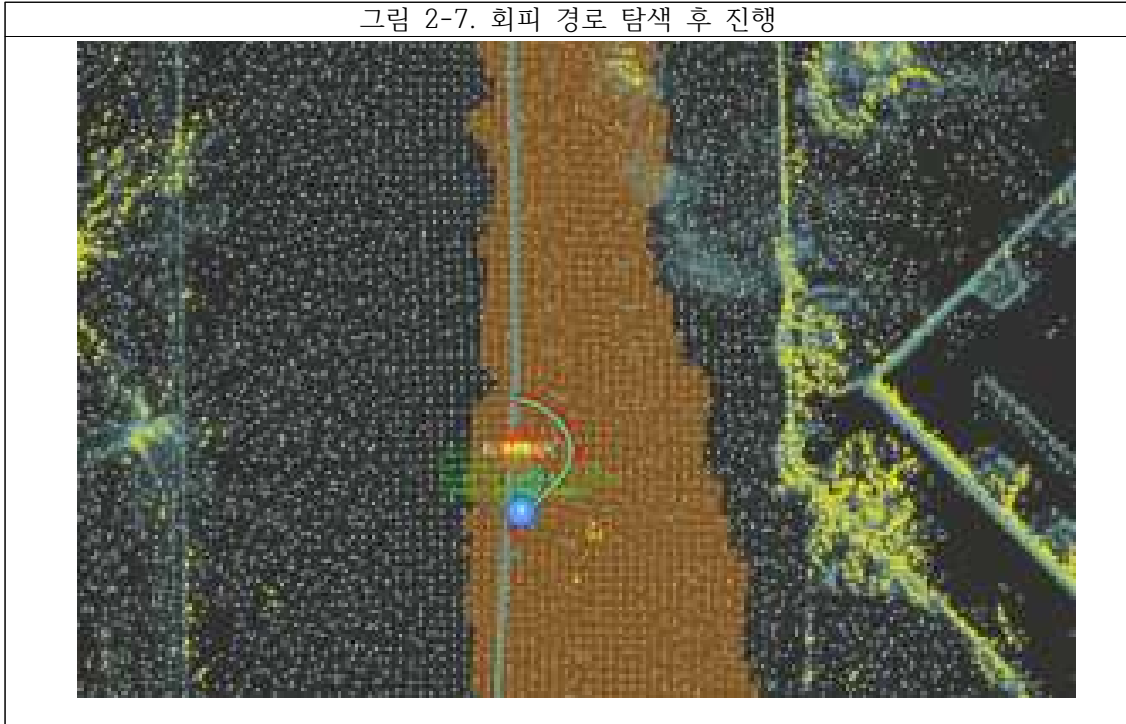
2) 지역 회피 경로 생성

전역 경로가 장애물에 의해 차단된 것으로 판단되면, 시스템은 기존 전역 경로를 그대로 따라가지 않고 지역 회피 경로를 생성한다. 본 시스템에서는 우선 sidestep 방식의 회피 경로를 생성한다. sidestep 방식은 로봇을 기준으로 장애물의 상대 위치를 계산한 뒤, 장애물이 위치한 방향과 반대 방향 또는 더 안전한 방향으로 일정한 횡방향 이동 경로를 생성하는 방식이다.

이 과정에서 로봇은 단순히 좌우로 이동하는 것이 아니라, 장애물에 접근하는 구간, 장애물 옆을 통과하는 구간, 다시 기존 전역 경로로 복귀하는 구간을 포함하는 회피 경로를 생성한다. 회피 경로는 로봇의 차체 폭, 장애물과의 최소 안전거리, 주행 가능 영역, 회피 후 전역 경로 재합류 가능성을 함께 고려하여 생성된다.

생성된 회피 후보 경로는 주행 가능 영역 지도와 장애물 정보를 기준으로 다시 검증된다. 만약 회피 경로가 주행 불가능 영역을 지나가거나 장애물과 충돌할 가능성이 있는 경우 해당 경로는 사용하지 않는다. sidestep 방식으로 유효한 경로를 찾지 못하는 경우에는 A* 기반의 대체 경로 탐색을 통해 전역 경로의 앞쪽 지점으로 다시 합류할 수 있는 회피 경로를 추가로 탐색한다.

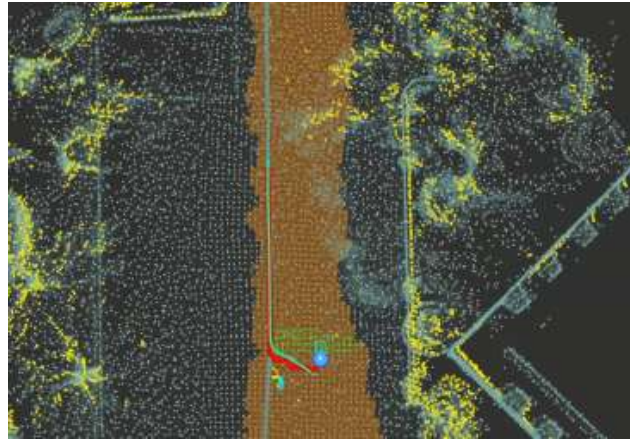
그림 2-7. 회피 경로 탐색 후 진행



3) 주행 모드 전환 및 전역 경로 복귀

본 시스템은 회피 주행 상황을 안정적으로 관리하기 위해 주행 모드를 구분하여 동작한다. 기본 주행 상태에서는 로봇이 전역 경로를 따라 이동한다. 주행 중 전방 경로에 장애물이 감지되고 유효한 회피 경로가 생성되면, 로봇은 지역 회피 경로를 따라 이동한다.

그림 2-8. 회피 후 전역 경로로 복귀



회피 주행이 완료되면 로봇은 다시 전역 경로로 복귀한다. 이때 단순히 장애물을 지나쳤다는 이유만으로 바로 복귀하는 것이 아니라, 회피 경로의 끝점에 충분히 가까워졌는지, 기존 전역 경로 주변이 다시 안전한지, 이전 회피 경로가 여전히 유효한지 등을 확인한다. 이를 통해 회피 도중 경로가 갑자기 바뀌거나, 장애물이 아직 남아 있는 상태에서 무리하게 전역 경로로 복귀하는 상황을 방지한다.

또한 로봇이 회피 과정에서 전역 경로에서 일정 거리 이상 벗어난 경우에는 전역 경로로 다시 합류하기 위한 복귀 경로를 계산한다. 이를 통해 장애물을 피한 이후에도 로봇이 주행 구역을 벗어나지 않고, 기존 목표 지점까지 안정적으로 이동할 수 있도록 한다.

4) 안전 장치

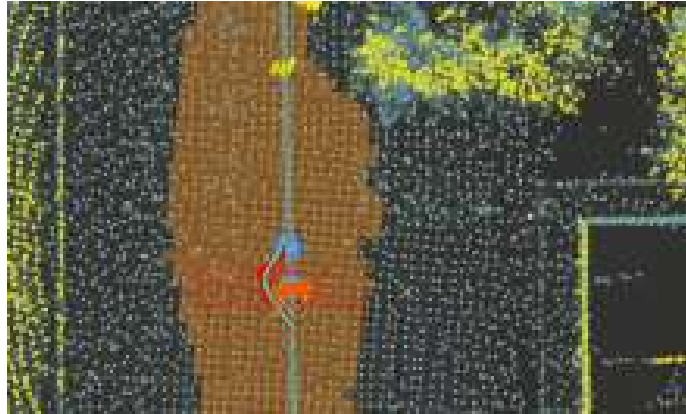
본 시스템은 실외 환경에서 발생할 수 있는 예기치 못한 상황에 대응하기 위해 여러 단계의 안전 장치를 포함한다. 첫 번째로, 주행 가능 영역 지도에서 확인되지 않은 영역은 기본적으로 주행 위험 영역으로 처리한다. 이를 통해 로봇이 센서나 지도에서 확인되지 않은 공간으로 무리하게 진입하는 것을 방지한다.

두 번째로, 장애물 주변에는 로봇의 실제 크기와 여유 간격을 고려한 안전 마진을 적용한다. 즉, 장애물을 점 하나로만 판단하지 않고 로봇의 폭과 회전 가능성을 고려하여 장애물 주변 영역을 확장한 뒤 경로를 검증한다. 이를 통해 로봇이 실제로는 지나가기 어려운 좁은 공간을 통과하려는 상황을 줄일 수 있다.

세 번째로, 유효한 회피 경로가 존재하지 않는 경우 로봇은 무리하게 전진하지 않고, 정지 모드인 HOLD 상태로 전환된다. HOLD 상태에서는 경로를 따라 계속 주행하는 대신 정지 상태를 유지하며, 이후 장애물이 사라지거나 새로운 회피 경로가 생성될 때 다시 주행을 수행한다. 이는 장애물이 있는 상태에서 잘못된 경로를 따라 주행하는 것을 방지하기 위한 핵심 안전 장치이다.

네 번째로, 로봇 근처의 장애물에 대한 긴급 정지 기능을 적용할 수 있다. 로봇 바로 앞 가까운 영역에서 장애물이 감지되면, 경로 계획 결과와 관계없이 정지 또는 저속 주행을 수행하도록 하여 충돌 가능성을 줄인다. 또한 회전 중 사각지대에 존재할 수 있는 장애물이나 센서에서 일시적으로 사라진 정적 장애물에 대해서도 일정 시간 동안 기억하고 보수적으로 대응할 수 있도록 구성하였다.

그림 2-9. 장애물이 너무 가까워서 긴급 정지를 진행한 모습



마지막으로, 이전에 생성된 회피 경로를 계속 사용할 때도 해당 경로가 현재 지도와 장애물 정보 기준에서 여전히 안전한지 재검증한다. 만약 기존 회피 경로가 새롭게 감지된 장애물이나 주행 불가능 영역과 충돌하는 경우, 해당 경로는 폐기되고 정지 또는 새로운 경로 탐색이 수행된다. 이를 통해 회피 주행 중 환경 변화가 발생하더라도 안전한 주행을 유지할 수 있다.

● VLM 추론 및 웹 관제 서비스 연동 구조

1) 자율주행 상태 정보 제공

본 시스템은 VLM 추론 및 웹 관제 서비스와 연동하기 위해 로봇의 주행 상태와 판단 결과를 외부에서 활용할 수 있는 형태로 제공한다. 자율주행 시스템 내부에서 생성되는 정보는 단순한 센서 데이터뿐만 아니라, 로봇의 현재 위치, 전역 경로, 지역 회피 경로, 주행 모드, 장애물 인식 결과, 정지 여부, 회피 방향 등으로 구성된다.

