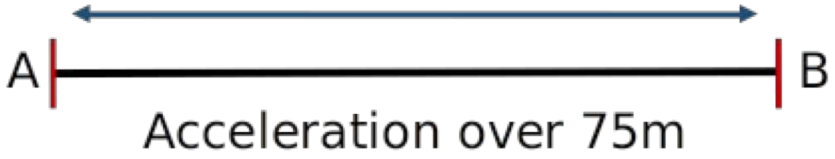
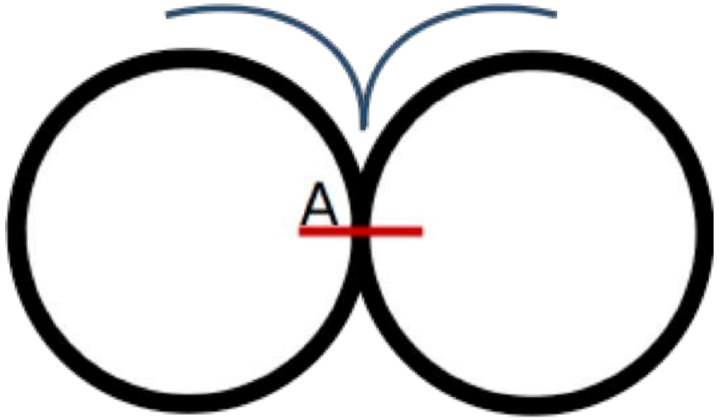
Creating MIT's First Student-led Driverless Car

July 9th, 2019 | Sibo Zhu, Kathleen Brandes, Kieran Strobel, Luke Kulik

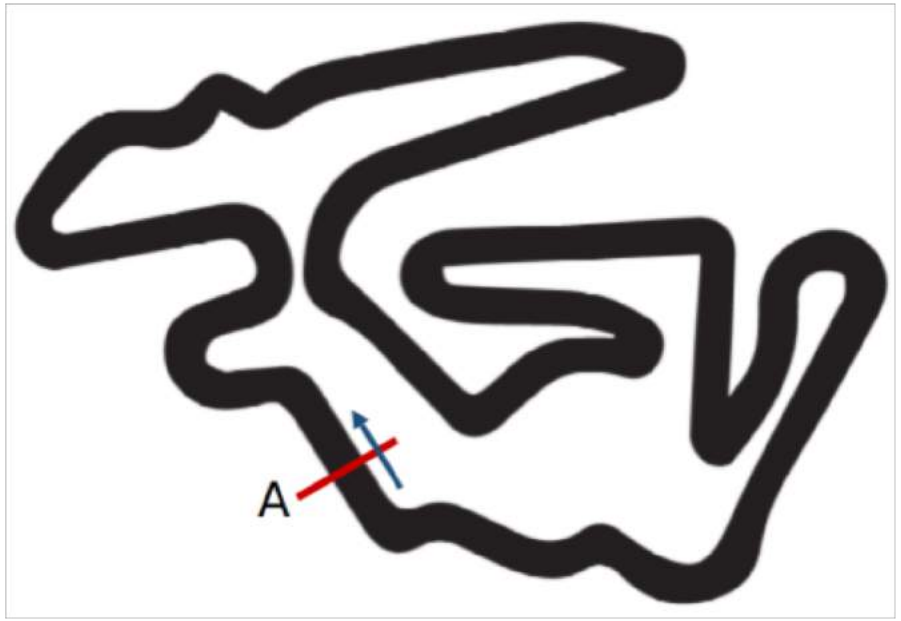

FORMULA STUDENT GERMANY
AN INTERNATIONAL DESIGN COMPETITION OF SKILLS, SPEED AND SPIRIT

Formula Student Competition Disciplines



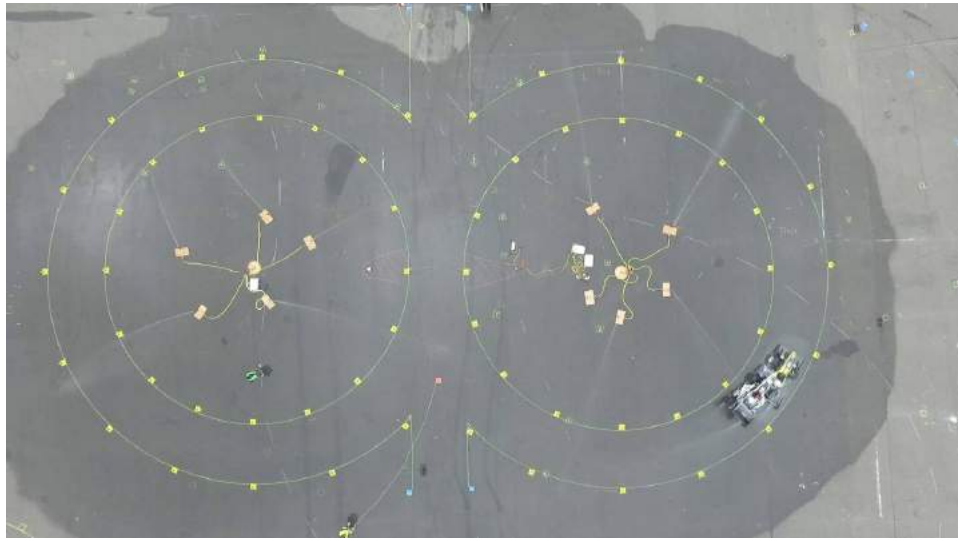


SkidPad



Autocross and Endurance

Formula Student Competition Disciplines



Design of an Autonomous Racecar: Perception, State Estimation and System Integration

Miguel I. Valls*, Hubertus F.C. Hendrikx*, Victor J.F. Reijgwart*, Fabio V. Meier*
Inkyu Sa, Renaud Dubé, Abel Gawel, Mathias Bürki, and Roland Siegwart

Abstract— This paper introduces *fläela driverless*: the first autonomous racecar to win a Formula Student Driverless competition. In this competition, among other challenges, an autonomous racecar is tasked to complete 10 laps of a previously unknown racetrack as fast as possible and using only onboard sensing and computing. The key components of *fläela*'s design are its modular redundant sub-systems that allow robust performance despite challenging perceptual conditions or partial system failures. The paper presents the integration of key components of our autonomous racecar, i.e., system design, EKF–based state estimation, LiDAR–based perception, and particle filter-based SLAM. We perform an extensive experimental evaluation on real–world data, demonstrating the system's effectiveness by outperforming the next-best ranking team by almost half the time required to finish a lap. The autonomous racecar reaches lateral and longitudinal accelerations comparable to those achieved by experienced human drivers.

I. INTRODUCTION

On August 13th 2017, *fläela driverless* became the first car to ever win the Formula Student Driverless (FSD) competition. The competition requires the car to race fully autonomously and consists of 4 dynamic and 4 static disciplines [1]. The dynamic disciplines test the system's reliability under general race conditions and at high lateral and longitudinal speeds. The static disciplines evaluate the system's design under aspects of software, hardware, costs, and business. While *fläela driverless* performed well in all categories, we this paper focuses on software and hardware designs.

The hardware platform for the project is *fläela*, an electric 4WD car with a full aerodynamic package, high wheel torque, and a lightweight design developed by AMZ[2] for Formula Student Electric 2015. The sensor outfit for autonomous operation and the software system are developed from scratch.

In our autonomous design, system reliability under high performance operation is chosen as the main design goals, since the FSD regulations allow no human intervention.

This paper presents the state estimation, LiDAR SLAM, and localization systems that were integrated in *fläela*. The autonomous system perceives its surroundings using a 3D LiDAR and a self-developed visual-inertial stereo camera system. Furthermore, a velocity sensor and an Inertial Navigation System (INS) combining an IMU and a GPS were added for state estimation. All the information is processed online by two computing units running Robot Operating System (ROS) and a real-time capable Electronic Control Unit (ECU). Figure 2 shows an overview of the hardware-software setup.

In order to reach *fläela*'s full potential when racing autonomously, the track must be known at least 2s ahead. At high speeds, this requires a perception horizon that exceeds the sensors' range. The car must thus drive carefully to discover and map the track. This mode will later be referred to as *SLAM Mode*. Once the map is known, the car can drive in *Localization Mode* which can exploit the advantage of planning on the previously mapped race-track.

The contributions of this paper are:

- A complete pipeline from perception to state estimation with on-board sensors and computation only, capable of driving an autonomous racecar close to a human driver's performance.
- Extensive experimental evaluation and demonstration in real-world racing scenarios.

The remainder of this paper is structured as follows. Section II introduces state-of-the-art work on autonomous racing, Section III describes the theoretical development for this project and Section IV the implementation details. We present our experimental results in Section V, and conclude in Section VI.

II. RELATED WORK/BACKGROUND

Autonomous racing is an emerging field within autonomous driving. In the last years, a few self-racing vehicles have been developed, both in academic and in the industrial research. The first known autonomous vehicle competition

* The authors contributed equally to this work.
Authors are with the Autonomous Systems Lab, ETH Zurich, Zurich, Switzerland. dnique@ethz.ch

[2] http://www.amzracing.ch/

Best Student Paper Award at ICRA 2018

Partnership with Delft University of Technology (DUT)

