

F1/10 Autonomous Racing

Course Introduction

Madhur Behl

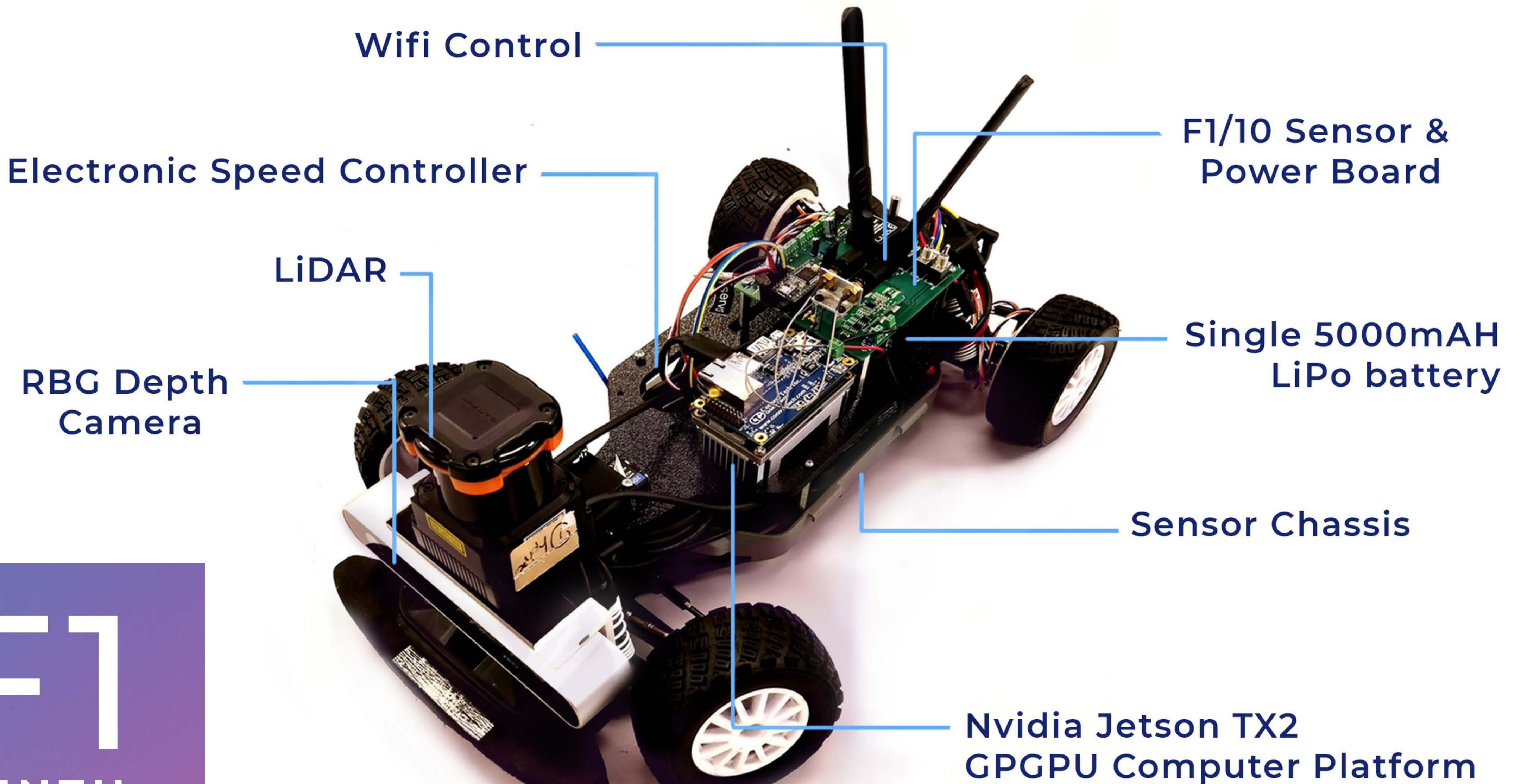
Computer Science | Engineering Systems and Environment | UVA Link Lab

CS 4501 -003 | 14th January 2020



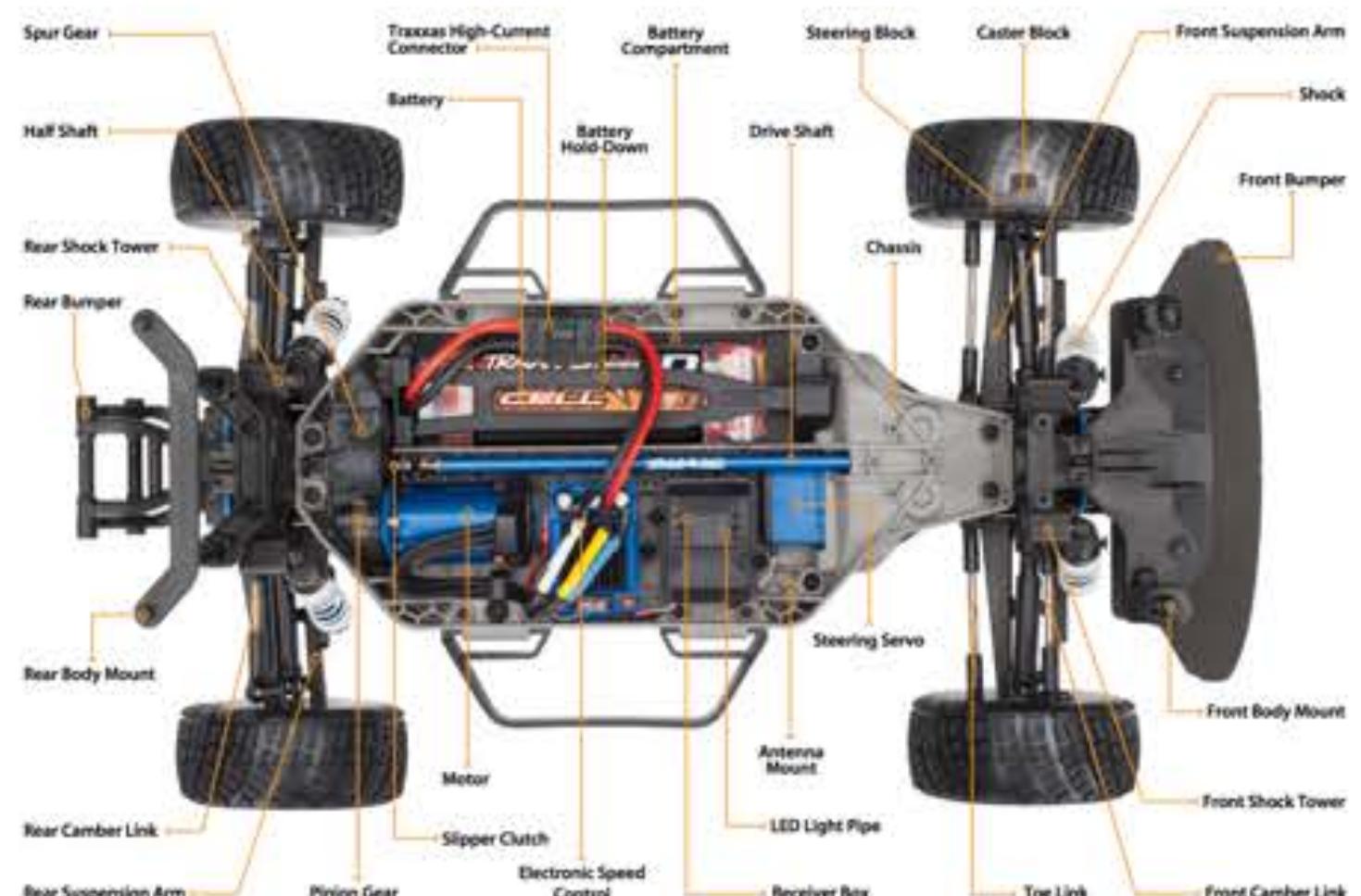
Overview

- Introduction to the Course and What is F1/10 ?
 - Logistics, grading, policies
- Course Support
 - Website, TAs, Piazza, Github
- Expectations
- Brief Introduction to Autonomous Racing topics
- Autonomous Racing Research @ UVA



