



한양대학교
HANYANG UNIVERSITY

Embedded System Project 2

미래자동차공학과 김나영 2022075478

미래자동차공학과 이호연 2020044466

미래자동차공학과 김경호 2020058513

Department of Automotive Engineering
Hanyang University

목차

1. 프로젝트 목표
2. 전체 흐름 요약
3. 주요 기술
4. 겪은 어려운 점

프로젝트 목표 – ADAS : ACC

- Project 1에서 아쉬웠던 점

“정확한 거리 인식의 한계”

bounding box의 세로 길이만을
기준으로 대략적으로 판단

- Project 1에서 심화 (확장, 개선)

정확한 거리 계산 응용 기술

→ ACC (Adaptive Cruise Control)



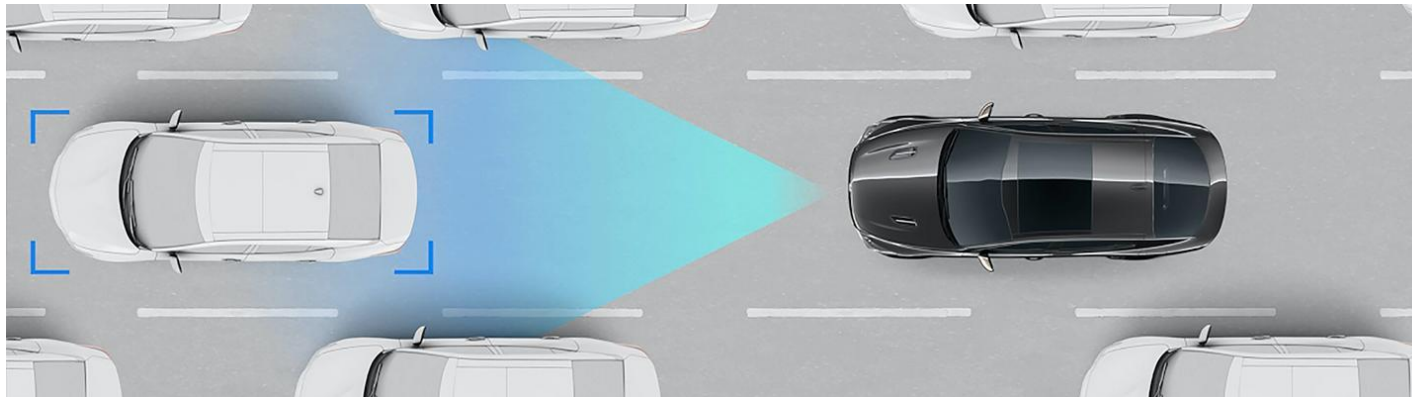
프로젝트 목표 – ADAS : ACC

- ADAS(Advanced Driver-Assistance System) system

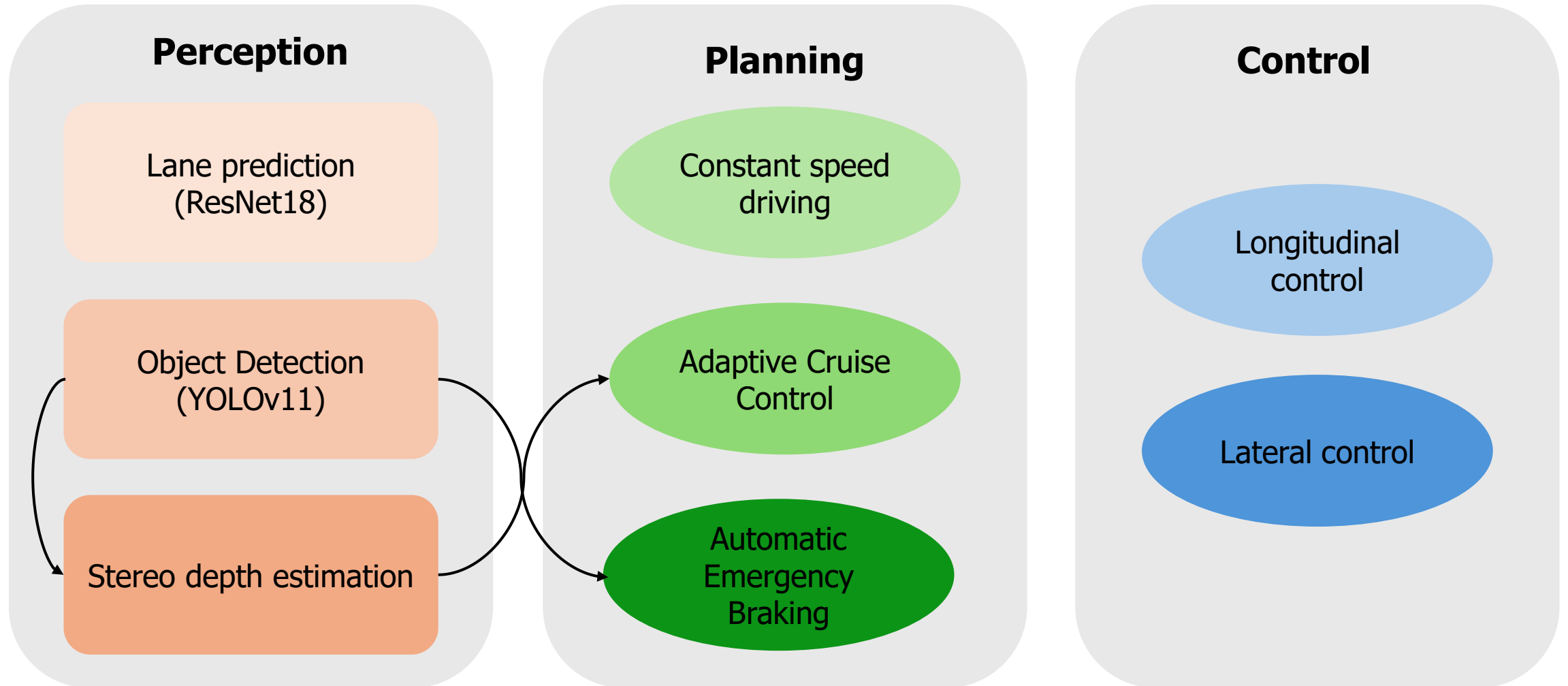
- ◆ Adaptive Cruise Control (ACC)

