[두산로보틱스] 지능형 로보틱스 엔지니어

Linetracing and pick and place (디지털 트윈 기반 서비스 로봇 운영 시스템 구성)

TEAM F-1조

조장: 한준모

조원: 김도엽, 김태영, 이종휘(중도포기)

K-Digital Training

목 차

K-Digital Training

01 프로젝트 개요

1

프로젝트 주제 및 선정 배경, 기획의도

디지털 트윈 기반
서비스 로봇
운영 시스템 구성

2

프로젝트 내용

TurtleBot3 기반
가상 및 실제 환경에서의
자율주행 시스템 구현

3

활용 장비 및 재료

Turtlebot 3
Open Manipulation

4

프로젝트 구조

자율 주행
Image→ Image preprocess
→Lane detect → control

Manipulation
Image → pick & place

5

활용방안 및 기대 효과

이미지, 라이다 센서 기반
자율주행 기술의 탐구

02 K-Digital Training

프로젝트 팀 구성 및 역할

훈련생	역할	담당 업무	
한준모	팀장	프로젝트 총괄	Manipulation
김도엽	팀원	레인 트래킹 최적화	GUI 제작
김태영	팀원	장애물 회피 구현(라이다)	파라미터 최적화

K-Digital Training

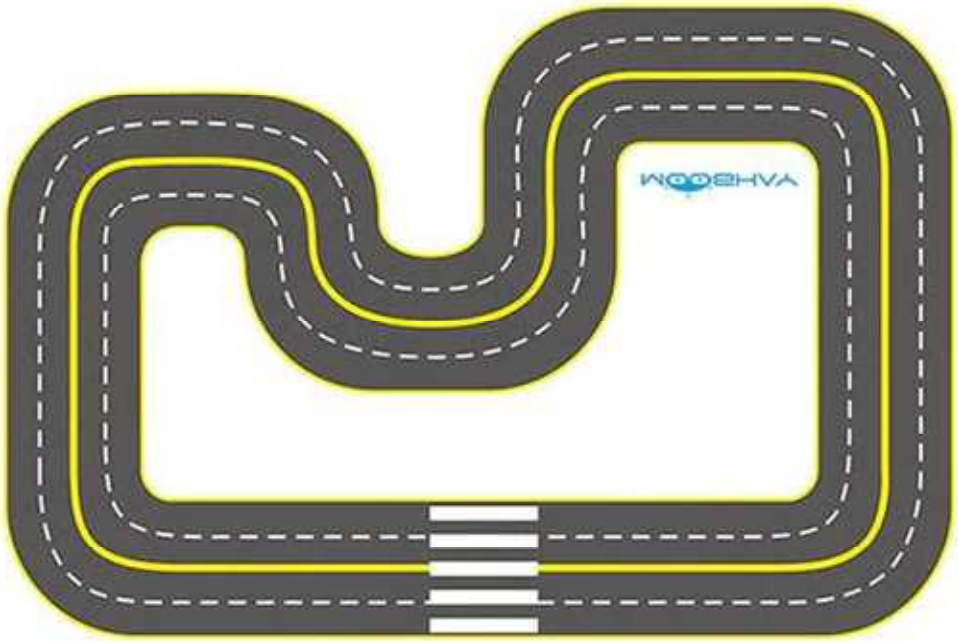
03 프로젝트 수행 절차 및 방법

▶ 프로젝트의 기획 및 과정

구분	기간	활동	비고
사전 기획	5/9(금)	프로젝트 기획 및 주제 선정 / 계획 수립 및 역할 분배	아이디어 선정
예제 코드 실습	5/12(월) ~ 5/13(화)	예제 코드 실습 / 관련 개념 학습 및 자료 조사	
가상 환경 내 구현	5/14(수) ~ 5/16(금)	Lane detection 파라미터 조정 / Lane tracking 최적화	
실제 주행 실습	5/19(월)	실제 환경 내 파라미터 조정	팀별 중간보고 실시
로봇팔 조작	5/20(화) ~ 5/22(목)	실제 환경 내 자율주행 최적화 / Manipulation 구현	최적화, 오류 수정
총 개발기간	5/9(금) ~ 5/22(목)(총 2주)		