2) 설명 이벤트 및 판단 로그 생성

회피 주행 또는 정지와 같은 주요 주행 상태 변화가 발생하면, 시스템은 해당 상황을 설명할 수 있는 이벤트 정보를 생성한다. 설명 이벤트에는 현재 주행 모드, 이벤트 종류, 판단 이유, 수행 동작, 회피 방향, 가장 가까운 장애물과의 거리 등이 포함된다.

이러한 설명 이벤트는 VLM이 자연어 설명을 생성할 때 중요한 보조 정보로 활용된다.

VLM은 카메라 영상을 통해 주변 장면을 이해하고, 설명 이벤트를 통해 로봇의 실제 판단 결과를 반영함으로써 보다 정확하고 신뢰성 있는 설명 로그를 생성할 수 있다.

이 정보는 VLM 기반 설명 생성뿐만 아니라 웹 관제 서비스에서 로봇의 현재 주행 상태를 시각화하는 데 활용된다.

3) 연동 구조의 확장성

본 시스템은 ROS 기반의 모듈화된 구조로 구성되어 있어, 향후 다양한 서비스와 쉽게 연동될 수 있다. 자율주행 모듈은 주행과 회피 판단을 수행하고, VLM 추론 서버는 영상과 판단 정보를 바탕으로 자연어 설명을 생성하며, 웹 관제 서버는 생성된 설명과 주행 상태를 사용자에게 시각적으로 제공한다.

이러한 구조는 특정 서비스에 한정되지 않고, 배달 로봇, 순찰 로봇, 안내 로봇 등 다양한 실외 자율주행 서비스로 확장될 수 있다. 예를 들어 배달 로봇으로 활용할 경우에는 목적지 정보, 배송 상태, 수령 여부 등을 추가로 연동할 수 있으며, 순찰 로봇으로 활용할 경우에는 순찰 경로, 이상 상황 감지 결과, 경고 알림 등을 추가할 수 있다.

결과적으로 본 시스템의 VLM 및 웹 관제 연동 구조는 자율주행 로봇의 주행 결과를 단순히 보여주는 수준을 넘어, 로봇이 어떤 상황을 인식하고 어떤 판단을 수행했는지를 사용자가 이해할 수 있도록 지원한다. 이를 통해 자율주행 시스템의 투명성, 신뢰성, 운영 편의성을 향상시킬 수 있다.

5-2 VLM을 활용한 설명 가능한 모델

● 활용 데이터 및 베이스 모델 정보

1) 활용 ROS 토픽

고성능 자율주행 시스템을 통해 나오는 토픽중 다음과 같은 7개 토픽을 구독한다. 이 토픽들을 활용하여 VLM입력을 제작한다. VLM추론 문장은 /xai/vlm_log(std_msgs/String)토픽으로 발행된다.

토픽명	주요 내용	활용 목적
/camera/color/image_raw	RGB 이미지 (640×480, rgb8)	VLM 입력 이미지
/camera/color/camera_info	fx, fy, cx, cy (카메라 내부 파라미터)	핀홀 투영으로 LiDAR → 픽셀 좌표 변환
/planning/near_future_id_stop_hits	로봇 기준 근거리 장애물 포인트 (x=전방m, y=좌측m)	장애물 centroid 좌표 추출
/xai/event_log	회피 활성 여부, 장애물 거리, 회피 방향, 경로 변경 정보	추론 트리거 및 문맥 정보
/planning/emergency_stop	긴급 정지 신호 (true/false)	긴급 정지 상황 감지
/planning/global_obstacle_overlay	전역 좌표계 장애물 점유 그리드	장애물 위치 보조 참조
/nav/odom	로봇 위치 (x, y) 및 방향 (yaw)	이동 이력 추적

2) VLM 모델 정보

- 베이스 모델: InternVL2.5-1B-MPO

OpenGVLab/InternVL2_5-1B-MPO는 OpenGVLab이 공개한 1B 파라미터 규모의 경량 Vision-Language Model이다.

항목	내용
전체 파라미터	약 1B (10억)
아키텍처	Vision Encoder (InternViT-300M) + LLM (Qwen2.5-0.5B-Instruct)
입력형태	이미지 + 텍스트 질문
출력형태	자연어 텍스트
학습방식	MPO (Mixed Preference Optimization) 사전학습
추론환경	NVIDIA Xavier AGX (CUDA)

● VLM 전체 학습 과정 구조

1) 데이터셋 구축 방법

본 프로젝트의 학습 데이터셋은 실제 자율주행 로봇 운용 중 녹화된 ROS bag 파일 10개를 프레임단위로 잘라서 만들어진 이미지 셋으로 구축되었다.

Step 1. 회피 이벤트 자동 추출

ROS bag에서 /xai/event_log 토픽을 파싱하여 자율주행 플래너가 장애물을 감지하고 회피 동작을 수행하는 순간(AVOIDANCE_ACTIVE 이벤트를) 자동으로 추출한다. 총 10개 백파일에서 장애물 167개 샘플이 추출되었으며, 각 샘플은 RGB 이미지, depth map, LiDAR 장애물 좌표, 회피 방향 정보를 포함한다.

Step 2. 핀홀 투영 BBox 오버레이

LiDAR에서 추출된 장애물 centroid 좌표를 카메라 내부 파라미터(f_x , f_y , c_x , c_y)를 이용한 핀홀 투영으로 2D 픽셀 좌표로 변환한다. depth map에서 목표 거리 $\pm 700\text{mm}$ 범위 픽셀을 클러스터링하여 바운딩 박스를 생성하고, RGB 이미지에 초록색으로 오버레이한다.



Step 3. 라벨 부여

각 샘플에 대해 사람/표지판/지키퍼 3개 클래스 중 하나를 할당하였다. 라벨은 bag 재생 영상을 직접 확인하는 방식으로 수작업 검수하였다. 사람 86개(51%)/표지판 47개(28%)/지키퍼 34개(20%)

Step 4. 학습 데이터 포맷 변환 및 증강

각 샘플을 VLM 파인튜닝용 대화 형식으로 변환하며, 질문은 LiDAR 위치·회피 방향을 포함해 샘플마다 동적으로 생성한다. 정답(GT)은 "정면Xcm 왼쪽Ycm에 [라벨]이 있습니다. / 회피경로를 장애물 [방향]으로 생성하여 주행합니다."형식으로 구성된다. 전체 167개 샘플 중 82%(137개)를 학습, 18%(30개)를 검증에 사용하였다. 이후 밝기·노이즈·블러·회전·좌우반전 등 Clean포함 8종 증강을 적용하여 최대 1,336(8종*167개)까지 확장하였다.



2) LoRA를 활용한 InternVL2.5-1B-MPO 학습

- 파인튜닝 방법: LoRA (Low-Rank Adaptation)

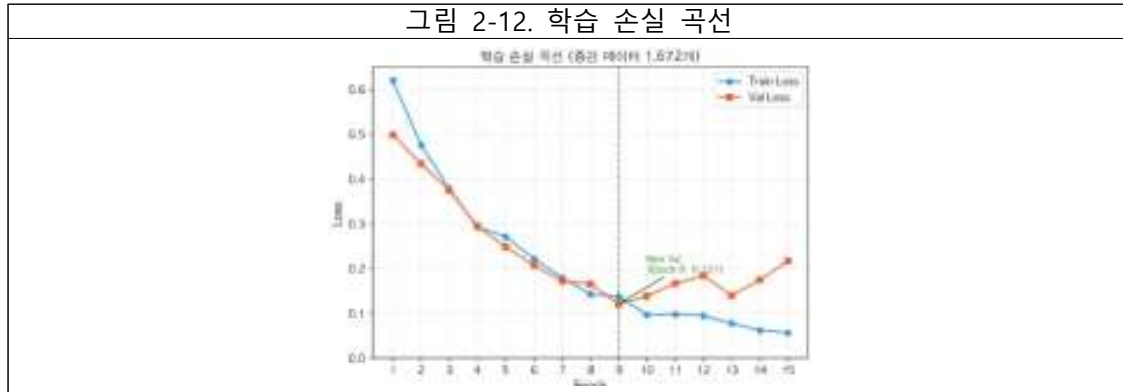
LoRA는 사전학습된 LLM의 가중치를 직접 수정하지 않고, 기존 가중치 행렬 옆에 저랭크 (low-rank) 행렬 쌍(A, B)을 추가하는 경량 파인튜닝 기법이다. 전체 파라미터의 1% 미만만 학습하므로 소규모 데이터셋에서도 과적합 없이 도메인 적응이 가능하다.

LoRA 학습 방식
$y = (W + \Delta W)x = Wx + ABx$

원래 가중치 W는 고정되며, LoRA 추가분 $\Delta W = A \times B$ (A: $r \times d$, B: $d \times r$, $r \ll d$)만 학습된다. 본 프로젝트의 LoRA 적용 설정은 다음과 같다.

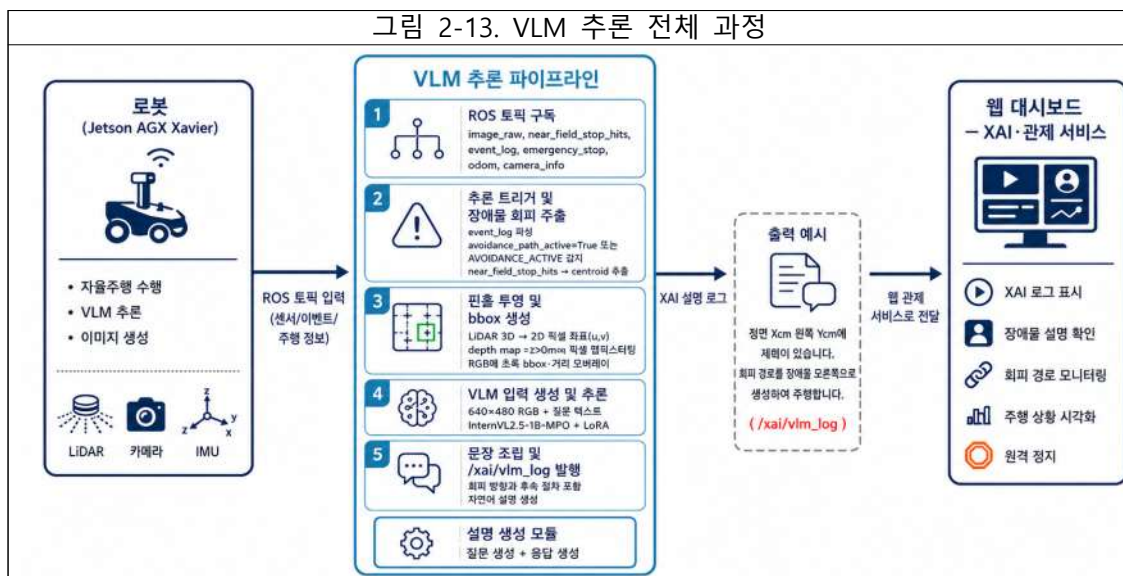
항목	내용
베이스 모델	OpenGVLab/InternVL2_5-1B-MPO
LoRA Rank (r)	8
LoRA Alpha (α)	16
Dropout	0.05
적용 레이어	q, k, v, o, up, gate, down proj
학습에폭	15
학습률	2×10^{-4}
학습 환경	Mac MPS (Apple Silicon)

다음은 학습손실곡선에 대한 그래프이다. Train Loss가 Epoch 1(0.62)부터 15(0.06)까지 꾸준히 감소하고 있어서 모델이 학습 데이터를 잘 학습하고 있음은 확인되었고, Best Val이 Epoch 9(0.121)에서 찍히고 이후 Val Loss가 올라가는 것을 확인 후 Epoch9체크 포인트를 최종 모델로 선택하였다.