F1
TENTH

f1tenth.org



F1/10 Platform Board



Chassis Design

Hokuyo
LiDAR

Zed
Depth Camera

FLIR Flea
Camera

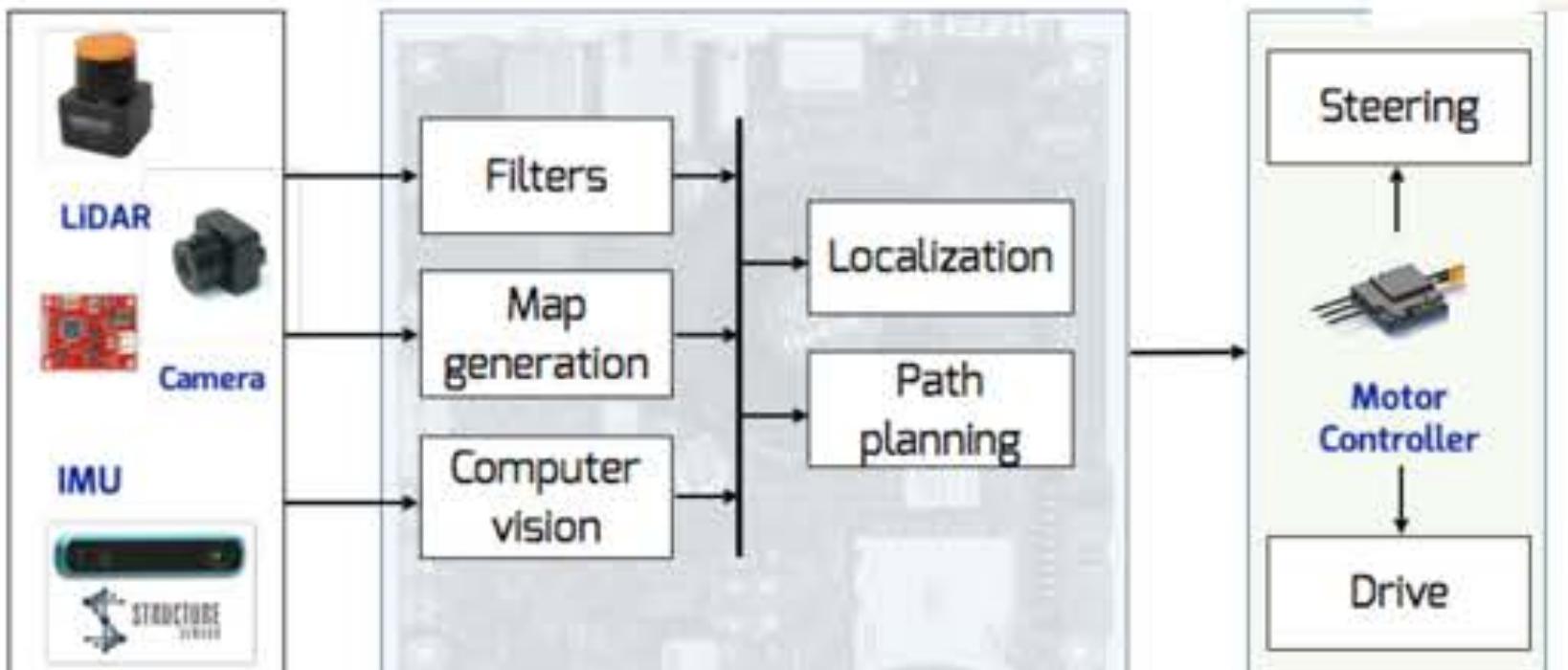
Electronic
Speed
Controller

WiFi
Telemetry

GPGPU Compute Platform
NVIDIA Jetson TX1/TX2

Software Architecture

ROS



Perception

Planning

Control

GPU Accelerated
Libraries

NVIDIA CUDA



Safe
Autonomy

Secure
Autonomy

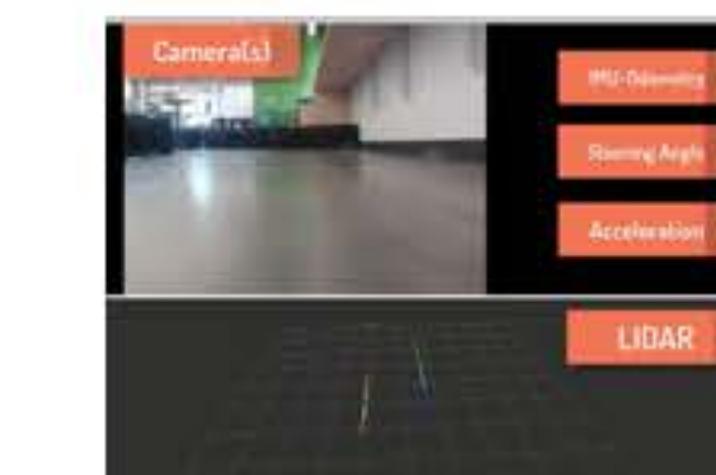
Coordinated
Autonomy

Efficient
Autonomy

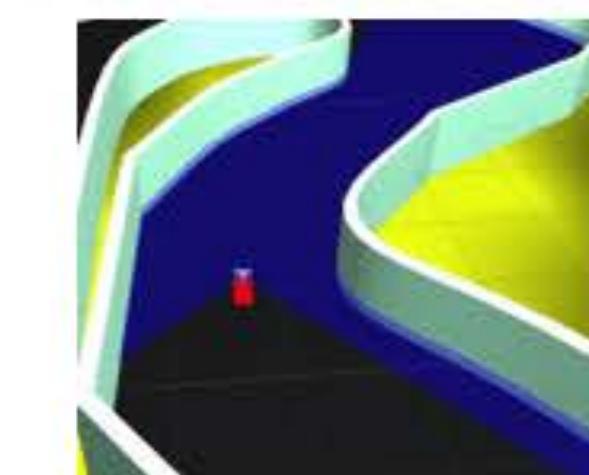
Lane Keeping Assist



AV Data Collection



F1/10 Simulator

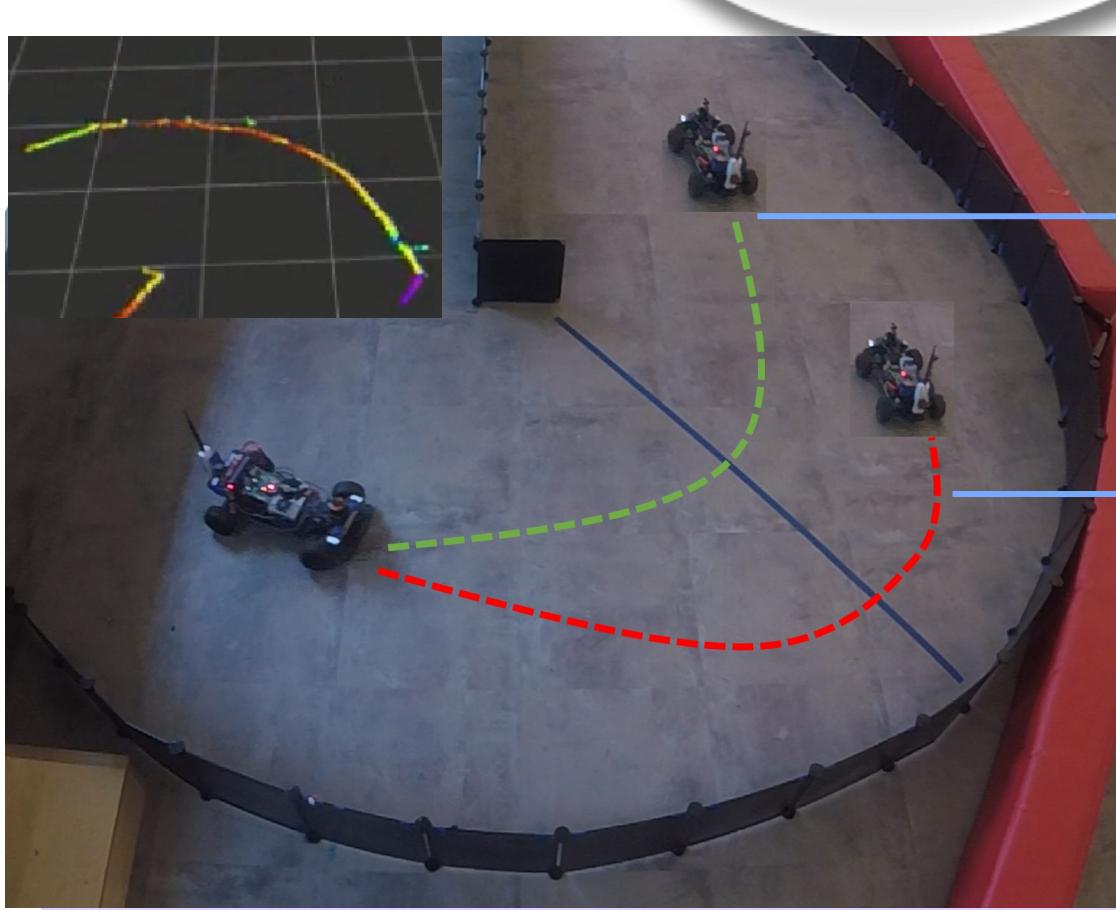


Model-Predictive
Control

Simultaneous Localization
And Mapping (SLAM)

DNN Racing

FOLLOW THE GAP METHOD



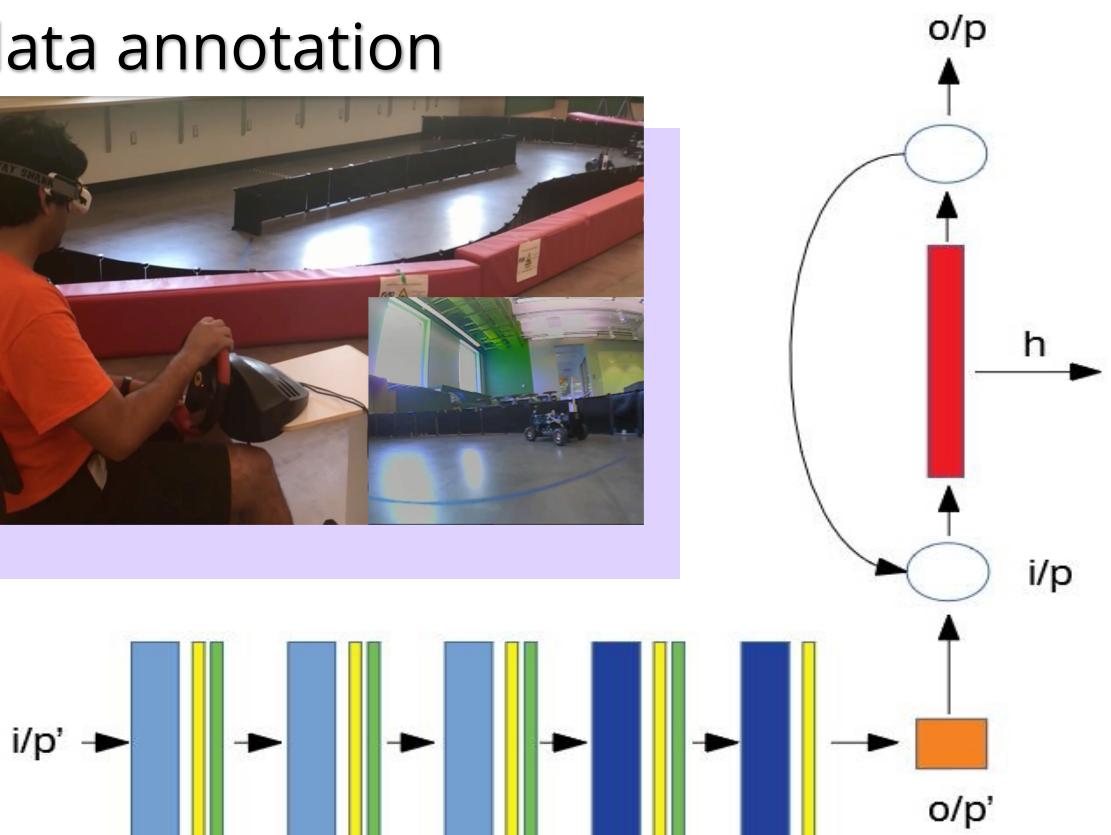
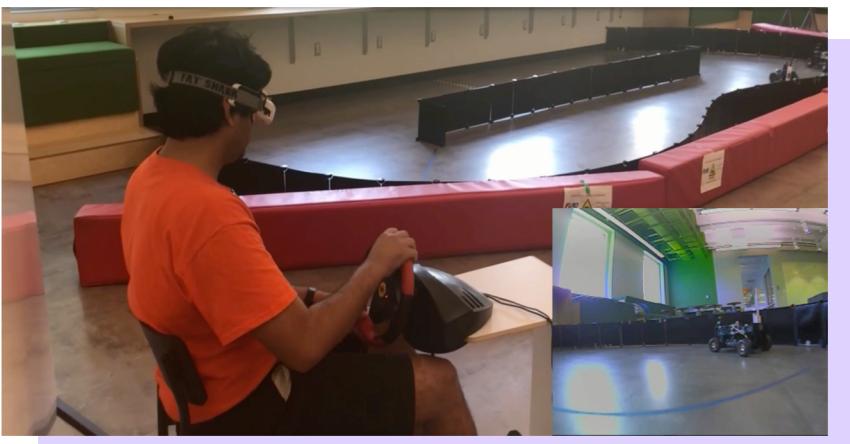
Follow the
gap method

Simple
obstacle
avoidance

F1
TENTH

END-TO-END DNN

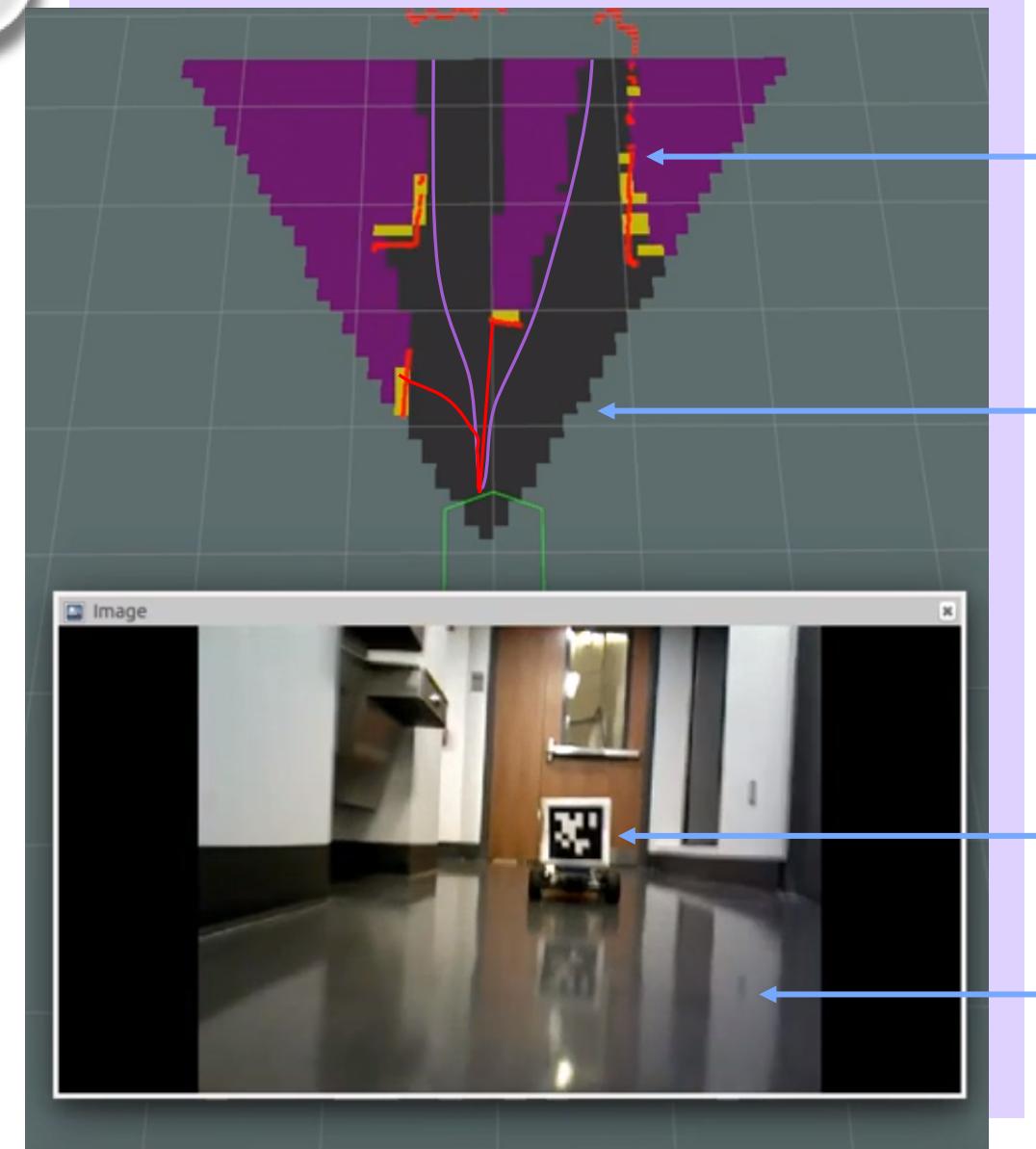
FPV data annotation



Predicted steering (blue)
Ground Truth (green)