03 K-Digital Training

프로젝트 수행 목적



레인 트레이싱 기능 구현

- Gazebo 가상환경에서 레인 감지 및 자율주행 알고리즘 구현

- 가상 환경과 실제 환경의 오차 보정 및 안정적인 주행 구현

Manipulation 기능 구현

- 가상환경 내에서 기본적인 pick & place 기능 구현

- 객체 인식 후 manipulator로 지정 위치로의 이동 구현

04 K-Digital Training 프로젝트 수행 경과

Lane tracking Debuging

기존 코드 활용

실제 라인은 모두 노란색

→ 방향 정보 X

흰색 점선 인식

최소 인식 크기 줄임

→ 노이즈 증가

ROI 및 회복 노드

조감도 → 관심 영역 설정

차선 감지 실패 시 정지 후 탐지

회복 노드 개선

차선 감지 실패 시
이전 데이터를 바탕으로 주행

차선 침범 감지

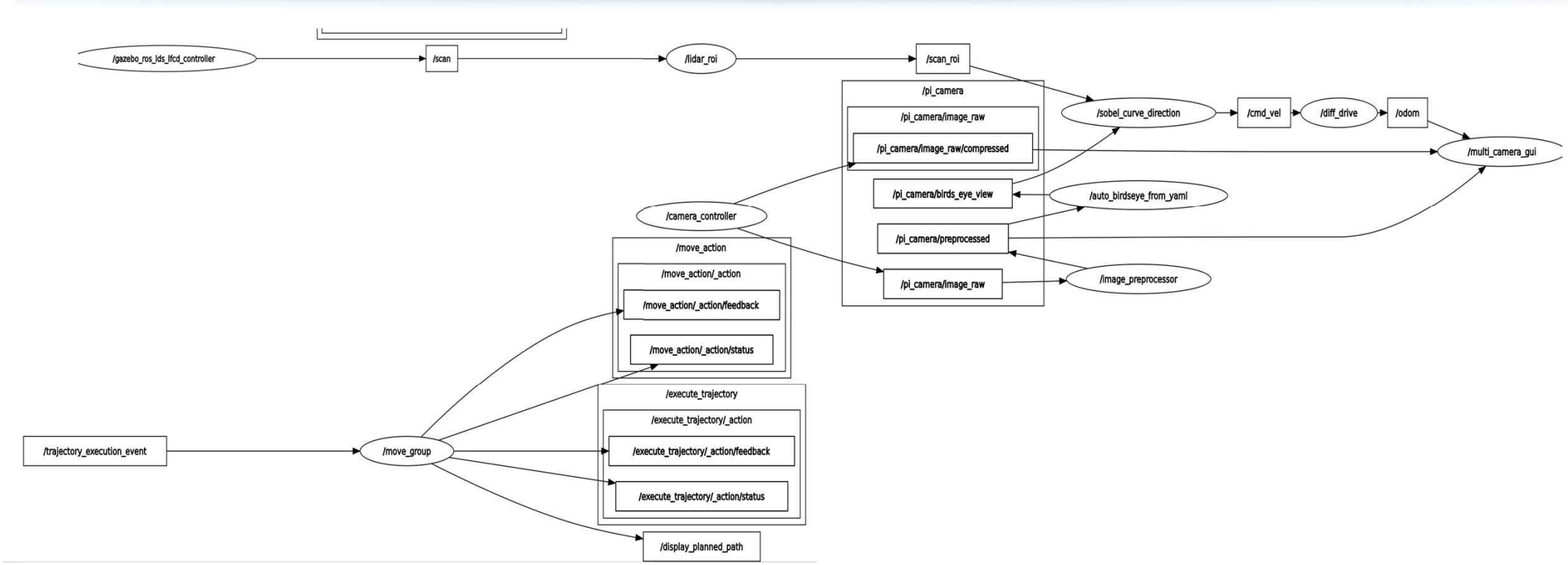
차선 침범 감지 시
이전 주행과 반대로 주행

Path 기반 주행

맵 크기를 2배 증가 후
두 라인 사이 중앙 path 설정

프로젝트 수행 경과

Skeletonization

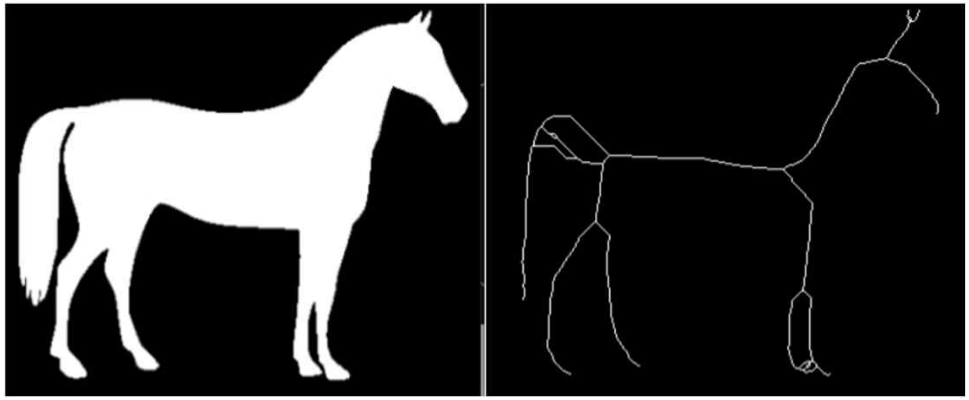


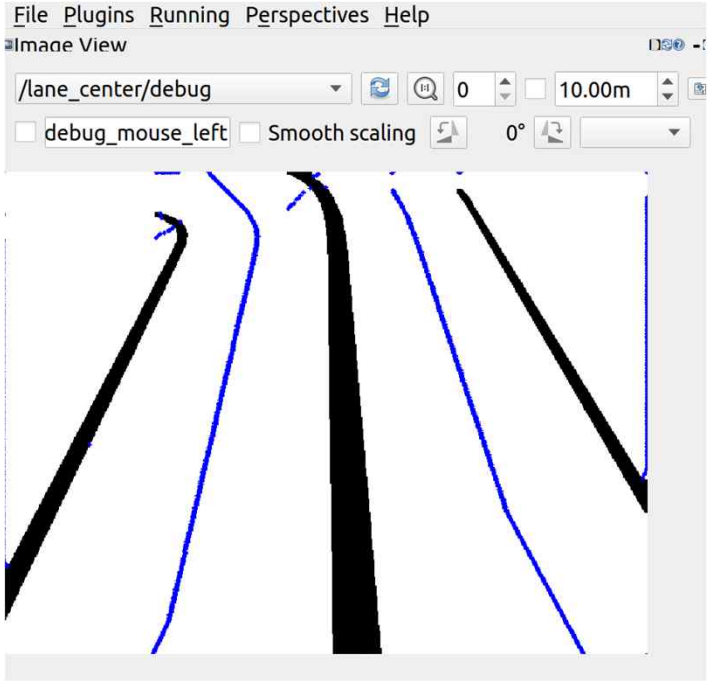
프로젝트 수행 경과

Skeletonization

Skeletonization 이란?

객체의 윤곽선을 유지하면서 두께를 줄이는 기법



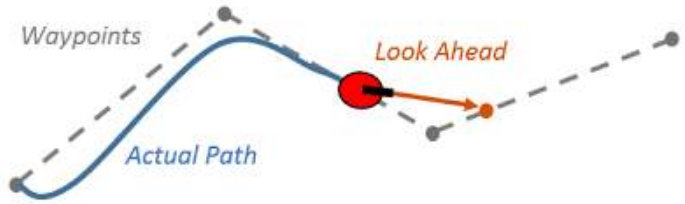


04 K-Digital Training 프로젝트 수행 경과

Pure pursuit

Pure pursuit 이란?