● VLM 추론 과정/결과 및 성능 평가

1) VLM 추론 전체 과정



본 시스템의 VLM 추론 파이프라인은 ROS 토픽 수신부터 자연어 결과 발행까지 총 5단계로 구성된다.

① ROS 토픽 구독

실시간 로봇 또는 ROS bag 재생 환경에서 7개 토픽을 동시에 구독한다. image_raw, near_field_stop_hits, event_log, emergency_stop, odom, camera_info 토픽이 각각 병렬로 수신된다.

② 추론 트리거 및 장애물 좌표 추출

/xai/event_log의 JSON 메시지를 파싱하여 avoidance_path_active=True 또는 AVOIDANCE_ACTIVE 이벤트가 감지될 때에만 VLM 추론을 실행한다. 이를 통해 불필요한 연산을 방지하고 회피 상황에 집중적으로 추론 자원을 투입한다. 트리거 발생 시 /planning/near_field_stop_hits의 포인트클라우드 데이터에서 장애물 centroid 좌표(x_fwd=전방 m, y_left=좌측 m)를 추출한다.

③ 핀홀 카메라 투영 및 바운딩 박스 생성

추출된 LiDAR 3D 좌표를 카메라 내부 파라미터(f_x , f_y , c_x , c_y)를 이용한 핀홀 투영 수식으로 2D 픽셀 좌표(u , v)로 변환한다. 수식은 다음과 같다.

핀홀 카메라 투영 식	
$u = f_x \cdot \frac{-W_z D}{x_{\text{world}}} + c_x$	$v = f_y \cdot \frac{0}{x_{\text{world}}} + c_y$

바운딩 박스는 depth map에서 목표 거리 $\pm 700\text{mm}$ 범위의 픽셀을 클러스터링하여 생성한다. 최종적으로 RGB 이미지에 초록색 bbox와 거리 정보를 오버레이한다.

④ VLM 입력 생성 및 추론

bbox가 오버레이된 640×480 RGB 이미지와, 실제 측정값(거리, 방향)이 동적으로 삽입된 질문 텍스트를 함께 VLM에 입력한다. 질문은 샘플마다 실제 측정값을 포함하여 생성되므로 모델이 이미지 내 특정 위치를 집중 분석하도록 유도된다.

질문 예시: "정면 Xcm 왼쪽 Ycm 위치의 장애물을 확인하세요. 오른쪽 회피 경로로 주행 중입니다. 이 장애물이 무엇인지 설명하세요."

InternVL2.5-1B-MPO + LoRA 모델은 Vision Encoder를 통해 이미지를 패치 토큰으로 변환한 뒤, Qwen2.5 LLM backbone이 이미지 토큰과 텍스트 토큰을 함께 처리하여 자연어 응답을 생성한다.

⑤ 출력 문장 조립 및 토픽 발행

추론 결과와 회피 방향 정보를 조합하여 최종 문장을 조립한 후 /xai/vlm_log 토픽으로 발행한다.

출력 예시: "정면 Xcm 왼쪽 Ycm에 사람이 있습니다. 회피 경로를 장애물 오른쪽으로 생성하여 수행합니다."

웹 관제 서비스 시스템은 해당 토픽을 구독하여 설명 가능한 자율주행(XAI) 로그로 활용할 수 있다.

2) VLM 추론 결과 및 성능평가

-평가방법: KoffVQA / 평가용 모델: Claude sonnet4.6

성능 평가는 LLM-as-Judge 방식인 KoffVQA를 채택하였다. 평가는 샘플별로 질문(question), 모델 출력(raw_output), 정답(gt_answer)을 평가 LLM에 함께 전달하는 방식으로 수행된다. 평가모델은 Claude sonnet4.6을 이용하여 진행하였다. 모델에 입력되는 질문은 실제 LiDAR 센서로 측정한 거리·방향 수치와 회피경로 및 카메라 이미지를 동적으로 삽입하는 형태로 구성된다.

평가 LLM은 모델 출력에서 ① 장애물 종류의 정확성, ② 회피경로 방향 포함 여부, ③ 불필요한 잉여 정보 포함 여부를 종합적으로 고려하여 채점하며, 거리·방향 수치는 센서 측정값으로 항상 정확하므로 채점 대상에서 제외한다. 채점 기준은 10점(장애물 종류 정확, 회피경로 방향 포함, 간결한 출력), 5점(장애물 종류는 맞지만 외형 묘사·부연 설명 등 잉여정보가 포함되거나 회피경로 방향 표현이 불명확한 경우), 0점(장애물 종류 오답 또는 미언급)으로 구분되며, 최종 KoffVQA 점수는 샘플별 점수의 평균에 10을 곱하여 0~100 범위로 산출한다. 아래 예시 샘플(sample_00008)에서 Base 모델은 장애물을 "주차선지"로 표현하는 장애물 종류 오류로 0점을 기록한 반면, LoRA 학습모델은 "정면 120cm 오른쪽 21cm 위치에 표지판이 있습니다. 왼쪽 회피경로로 주행합니다."로 간결하게 출력하여 10점을 획득하였다. 다른 샘플(sample_00063)에서도 Base 모델은 "83cm 차량의 앞을 나타냅니다."와 같은 잉여 정보로 인해 5점에 그쳤으나, LoRA 학습 모델은 간결한 형식으로 10점을 기록하였다.

그림 2-14. 평가 결과

		
		<pre> "results": { "sample_id": "sample_00008", "gt_answer": "정면 120cm 오른쪽 21cm에 표지판이 있습니다. / 왼쪽 회피경로로 주행합니다.", "question": "이미지에서 정면 120cm 오른쪽 21cm 위치의 장애물을 확인하세요. 왼쪽 회피 경로로 주행합니다. 이 장애물이 무엇인지 설명하세요.", "image": "selected_clean/sample_00008_clean.jpg", "InternVL2.5-18_base": { "raw_output": "이미지에서 회피 경로로 주행 중인 동안 확인한 장애물은, 정면 120cm 오른쪽 21cm 위치에 위치한 이 장애물은 보통 차량이 차선을 정지시키거나 차선을 변경할 때 나타나는 붉은 구멍을 가리킵니다.", "judge_score": 0, }, "InternVL2.5-18_lora": { "raw_output": "정면 120cm 오른쪽 21cm 위치에 표지판이 있습니다. 왼쪽 회피경로로 주행합니다.", "judge_score": 10, } } </pre>

그림 2-15. 평가 결과

		
		<pre> "sample_id": "sample_00063", "gt_answer": "정면 83cm에 사람이 있습니다. / 오른쪽 회피경로로 주행합니다.", "question": "이미지에서 정면 83cm 위치의 장애물을 확인하세요. 오른쪽 회피경로로 주행합니다. 이 장애물이 무엇인지 설명하세요.", "image": "selected_clean/sample_00063_clean.jpg", "InternVL2.5-18_base": { "raw_output": "이미지에서 장애물은 오른쪽으로 회피 경로로 주행 중인 사람이 있습니다. 장애물은 83cm 차선의 앞을 나타냅니다.", "judge_score": 3, }, "InternVL2.5-18_lora": { "raw_output": "이미지에서 정면 83cm 위치에 사람이 있습니다. 오른쪽 회피경로로 주행합니다.", "judge_score": 10, } } </pre>

-성능평가결과

본 연구에서는 전체 167개 샘플 중 137개(82%)를 학습에 사용하고, 나머지 30개(18%)를 학습 과정에서 완전히 배제한 평가 전용 데이터셋으로 확보하였다. 성능 평가는 해당 30개 샘플에 대해 KoffVQA 방식으로 수행하였다. LoRA 파인튜닝 적용 후 전체 KoffVQA 점수가 33.3점에서 90.0점으로 56.7점 향상되었다. 사람 클래스는 95.0점으로 가장 높은 인식률을 보였으며, 표지판과 지쿠터 클래스도 각각 90.0점, 87.5점으로 파인튜닝 효과가 세 클래스 전반에 걸쳐 고르게 확인되었다.

모델 비교 결과				
InternVL2.5-1B (Base)	33.3	40.0	30.0	25.0
InternVL2.5-1B (LoRA)	90.0	95.0	90.0	87.5
향상폭	+56.7	+55.0	+60.0	+62.5

점수 분포 측면에서 Base 모델은 0점 15개, 5점 10개, 10점 5개였던 반면, LoRA 모델은 0점 0개, 5점 6개, 10점 24개로 장애물 종류 분류 정확도가 전반적으로 크게 향상되었음을 확인하였다. 또한 과적합 검증을 위한 Train/Val 데이터셋에 대한 분리평가에서도 Train과 Val 점수 차이가 0.5점으로 매우 작아, LoRA 파인튜닝이 평가 전용 데이터셋에도 안정적으로 일반화되었음을 확인하였다.