Two top technology programs, collaborating across the Atlantic

Waymo shop visit – February 16, 2019

1. Sensors (“Perceive”)

VLP32C LiDAR

Stereo + Mono cameras

Tactical-grade IMU

Dual-antenna GPS

Kistler ground speed sensor

2. Compute (“Plan”)

Intel i7 CPU

Nvidia GPU

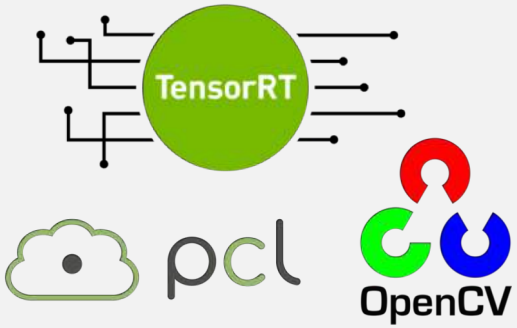
Stereo-matching FPGA

3. Actuation (“Execute”)

- Steering actuator
- Fail safe emergency brake
- 4WD motor controller interface

Perception

- Interpret the environment using redundant Vision and LiDAR systems



State estimation

- Estimate the state of the vehicle and its environment

CERES SOLVER

ROVIO

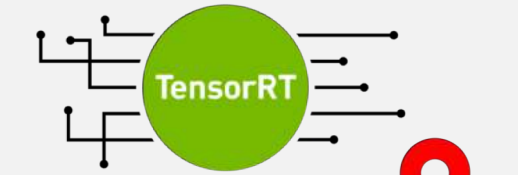
Planning & Controls

- Generate feasible driving region and issue optimal control commands

CVXGEN

Perception

- **Interpret the environment using redundant Vision and LiDAR systems**



State estimation

- Estimate the state of the vehicle and its environment

CERES SOLVER

ROVIO

Planning & Controls

- Generate feasible driving region and issue optimal control commands

CVXGEN

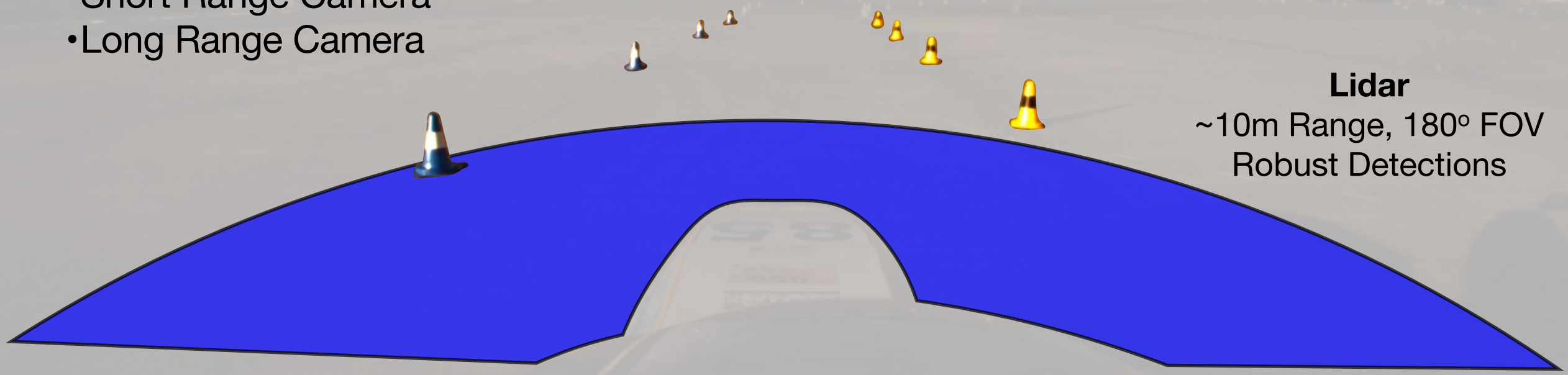
INTEL MATH KERNEL LIBRARY


85

1. Lidar Based Perception
2. Camera Based Perception
    - Short Range Camera
    - Long Range Camera

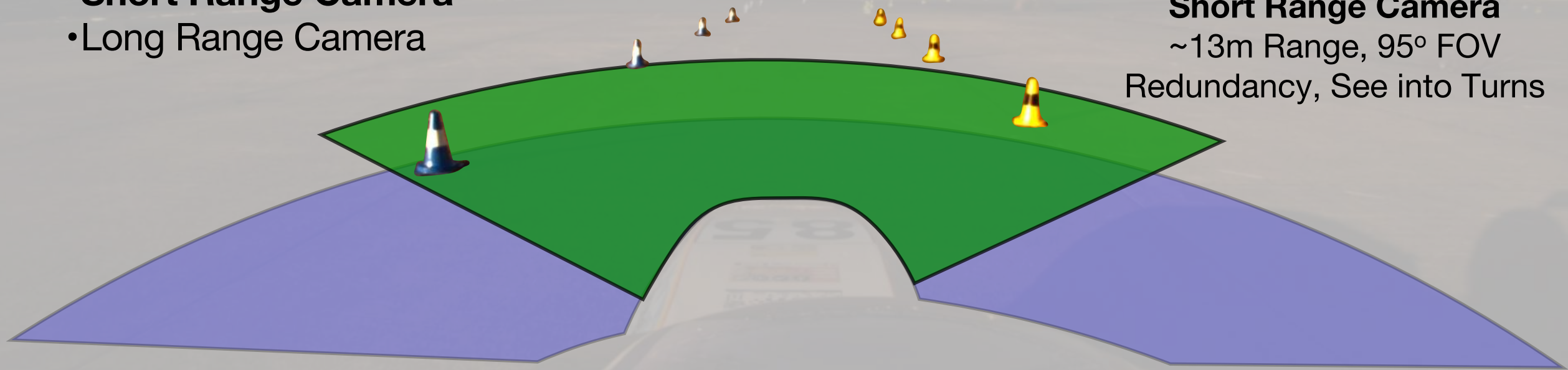
1. Lidar Based Perception

2. Camera Based Perception
   - Short Range Camera
   - Long Range Camera



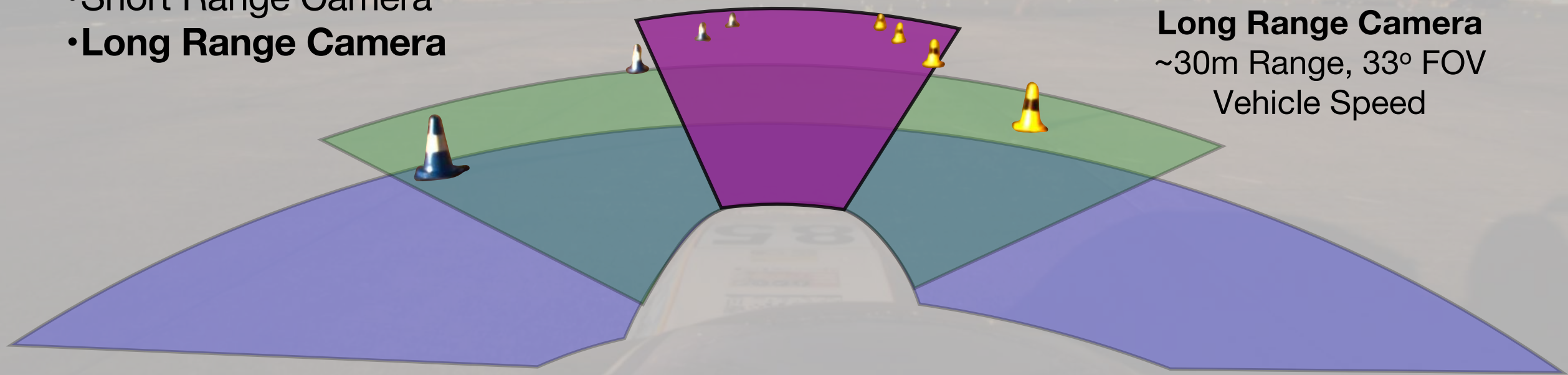
Lidar
~10m Range, 180° FOV
Robust Detections

1. Lidar Based Perception
2. **Camera Based Perception**
   - **Short Range Camera**
   - Long Range Camera

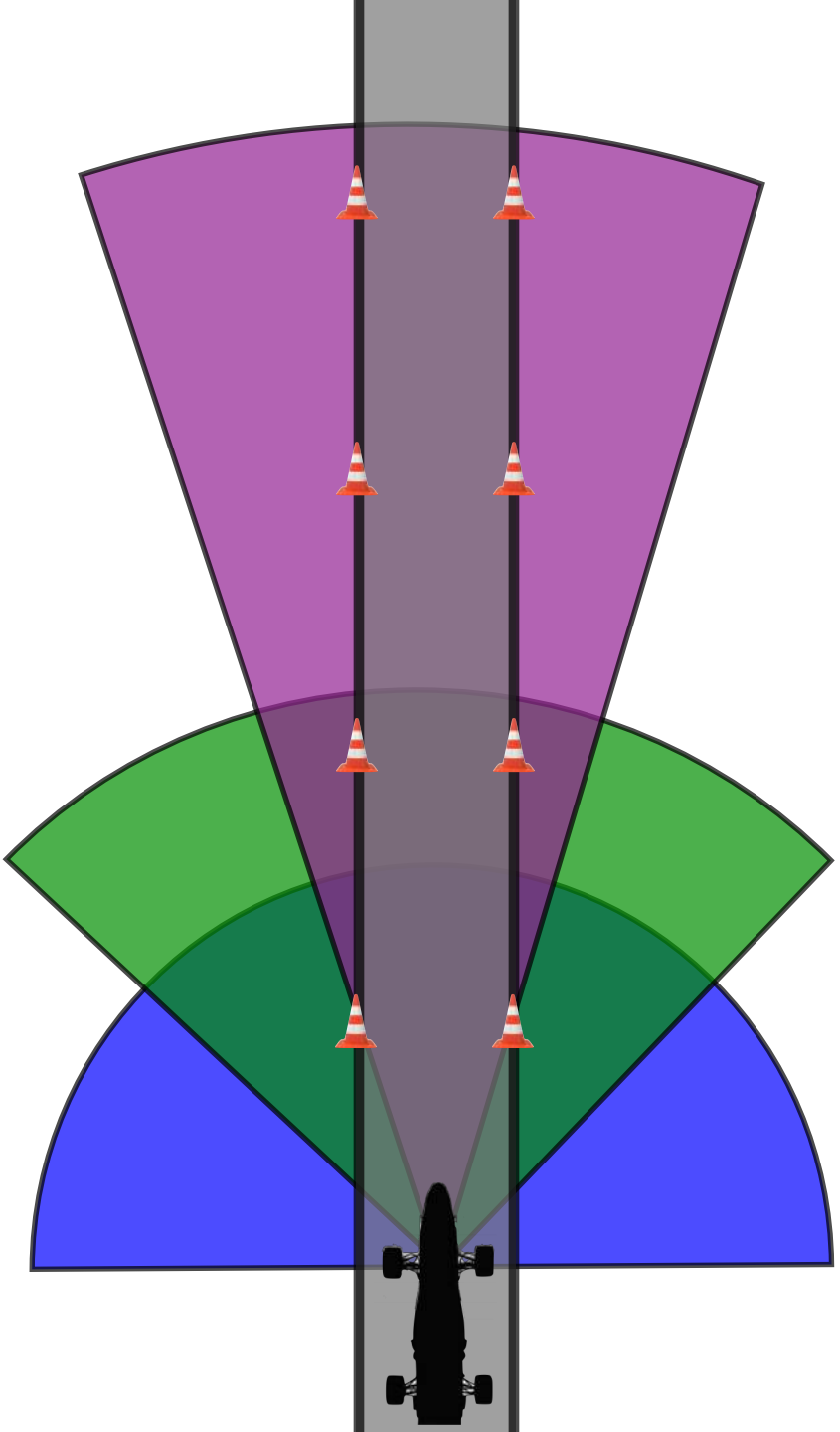


Short Range Camera
~13m Range, 95° FOV
Redundancy, See into Turns

1. Lidar Based Perception
2. **Camera Based Perception**
   - Short Range Camera
   - **Long Range Camera**



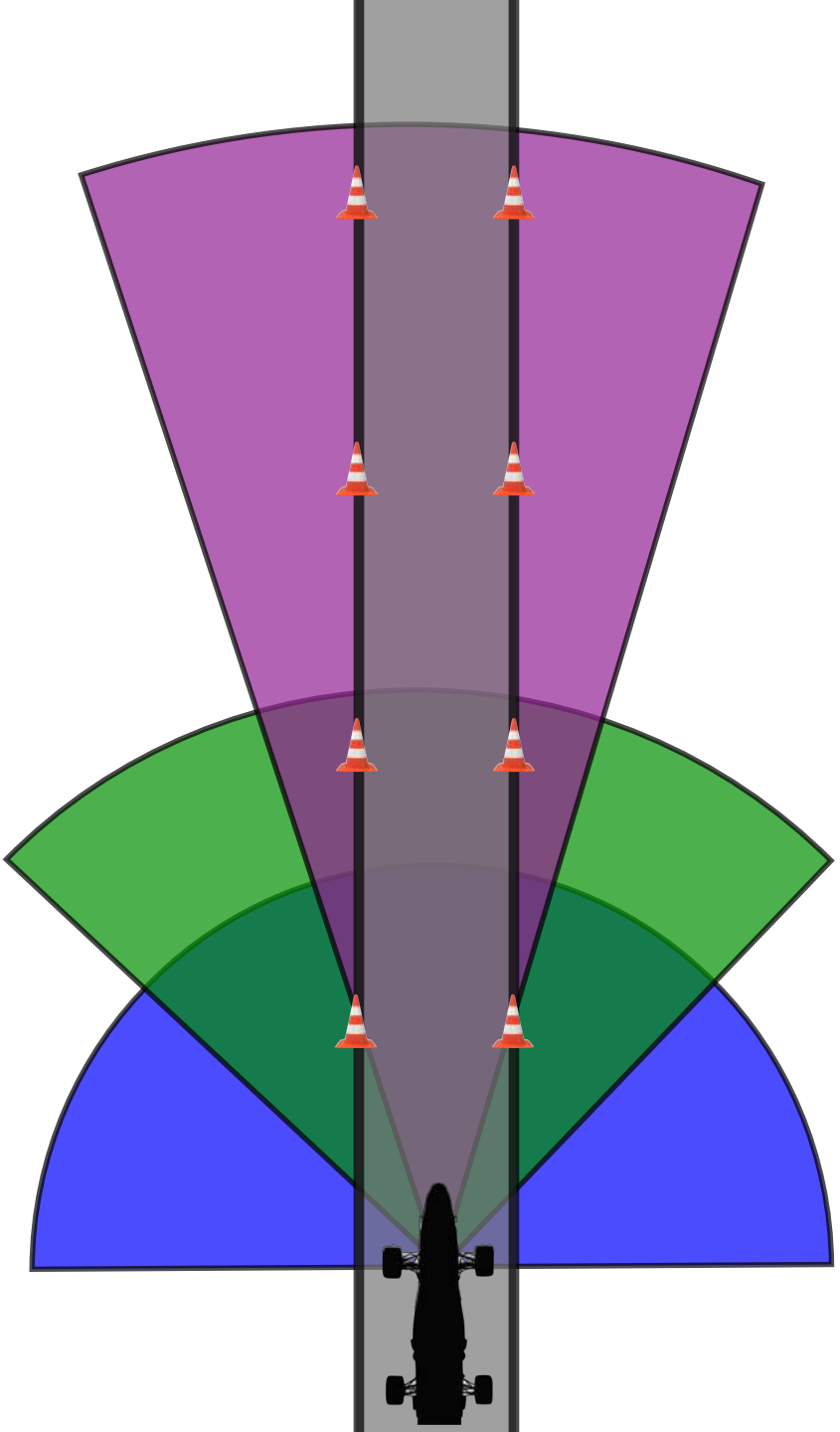
Long Range Camera
~30m Range, 33° FOV
Vehicle Speed