MODEL PREDICTIVE CONTROL



Generated
paths

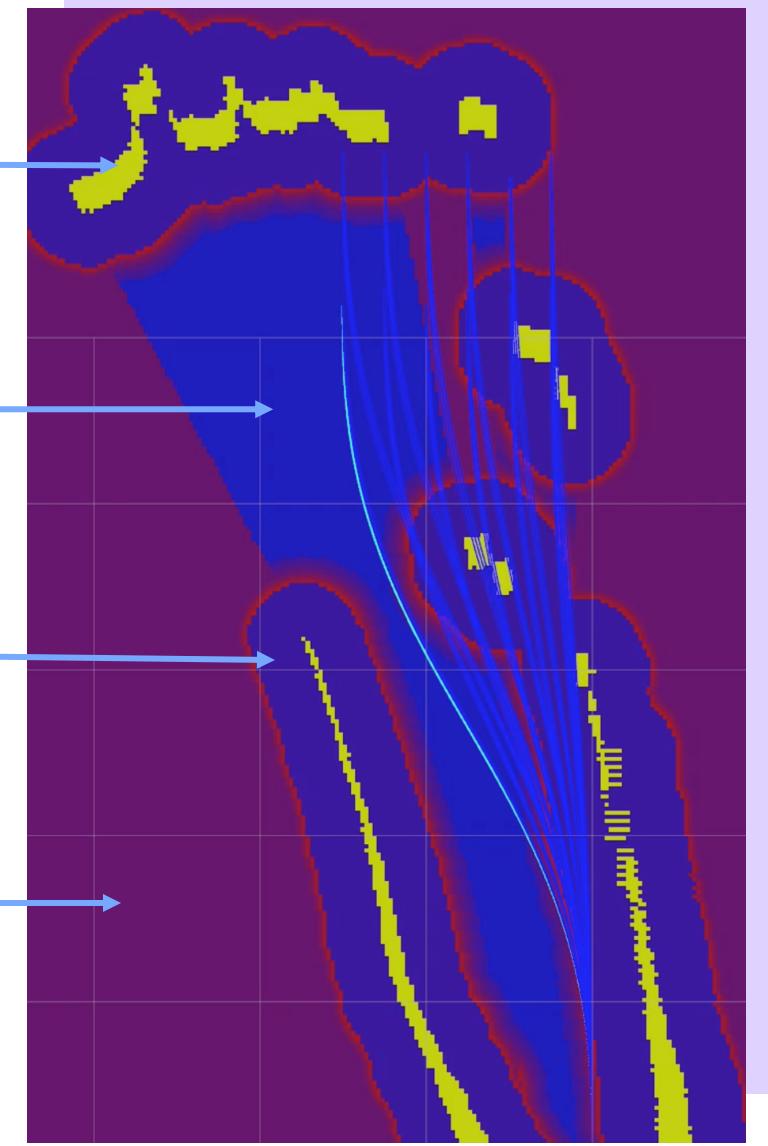
Drivable
region

Medium
cost

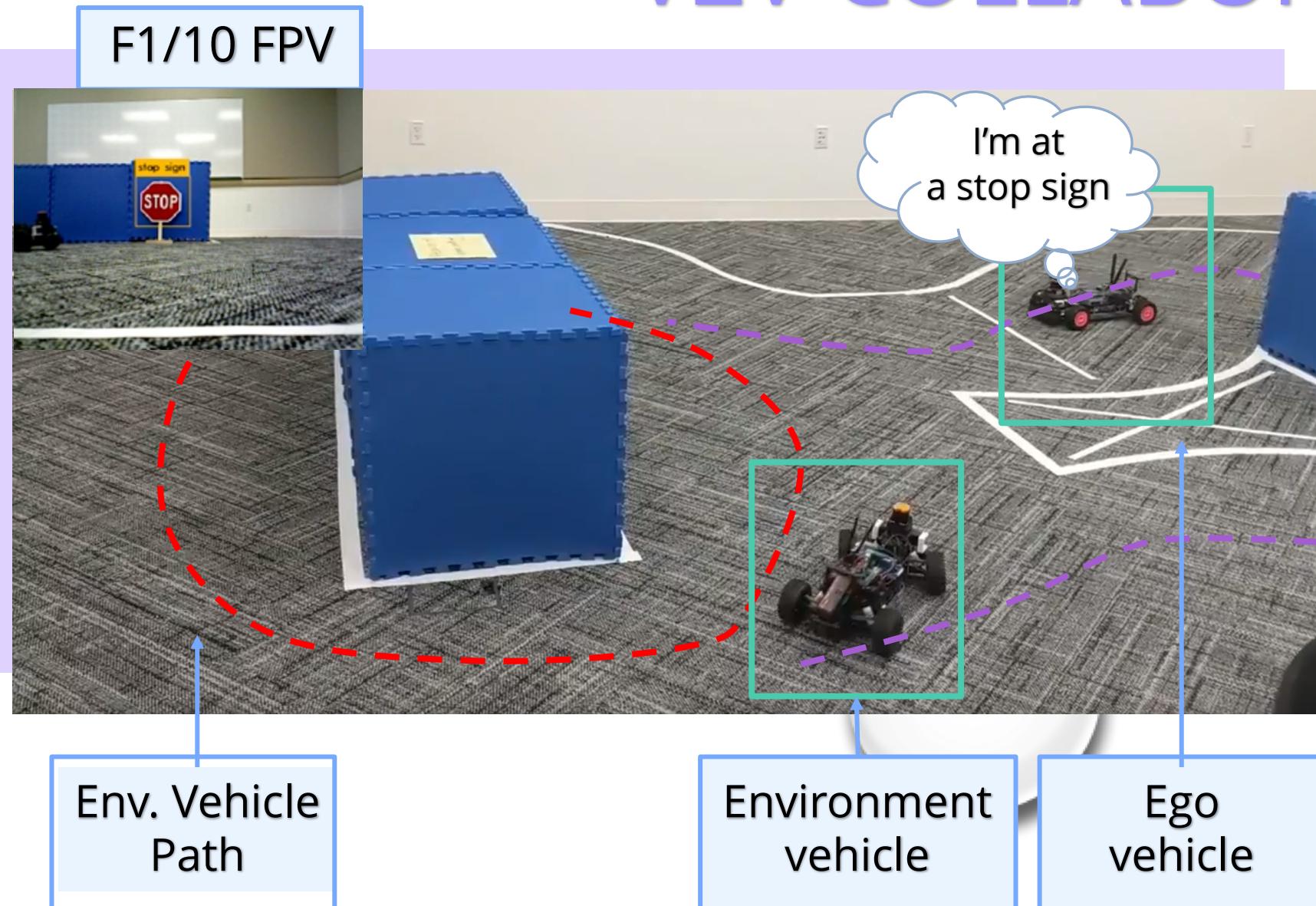
Unexplored
space

F1/10
RGB FPV

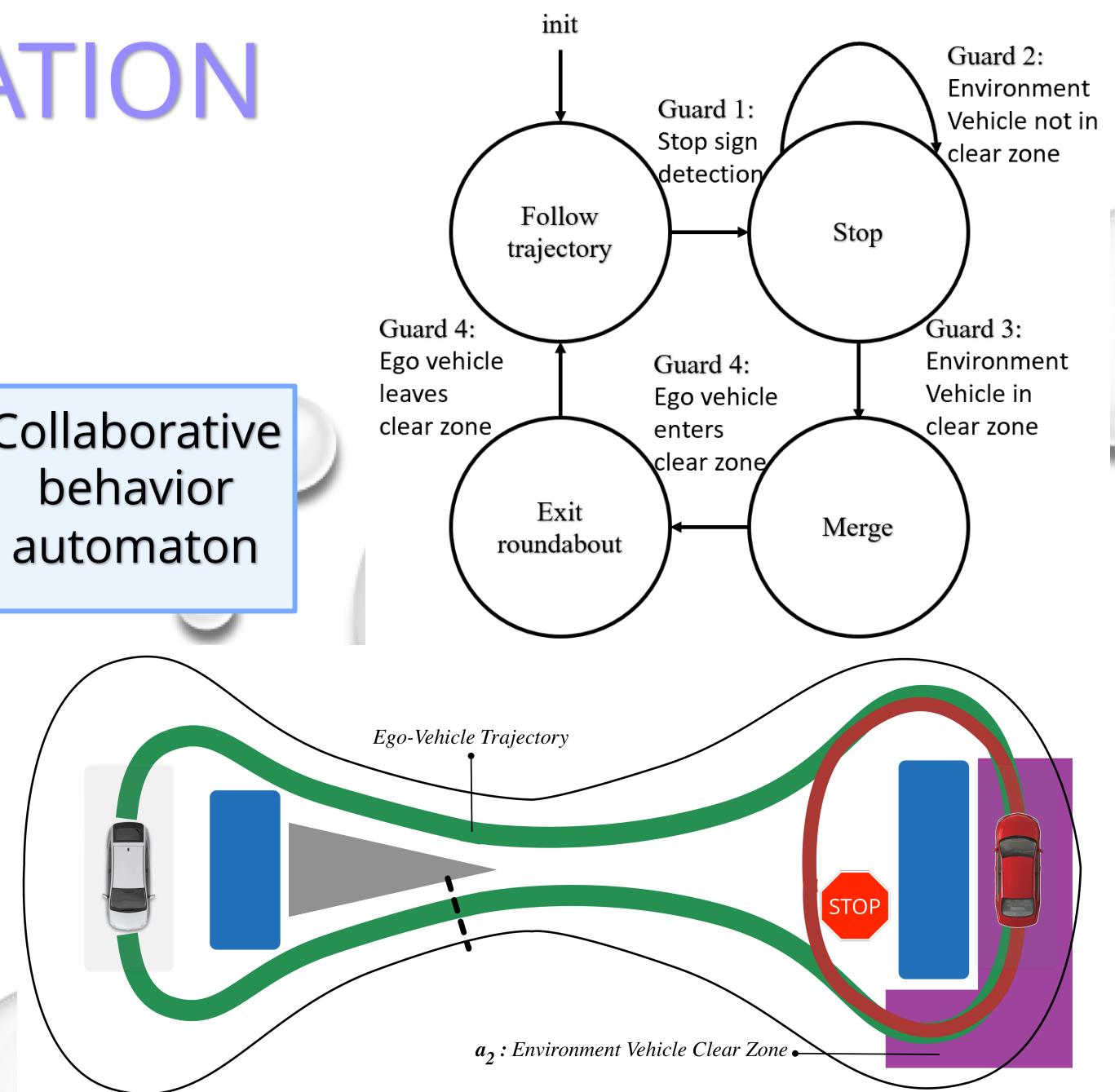
Cost Map Legend



V2V COLLABORATION

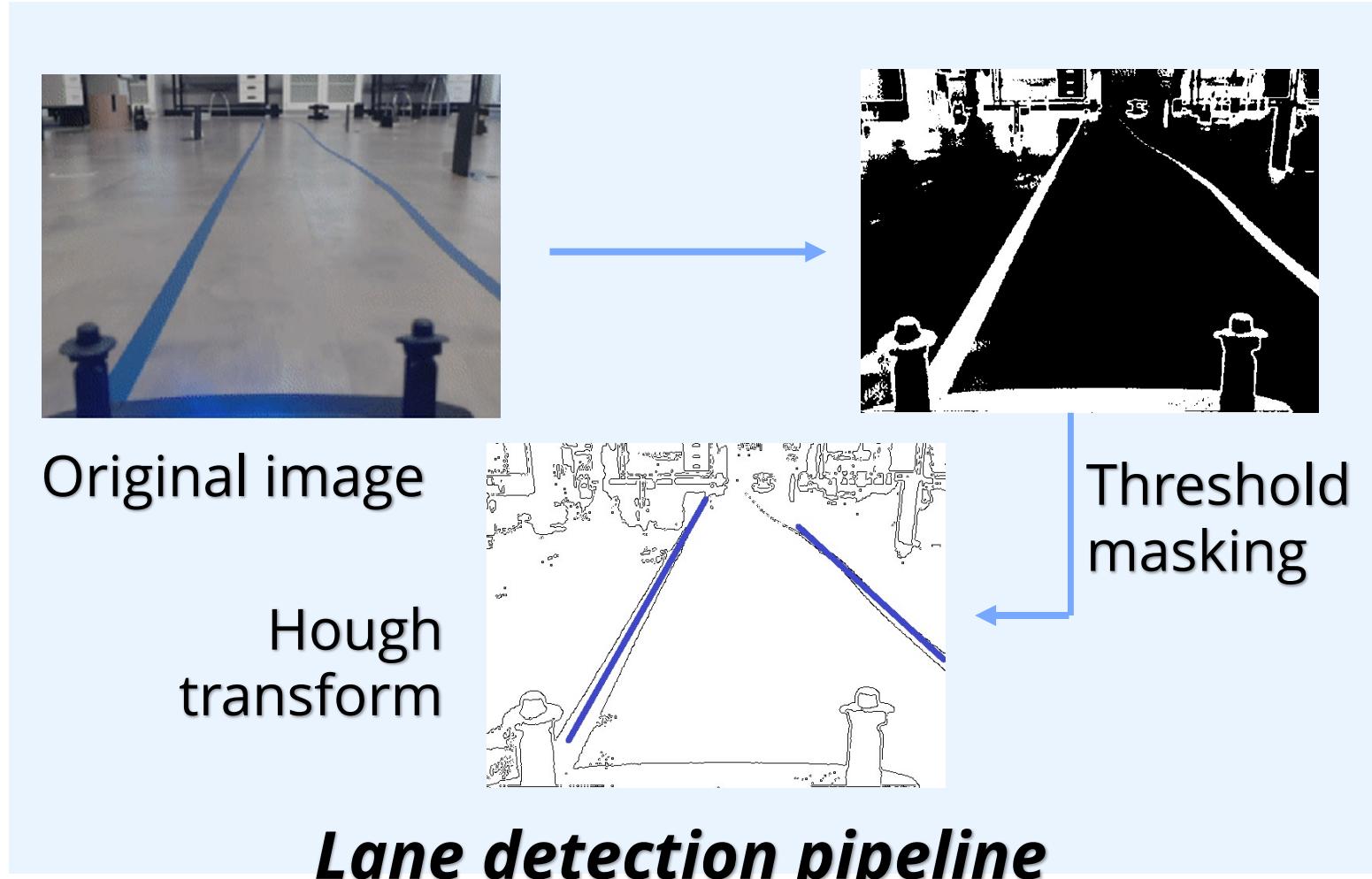


Collaborative
behavior
automaton

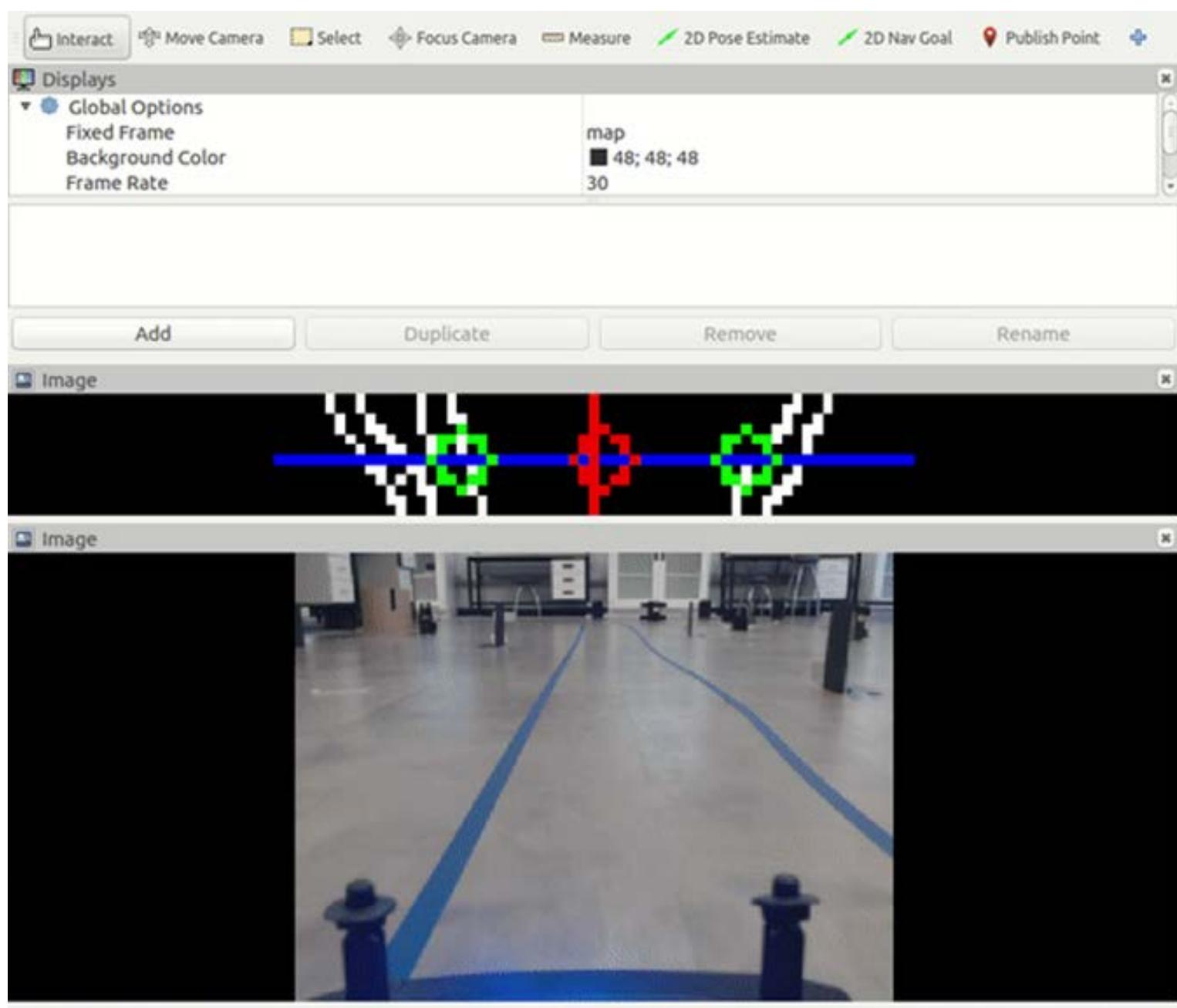