과적합 검증(Train/Val 분리 평가)			
모델	Train	Val	차이폭
InternVL2.5-1B (Base)	34.0	33.3	-0.7
InternVL2.5-1B (LoRA)	90.5	90.0	-0.5

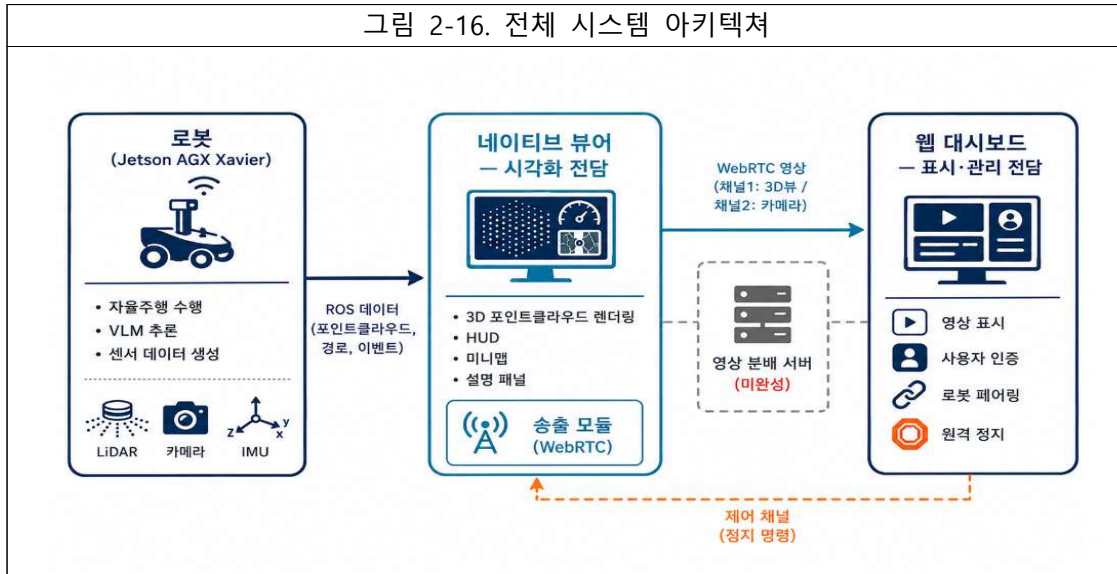
5-3 웹 관제 서비스

● 서비스 개요 및 시스템 구조

웹 관제 서비스는 실외 자율주행 로봇의 주행 상태, 장애물 인식 결과, VLM이 생성한 설명 문장을 사용자가 원격에서 실시간으로 확인할 수 있도록 하는 서비스이다. 단순히 로봇이 어디에 있는지를 보여주는 것에 그치지 않고, 로봇이 어떤 상황을 인식하였으며 왜 해당 판단을 수행하였는지까지 함께 확인할 수 있도록 하는 것을 목표로 설계되었다. 본 서비스는 자율주행과 VLM 추론을 수행하는 로봇, 로봇 내부에서 주행 상황을 화면으로 그려내는 네이티브 뷰어, 그리고 사용자가 웹 브라우저로 접속하여 그 화면을 확인하는 웹 대시보드로 구성된다. 설계의 핵심은 역할 분리이다. 화면을 그리는 일은 전부 네이티브 뷰어가 담당하고, 웹 대시보드는 완성된 화면을 영상으로 전달받아 표시하는 역할과 사용자 인증, 로봇 등록·관리 기능만을 담당한다. 이렇게 나눈 이유는 LiDAR 센서가 만들어내는 3차원 점 데이터가 회전 1회당 십만 개를 넘어, 웹 브라우저에서 직접 그리기에는 성능 부담이 크기 때문이다. 따라서 성능이 보장되는 로봇 쪽 프로그램에서 화면을 완성한 뒤, 그 결과만 영상으로 전송하는 방식을 채택하였다. 화면 전송에는 WebRTC라는 실시간 영상 통신 기술을 사용한다. WebRTC는 화상회의 서비스에서 널리 쓰이는 기술로, 별도의 프로그램 설치 없이 웹 브라우저만으로 지연이 적은 실시간 영상을 받아볼 수 있다는 장점이 있다. 당초 설계에서는 여기에 더해 Jetson을 영상 분배 서버로 두어, 네이티브 뷰어의 영상을 한 번 받아 다수의 웹 접속자에게 나누어 보내는 구조를 계획하였다. 그러나 해당 부분은 프로젝트 기간 내에 완성하지 못하였으며, 현재는 네이티브 뷰어가 웹 대시보드에 영상을 직접 송출하는 1:1 구조로 동작한다.

구성 요소	역할	사용 언어·기술	비고
로봇	자율주행 수행, VLM 추론, 센서 데이터 생성	C++, Python (ROS)	
네이티브 뷰어	시각화 전달 — 3D 화면·계기판·미니맵 렌더링, 영상 송출	C++ (Qt, OpenGL), Python (WebRTC 송출)	
중계서버	영상 중계 — 네이티브 뷰어의 영상을 받아 다수의 웹 접속자에게 분배	Python	미완성 — 현재는 네이티브 뷰어가 웹에 직접 송출
웹 대시보드	중계 전달 — 영상 수신 및 표시, 사용자 인증, 로봇 등록·관리	JavaScript (React), Python (FastAPI)	

그림 2-16. 전체 시스템 아키텍처



● 네이티브 뷰어 구조

네이티브 뷰어는 로봇에 탑재되어 동작하는 시각화 프로그램으로, 화면을 그리는 렌더링 모듈과 그 화면을 영상으로 내보내는 송출 모듈 두 부분으로 구성된다.

(1) 렌더링 모듈 — 시각화 전담

렌더링 모듈은 자율주행 시스템이 발행하는 데이터를 실시간으로 받아 하나의 통합 화면으로 그려낸다. 수신하는 주요 데이터는 다음과 같다.

데이터	내용
LiDAR 포인트클라우드	주변 환경을 표현하는 3차원 점 데이터
로봇 위치·자세	위치 추정 모듈이 계산한 현재 위치와 방향
주행 경로	현재 추종 중인 경로와 회피 시 탐색된 후보 경로
주행 상태	경로 차단 여부, 대체 경로 전환 여부, 비상정지 여부
회피 이벤트 설명	VLM이 생성한 주행 판단에 대한 자연어 설명
카메라 영상	로봇 전방 카메라의 실시간 영상

이 데이터를 바탕으로 렌더링 모듈은 네 가지 화면 요소를 그린다. 주변 환경을 입체적으로 보여주는 3D 포인트클라우드 뷰, 로봇의 상태를 자동차 계기판처럼 표시하는 HUD(헤드업 디스플레이), 경로와 주행 상태를 위에서 내려다본 형태로 요약하는 미니맵, 그리고 회피·정지 판단의 이유를 자연어 문장으로 보여주는 설명 패널이다. 각 화면 요소의 상세 기능은 4)에서 설명한다.

그림 2-17. 네이티브 뷰어 전체 실행 화면



(2) 송출 모듈 — 영상 전송 담당

송출 모듈은 렌더링 모듈이 완성한 화면을 WebRTC 영상으로 변환하여 웹 대시보드에 전송한다. 이때 두 개의 독립된 영상 채널을 사용하는데, 첫 번째 채널로는 3D 시각화 화면을, 두 번째 채널로는 로봇 전방 카메라 영상을 각각 송출한다. 영상 압축에는 하드웨어 가속 인코딩을 우선 사용하여 전송 지연을 최소화하였다.

네트워크 환경에 따라 연결 방식도 자동으로 전환된다. 로봇과 사용자가 같은 네트워크에 있으면 두 기기를 직접 연결하고, 서로 다른 네트워크에 있으면 통신 중계(TURN) 서버를 경유하도록 구성하여, 어떤 네트워크 환경에서도 영상이 전달될 수 있도록 하였다. (이 통신 중계는 WebRTC 연결 수립을 위한 표준 기술로, 1)에서 언급한 미완성 영상 분배 서버와는 별개이다.)

영상과 별도로, 웹 대시보드에서 로봇으로 명령을 보내는 제어 채널도 함께 운영된다. 제어 채널의 모든 명령에는 고유 번호가 부여되고 수신 확인 응답을 주고받도록 설계하여, 네트워크 불안정으로 같은 명령이 중복 전달되더라도 한 번만 실행되도록 안전하게 처리하였다.

보안 측면에서는 로봇과 웹 대시보드 간 연결 수립 시 사전에 발급된 인증 키를 검증하도록 구성하였으며, 인증 키는 소스 코드와 분리하여 관리한다. 이를 통해 허가되지 않은 사용자가 로봇의 영상이나 제어 채널에 접근하는 것을 차단하였다.

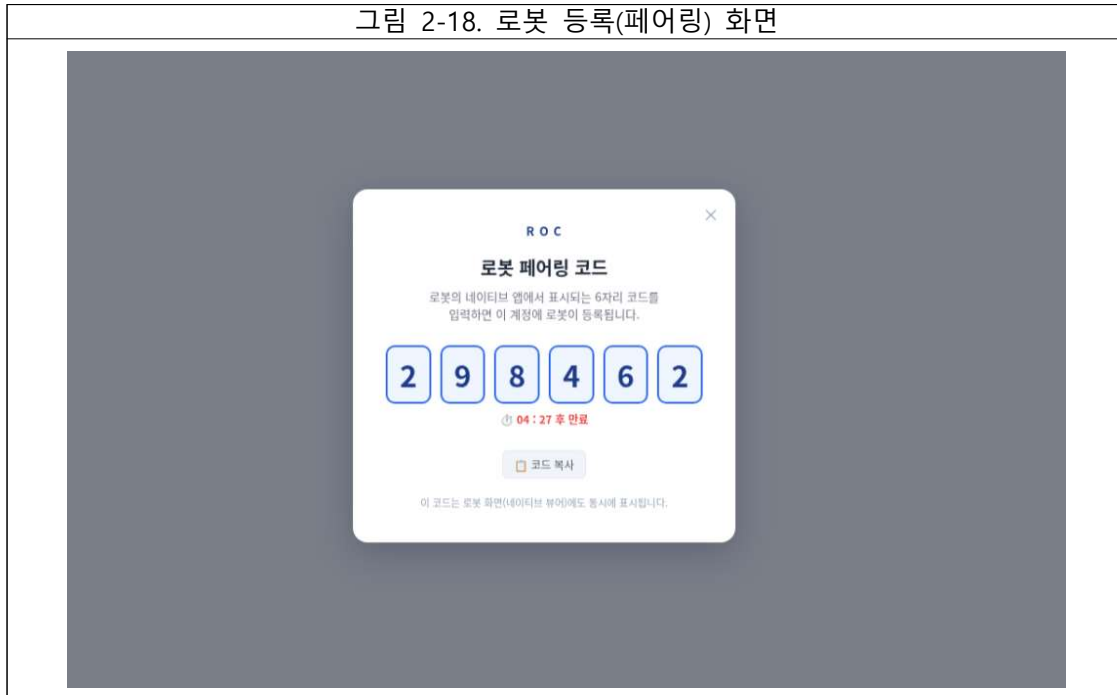
● 웹 대시보드 구조

웹 대시보드는 사용자가 웹 브라우저로 접속하는 화면으로, 네이티브 뷰어가 보내온 영상을 표시하고 운영에 필요한 관리 기능을 제공한다. 시각화 연산은 수행하지 않으며, 완성된 영상을 받아 보여주는 역할에 집중한다.