Velodyne VLP32

Pointgrey Grasshopper3

Nerian Stereovision Pair



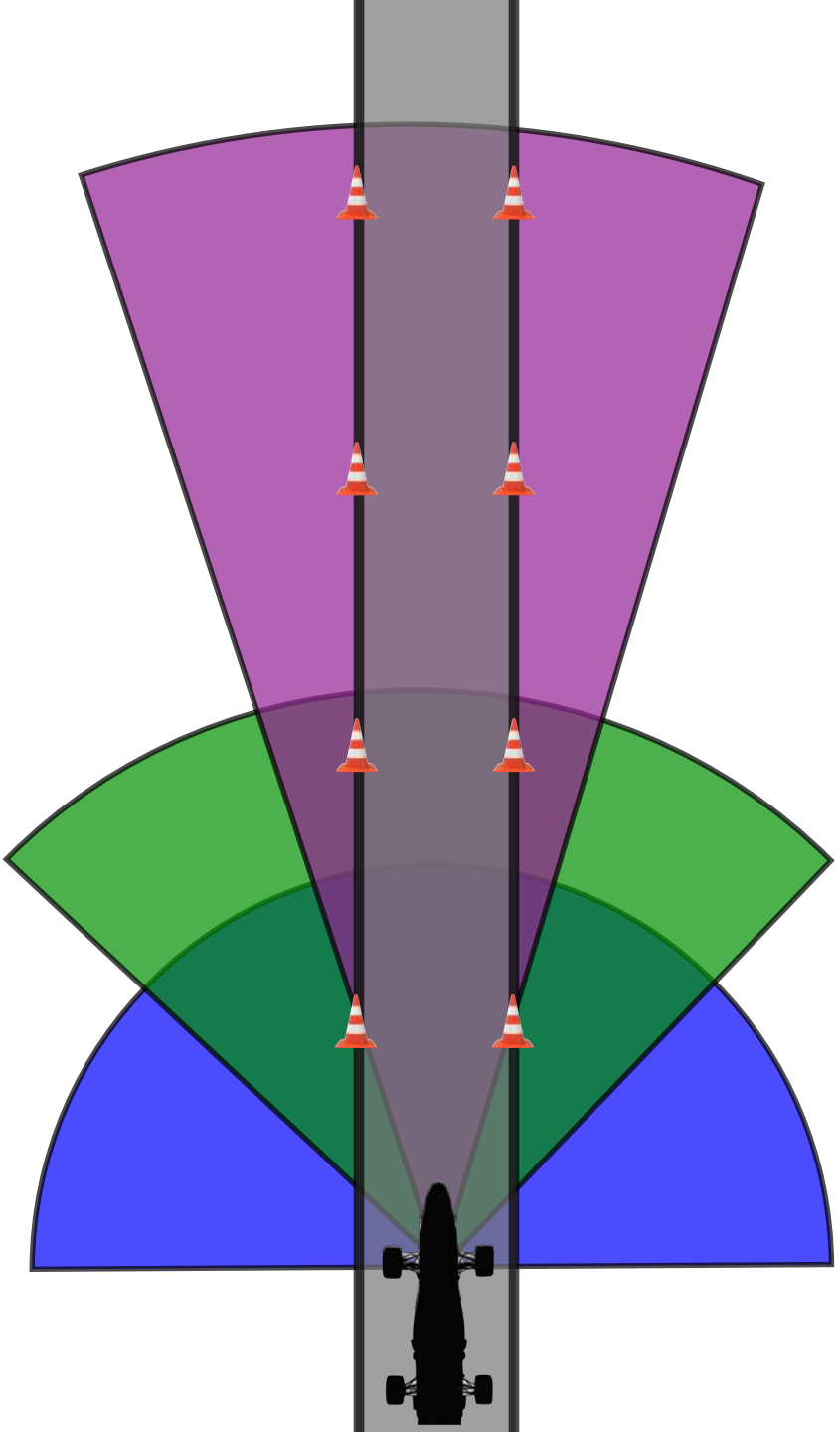
Velodyne VLP32

Pointgrey Grasshopper3

Nerian Stereovision Pair

Ethernet

Main Compute



Velodyne
VLP32

Pointgrey
Grasshopper3

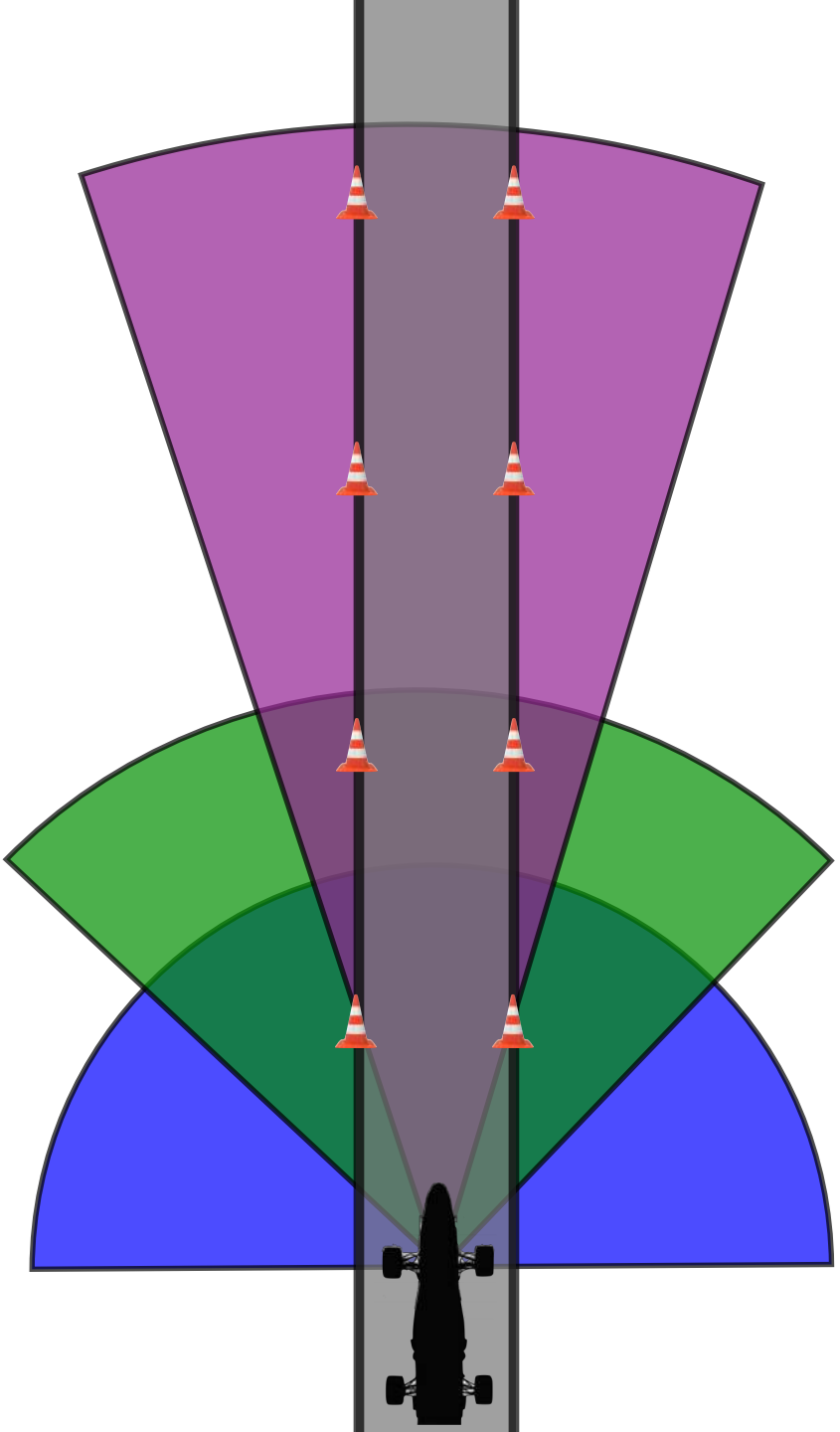
Nerian
Stereovision Pair

Ethernet

USB3

Nvidia Jetson
Xavier
Embedded GPU

Main Compute



Velodyne
VLP32

Pointgrey
Grasshopper3

Nerian
Stereovision Pair

USB3

StereoVision
Processor

Ethernet