a_2 : Environment Vehicle Clear Zone

LANE KEEPING ASSIST

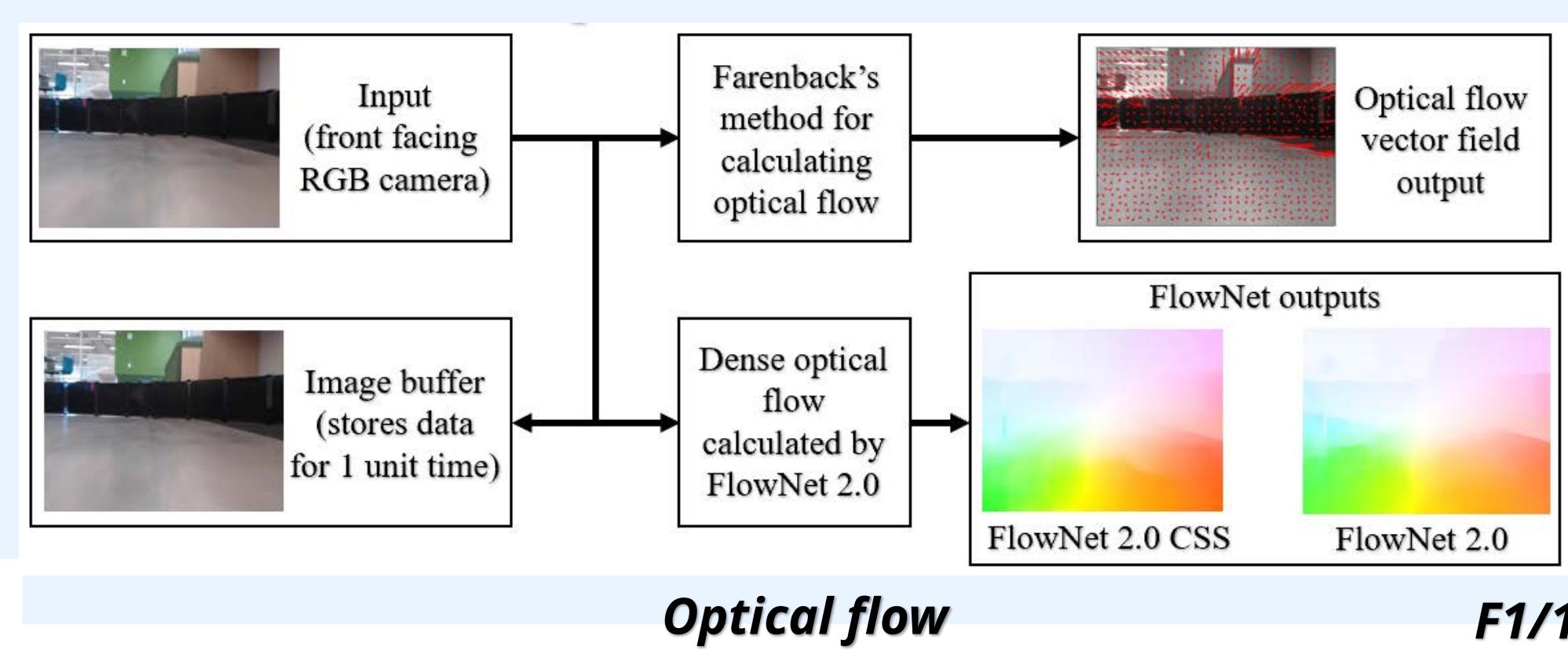


Lane detection pipeline



rviz visualization

COMPUTER VISION EXAMPLE



F1/10 online semantic segmentation

F1
TENTH

Currently tested
On F1/10

Hector
SLAM

AMCL

Google
Cartographer

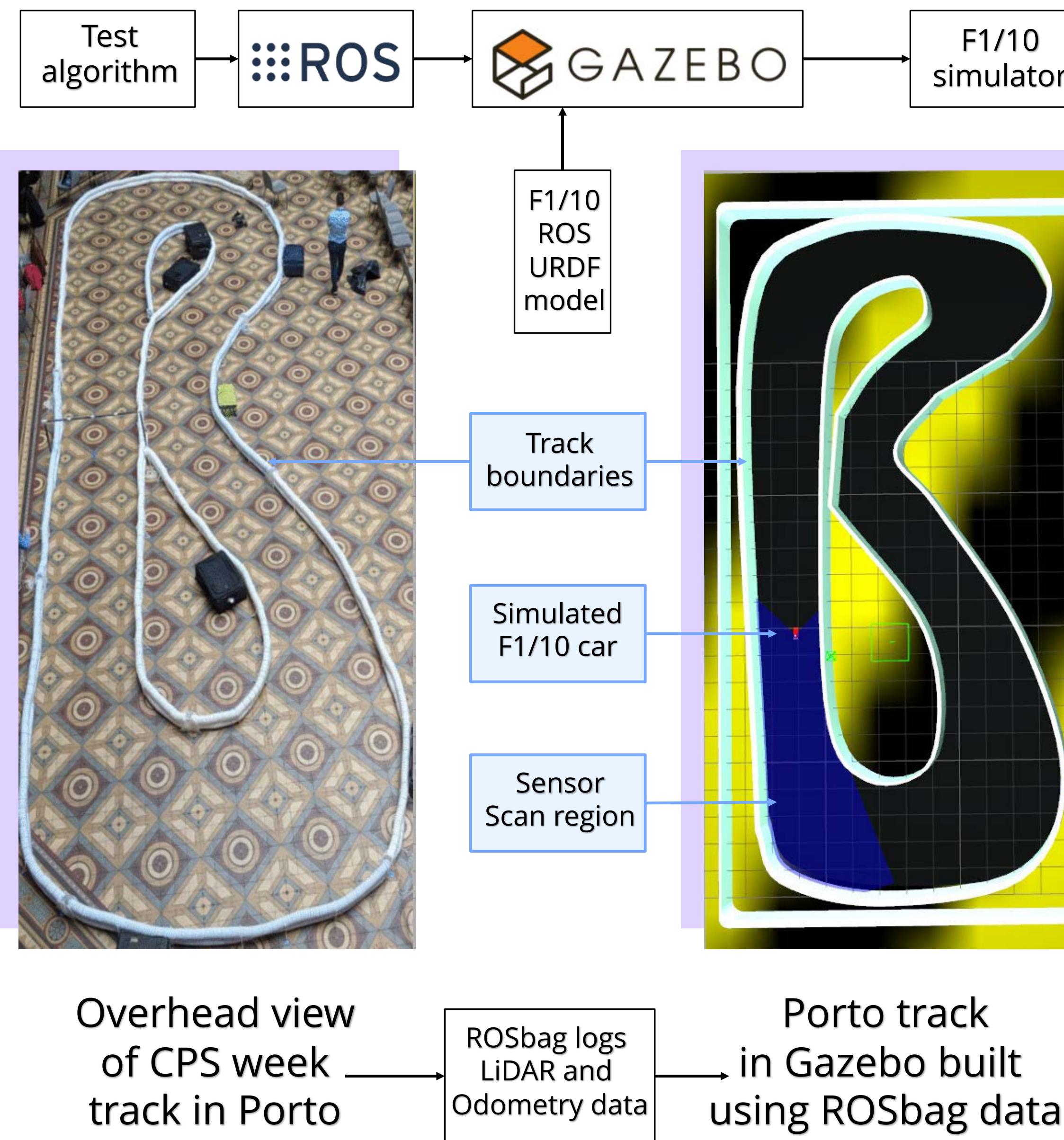
ORB-
SLAM



Global planning using rviz