차량이 목표 지점을 추종하도록 조향각을 계산하는 알고리즘



조향각 계산

$$\delta = \tan^{-1}\left(\frac{2L \cdot y}{L_d^2}\right)$$

- L: 휠베이스
- y: 로컬 좌표계 기준 목표점의 y값 (횡방향 거리)
- L_d: lookahead 거리

Pure pursuit workflow

현재 위치 인식 및 헤딩

↓

Look Ahead point 지정

↓

목표점까지 궤적 계산

↓

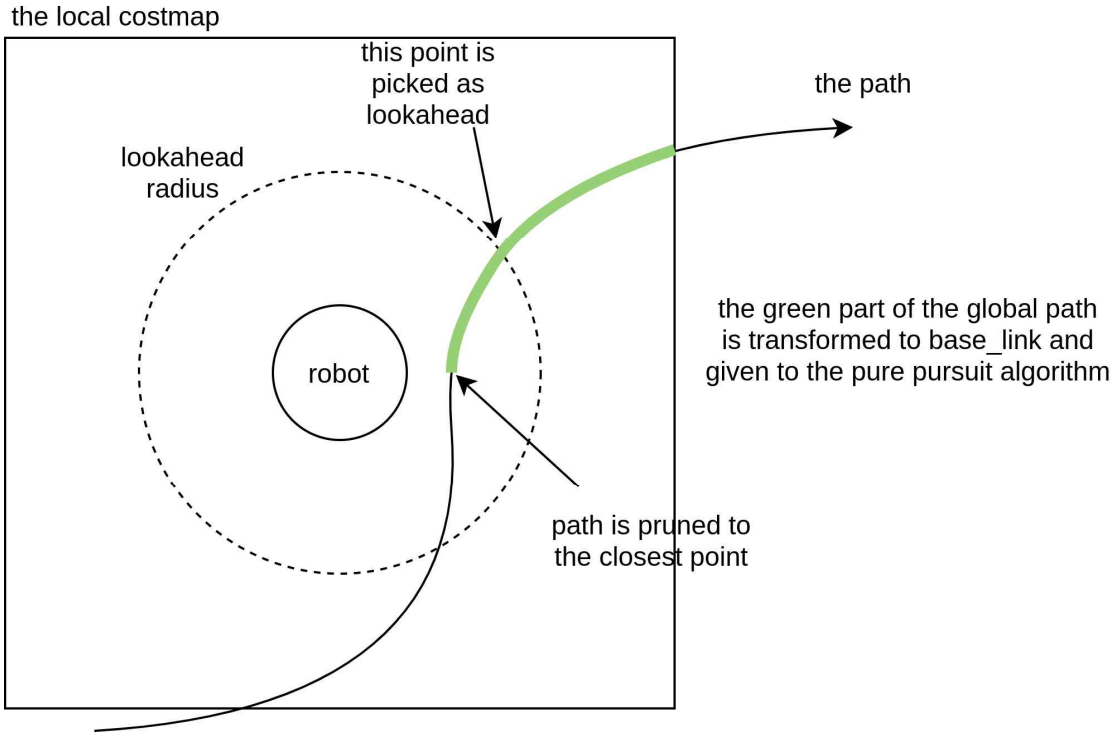
조향각 계산

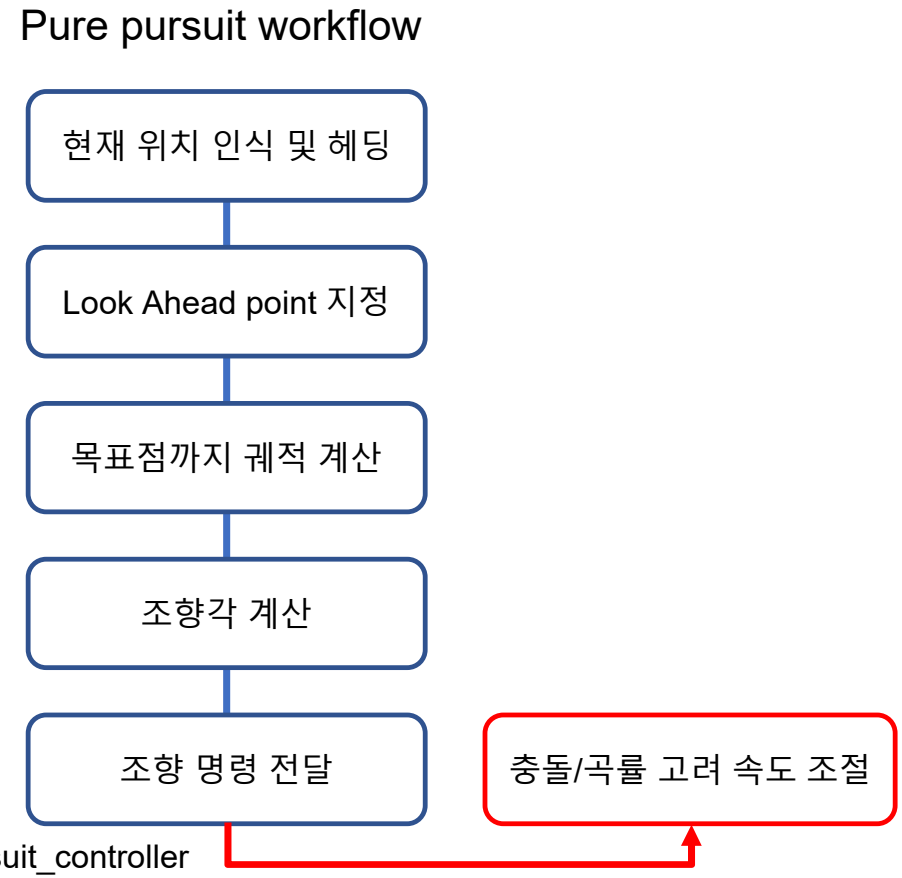
↓

조향 명령 전달

프로젝트 수행 경과

Regulated Pure pursuit





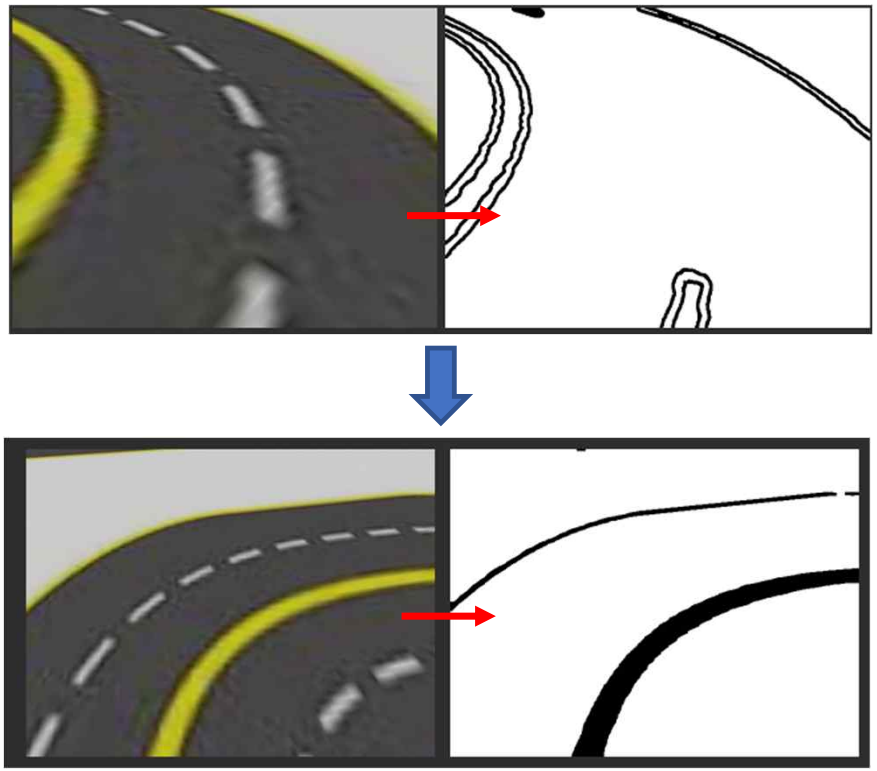
출처: https://github.com/ros-navigation/navigation2/tree/main/nav2_regulated_pure_pursuit_controller