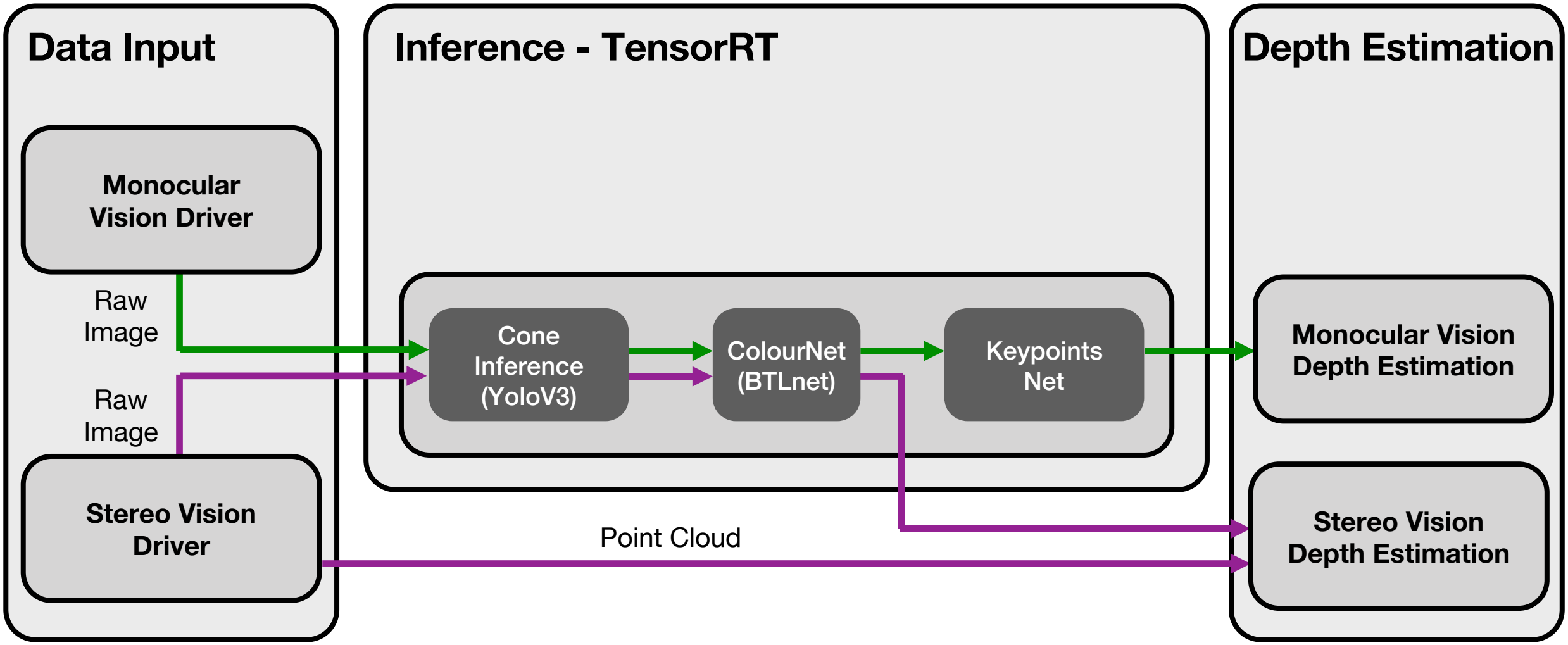
USB3

Ethernet

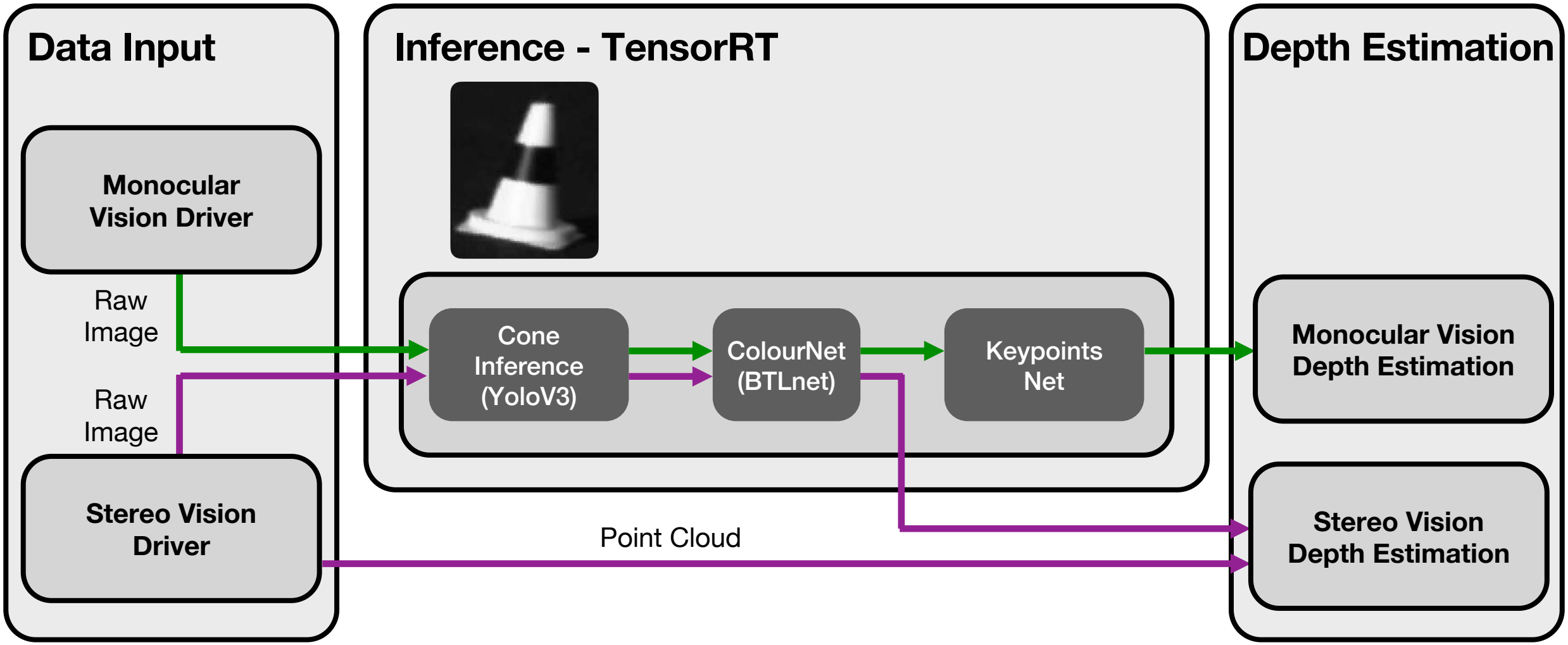
Nvidia Jetson
Xavier
Embedded GPU

Main Compute

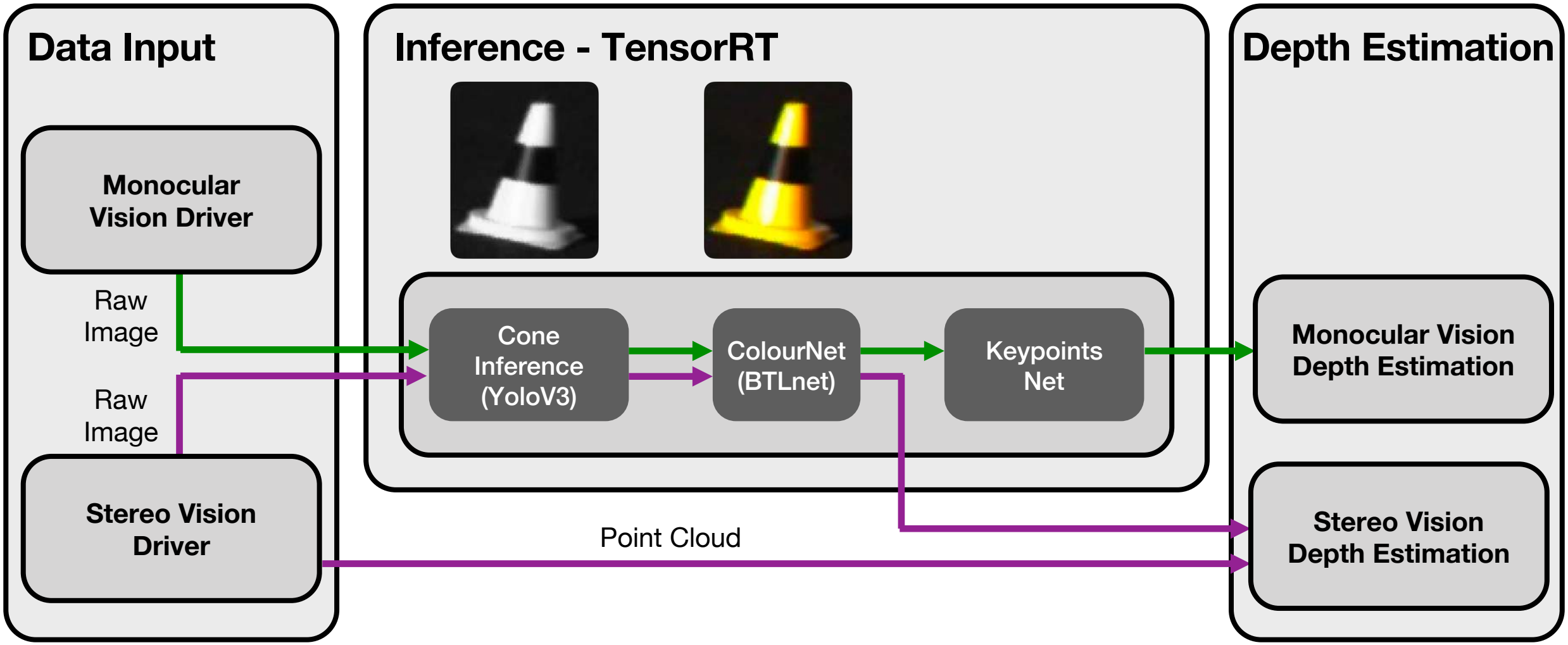
Camera Perception - ROS



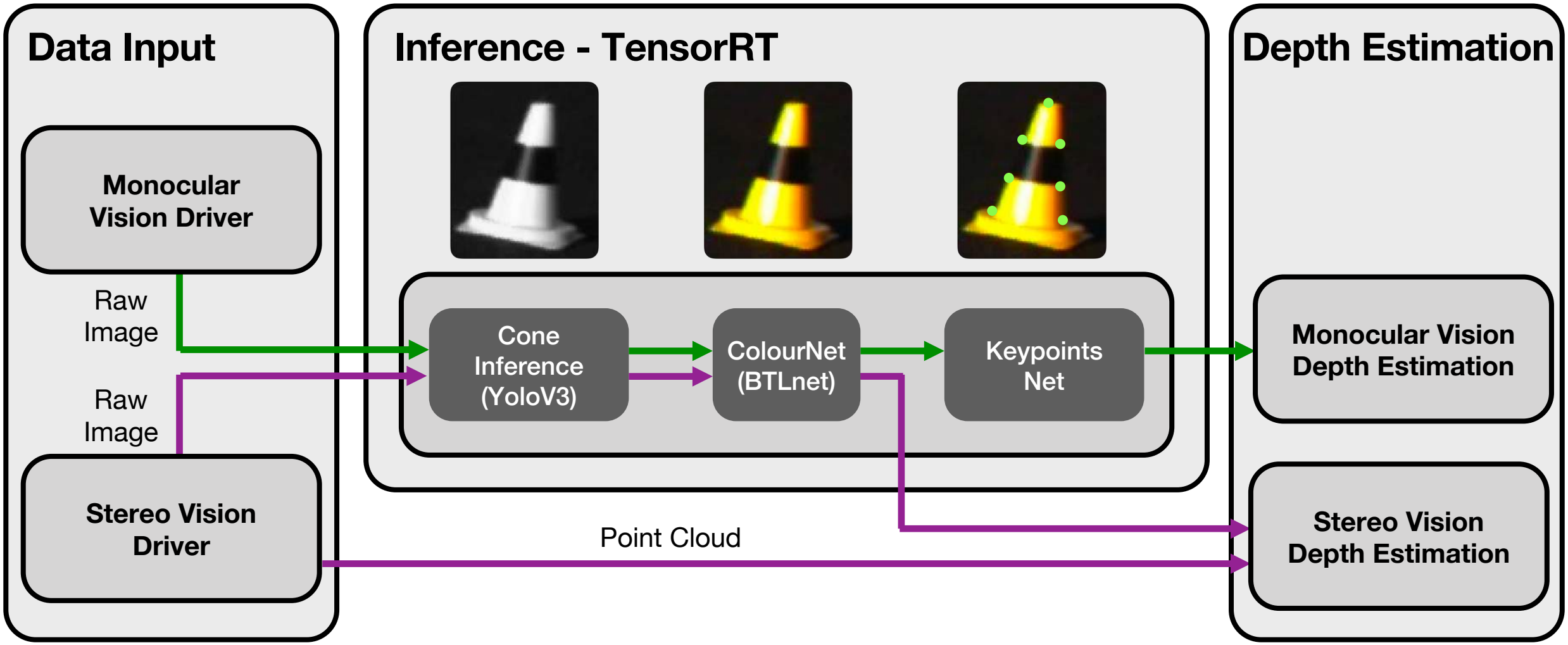
Camera Perception - ROS



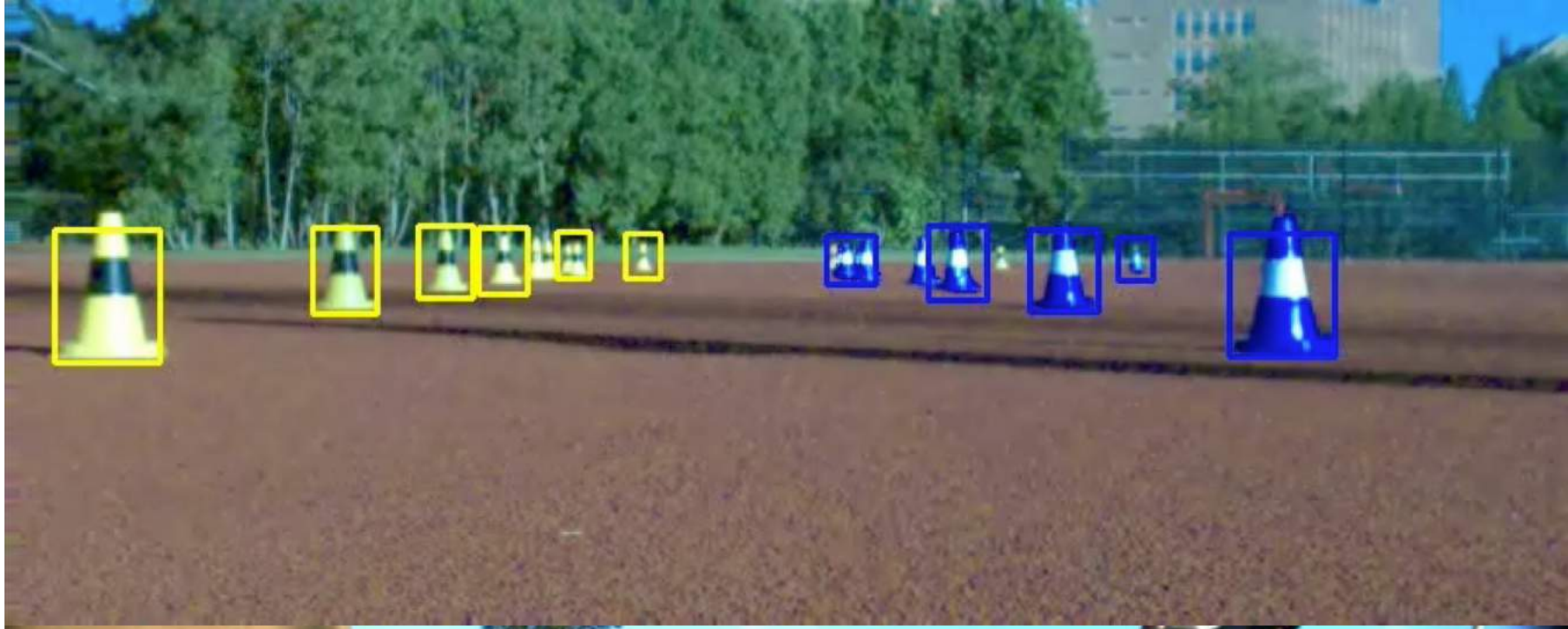
Camera Perception - ROS



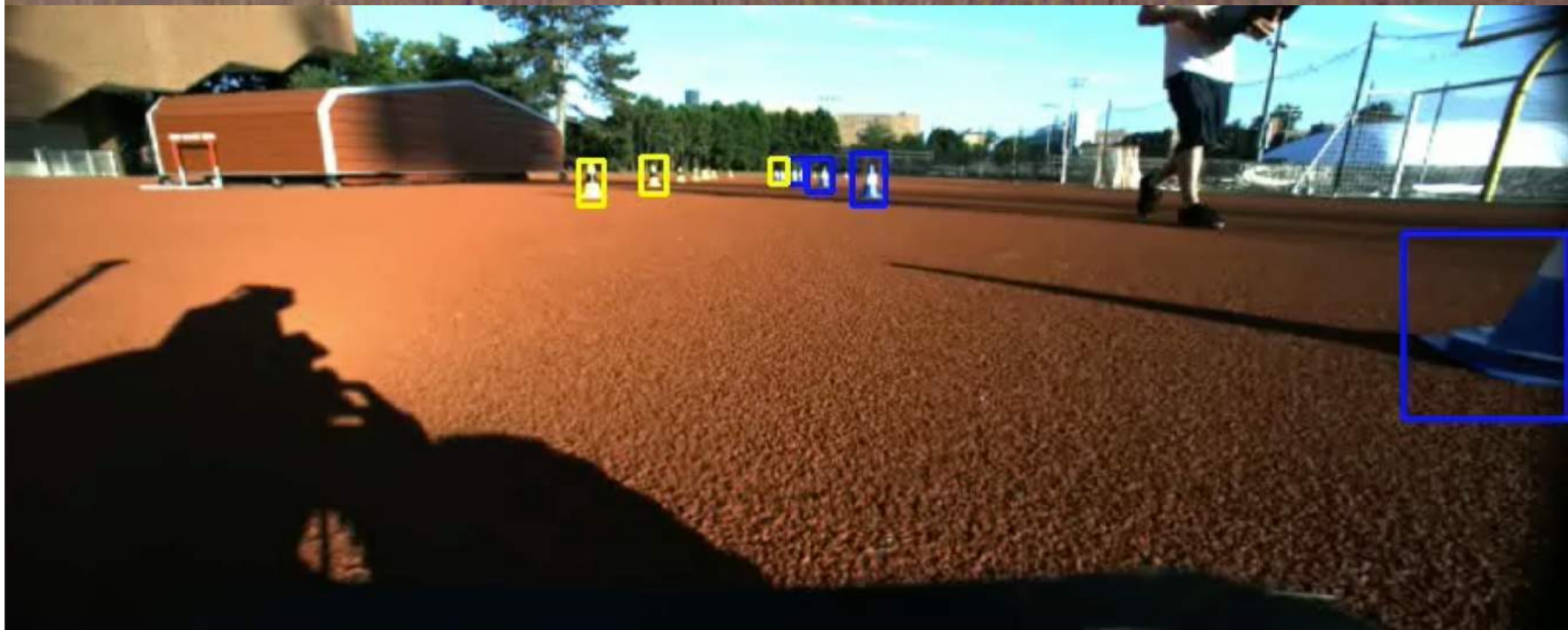
Camera Perception - ROS



Long Range Detections

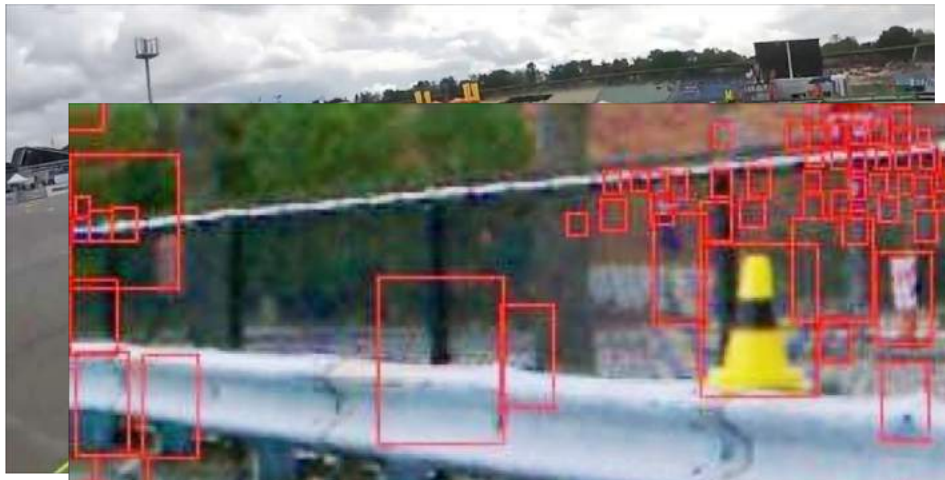


Short Range Detections