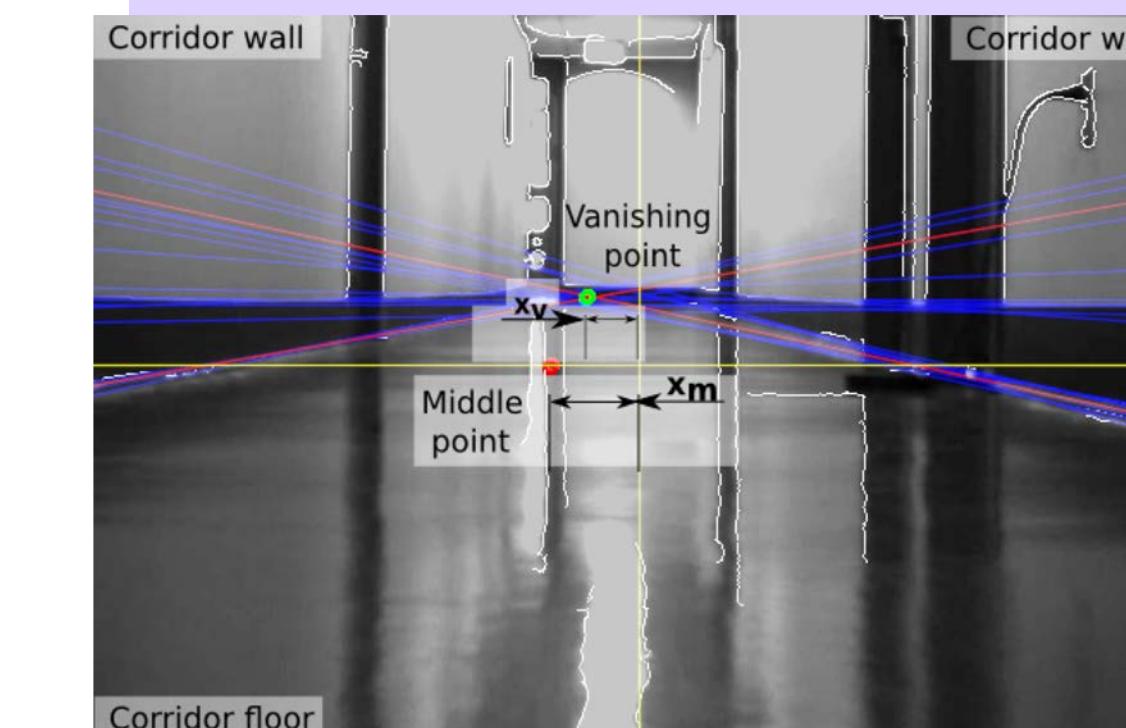
ROS transform frame

F1TENTH GAZEBO SIMULATOR

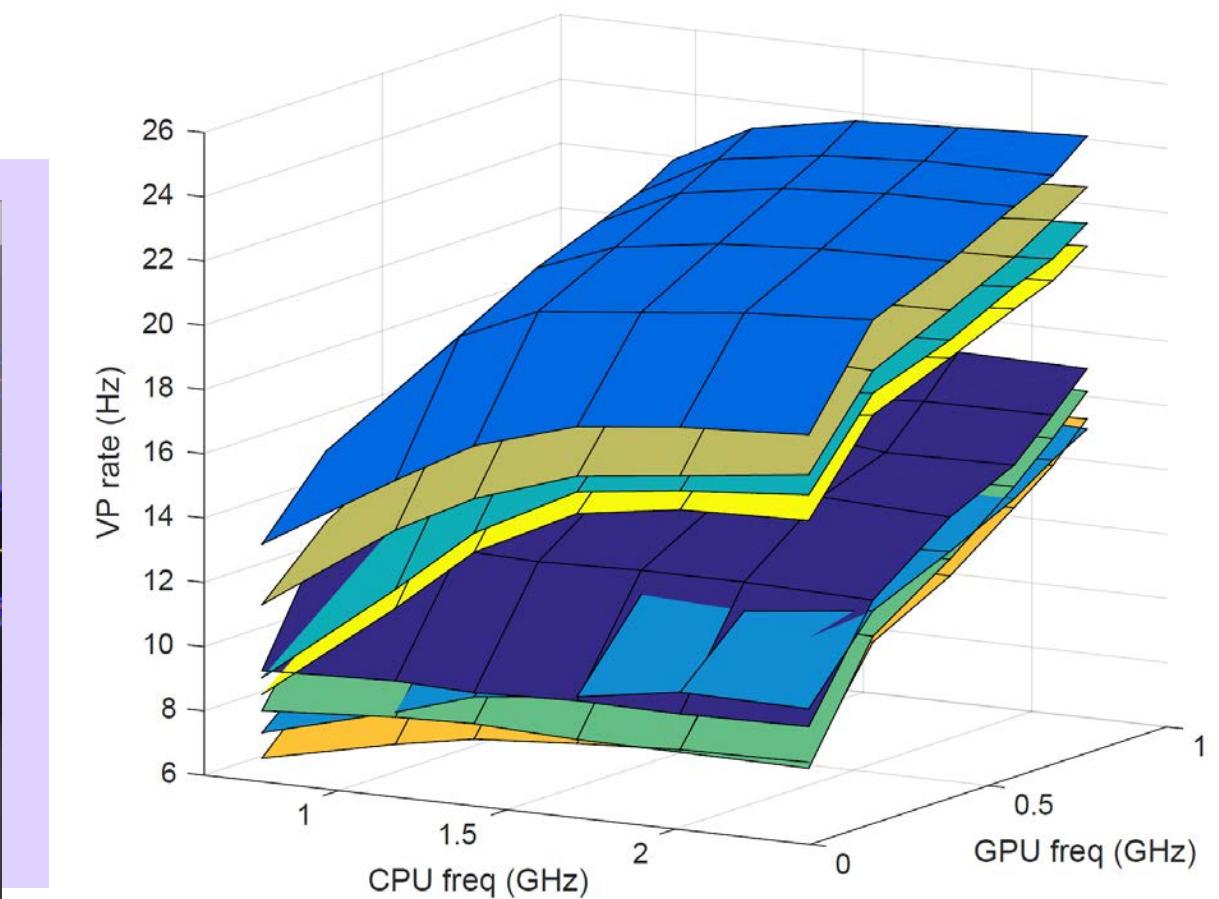


REAL TIME SCHEDULING

Vanishing point (VP) algorithm implemented on F1/10



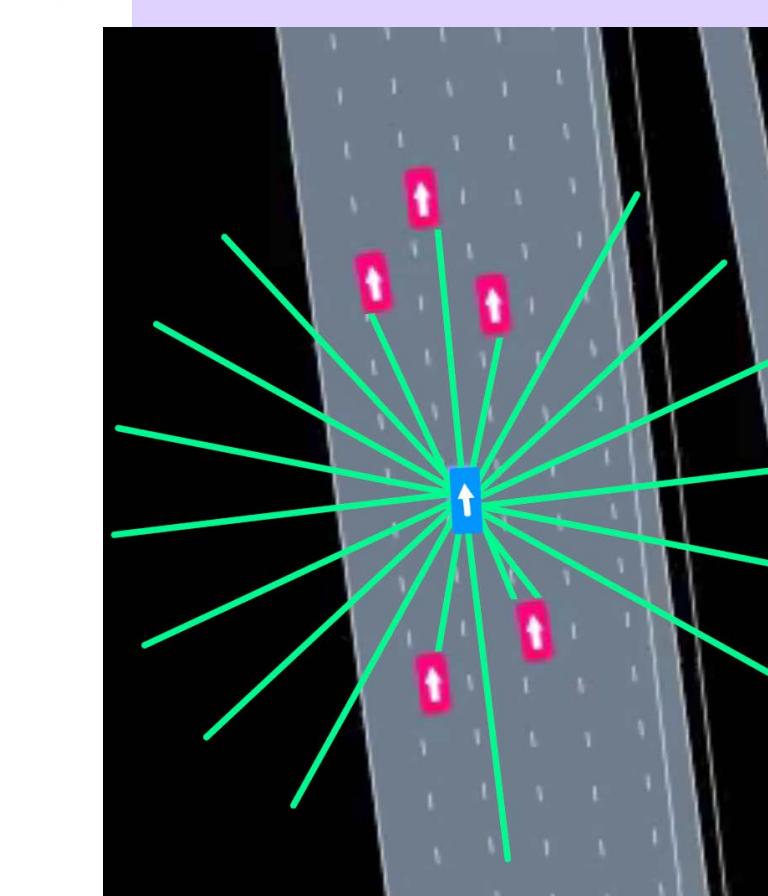
Green dot (vanishing point)
Red dot (middle point)



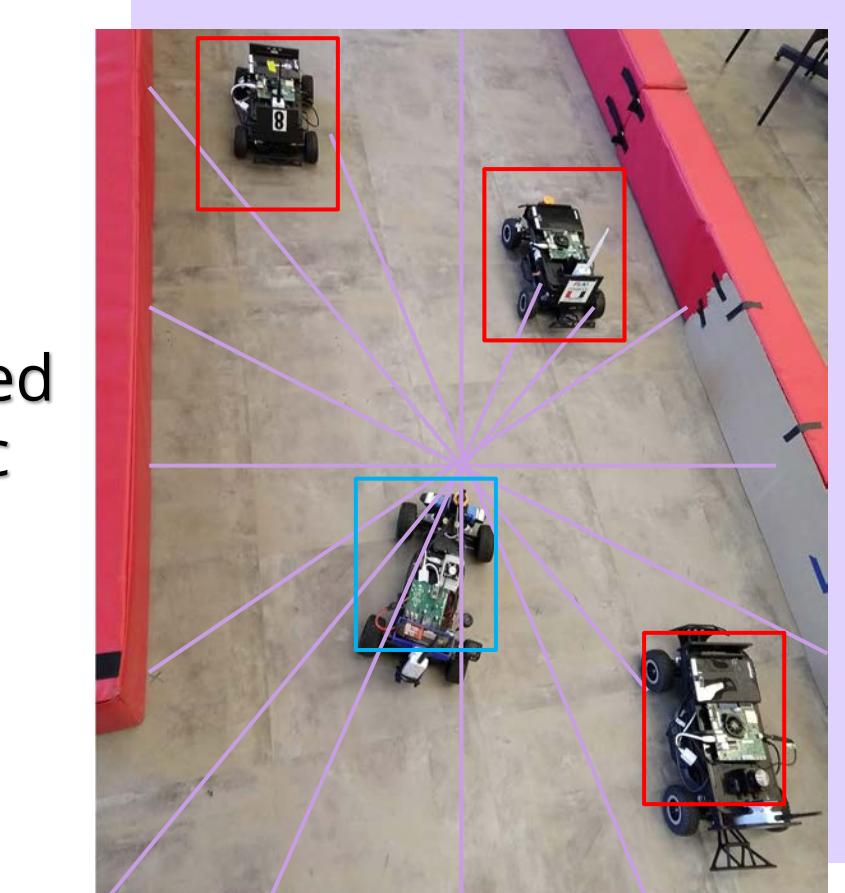
VP throughput based on CPU & GPU frequencies