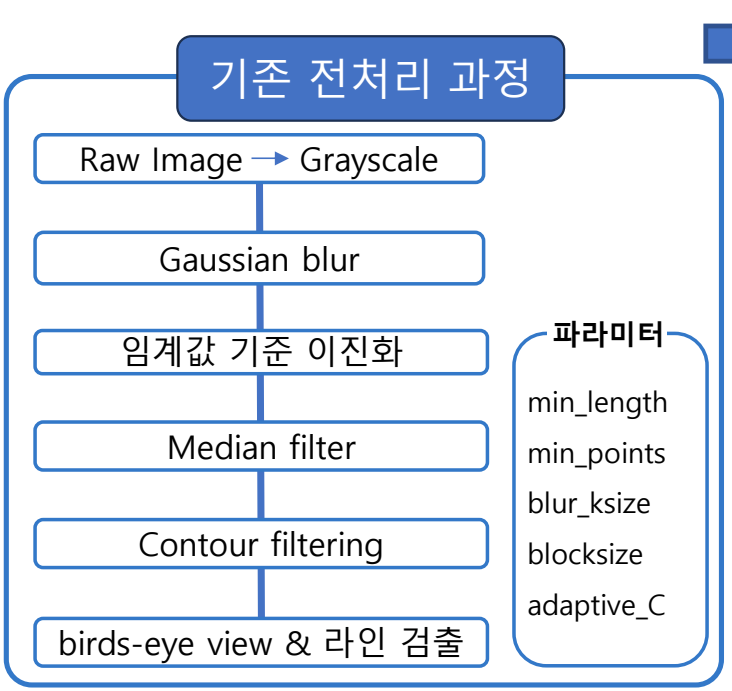
K-Digital Training

04 프로젝트 수행 경과

▸ 전처리 Node 구조 단순화

• 효율적인 데이터 추출을 위한 이미지 전처리 과정

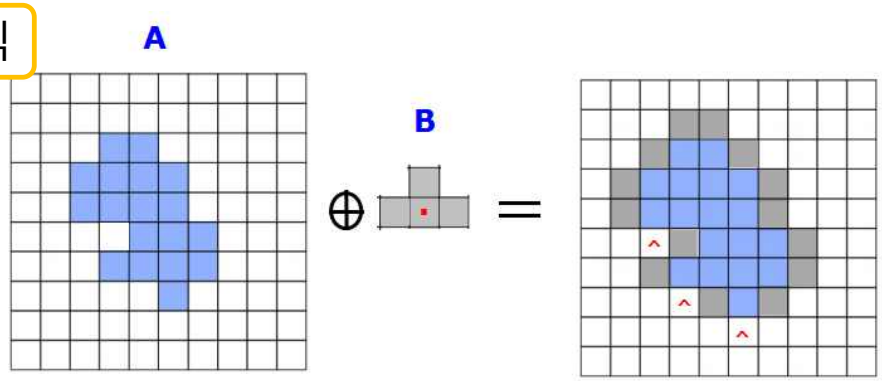




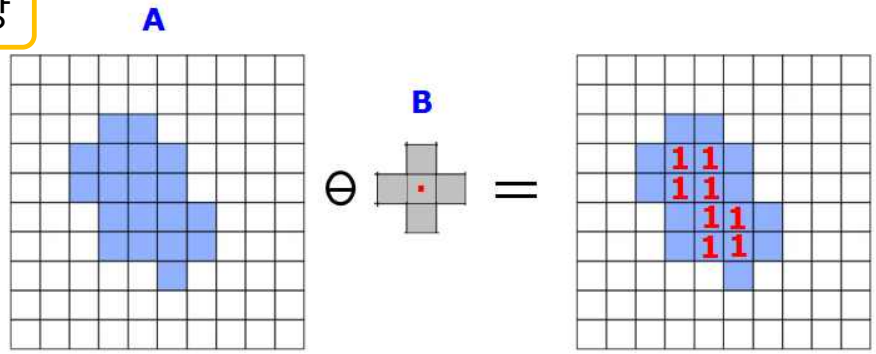
프로젝트 수행 경과

Morphology

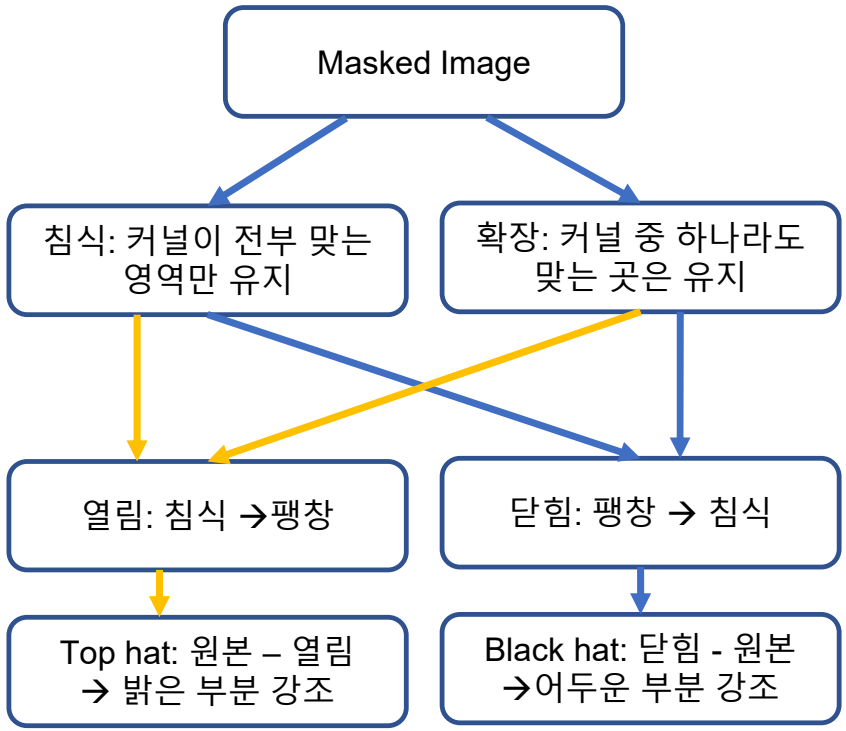
침식



확장



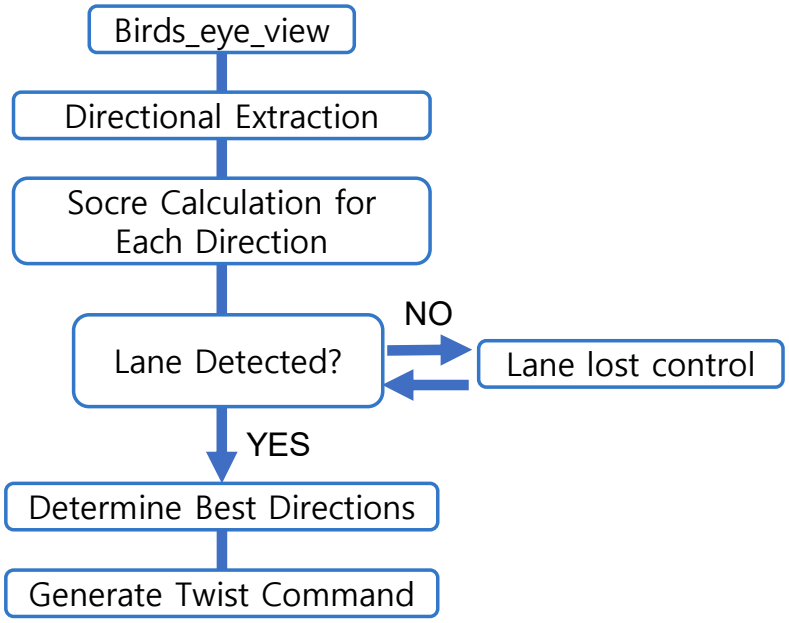
Morphology workflow



04 프로젝트 수행 경과

Lane Control Node

- 탑뷰로 변환된 이미지에서 선의 방향을 분석하여 로봇의 진행방향을 결정, 속도 제어



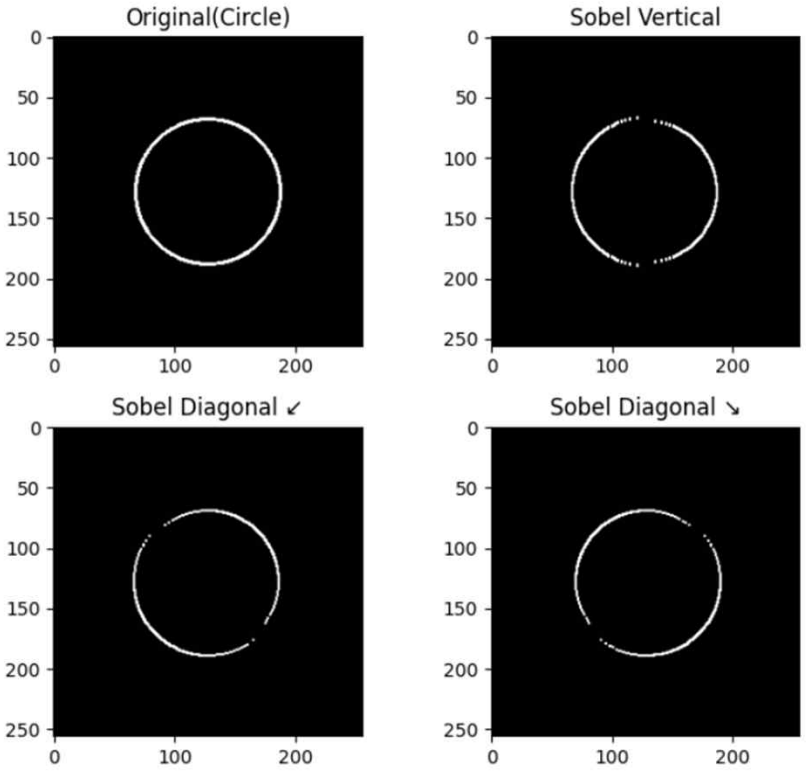
파라미터	역할
score threshold	주행 중 레인 인식 실패로 간주하는 **기준**
max_arg	회전 시 최대 **각속도**
base_lin	**기본 전진 속도**
gain	각속도를 조절하기 위한 **가중치**
lost_rotate_gain	레인 인식 실패 시 복원용 **각속도**