Modifications on YoloV3

+ 5% mAP



mAP: 75% → 89%
Recall: 88% → 91%
Precision: 18% → 88%

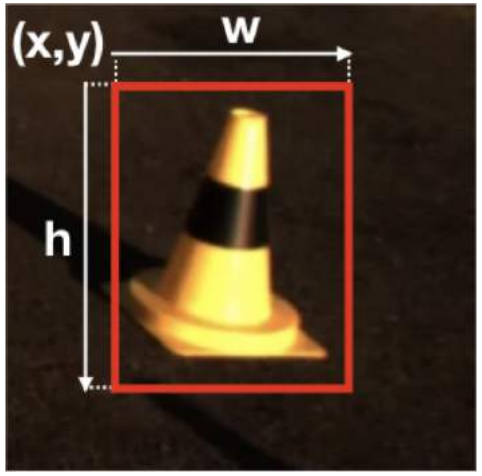
+ 6% mAP

$loss_{conf}$

$loss_w$ $loss_h$

Modifications on YoloV3

+ 3% mAP

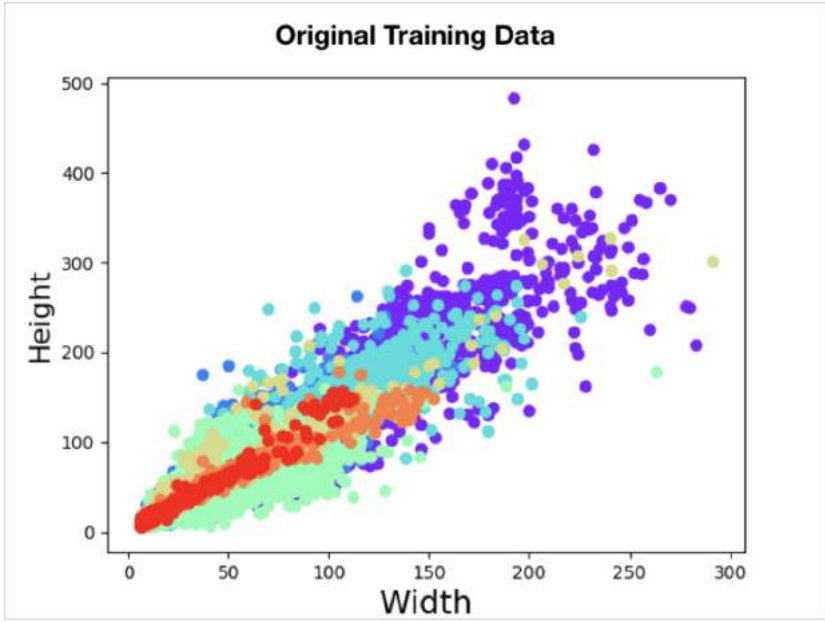


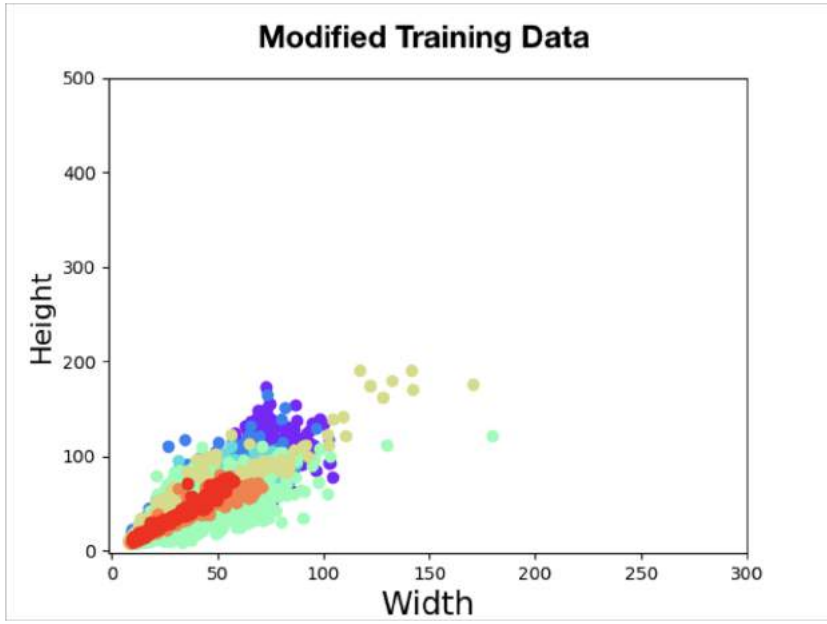
+ 6% mAP

mAP: 75% → 89%
Recall: 88% → 91%
Precision: 18% → 88%

$$loss_x \quad loss_y \quad loss_{conf} \quad loss_w \quad loss_h$$

+ 5% mAP





Perception

- Interpret the environment using redundant Vision and LiDAR systems