F1
TENTH

TESTING & VERIFICATION

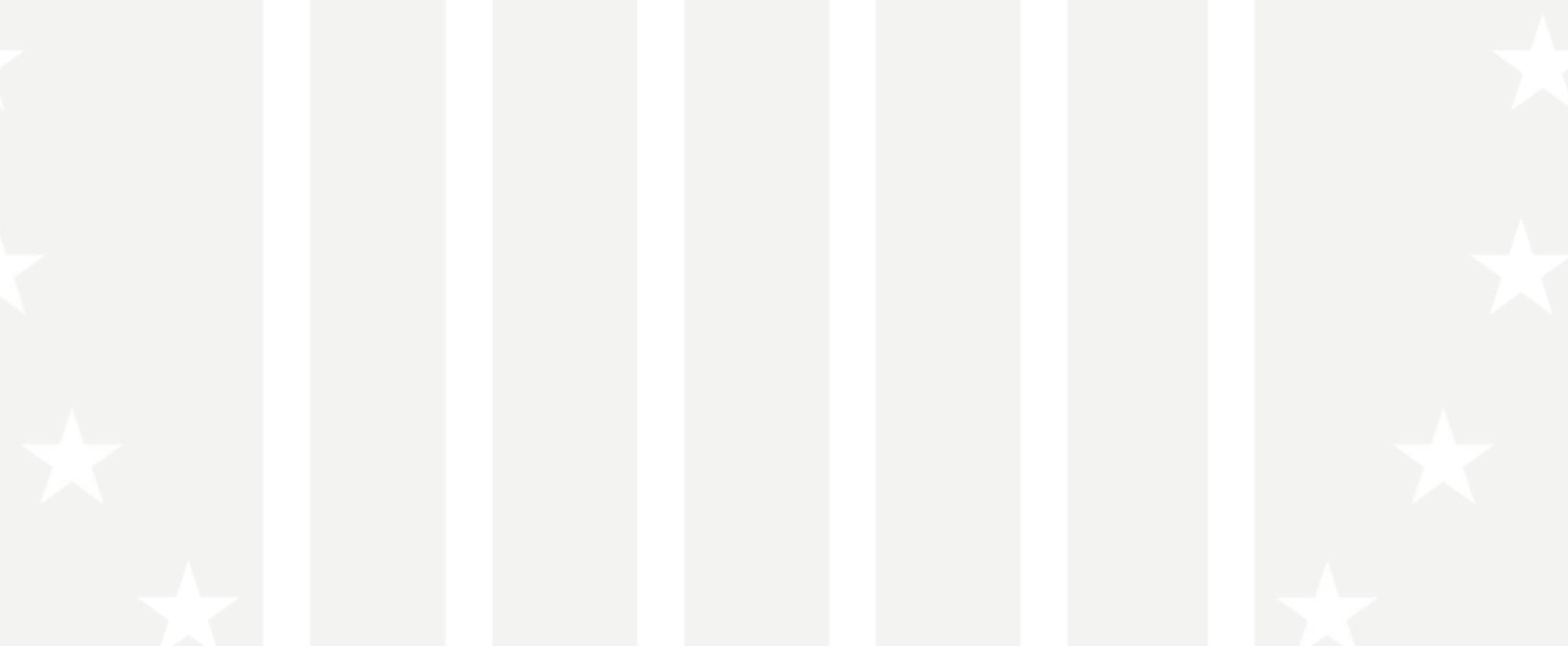


GAIL model: Outputs include mean & variance of steering & throttle



F1/10 helps recreate traffic scenarios to verify model

Course logistics



Instructor

Prof. Madhur Behl

Assistant Professor

Computer Science

Engineering Systems and Environment

Founder: F1/10 International Autonomous Racing Competition

Director: Cavalier Autonomous Racing

How to reach me:

Office: Olsson 265 (Link Lab)

Office Hours: Every Monday 2-3pm in my office (or by appt.)

Webpage: <http://www.madhurbehl.com/>

Email: madhur.behl@virginia.edu



Teaching Assistants

Varundev SureshBabu

PhD Candidate: Computer Engineering

vss8sm@virginia.edu



Sandesh Banskota

Mechanical Engineering Major & Computer Science Minor

sb3vf@virginia.edu



Mert Karakas

Computer Engineering Major

mmk2nh@virginia.edu



Course logistics

Lectures:

3:30-4:45p Tuesday & Thursday
Rice 120

Lab Demos and Practice:

Rice 242

Course Website:

<https://linklab-uva.github.io/autonomoustracing/>
UVA Collab for assignment submission

Piazza: TBD

Textbooks: None you have to buy

Github: You will have access to template code for labs and assignments via Git

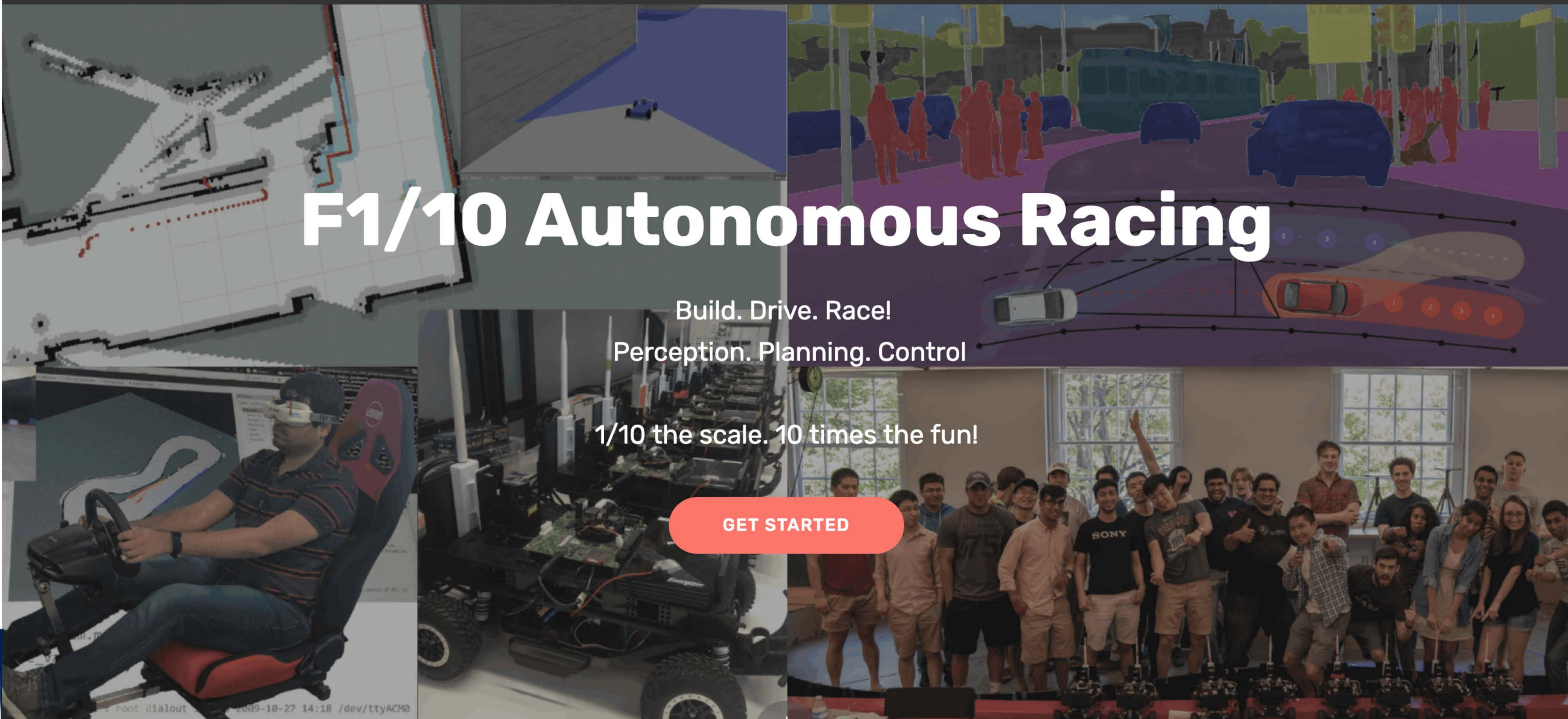
[Home](#)[Logistics](#)[Syllabus](#)[Resources](#)

F1/10 Autonomous Racing

Build. Drive. Race!

Perception. Planning. Control

1/10 the scale. 10 times the fun!

[GET STARTED](#)



f1tenth.org

Autonomous Racing Competition

1/10th the size. 10 times the fun!

[Start Your Engines!](#)

Stay in the loop with email updates!

[Subscribe](#)

Topics to be covered

Week 1

Course Introduction

Week 2

Introduction to Robot Operating Systems

Week 3

ROS deep dive + TurtleSim

Week 4

Pulse Width Modulation - Teensy

Week 5

Perception – Sensing your surroundings

Week 6

PID Control and path following

Week 7

Wall following, boundary detection

Topics to be covered

Spring Break

Week 8

Basics of Navigation - Localization

Week 9

Mapping - SLAM

Week 10

Trajectory planning

Week 11

Open CV, Visual Odometry

Week 12

Advanced topics in scene understanding

Week 13

Racing strategy

Lab exercises

P1

19°C

88%

P2

19°C

88%

Q

24°C

55%

R

24°C

0%

Practice Session 1:

Understanding ROS | Setting up the Car

Practice Session 2:

Perception/Sensing | Driving Straight

20
TURNS

3.4
MILES

LAPS

Practice Session 3:

Driving in a loop | Visual Odometry

Practice Session 3:

Race | Time trials | Head-to-head racing

Grading Criteria

This course does not have a midterm or a final exam.

The final grade will be determined on the basis of:

- Weekly assignments
- Lab exercises
- Short project
- Final race performance

Students will be graded individually and in teams.



This is an **advanced** special topics class. Which means, that although, we will cover everything which is required to build, drive, and race the cars; it is assumed that students have some familiarity with the topics highlighted below.

To succeed in this course, you need to have some experience with the following topics:

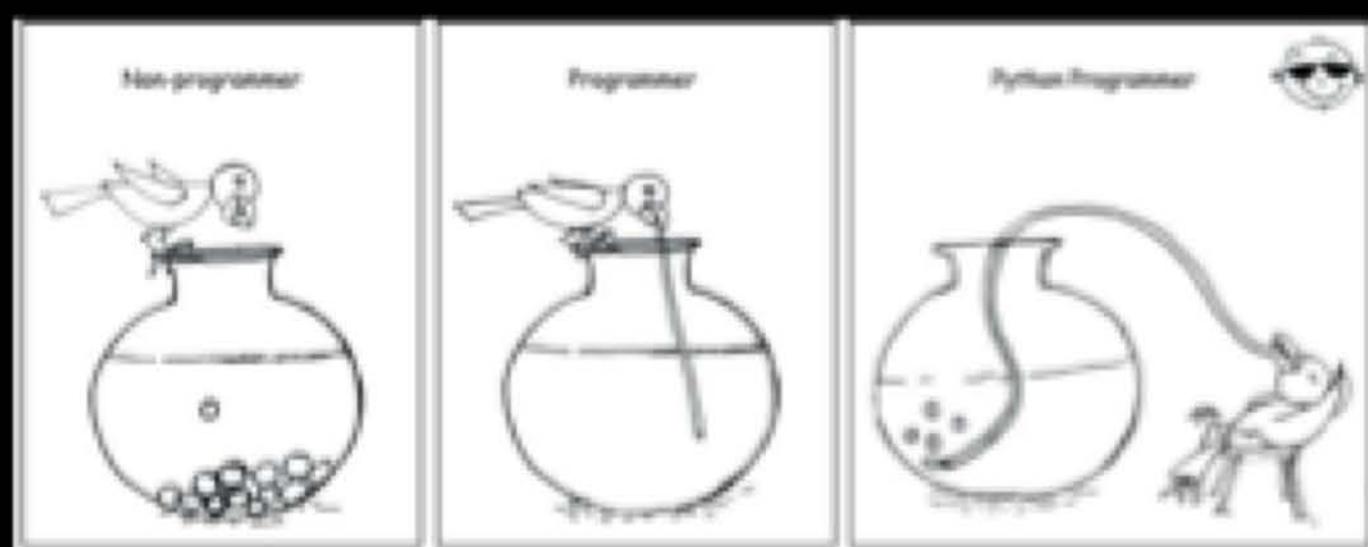
- **Calculus and linear algebra:**
 - Single variable calculus and differential equations.
 - Matrix operations- transformations and rotations.
- **Basic statistics and probability:**
 - What is a probability distribution ?
 - What is sampling ? Mean, and variance of samples.
- **Intermediate Python:**
 - Function calls, conditional statements, loops and recursion
- **Unix/linux command line/shell basics**
 - File commands/file permissions: *ls, cd, pwd, mkdir, rm, cp, mv, touch, chmod, tar*
 - Process management: *ps, top, kill pid*
 - *ssh user@host., grep, locate, echo*
 - Installation: *./configure, make, make install*
 - Ports: */dev/ttyACM**
- **Basic physics (Newtonian mechanics)**

Background in the following is recommended, but not required:

- Intermediate C++
- ROS programming
- Integrating sensors with microcontrollers (master/slave configurations)
- Machine learning

If upon looking at these topics, your conclusion is: *I'm qualified, but not that qualified.* Then it is likely that you are qualified.