- ◆ 앞 차와의 거리를 유지하면서 자동으로 속도 조절

- 사용자가 설정한 속도로 정속 주행이 가능하도록 제어
 - 전방 차량이 감지될 경우, TTC(Time to collision)를 유지하며 감속
 - 전방 차량이 사라지면 다시 설정 속도로 회복



전체 흐름 요약



주요 기술 – Perception

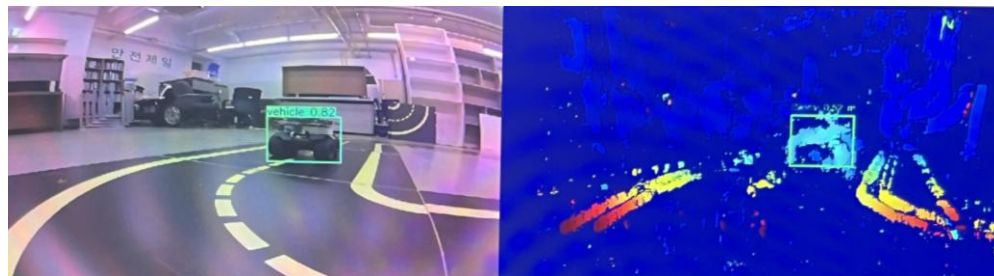
● Stereo depth estimation

◆ Camera Calibration

- OpenCV의 stereoCalibrate() 사용
- 두 카메라의 Intrinsic/Extrinsic matrix -> 카메라 정렬 보정

◆ Depth estimation

- OpenCV의 StereoBM (Block Matching) 알고리즘 사용
- Disparity로부터 Depth 계산 $Depth = \frac{f_x \times B}{disparity}$, B: baseline
 - YOLO detection bbox에 해당하는 disparity에서 계산



● Lane prediction

◆ Alexnet → ResNet18 모델 사용

- 일반화 성능이 높고 추론 속도 빠름

모델	Test MSE	FLOPs	파라미터 수	추론 시간
Alexnet	0.0104401643	7.27 GMAC	57.01 M	0.01907 s
ResNet18	0.001960151	18.95 GMAC	11.18 M	0.02064 s
ResNet50	0.0021892965	42.92 GMAC	23.51 M	0.03133 s

겪은 어려운 점 – Perception

- **Disparity Map의 잡음이 심함**

- ◆ block window의 크기: 5 → 17
- ◆ bounding box에 해당하는 disparity 중 outlier data 샘플링
 - NaN 값 제거, bounding box의 중앙 부분만 평균 계산