웹 대시보드가 담당하는 기능은 다음과 같다.

- 사용자 인증: 운영자 로그인 및 권한 관리
- 로봇 등록(페어링): 로봇이 화면에 표시하는 6자리 코드를 웹에 입력하면 해당 로봇과 웹 대시보드가 연결되는 방식으로, 허가된 사용자만 로봇 영상에 접근할 수 있도록 함
- 영상 표시: 네이티브 뷰어로부터 수신한 3D 시각화 영상과 카메라 영상을 브라우저에 표시
- 원격 정지 명령: 제어 채널을 통해 로봇에 정지 명령을 전송

그림 2-18. 로봇 등록(페어링) 화면



● 주요 기능

(1) 실시간 3D 포인트클라우드 뷰 및 HUD

LiDAR가 수집한 3차원 점 데이터를 실시간으로 렌더링하여 로봇 주변 환경을 입체적으로 보여준다. 화면 위에는 자동차 계기판 형태의 HUD가 함께 표시되며, 현재 속도, 배터리 잔량, 주행 모드, 목표 지점까지의 잔여 거리, 인식 중인 주변 객체 수를 실시간으로 확인할 수 있다.

그림 2-19. 3D포인트클라우드 뷰 및 HUD



(2) 미니맵 및 경로 시각화

미니맵은 로봇의 주행 상황을 위에서 내려다본 형태로 요약하여 보여준다. 현재 따라가고 있는 경로와 회피 시 탐색된 후보 경로를 함께 표시하며, 주행 상태에 따라 색상을 다르게 표현하여 운영자가 한눈에 상황을 파악할 수 있도록 하였다.

상태	색상	의미
정상 주행	초록	현재 로봇이 따라가는 경로
후보 경로	회색	회피를 위해 탐색된 대안 경로
경로 차단	빨강 테두리	전방 경로가 장애물로 막힌 상태
대체 경로 전환	노랑 테두리	기존 경로 대신 대체 경로로 주행 중인 상태
비상정지	빨강	긴급 정지가 발생한 상태

그림 2-20. 정상주행

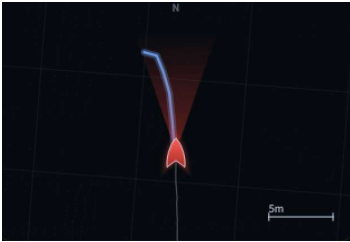
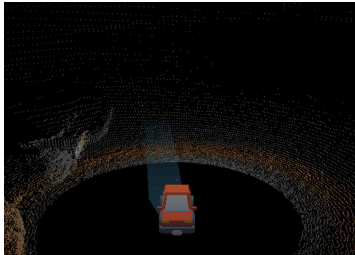
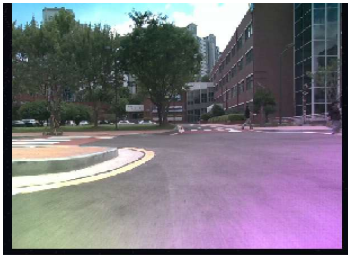
미니맵	클라우드포인트	카메라
		

그림 2-21. 경로차단상태

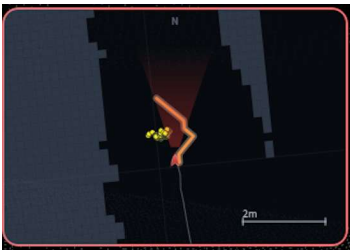
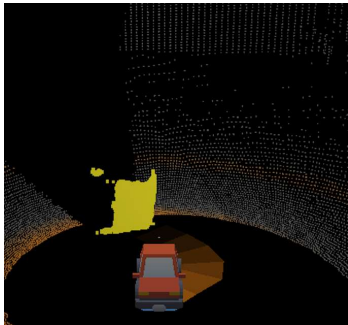
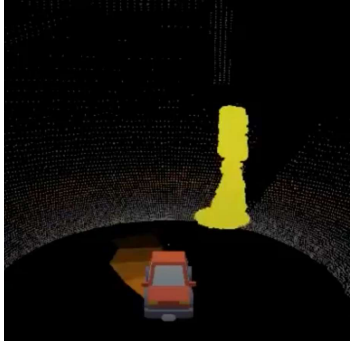
미니맵	클라우드포인트	카메라
		

그림 2-22. 회피경로표시

미니맵	클라우드포인트	카메라
		

(3) VLM 설명 패널

로봇이 장애물을 회피하거나 정지하는 판단을 내리면, 그 판단 정보가 VLM에 전달되어 "전방에 보행자가 감지되어 우측으로 회피하였습니다"와 같은 자연어 설명 문장이 생성된다. 설명 패널은 이 문장을 화면에 실시간으로 표시하여, 운영자가 로봇이 왜 멈추었는지, 왜 경로를 바꾸었는지를 즉시 이해할 수 있도록 한다. 이는 자율주행의 판단 과정이 블랙박스처럼 보이는 문제를 줄이는 본 프로젝트의 핵심 기능이다.

그림 2-23. VLM설명패널



(4) 원격 정지 제어

웹 대시보드의 정지 버튼을 누르면 제어 채널을 통해 로봇에 정지 명령이 전달된다. 명령은 고유 번호와 수신 확인 응답으로 관리되어, 네트워크가 불안정한 상황에서도 명령이 유실되거나 중복 실행되지 않도록 처리된다. 이를 통해 운영자는 위험 상황 발생 시 원격에서 즉시 로봇을 멈출 수 있다.

그림 2-24. 원격 정지 제어



● 성능 측정 및 표시 설정 결정

성능 측정에 사용한 환경은 다음과 같다.

구분	장비	사양	용도
개발용 보드	NVIDIA Jetson Orin Nano Developer Kit	6코어 Arm Cortex-A78AE CPU, Ampere 아키텍처 GPU (CUDA 코어 1,024개), 8GB LPDDR5 메모리	영상 전송 지연 측정, 포인트 수별 렌더링 성능 측정
로봇 탑재 PC	NVIDIA Jetson AGX Xavier	8코어 Carmel(ARM v8.2) CPU, Volta 아키텍처 GPU, 16GB LPDDR4x 메모리	실차 연결 테스트

(1) 영상 전송 지연 측정

개발용 보드 환경에서 네이티브 뷰어와 웹 대시보드를 연결하여 WebRTC 영상 전송 지연을 측정하였다.

측정 항목	측정값	비고
영상 지연 (왕복, RTT)	12 ms	
영상 지연 (편도)	6 ms	RTT의 1/2로 환산

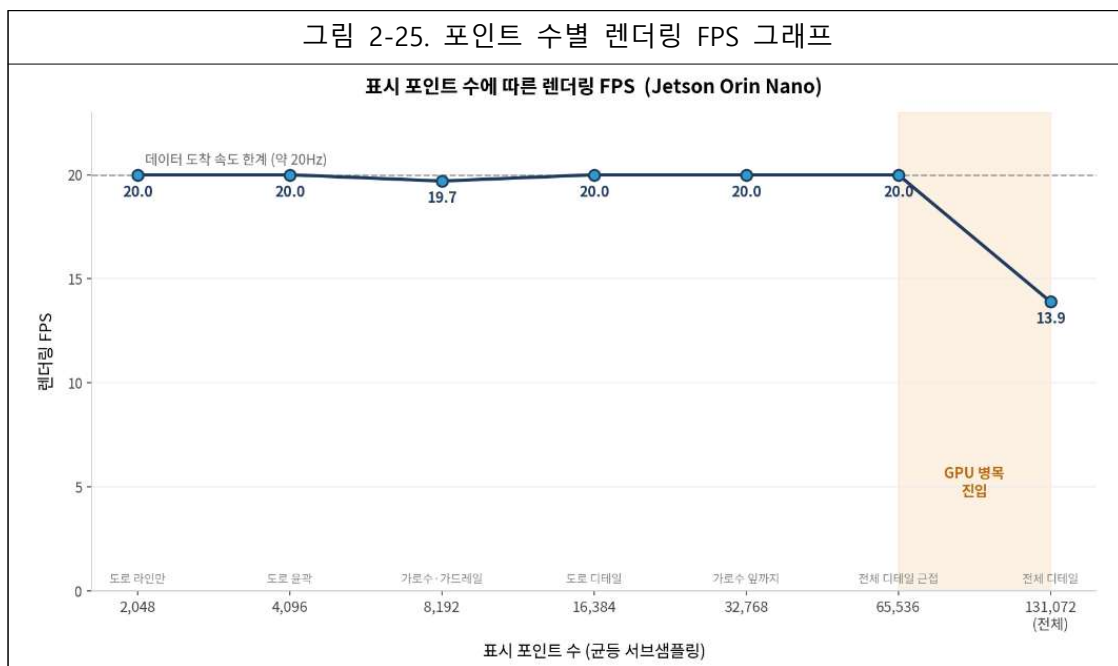
같은 네트워크 환경 기준 편도 약 6ms의 지연으로, 실시간 관제에 충분한 수준임을 확인하였다. 다만 외부망 환경에서의 지연은 프로젝트 기간 내 직접 측정하지 못하였으며, 통신 중계(TURN) 서버를 경유하는 일반적인 인터넷 환경 특성을 고려하면 편도 30~60ms 수준으로 추정된다.

(2) 포인트 수별 렌더링 성능 측정

Jetson Orin Nano 개발용 보드 환경에서 표시 포인트 수를 단계적으로 늘려가며 렌더링 FPS(초당

화면 갱신 횟수)를 측정하였다. 표시 포인트 수는 전체 131,072점의 LiDAR 데이터에서 일정 간격으로 점을 추출하는 균등 서브샘플링 방식으로 조절하였으며, 추출 간격을 절반씩 줄여가며 총 7단계로 측정하였다. 이 방식은 점 수가 줄어도 장면 전체의 형태가 고르게 유지되기 때문에, 단계별 화면 품질을 함께 비교할 수 있다.

표시 포인트 수	FPS	화면 품질
2,048	20.0	도로 좌우 라인만 식별 가능
4,096	20.0	도로 윤곽이 어렴풋이 보임
8,192	19.7	가로수·가드레일 식별 가능
16,384	20.0	도로 디테일 표현
32,768	20.0	가로수 앞까지 표현
65,536	20.0	전체 디테일에 근접
131,072 (전체)	13.9	전체 디테일 표현, GPU 병목 진입



측정 결과에서 다음과 같은 특성을 확인하였다.