If you are still unsure, feel free to email the instructor for permission to register.



```
int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
              // guaranteed to be random.
}
```

Expectations

- Maturity in programming, linear algebra, calculus, and statistics
- Proficiency in Linux, command line, Github,
- Familiarity with computer vision, and robotics is a plus.
- If you don't have the background, come back another semester. This course is offered in each Spring semester.
- Course is listed as a specials topics course – we may make adjustments along the way

What this course is not

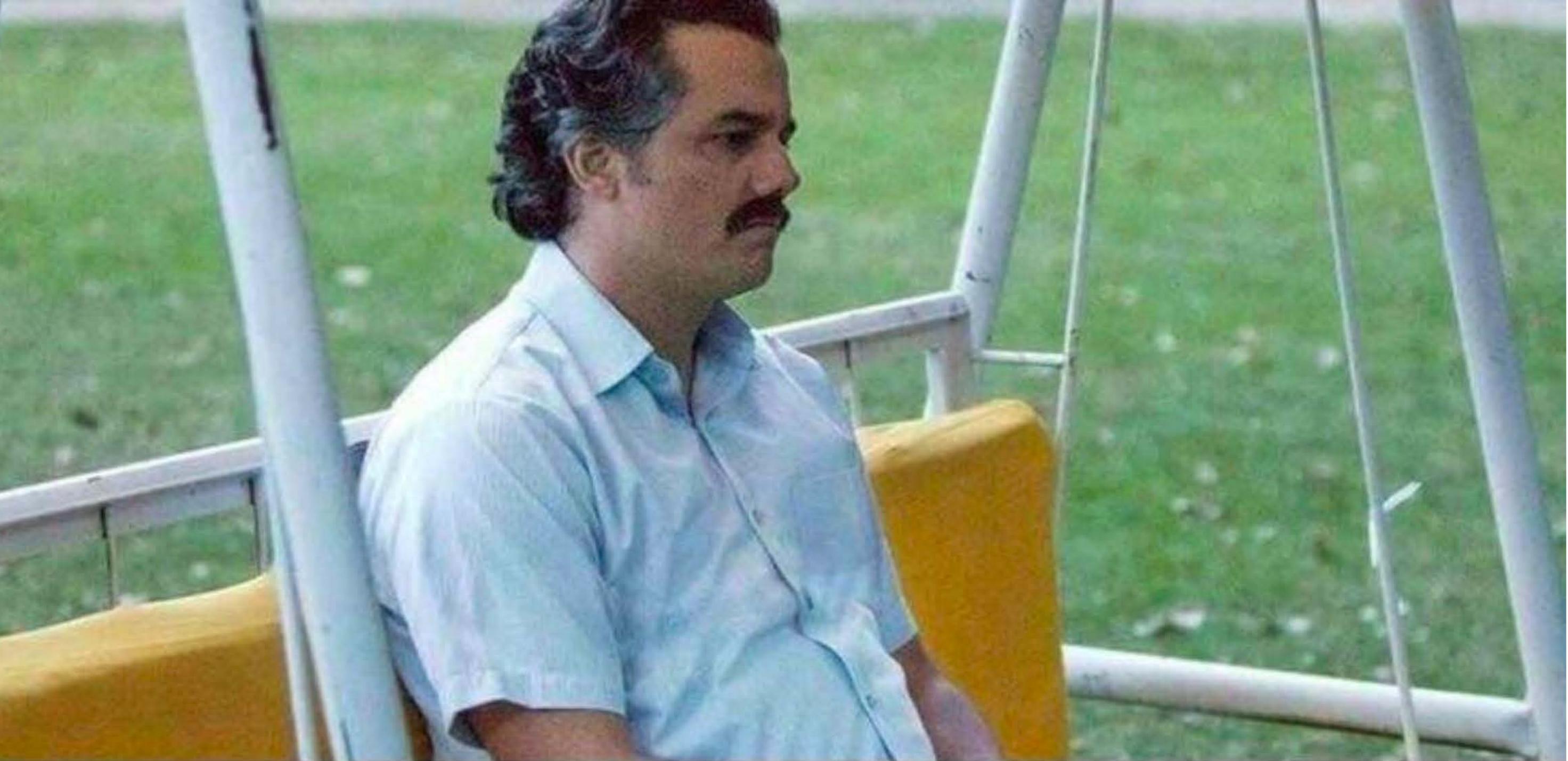
- Mechatronics
- Mechanical Design
- Deep Learning
- Computer Vision
- Your first/second programming, and statistics course
- A replacement for semester long courses in the topics we cover
- **Easy**

Waiting List

Enrolment is restricted to:

48 students

- Hardware/autonomous cars available
- Max capacity of Rice 120



We will use Ubuntu 16.04 LTS on the F1/10 car

Everyone must have access to Ubuntu **16.04 (Xenial)** or **18.04 (Bionic)**

Options:

- Install an Ubuntu virtual machine on any OS.
Windows: VirtualBox
OS X: Parallels (paid software)
- Dual boot your laptop. (careful)

No lab session this week.

Bring your laptop to the lab session next Thursday.

So what is



?

TENTH

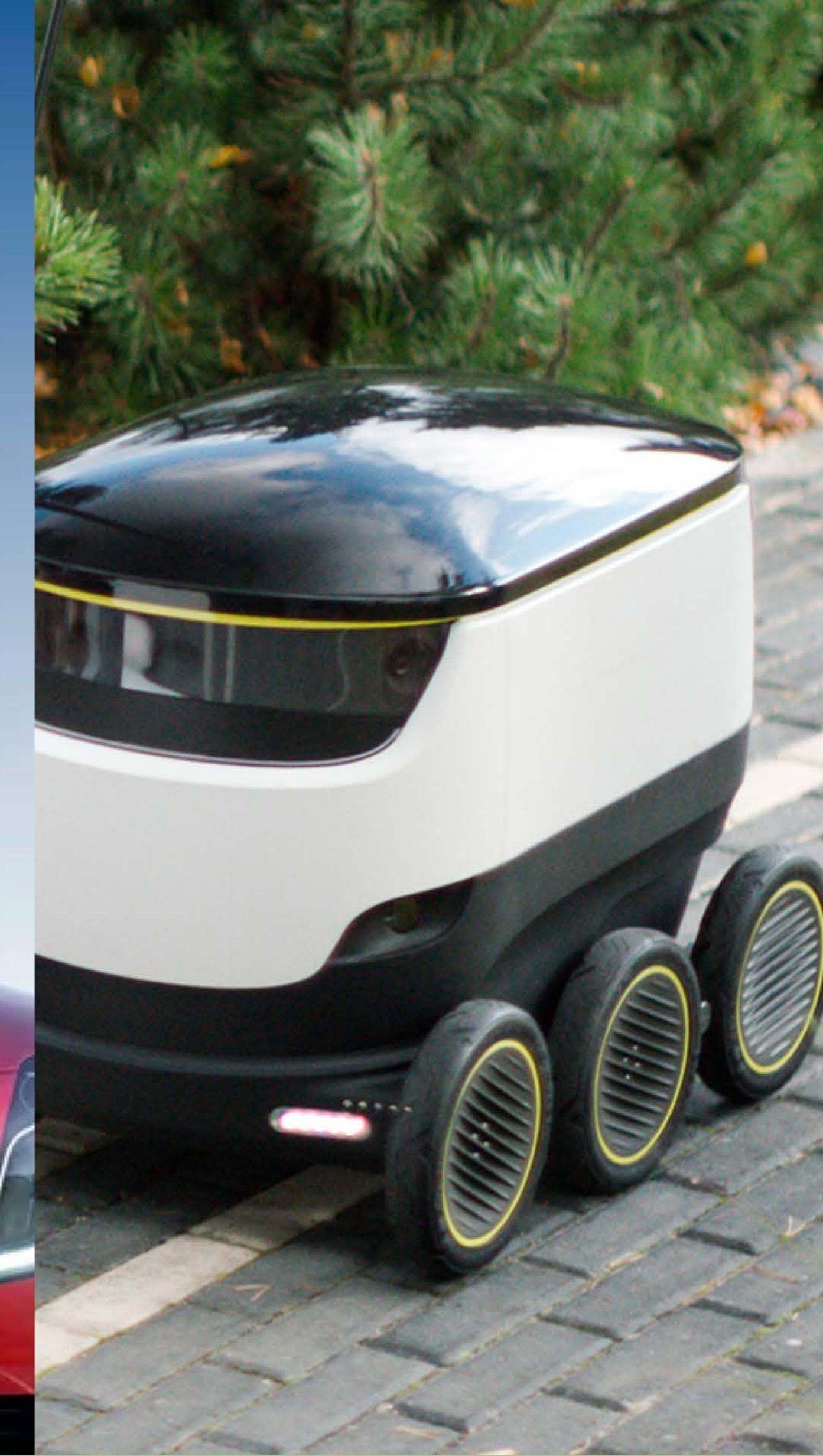
Everything that moves will go autonomous



Cars



Trucks



Carts



Drones



THE 6 LEVELS OF

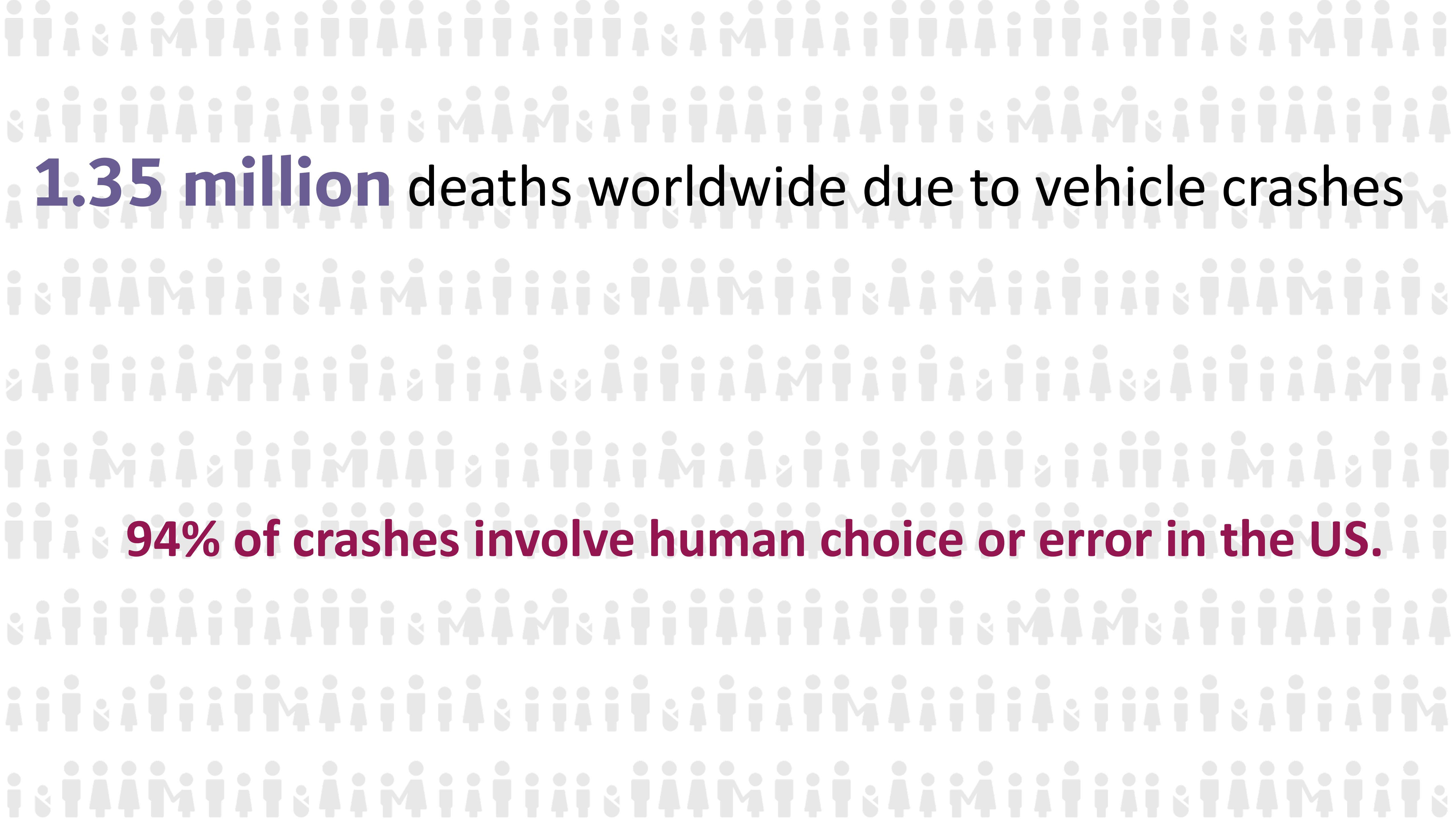
AUTONOMOUS DRIVING

L0 No Automation	L1 Driver Assistance	L2 Partial Automation	L3 Conditional Automation	L4 High Automation	L5 Full Automation
-------------------------------	-----------------------------------	------------------------------------	--	---------------------------------	---------------------------------

SOCIETY OF AUTOMOTIVE ENGINEERS (SAE) AUTOMATION LEVELS

Full Automation

0	1	2	3	4	5
No Automation Zero autonomy; the driver performs all driving tasks.	Driver Assistance Vehicle is controlled by the driver, but some driving assist features may be included in the vehicle design.	Partial Automation Vehicle has combined automated functions, like acceleration and steering, but the driver must remain engaged with the driving task and monitor the environment at all times.	Conditional Automation Driver is a necessity, but is not required to monitor the environment. The driver must be ready to take control of the vehicle at all times with notice.	High Automation The vehicle is capable of performing all driving functions under certain conditions. The driver may have the option to control the vehicle.	Full Automation The vehicle is capable of performing all driving functions under all conditions. The driver may have the option to control the vehicle.