State estimation

- **Estimate the state of the vehicle and its environment**

CERES SOLVER

ROVIO

Planning & Controls

- Generate feasible driving region and issue optimal control commands

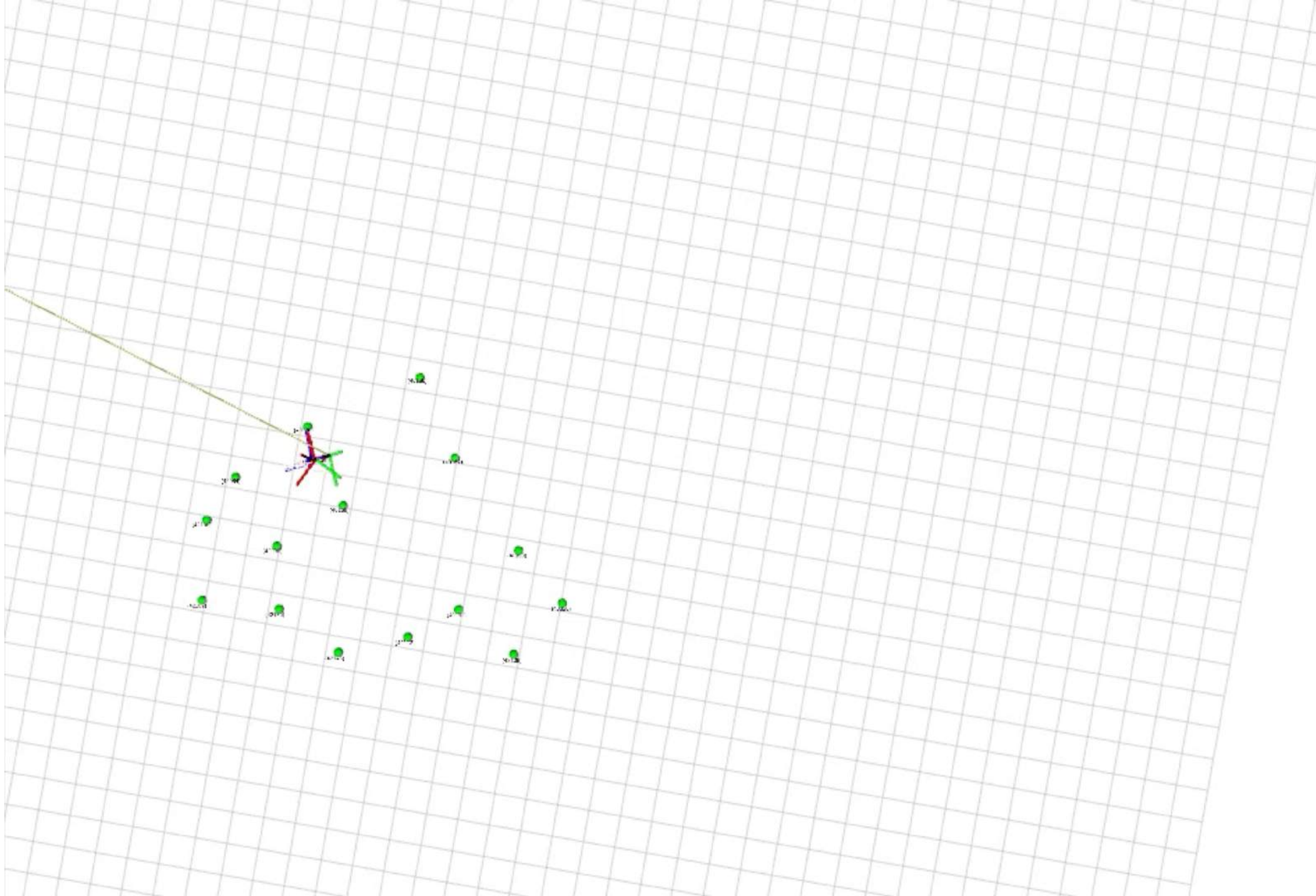
CVXGEN

mosek

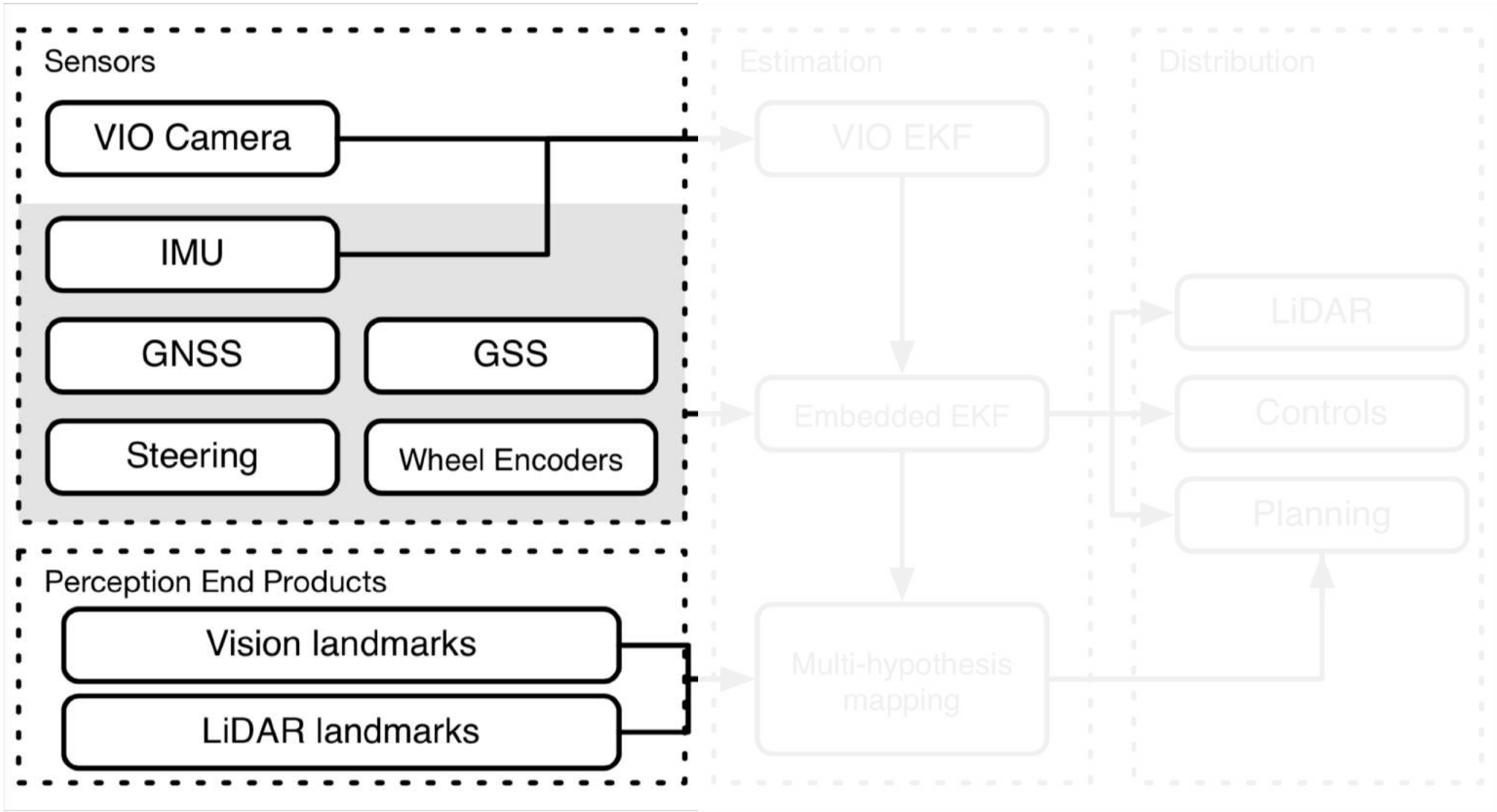
ROS

ubuntu

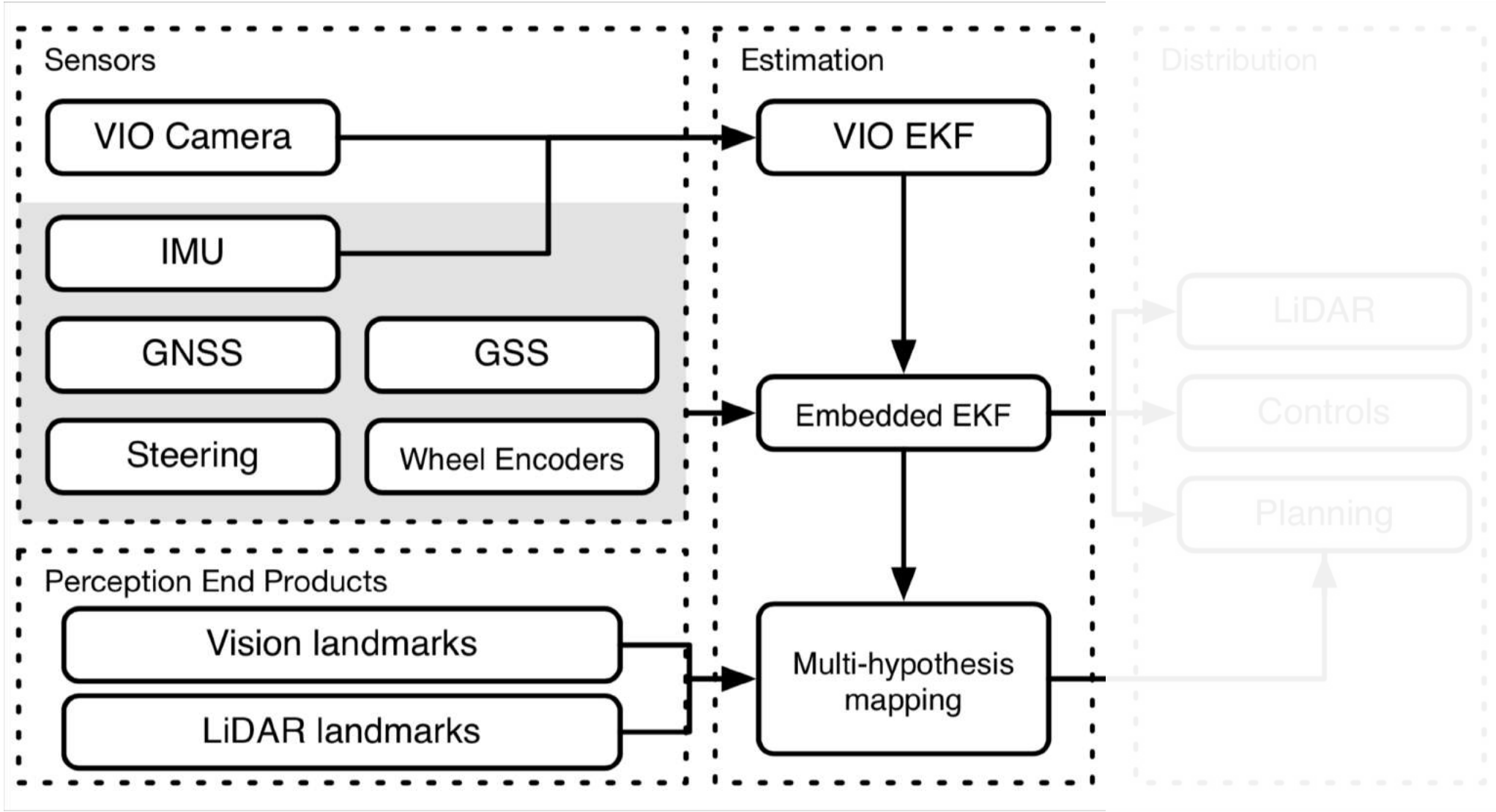
INTEL MATH KERNEL LIBRARY



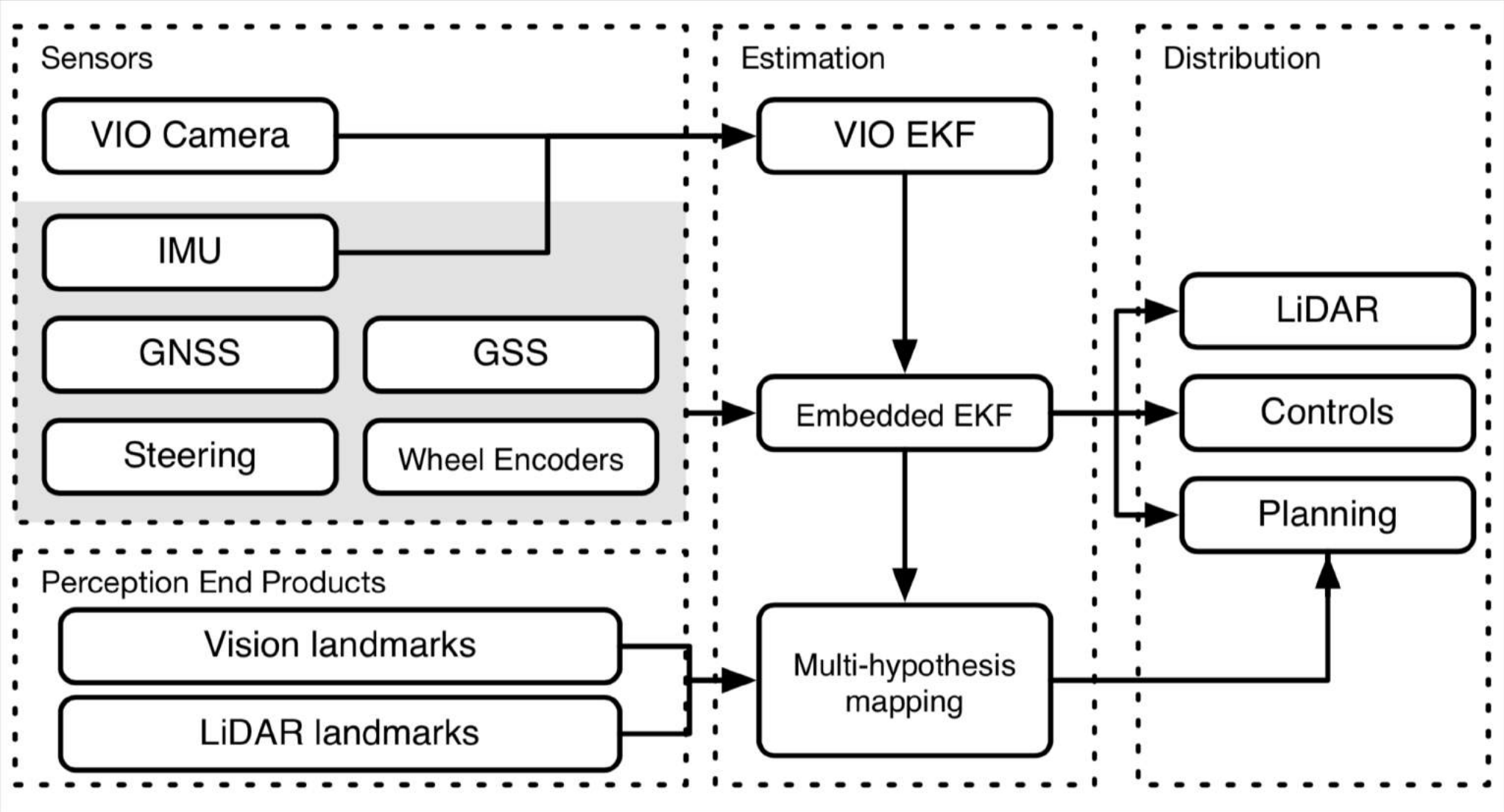
State estimation and mapping software architecture