첫째, 약 6.5만 점(65,536점)까지는 FPS가 약 20으로 일정하게 유지되었다. 이는 LiDAR 데이터가 초당 약 20회 도착하기 때문으로, 이 구간에서는 GPU에 충분한 여유가 있어 화면 갱신 속도가 데이터 도착 속도에 의해 결정된다. 즉 포인트 수를 더 줄여도 FPS가 올라가지 않는, 데이터 도착 속도가 병목인 구간이다.

둘째, 전체 131,072점을 표시하는 경우에만 FPS가 13.9로 하락하며, 이 지점에서 GPU의 그리기 연산이 병목으로 전환된다. 다만 하락 폭은 최대 성능 대비 약 30% 수준으로, 전체 포인트를 표시하더라도 관제에 필요한 화면 갱신 속도는 유지되는 것을 확인하였다.

셋째, 균등 서브샘플링 특성상 점 수가 줄어도 장면의 전체 구조는 유지되지만, 세부 표현력은 점 수에 비례하여 뚜렷하게 달라졌다. 약 2천 점에서는 도로의 좌우 라인 정도만 식별 가능한 반면, 3만 점 이상에서는 가로수 잎과 같은 세부 요소까지 표현되었다.

다만 측정 과정에서 동일한 데이터를 표시하더라도 개발용 보드와 로봇 탑재 PC 간에 포인트클라우드가 일부 다르게 렌더링되는 현상이 확인되었다. 두 환경의 그래픽 처리 방식 차이에서 비롯된 것으로 보이나, 정확한 원인 분석과 해결은 프로젝트 기간 내 완료하지 못하여 향후 개선 과제로 남겨두었다.

(3) 표시 포인트 수 결정 — 최저 FPS, 최고 가시성

위 측정을 통해 표시 포인트 수와 화면 갱신 속도 사이의 상충 관계를 확인하였다. 약 6.5만 점 이하로 제한하면 20 FPS의 부드러운 화면을 얻을 수 있는 반면 세부 표현력이 낮아지고, 전체 포인트를 표시하면 환경을 가장 상세하게 표현할 수 있지만 FPS는 13.9로 하락한다. 어느 쪽을 선택할지 판단하기 위해 동일한 장면을 7단계 포인트 수로 표시한 화면을 비교하였다.

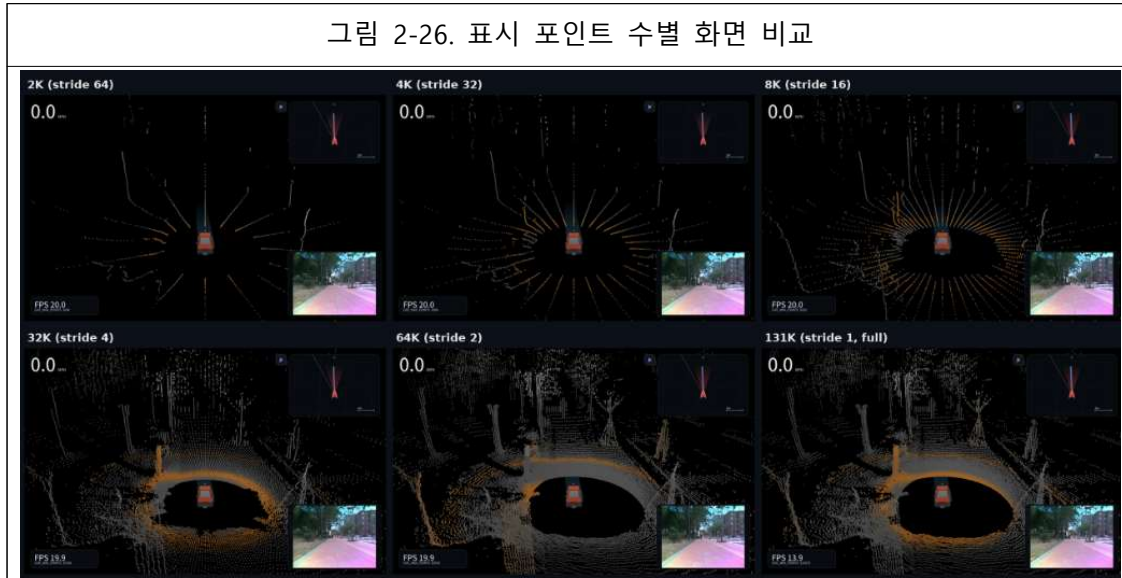


그림 2-26에서 확인할 수 있듯이, 약 2천 점에서는 LiDAR의 방사형 스캔 라인 정도만 표시되어 주변 환경을 파악하기 어렵고, 점 수가 늘어날수록 도로면, 연석, 가로수, 건물 윤곽이 점차 채워진다. 약 3만 점 이상에서 주행 환경의 구조가 뚜렷해지며, 전체 포인트(131,072점)에서는 노면의 형태와 주변 구조물까지 가장 상세하게 표현된다.

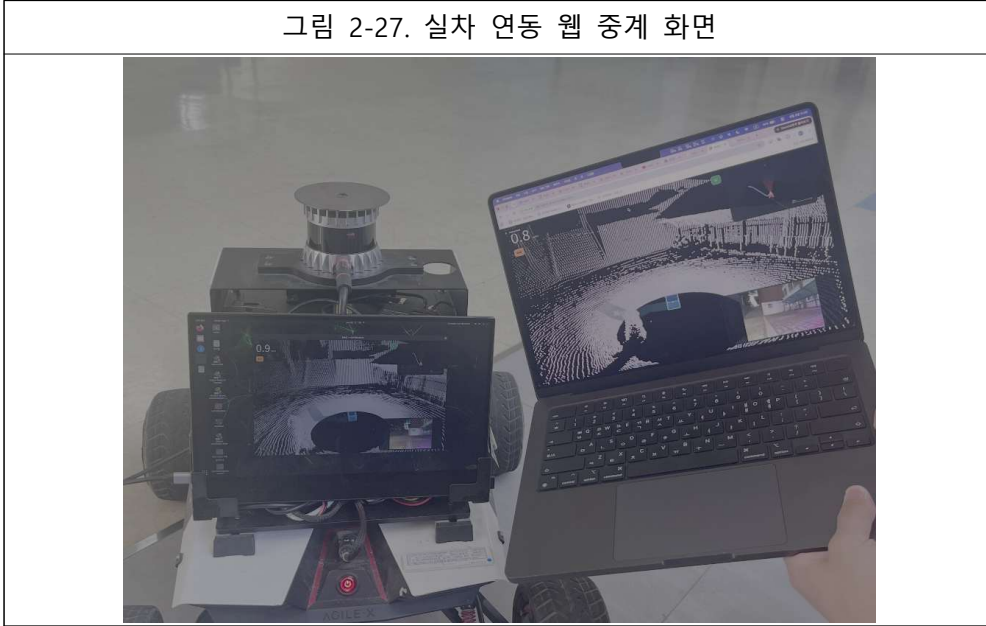
반면, 13.9 FPS와 20 FPS의 차이는 실제 관제 화면에서 체감되는 부드러움의 차이가 크지 않았으며, 13.9 FPS에서도 주행 상황을 파악하는 데 불편함이 없는 수준이었다. 반면 포인트 수에 따른 환경 표현의 차이는 화면에서 뚜렷하게 드러났다. 관제 서비스의 목적이 빠른 화면 갱신보다 주변 환경과 장애물을 정확하게 파악하는 것에 있다는 점을 고려하여, 본 시스템은 최저 FPS·최고 가시성, 즉 전체 포인트(131,072점)를 표시하는 구성을 채택하였다. 측정으로 도출한 6.5만 점 기준은 향후 더 빠른 화면 갱신이 필요한 상황에서 포인트 수를 조절하기 위한 운영 기준으로 활용할 수 있다.

● 자율주행·VLM 모듈과의 연동 동작 흐름

전체 시스템이 함께 동작하는 과정을 하나의 시나리오로 설명하면 다음과 같다.

1. 자율주행 시스템이 LiDAR 데이터를 분석하여 전방 경로 상에 장애물을 감지한다.
2. 회피 경로를 생성하고 주행 모드를 회피 상태로 전환하며, 이때 발생한 판단 정보(회피 방향, 장애물과의 거리 등)를 기록한다.
3. VLM이 카메라 영상과 판단 정보를 입력받아 자연어 설명 문장을 생성한다.
4. 네이티브 뷰어가 설명 문장을 설명 패널에 표시하고, 미니맵에 회피 경로를 반영한 화면을 완성한다.
5. 완성된 화면이 WebRTC 영상으로 웹 대시보드에 전송된다.
6. 운영자는 웹 브라우저에서 3D 화면, 카메라 영상, 회피 판단의 이유를 동시에 확인한다.

그림 2-27. 실차 연동 웹 중계 화면



이러한 구조를 통해 본 웹 관제 서비스는 운영자가 현장에 가지 않고도 로봇의 주행 상태와 판단 과정을 투명하게 모니터링할 수 있는 환경을 제공한다. 또한 시각화는 네이티브 뷰어가, 표시와 관리 는 웹이 담당하는 명확한 역할 분리를 통해, 향후 기능을 추가하거나 다른 서비스로 확장할 때에도 각 부분을 독립적으로 발전시킬 수 있는 기반을 마련하였다.