인식 성공	인식 실패
	

K-Digital Training

04 프로젝트 수행 경과

Sobel filter based twist



Sobel filter

방향에 따른 점수 부여

Vertical → 수직선

좌하향 대각선(↙)
→ 우회전 방향감지

우하향 대각선(↘)
→ 좌회전 방향 감지

주행 판단

점수를 기준으로
주행 방향 결정

↙ ↘ 점수 차이로
각속도 결정

차선 상실

점수 < 임계값

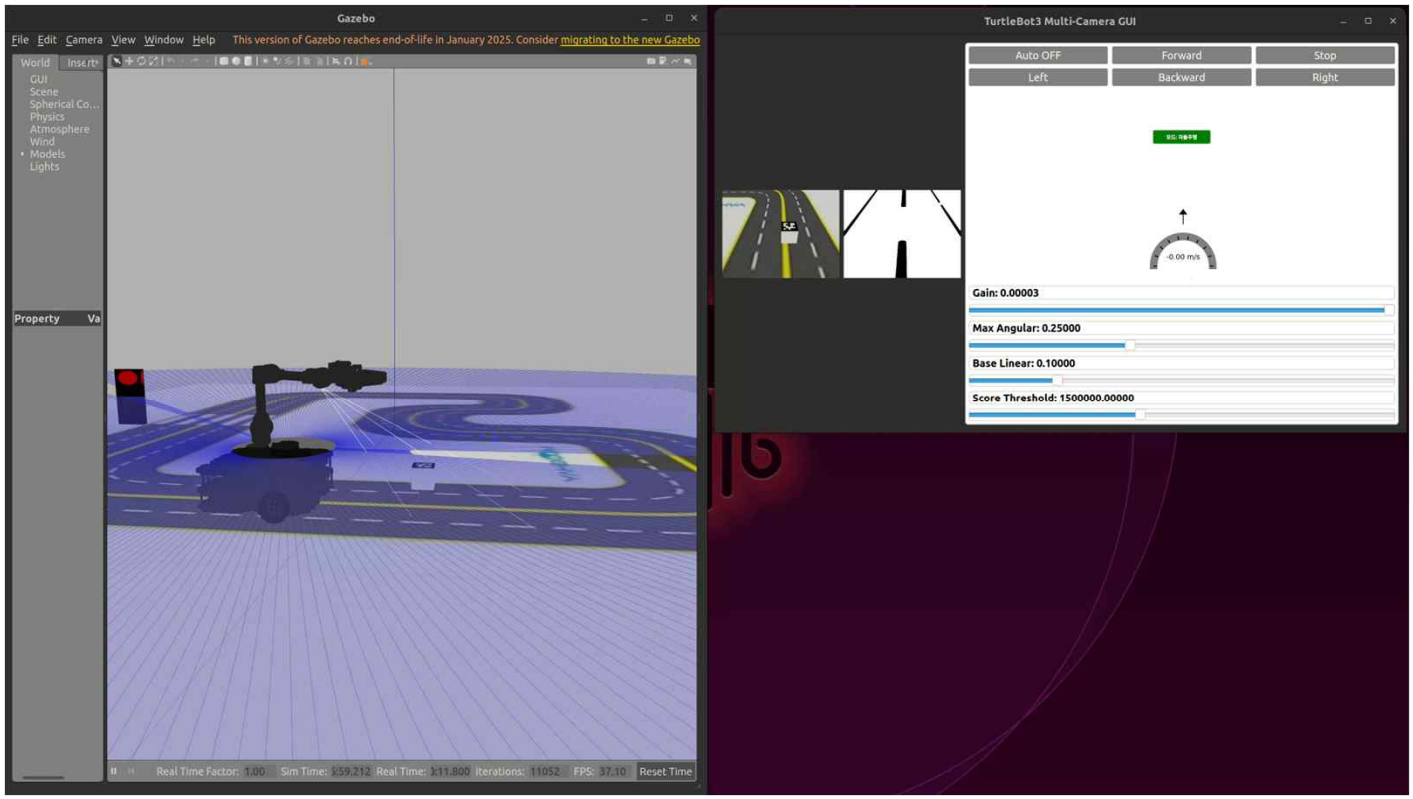
→ 차선 상실 판단

→ 과거 데이터로 주행

04 K-Digital Training

프로젝트 수행 경과