- **Stereo depth estimation 연산 시간**

- ◆ OpenCV CUDA 활성화 : 0.09 s → 0.06 s
- ◆ 5 frame에 한 번 실행되도록



```
jtop MAXN SUPER|CPU 7.4%|GPU 0.0%
jtop 4.3.2 - (c) 2024, Raffaello Bonghi [raffaello@rnext.it]
Website: https://rnext.it/jetson_stats

Platform
Machine: aarch64
System: Linux
Distribution: Ubuntu 22.04 Jammy Jellyfish
Release: 5.15.148-tegra
Python: 3.10.12

Serial Number: [s]XX CLICK TO READ XXX
Hardware
Model: NVIDIA Jetson Orin Nano Engineering Reference Developer Kit Super
699-level Part Number: 699-13767-0005-300 L.1
P-Number: p3767-0005
Module: NVIDIA Jetson Orin Nano (Developer kit)
SoC: tegra234
CUDA Arch BIN: 8.7
L4T: 36.4.3
Jetpack: 6.2

Libraries
CUDA: 12.6.68
cuDNN: 9.3.0.75
TensorRT: 10.3.0.30
VPI: 3.2.4
Vulkan: 1.3.204
OpenCV: 4.12.0-dev with CUDA: YES

Hostname: ubuntu
Interfaces
wlp1p1s0: 192.168.0.47
docker0: 172.17.0.1
```



주요 기술 – Planning & Control

- **Adaptive Cruise Control (ACC)**

- ◆ 앞 차와의 간격을 적절하게 유지하는 시스템
- ◆ Time To collision (TTC) 기반 거리 유지
 - $\text{target distance} = \text{default distance} + 0.3 * \text{current speed}$
- ◆ 커브구간에서는 앞 차량을 지나치게 의식하지 않도록 ROI 설정
- ◆ 신호등이 존재할 때, 신호 위반하는 차량을 계속 따라가지 않도록 함

- **Autonomous Emergency Braking (AEB)**

- ◆ 교차로에서 가로질러오는 차량 발견 시 정지
- ◆ 전방 차량과의 적정 안전거리 유지 실패 시 정지

- **횡방향 PI 제어**

- ◆ waypoint와 이미지의 중심선과의 error → error가 0이 되도록 PI제어 활용
- ◆ Throttle 값에 steering 비율을 반영하여 L, R 모터에 다른 출력으로 전달

- **종방향 PI 제어**

- ◆ target distance와 추정된 depth의 error → error 0이 되도록 PI 제어 활용

겪은 어려운 점 – Planning & Control

● 극저속 상황에서 제어의 불안정 극복

◆ 극저속 주행 시 throttle을 출력하여도 차량이 움직이지 않는 현상 발생

- I항을 통해 이를 극복하려 하였으나 한계 존재

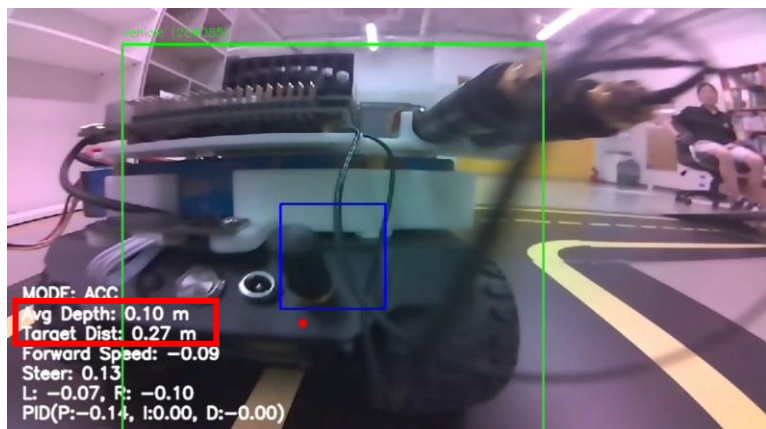
➔ 차가 움직이지 않는 최대 throttle값을 실험적으로 측정, 제어 command 발행 직전 default 값 합산 후 출력