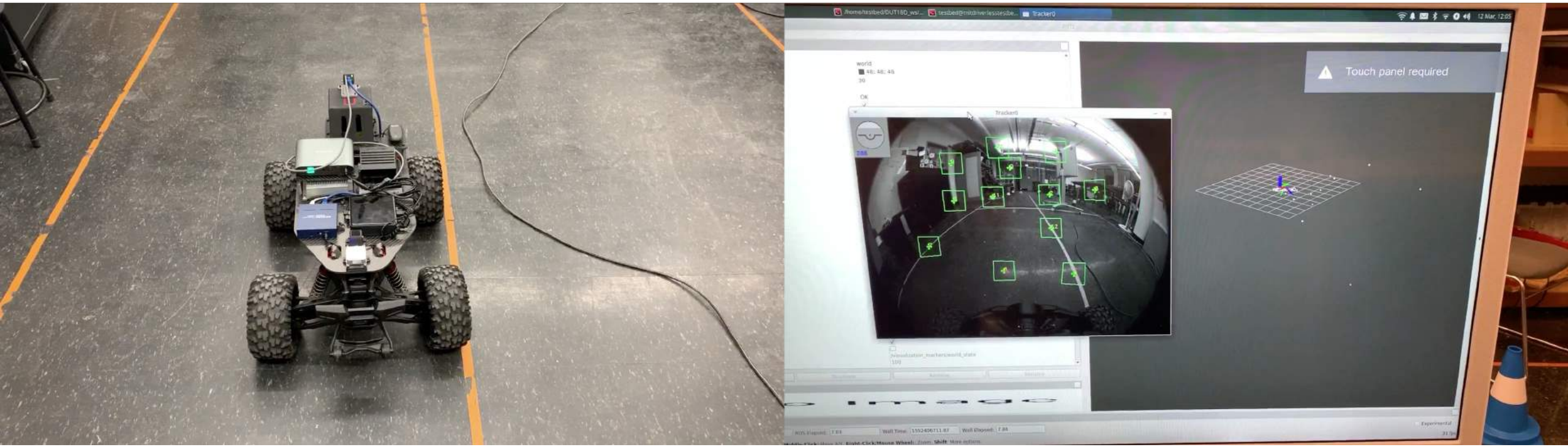


State estimation and mapping software architecture

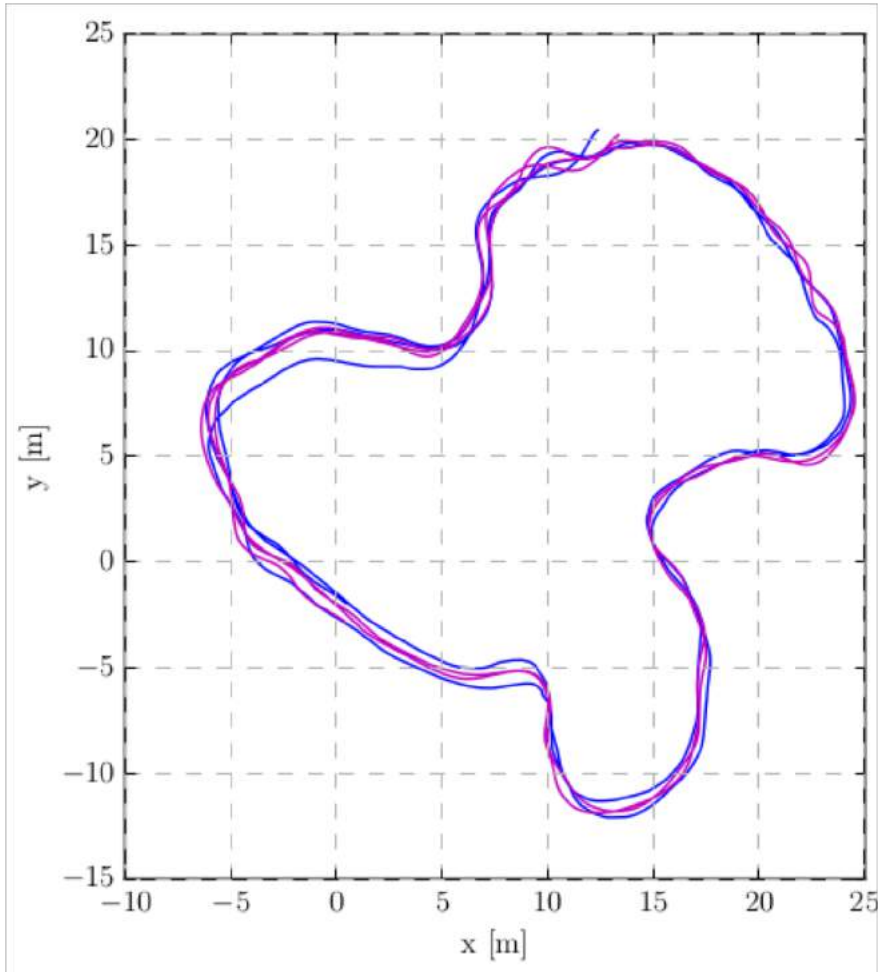


State estimation and mapping software architecture




Touch panel required

VIO vs. GPS deviation is under 1m



Perception

- Interpret the environment using redundant Vision and LiDAR systems

State estimation

- Estimate the state of the vehicle and its environment

CERES SOLVER

ROVIO

Planning & Controls

- **Generate feasible driving region and issue optimal control commands**

CVXGEN

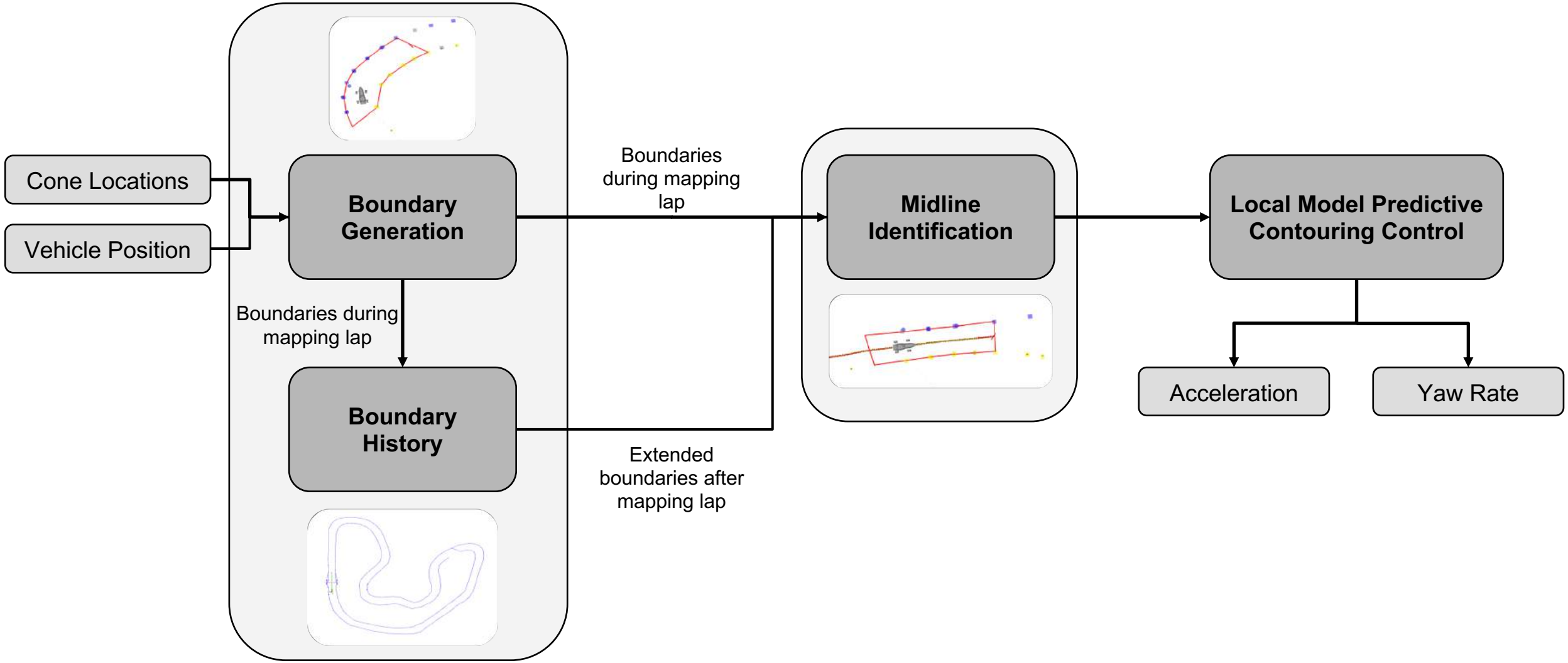
mosek

ROS

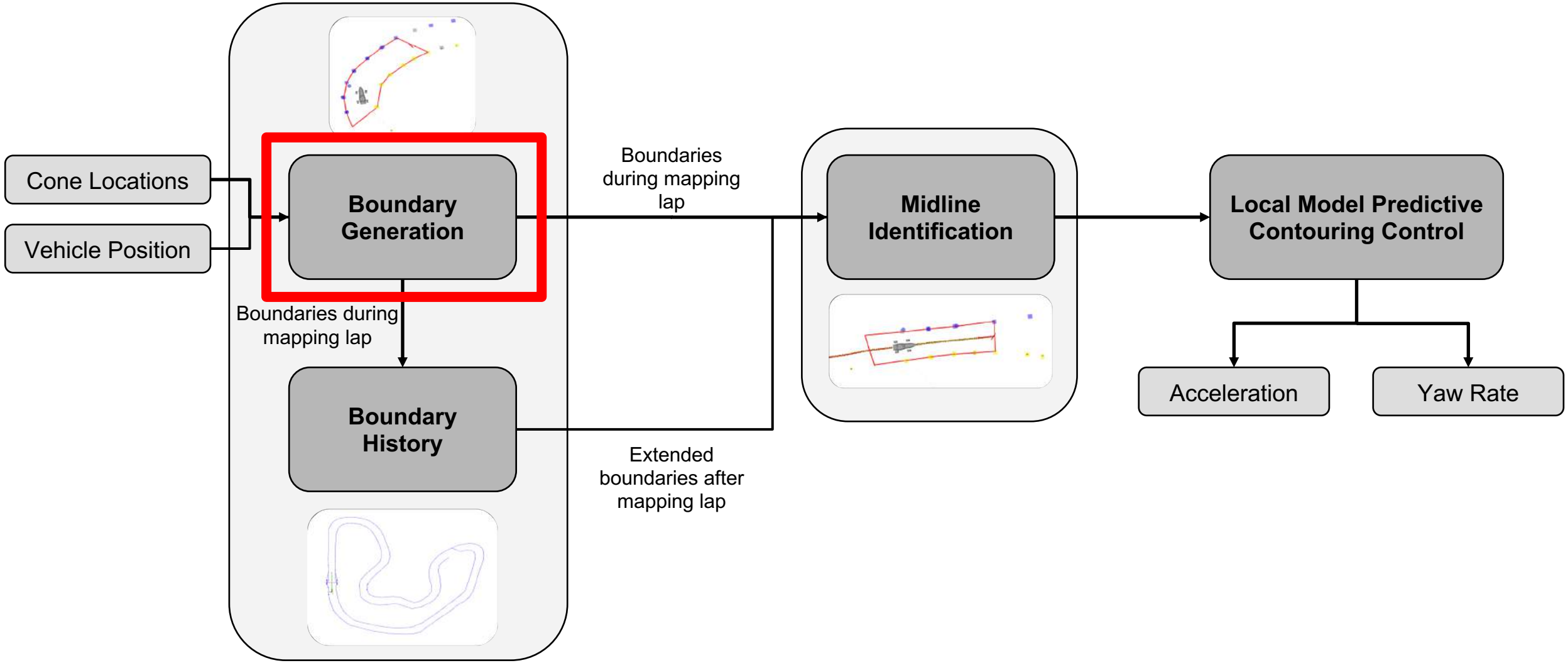
ubuntu

INTEL MATH KERNEL LIBRARY

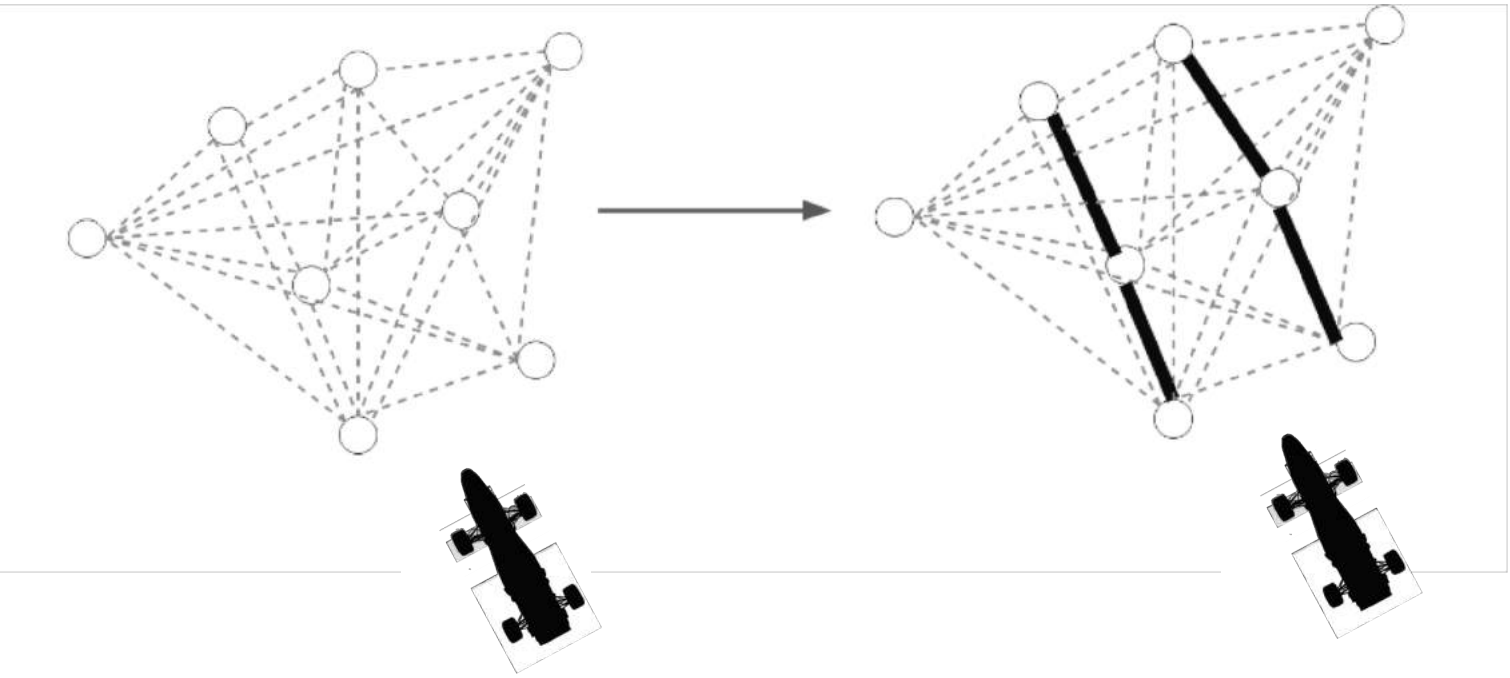
Path Planning: System Design



Path Planning: System Design

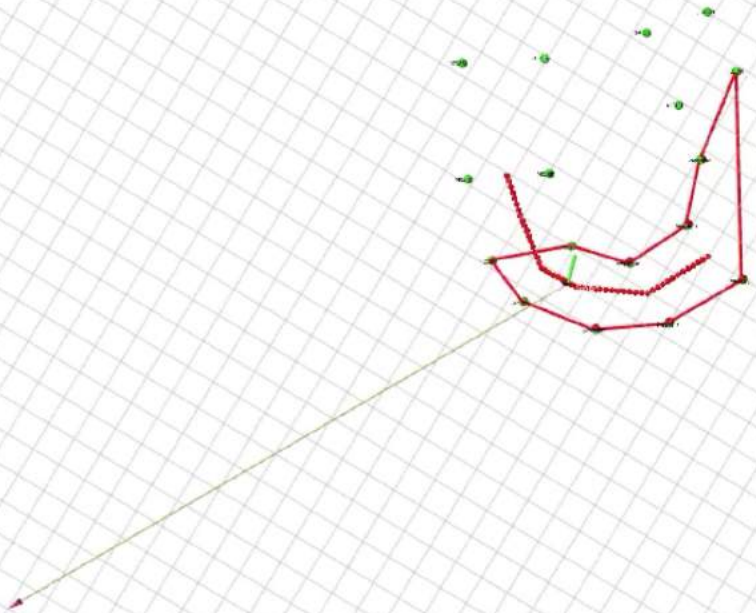


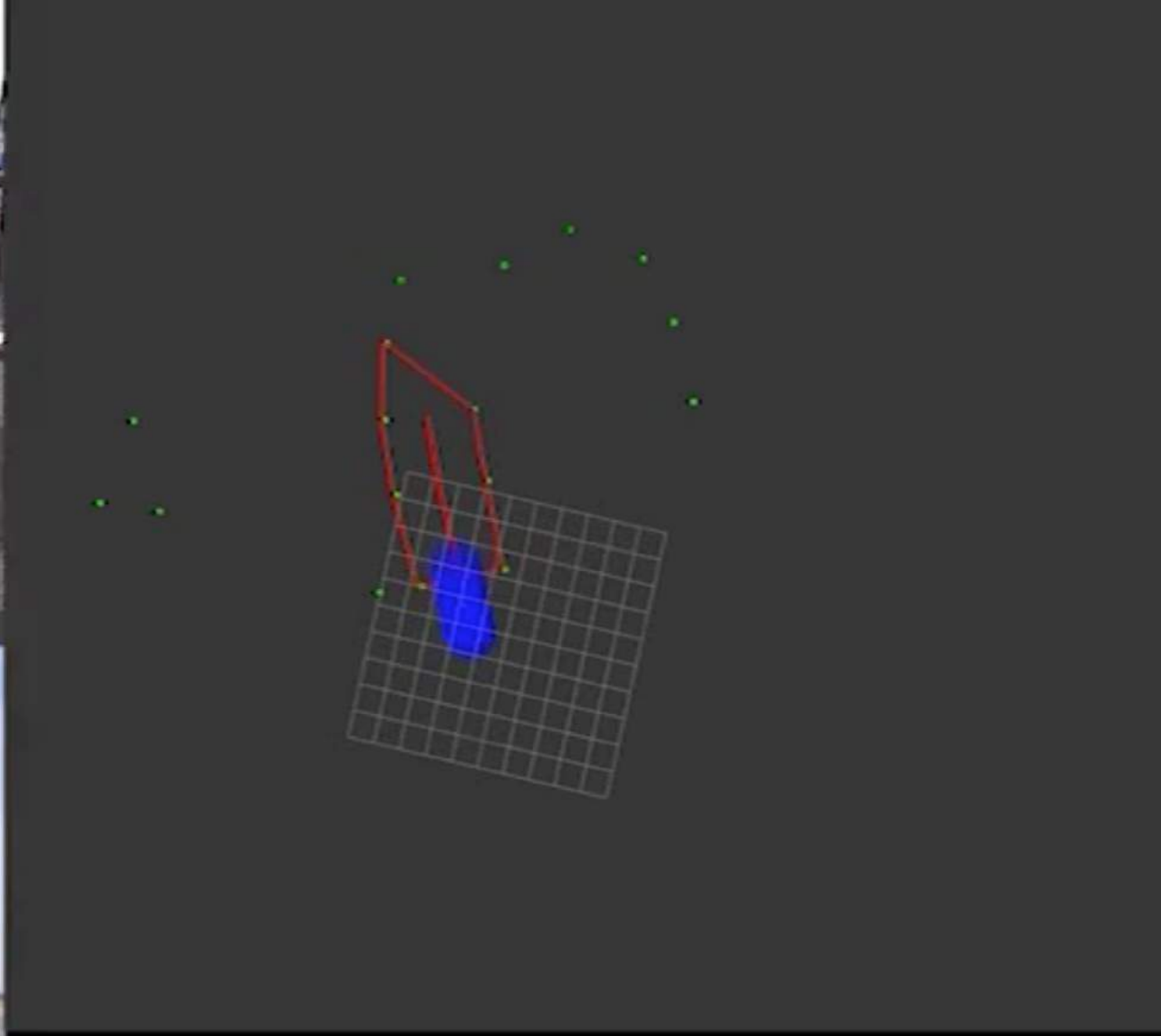
Path Planning: Boundary Generation



OPTIMIZATION DETAILS