Simulation

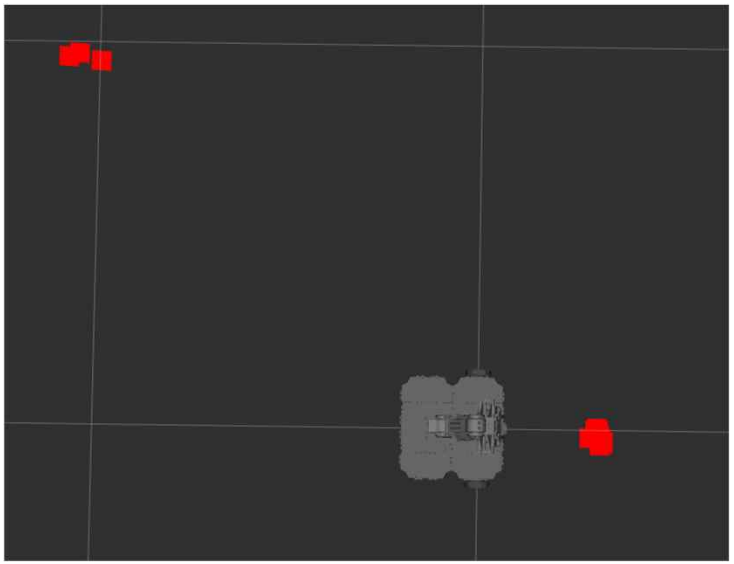


K-Digital Training

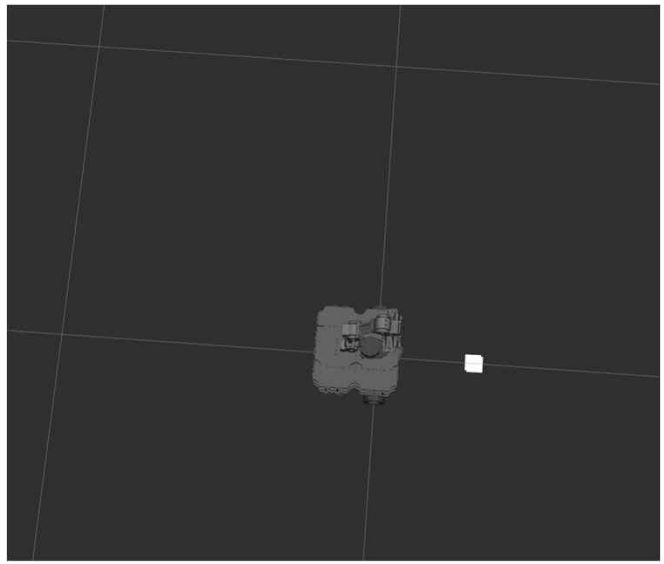
04 프로젝트 수행 경과

Simulation

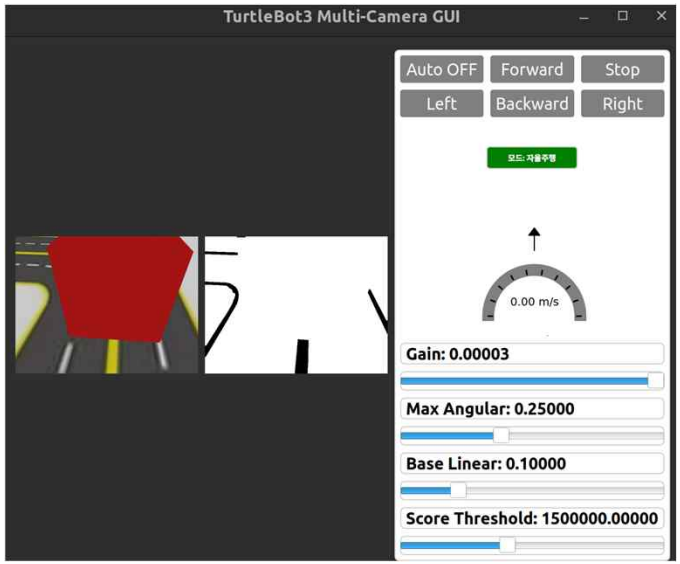
라이다 센서



라이다 센서 ROI



장애물 감지 시 정지

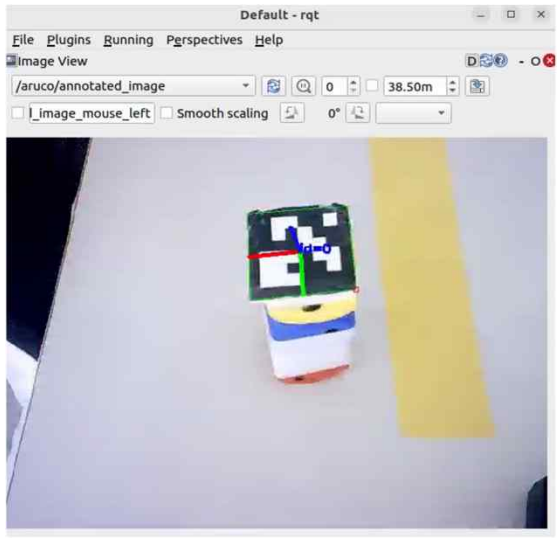


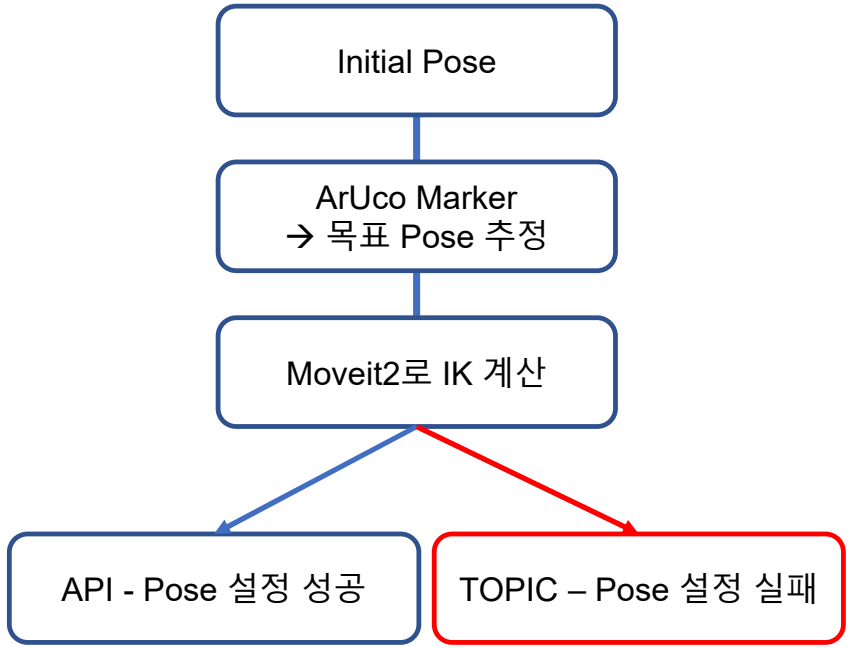
프로젝트 수행 경과

Manipulator

<Init pose>