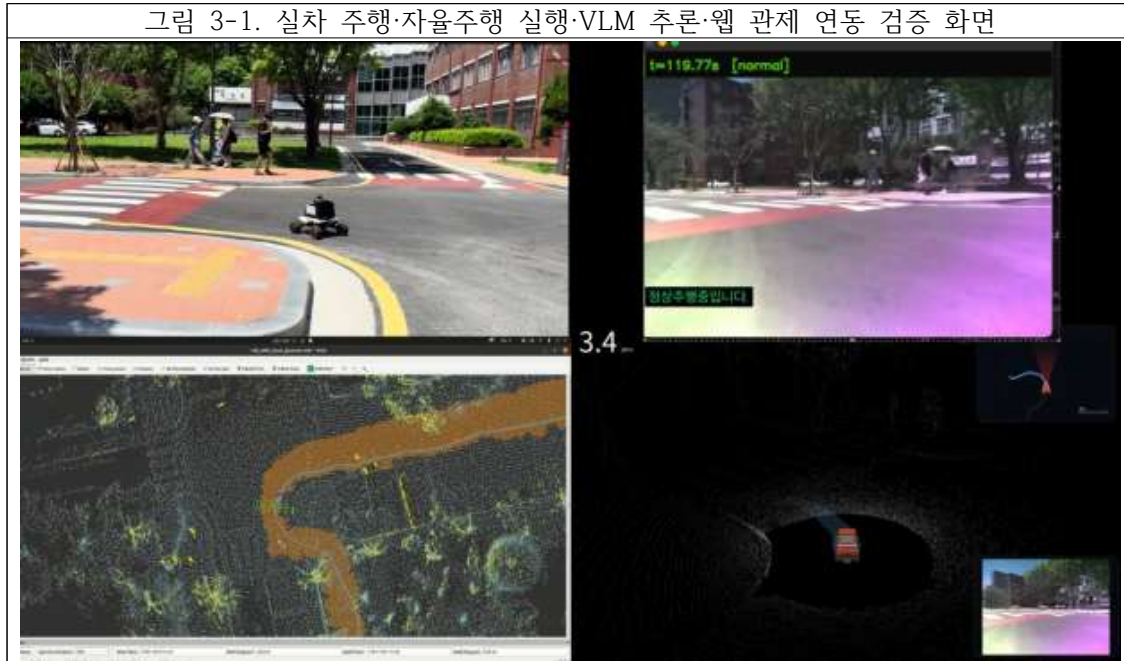
● 관제 서비스 차수별 개발목표 달성 현황

차수	계획 항목	달성 여부	비고
1차	웹-네이티브 뷰어 각각의 역할 정의	달성	시각화(네이티브 뷰어) / 중계-인증-관리(웹)로 명확히 분리
	포인트클라우드 → 메시 변환기 구현	미달성	3D 점 데이터를 면 단위 모델로 변환하는 기능은 미구현
	WebRTC 보안 검토 및 보완	달성	네트워크 환경별 연결 방식 구성, 인증 키 분리 관리 적용
	네이티브 뷰어를 배포 가능한 형태로 패키징	달성	설치 패키지 제작 및 자동 실행 환경 구성, 배포 자동화
중간	실제 로봇과 웹 관제 서비스 연결 및 실시간성 테스트	달성	로봇 탑재 환경에서 연결 확인 및 테스트 수행
	WebRTC 연결 유지 방안 구현	달성	명령 수신 확인 구조, 외부망 중계 연결 대응 구현
	실제 로봇 경로 제어 구현	부분 달성	경로 제어 전반은 미구현, 원격 정지 명령만 구현
3차	로봇 탑재 PC 이식 시 포인트클라우드 렌더링 차이 해결	미달성	개발 PC와 로봇 탑재 PC 간 렌더링 차이 미해결, 향후 개선 필요
	회피 경로 시각화 강화	달성	미니맵에 추종-후보-회피 경로 표시, 주행 상태별 색상 구분 구현
	긴급 정지 버튼 구현 및 테스트	달성	원격 정지 명령 구현 및 테스트 완료

5-4 통합 데모 시연 평가

● 통합 시스템 시연 구성

본 프로젝트의 최종 데모는 실외 자율주행 로봇, 자율주행 실행 시스템, VLM 기반 설명 모델, 웹 관제 서비스가 하나의 흐름으로 연동되는지를 확인하는 방식으로 진행하였다. 데모에서는 로봇이 실제 실외 환경에서 주행을 수행하는 모습과 함께, 자율주행 실행 화면, 카메라 기반 VLM 추론 화면, 웹 관제 서비스 출력 화면을 동시에 확인하였다.



통합 데모 과정에서 각 모듈의 동작을 함께 확인하기 위한 검증 화면이다. 좌측 상단은 실제 실외 환경에서 로봇이 주행하는 모습을 외부 카메라로 촬영한 영상이며, 좌측 하단은 RViz 기반 자율주행 실행 화면으로 전역 지도, 주행 가능 영역, 경로, 로봇 위치를 확인할 수 있다. 우측 상단은 로봇 전방 카메라 영상과 VLM 설명 결과가 표시되는 화면이며, 우측 하단은 웹 관제 서비스에서 출력되는 3D 포인트클라우드 기반 관제 화면이다. 이를 통해 실제 로봇 주행 상황, 내부 자율주행 판단 과정, VLM 설명 생성 결과, 관제 화면 출력이 서로 연동되어 동작함을 확인하였다.

그림 3-2. 웹 관제 서비스 최종 출력 화면



본 프로젝트에서 최종적으로 구현한 웹 관제 서비스의 출력 화면이다. 해당 화면은 로봇의 주행 상태를 운영자가 직관적으로 확인할 수 있도록 3D 포인트클라우드 화면, 로봇 속도, 주행 모드, 미니맵, 전방 카메라 영상, VLM 기반 설명 문장을 하나의 관제 화면에 통합하여 표시한다.

화면 중앙에는 LiDAR 포인트클라우드를 기반으로 구성된 주변 환경과 로봇의 현재 위치가 표시되며, 전방의 장애물은 강조되어 표현된다. 우측 상단의 미니맵은 로봇의 현재 위치, 주행 경로, 회피 경로를 위에서 내려다본 형태로 보여준다. 우측 하단에는 로봇 전방 카메라 영상이 함께 표시되어, 운영자가 실제 영상과 포인트클라우드 기반 인식 결과를 동시에 비교할 수 있도록 하였다.

또한 화면 좌측에는 현재 속도와 주행 상태가 표시되며, 장애물이 감지되어 회피 판단이 수행되는 경우 '회피' 상태가 나타난다. 하단에는 VLM이 생성한 자연어 설명 문장이 표시된다. 예를 들어 그림 3-2에서는 "정면 209cm 오른쪽 9cm에 사람이 있습니다. 회피 경로를 장애물 오른쪽으로 생성하여 주행합니다."라는 설명을 통해, 로봇이 어떤 객체를 인식했고 어떤 방향으로 회피 경로를 생성했는지를 사용자가 즉시 이해할 수 있다.

구분	기능정의	세부기능 설명
1	로봇 플랫폼 구성 기능	Scout mini 로봇 차체에 LiDAR, IMU, 카메라, 연산 PC를 탑재하여 실외 자율주행과 VLM 추론, 웹 관제 연동이 가능한 하드웨어 환경을 구성한다.
2	실외 자율주행 및 회피 기능	LiDAR와 IMU 기반 위치 추정, 전역 경로 생성, 경로 추종을 수행하며, 주행 중 장애물이 감지되면 회피 경로 생성 또는 정지 판단을 통해 안전하게 주행한다.
3	VLM 기반 설명 생성 기능	카메라 영상과 로봇의 주행 판단 정보를 활용하여 장애물 종류, 위치, 회피 방향 등을 자연어 문장으로 생성한다.
4	웹 관제 서비스 기능	로봇의 주행 상태, 3D 포인트클라우드, 미니맵, 카메라 영상, VLM 설명 로그를 웹 화면에서 실시간으로 확인할 수 있도록 제공한다.

6. 프로젝트 진행내용

1) 참여인원 및 담당 역할

연번	소속학과	성명	수행역할 분담내용
1	인공지능학부	고병재	팀장 및 고성능 자율주행 모델 제작
2	인공지능학부	김주완	팀원 및 VLM을 활용한 설명 가능한 모델 제작
3	인공지능학부	이승원	팀원 및 웹 관제 서비스 제작

2) 회의 및 SW멘토링 진행

번호	일시/장소	회의/멘토링 내용(상세히 작성)	관련 사진
1	2026. 4. 13. (11:00~13:00) / AI 융합대학	이번 회의에서는 매핑용 주행 데이터 수집 진행 상황을 공유하였다. AI융합대학, 스토리움, 학군단 일대를 중심으로 실제 주행을 수행하며, 지도 생성을 위한 환경 데이터를 수집하였다. 주행 중에는 주변 구조를 반영할 수 있는 LiDAR 데이터와 전방 및 주변 상황 확인을 위한 카메라 영상, 자세 변화 확인을 위한 IMU 데이터, 위치 확인을 위한 GPS 및 odometry 정보를 함께 기록하였다. 이러한 데이터는 단순 주행 기록이 아니라, 이후 지도 생성과 맵 정합, 반복 주행 시 위치 추정 정확도 검토에 활용하기 위한 목적을 가지고 수집되었다. 또한 서로 다른 건물 배치와 주행 환경을 가진 구간을 포함함으로써, 다양한 조건에서의 매핑 가능성을 확인할 수 있도록 하였다.	
2	2026. 5. 04. (11:00~13:00) / AI 융합대학	수집된 매핑용 주행 데이터를 기반으로 실제 주행 평가를 진행한 내용을 공유하였다. 구축한 주행 구간에서 로봇이 경로를 따라 안정적으로 이동하는지 확인하고, 위치 추정 오차, 경로 이탈 여부, 장애물 주변 주행 안정성 등을 중점적으로 검토하였다. 또한 LiDAR, 카메라, IMU, GPS, odometry 데이터가 정상적으로 기록되는지 확인하여 향후 지도 생성과 위치 추정 성능 평가에 활용 가능한 데이터 품질을 점검하였다. 특히 AI융합대학, 스토리움, 학군단 일대 등 서로 다른 주행 환경에서 테스트를 수행함으로써 다양한 조건에서의 주행 가능성과 매핑 데이터 활용 가능성을 함께 평가하였다. 이번 결과는 이후 주행 경로 보완, 센서 데이터 동기화 점검,	