- Mixed Integer Quadratic Program optimizing for:
  - Approximately straight edges
  - Edges with the expected cone spacing of the track
  - Edges near the angle of the heading
  - Edges connecting cones of the same color
- Constrained to find 2 non-crossing edges with no cycles
- Solved in CVXpy with Mosek





Thank you to our 2018-19 Season Sponsors!

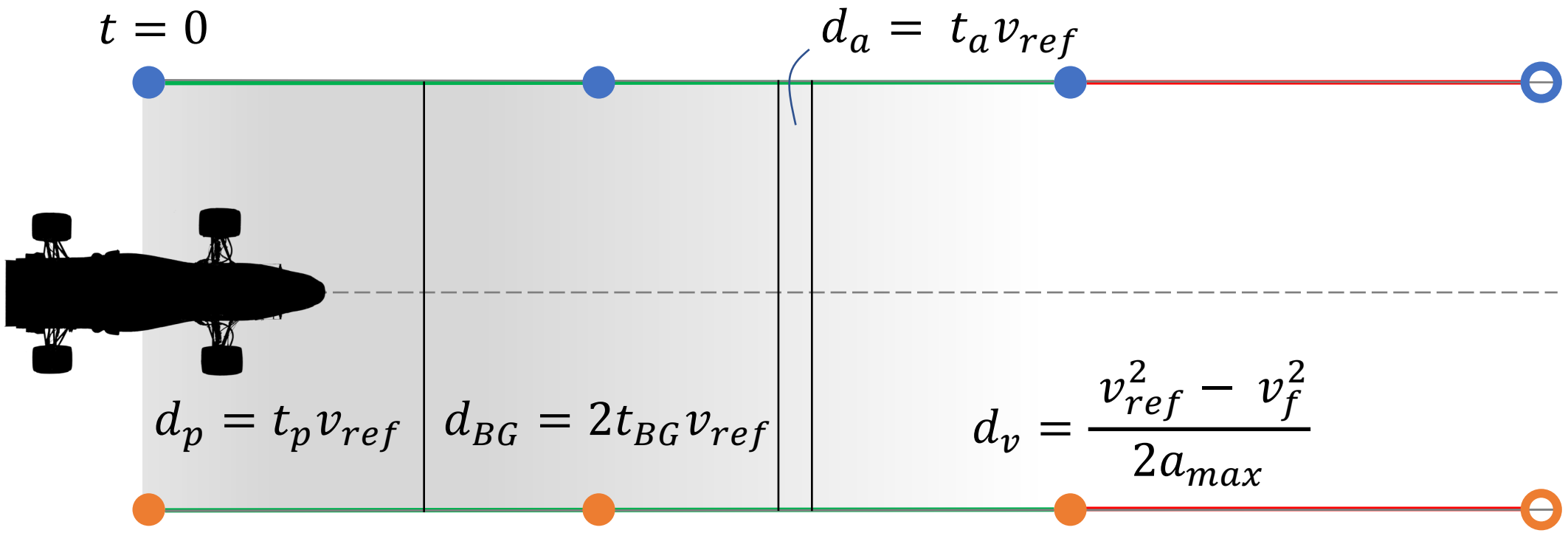
WAYMO

Altium

ANDREESSEN HOROWITZ

APTIV

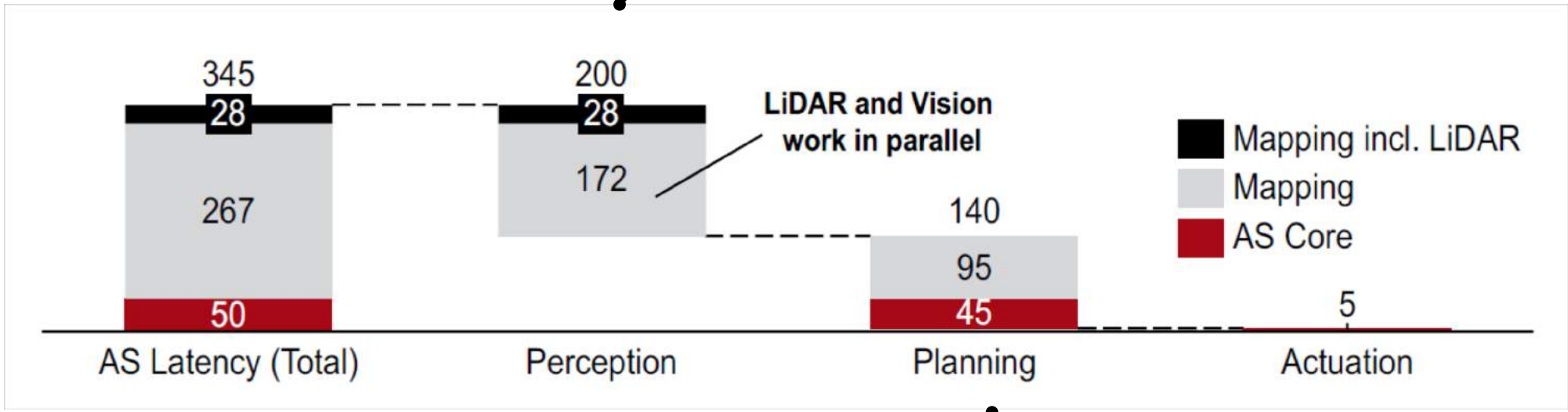
Latency and lookahead are the constraining factors to fast lap times (1/2)



$$V_{\text{ref}} \leq a_{\max}\left(\sqrt{(t_{\text{p}} + 2t_{\text{BG}} + t_{\text{act}})^2 + 2\frac{d_l}{a_{\max}} + \frac{v_f^2}{a_{\max}^2}} - (t_{\text{p}} + 2t_{\text{BG}} + t_{\text{act}})\right)$$

Latency and lookahead are the constraining factors to fast lap times (2/2)

Driven by CNN inference, depth stereo matching and point cloud clustering



Driven by boundary generation