● 근거리에서의 불안정한 stereo depth estimation 결과

◆ 근거리에서의 disparity 불안정한 issue → estimation error

◆ 근거리 상황 판단 보조 → bbox size 비교를 추가

- 다만, false positive 상황에서도 신뢰성을 확보하기 위해 confidence 기준을 크게 설정



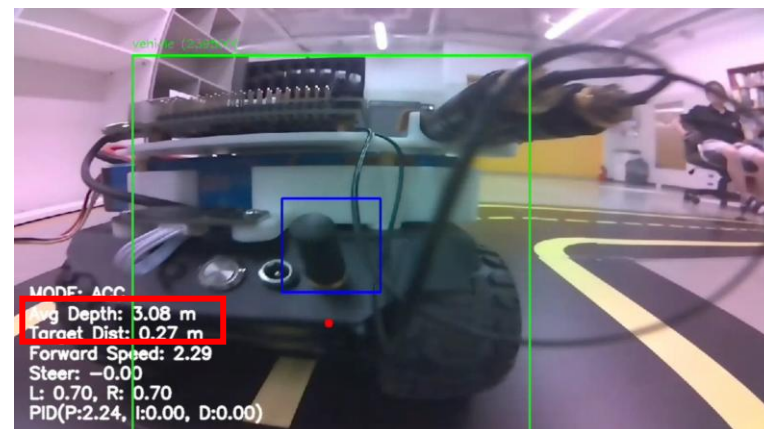
• 좌측 : 0.1 m

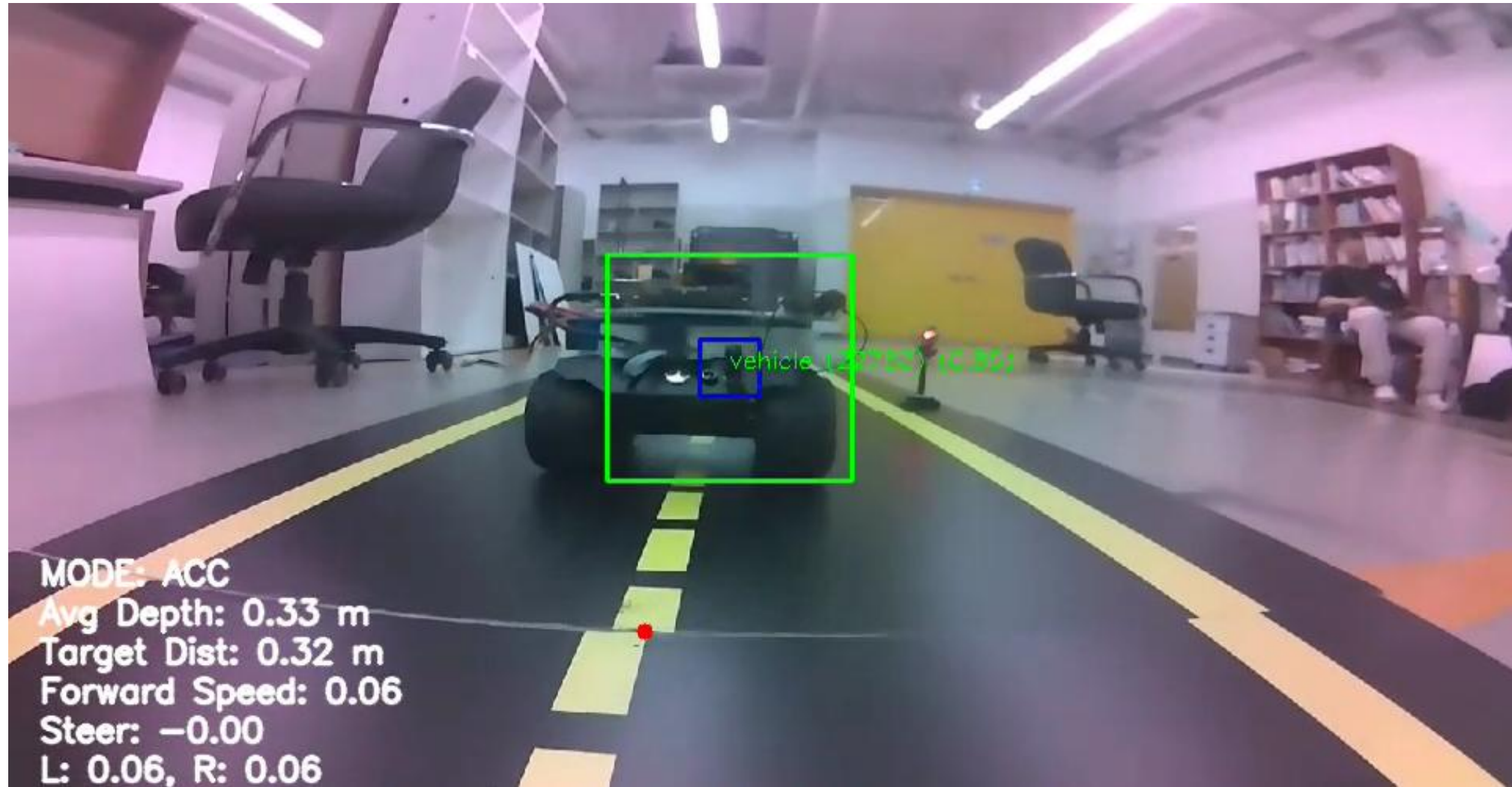


(1 frame 후)



• 우측 : 3.08m





A photograph of a modern building with a blue glass facade and a white curved section at the bottom. The building is the background of the slide. The text "Thank you!" is overlaid in the center.

Thank you!

Chang Ming Ren
Automation Research Center