		지도 생성 정확도 개선을 위한 기초 자료로 활용될 예정이다.	
3	2026. 5. 20. (11:00~13:00) / AI 융합대학	이번 회의에서는 우천으로 인해 기존에 계획했던 실외 매핑용 주행 데이터 수집 대신, 실내 환경에서 주행 테스트를 진행한 내용을 공유하였다. 실내 주행은 건물 내부 복도와 비교적 제한된 공간을 중심으로 수행하였으며, 좁은 통로, 회전 구간, 정지 및 재출발 상황에서 로봇이 안정적으로 주행할 수 있는지를 확인하였다. 주행 중에는 LiDAR 데이터를 통해 실내 구조물과 벽면, 장애물 정보를 수집하였고, 카메라 영상을 함께 기록하여 주행 중 전방 및 주변 상황을 확인할 수 있도록 하였다. 또한 IMU와 odometry 데이터를 활용하여 실내 주행 중 로봇의 자세 변화와 이동 경로를 확인하였다. 실내 환경은 GPS 신호가 불안정하거나 수신이 어려운 특성이 있기 때문에, 이번 테스트에서는 GPS보다는 LiDAR, IMU, odometry 기반의 주행 상태 확인에 중점을 두었다. 이를 통해 실외 환경과는 다른 조건에서 센서 데이터가 어떻게 기록되는지 확인하였으며, 반복 주행 시 경로 추정과 맵 정합에 활용 가능한 데이터를 확보할 수 있었다. 특히 실내 구조물처럼 특징점이 비교적 뚜렷한 환경에서 LiDAR 기반 매핑 가능성을 검토하고, 주행 중 센서 데이터가 정상적으로 저장되는지 점검하였다. 이번 실내 주행 테스트는 기상 상황으로 인한 대체 실험이었지만, 실외 매핑 데이터 수집과는 다른 관점에서 로봇의 주행 안정성, 센서 기록 상태, 실내 환경에서의 위치 추정 가능성을 확인할 수 있었다. 이후 날씨가 안정되면 AI융합대학, 스토리움, 학군단 일대의 실외 주행 데이터를 추가로 수집하고, 기존 데이터와 비교하여 실내·실외 환경별 매핑 결과와 위치 추정 성능을 함께 검토할 예정이다.	
4	2026. 4. 14. (10:00~12:00) / 공대 7호관	- 기 개발된 LiDAR, 카메라, IMU, GPS 적용된 자율주행 로봇에 VLM 기반의 개발 기획 필요성과 목표에 대한 내용 점검 - 기존 실내 위주의 LiDAR, 카메라 등의 기술이 적용된 자율주행 로봇의 개념상 AMR 자체의 외부 환경에 적용하기 위	

		한 VLM 적용을 위한 데이터셋 구축과 전처리에 대한 추진 방안 안내 - VLM을 적용 할 수 있는 장면 이해, 질의응답, 상황 설명 등 다양한 멀티모달 응용 분야에서 성능과 자율주행 분야에서도 카메라, LiDAR, 등 멀티모달 데이터를 활용하여 주행 상황을 해석하거나, 주행 판단 과정에서 XAI에 대한 자율주행 시스템 내부의 의사결정 과정을 사용자 친화적인 형태로 전환 방향성 안내 등	
5	2026. 5. 22. (10:00~12:00) / 공대 7호관	- LiDAR, 카메라, IMU, GPS 적용된 자율주행 로봇에 VLM 기반의 개발 관련 진행 사항 검토 및 향후 프로젝트 마무리를 위한 사전 일정 점검 - 로봇 주행 중 수집한 bagfile 데이터(Camera, LiDAR, IMU, GPS), 주행 local path planning, obstacle, event, VLM 학습용 설명 데이터셋 분석 및 구축과 모델 학습 방향성 협의 - 교내 융합 기술원, 학군단, 스토리움 건물 주변 실외 구간 주행 예정에서 실증 PoC를 위한 안전 표지판 및 횡단 보도 등의 검증 방안 협의 - 기존 모듈형 자율주행 구조 위에 설명 기능과 웹 관제 기능을 연계하는 방식으로 구현하여 난이도를 조정 및 설명 가능성과 데이터 신뢰성을 확보 방안 협의 등	

7. 프로젝트 세부일정 및 내용

No.	작업 내용	4월				5월				6월				7월				담당자	비고
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4		
1	자율주행 베이스라인 모델 개발																	고병재	
	고성능 모빌리티 로봇 주행환경 설정 및 LiDAR 기반 맵 구축																		
2	자율주행 베이스라인 모델 적용 및 기본 주행 성능 검증																	고병재	
3	자율주행 수행을 통한 주행 로그 및 설명 학습용 데이터 수집																	고병재	
4	설명가능 자율주행을																	김주완	

[illegible]

8. 결과물에 대한 향후 활용계획

8-1. 기대효과 및 개선 방향

본 프로젝트는 특정 목적만을 위한 단일 서비스 로봇이 아니라, 실외 환경에서 자율주행과 장애물 회피가 가능하고, 주행 판단 과정을 설명할 수 있는 범용 모빌리티 로봇 플랫폼을 구축한다는 점에서 다양한 활용 가능성을 가진다. 기존 자율주행 로봇이 단순히 목적지까지 이동하거나 장애물을 회피하는 기능에 집중했다면, 본 시스템은 로봇이 어떤 상황을 인식하고 어떤 근거로 주행 판단을 수행했는지를 자연어 형태로 설명할 수 있다는 점에서 차별성을 가진다.

첫 번째 기대효과는 다양한 서비스 환경으로의 확장 가능성이다. 본 프로젝트에서 개발하는 자율주행 로봇은 특정 서비스에만 제한되지 않고, 배달, 순찰, 안내, 물류 운반 등 여러 분야에 적용될 수 있다. 예를 들어 야구장과 같은 대형 실외 시설에서는 관람객에게 음식이나 물품을 전달하는 배달 로봇으로 활용될 수 있으며, 기아 타이거즈와 같은 지역 스포츠 구단 또는 경기장 운영 기관과의 제휴를 통해 경기장 내 이동형 서비스 로봇으로 발전할 가능성도 있다. 또한 캠퍼스, 공원, 산업단지, 주차장, 행사장 등에서는 정해진 구역을 순회하며 이상 상황을 확인하는 순찰 로봇이나 안내 로봇으로도 활용될 수 있다.

두 번째 기대효과는 자율주행 시스템의 신뢰성 향상이다. 자율주행 로봇이 장애물을 회피하거나 정지할 때, 사용자는 단순히 결과만 보는 것이 아니라 로봇이 어떤 객체를 인식했고 왜 해당 판단을 수행했는지 확인할 수 있다. 이를 통해 로봇의 주행 과정이 블랙박스처럼 보이는 문제를 줄이고, 사용자가 자율주행 시스템의 판단 과정을 보다 쉽게 이해할 수 있다. 특히 VLM 기반 설명 로그는 주행 중 발생한 문제를 분석하거나, 특정 상황에서 로봇이 올바르게 판단했는지 검토하는 데 유용하게 활용될 수 있다.

세 번째 기대효과는 웹 관제 서비스를 통한 운영 편의성 향상이다. 로봇의 주행 영상, 현재 상태, 장애물 인식 결과, 설명 로그를 웹에서 확인할 수 있다면 운영자는 현장에 직접 가지 않고도 로봇의 주행 상황을 모니터링할 수 있다. 또한 주행 데이터와 설명 로그가 축적되면, 이후 로봇의 오작동 원인 분석, 주행 성능 개선, 서비스 시나리오 검증 등에 활용할 수 있다. 이는 단순한 자율주행 로봇 개발을 넘어, 실제 서비스 운영까지 고려한 시스템으로 확장될 수 있다는 장점을 가진다.

향후 개선 방향으로서는 먼저 실외 주행 안정성을 더욱 높일 필요가 있다. 실제 실외 환경은 조명 변화, 날씨, 보행자 움직임, 임시 장애물, 주차 차량 등 다양한 변수가 존재하기 때문에, LiDAR와 카메라 데이터를 안정적으로 융합하고, 변화하는 환경에서도 로봇이 안전하게 주행

할 수 있도록 인식 및 제어 알고리즘을 개선해야 한다. 특히 장애물 회피 과정에서 경로 추종과 제어 지연이 발생하지 않도록 local path planning과 control 모듈의 연동 성능을 고도화할 필요가 있다.

또한 VLM 기반 설명 기능의 정확도와 실시간성을 개선하는 것이 중요하다. 설명가능 자율주행에서 중요한 것은 단순히 자연어 문장을 생성하는 것이 아니라, 실제 로봇의 인식 결과와 주행 판단에 맞는 정확한 설명을 제공하는 것이다. 따라서 카메라 영상과 LiDAR 기반 장애물 정보를 함께 활용하여, 로봇이 실제로 회피 판단에 사용한 객체와 상황을 더 정확하게 설명할 수 있도록 개선해야 한다. 동시에 VLM 추론 속도를 최적화하여 실시간 주행 상황에서도 설명 로그가 지연 없이 제공될 수 있도록 해야 한다.

웹 관제 서비스 또한 향후 고도화가 필요하다. 현재는 주행 영상과 설명 로그를 확인하는 형태로 설계할 수 있지만, 이후에는 로봇의 위치, 주행 경로, 장애물 발생 위치, 회피 판단 시점, 시스템 상태 등을 시각화하는 기능을 추가할 수 있다. 또한 주행 이력을 저장하고 특정 시점의 영상과 설명 로그를 함께 조회할 수 있도록 하면, 운영자 입장에서 사후 분석과 유지보수가 더욱 편리해질 수 있다.

마지막으로 서비스 적용 분야에 맞춘 기능 확장이 필요하다. 배달 로봇으로 활용할 경우에는 목적지 설정, 주문 연동, 수령 인증, 적재함 제어 기능이 추가되어야 하며, 순찰 로봇으로 활용할 경우에는 이상 상황 감지, 순찰 경로 관리, 알림 기능 등이 필요하다. 안내 로봇으로 확장할 경우에는 사용자와의 상호작용 기능과 목적지 안내 기능이 요구될 수 있다. 따라서 본 프로젝트를 기반 플랫폼으로 삼고, 향후 적용 분야에 따라 서비스 기능을 모듈화하여 확장한다면 다양한 실외 자율주행 서비스로 발전할 수 있을 것으로 기대된다.

9. 참고자료 및 문헌

- [1] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.
- [2] Tixiao Shan, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," GitHub Repository, TixiaoShan/LIO-SAM.
- [3] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," IEEE International Conference on Robotics and Automation (ICRA), 2011.
- [4] ROS.org, "Robot Operating System (ROS) Documentation," ROS Noetic API Documentation.
- [5] ROS.org, "sensor_msgs/PointCloud2 Message Documentation," ROS Noetic API Documentation.
- [6] ROS.org, "nav_msgs/OccupancyGrid Message Documentation," ROS Noetic API

Documentation.

[7] ROS.org, "nav_msgs/Path Message Documentation," ROS Noetic API Documentation.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, Vol. 4, No. 2, pp. 100–107, 1968.

[9] OpenStreetMap Wiki, "OSM XML," OpenStreetMap Documentation.

[10] D. Fox, W. Burgard, and S. Thrun, "The Dynamic Window Approach to Collision Avoidance," IEEE Robotics & Automation Magazine, Vol. 4, No. 1, pp. 23–33, 1997.

[11] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated Online Trajectory Planning and Optimization in Distinctive Topologies," Robotics and Autonomous Systems, Vol. 88, pp. 142–153, 2017.

[12] Chen, Zhe, et al. "Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling." arXiv preprint arXiv:2412.05271 (2024).

[13] Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." Iclr 1.2 (2022): 3.

[14] Kim, Yoonshik, and Jaeyoon Jung. "Koffvqa: An objectively evaluated free-form vqa benchmark for large vision-language models in the korean language." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2025.