<Aruco Marker Detection>



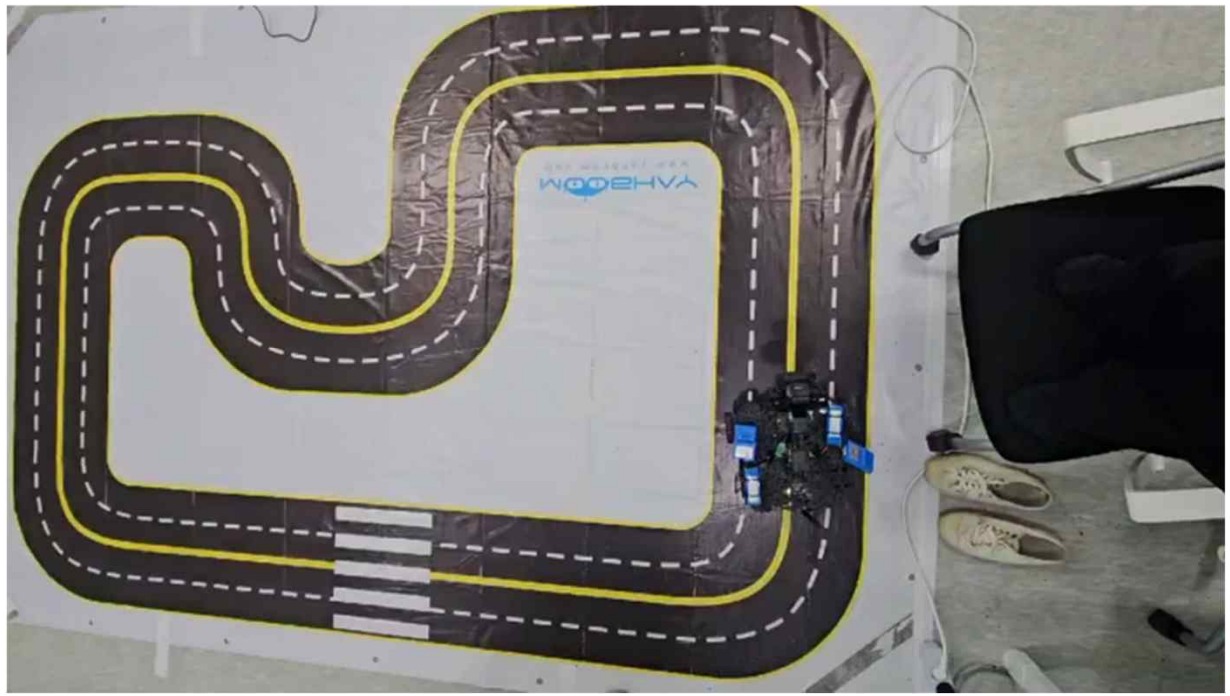


프로젝트 수행 경과

Manipulator

프로젝트 수행 경과

Lane control

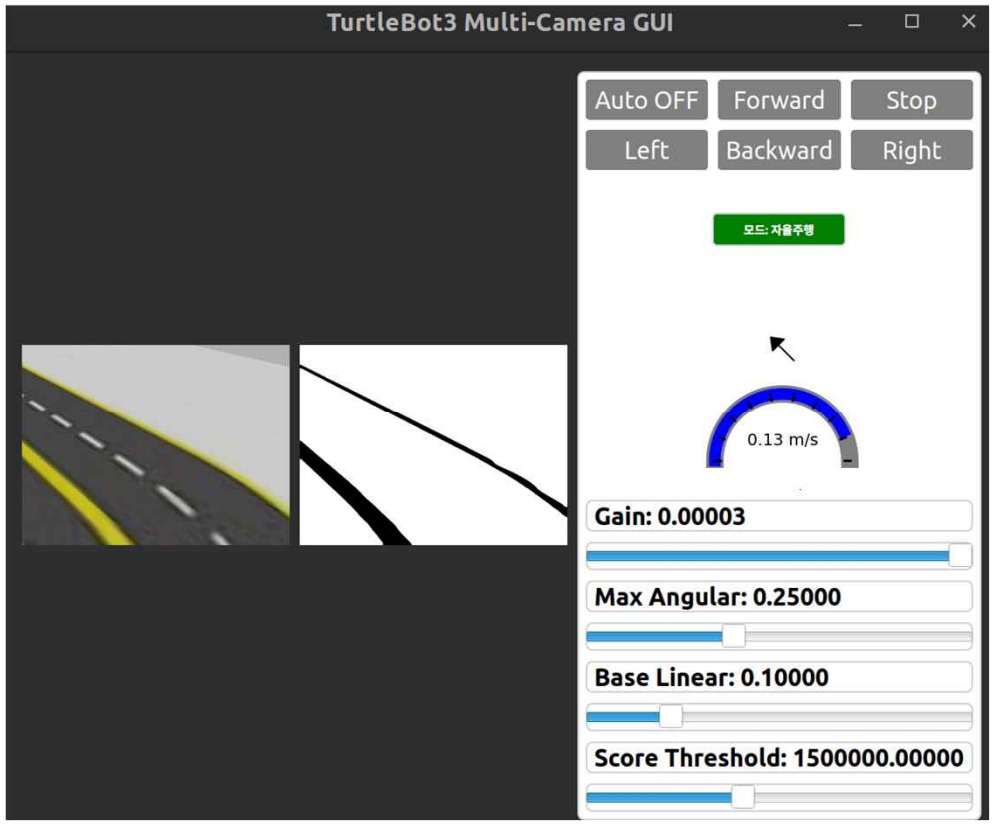


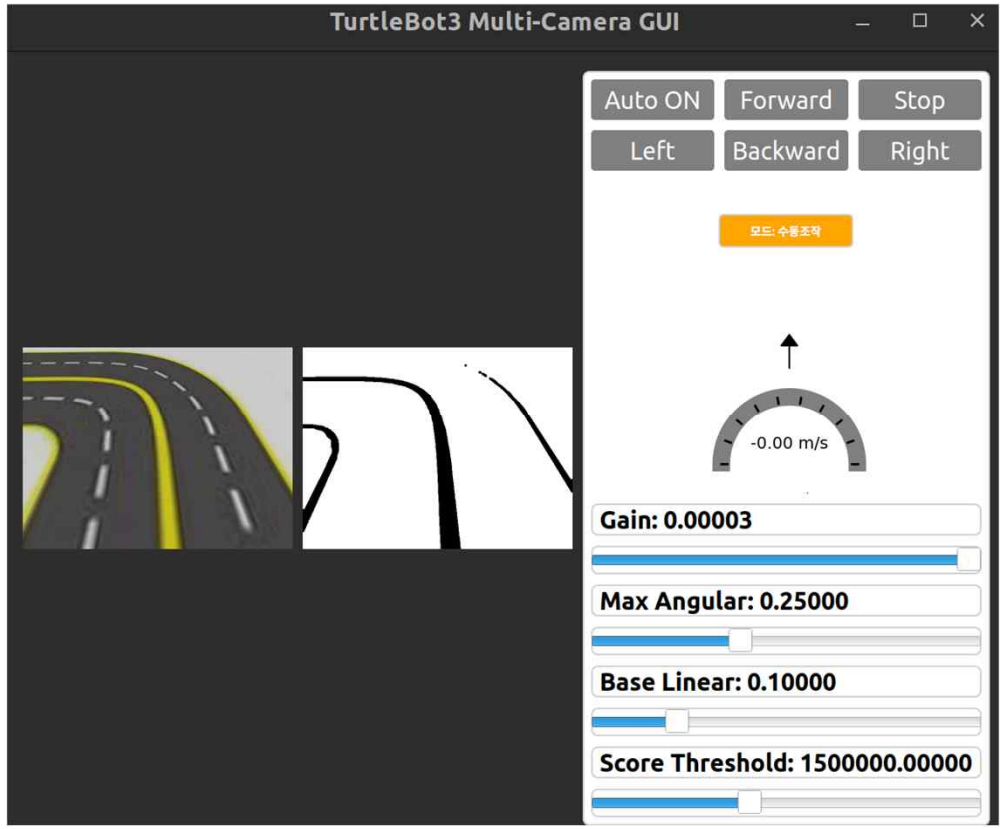
배속 아닙니다.

K-Digital Training

04 프로젝트 수행 경과

GUI

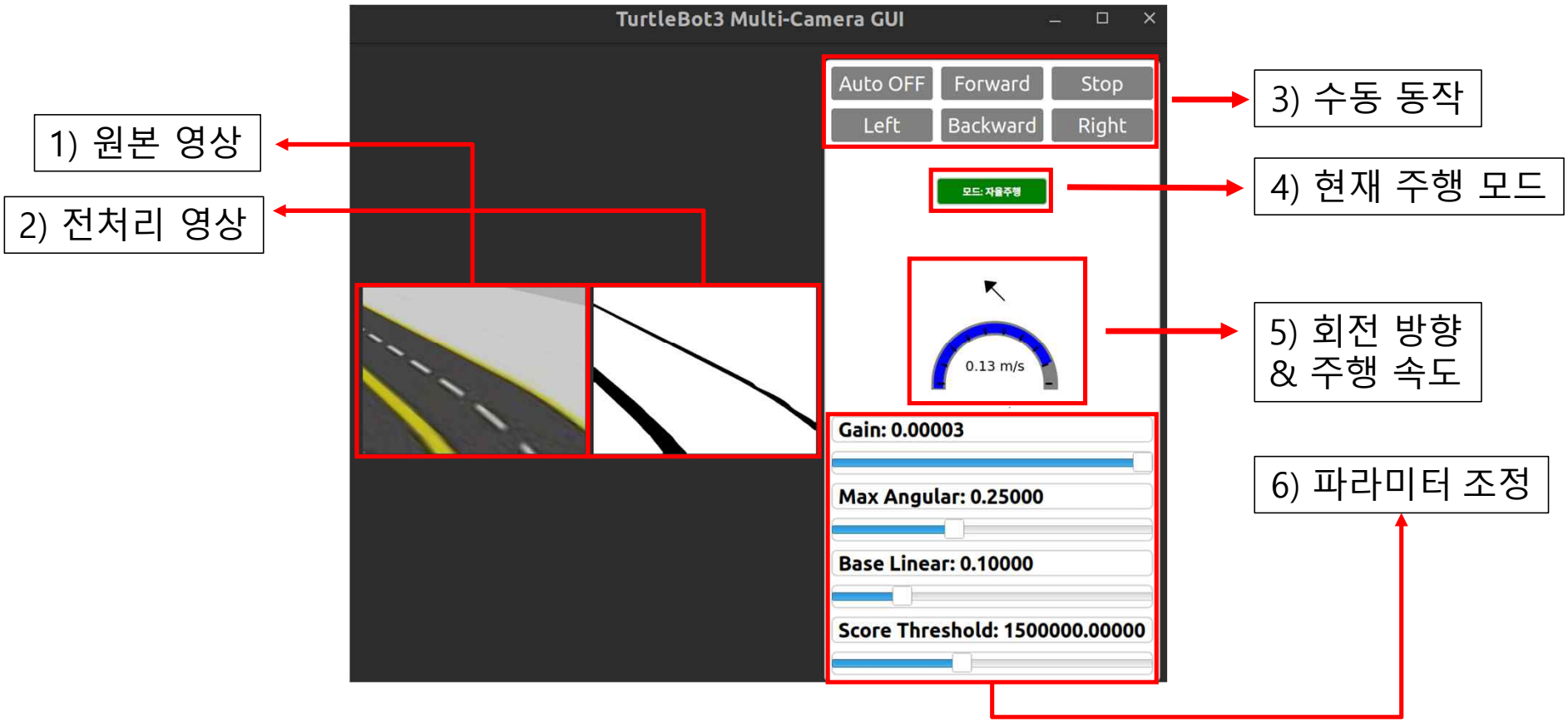




K-Digital Training

04 프로젝트 수행 경과

GUI



K-Digital Training

05 자체 평가 의견

완성도 평가

7.8 / 10

개선점

라이다 센서를 활용한 장애물 탐지 기능 구현

Aruco marker를 활용한 manipulator 기능 구현

개인 또는 우리 팀이 잘한 부분과 아쉬운 점

Lane detect 알고리즘을 바탕으로
빠르고 안정적인 주행 구현

터틀봇 기기, 카메라 위치, 매니퓰레이터 등의 이슈로
Pick & Place 구현 미흡

느낀 점이나 경험한 성과

기존 **가상환경**에서 진행하던 실습을 **실제 환경**으로
확장하여 실제의 다양한 **변수**와 **오차**에 대해 직접
체감할 수 있었음.