1.35 million deaths worldwide due to vehicle crashes

94% of crashes involve human choice or error in the US.

3 million

Americans age 40 and older are blind or have low vision

79%

of seniors age 65 and older living in car-dependent communities

42 hours

wasted in traffic each year per person

Localization and Mapping

Where am I ?

Scene Understanding

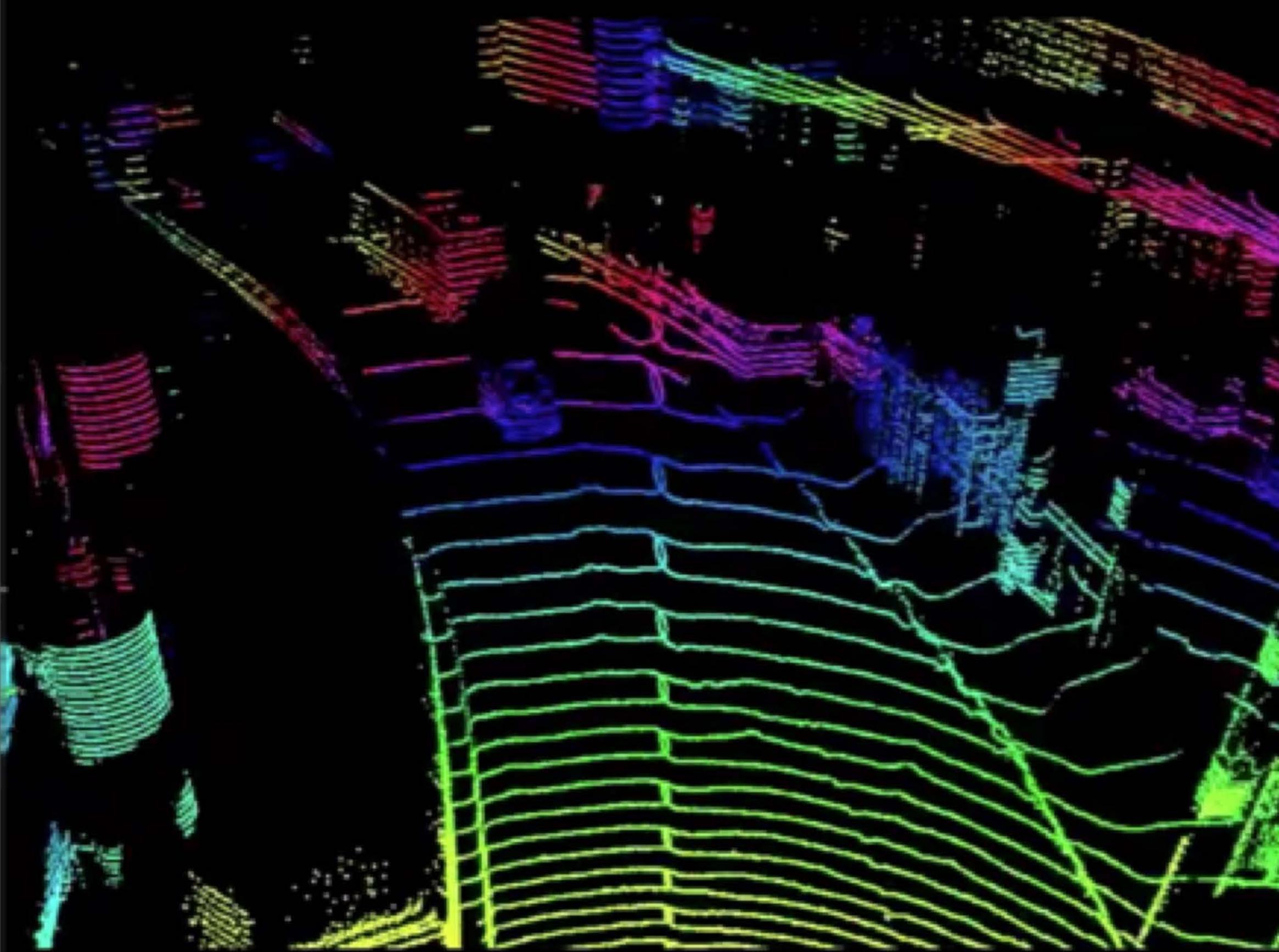
Where/who/what/why of everyone/everything else ?

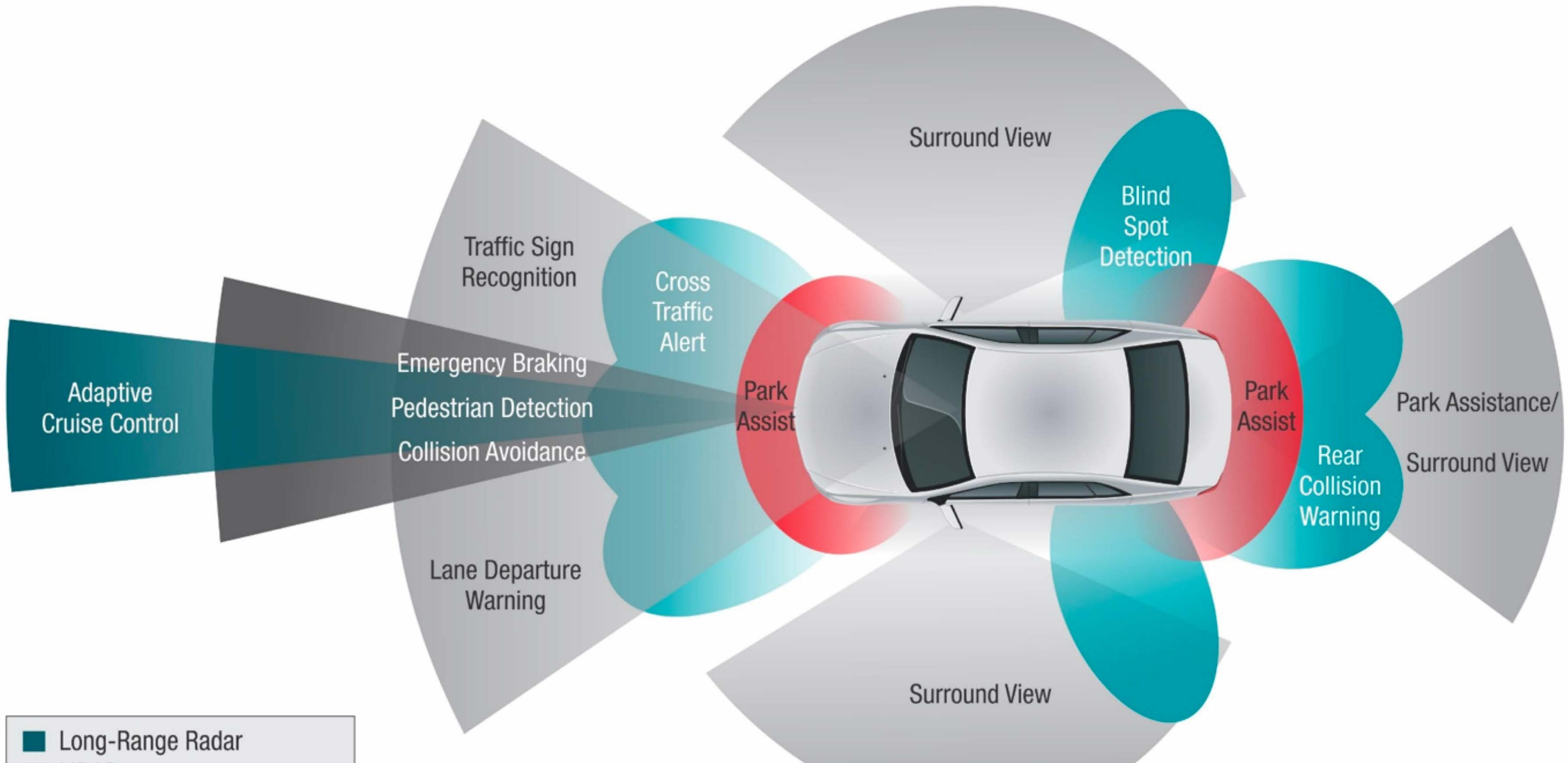
Trajectory Planning and Control

Where should I go next ?
How do I steer and accelerate ?

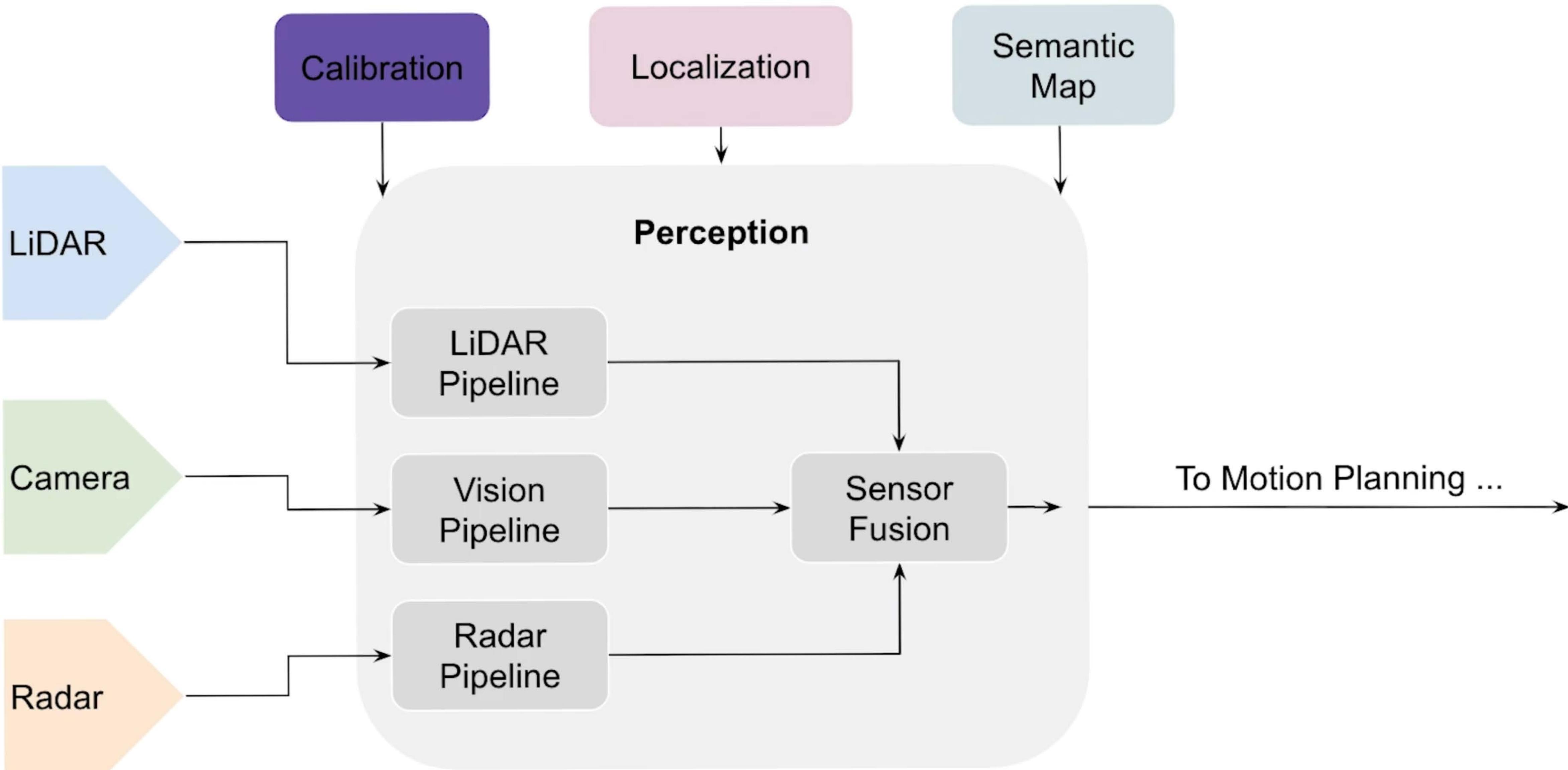
Human Interaction

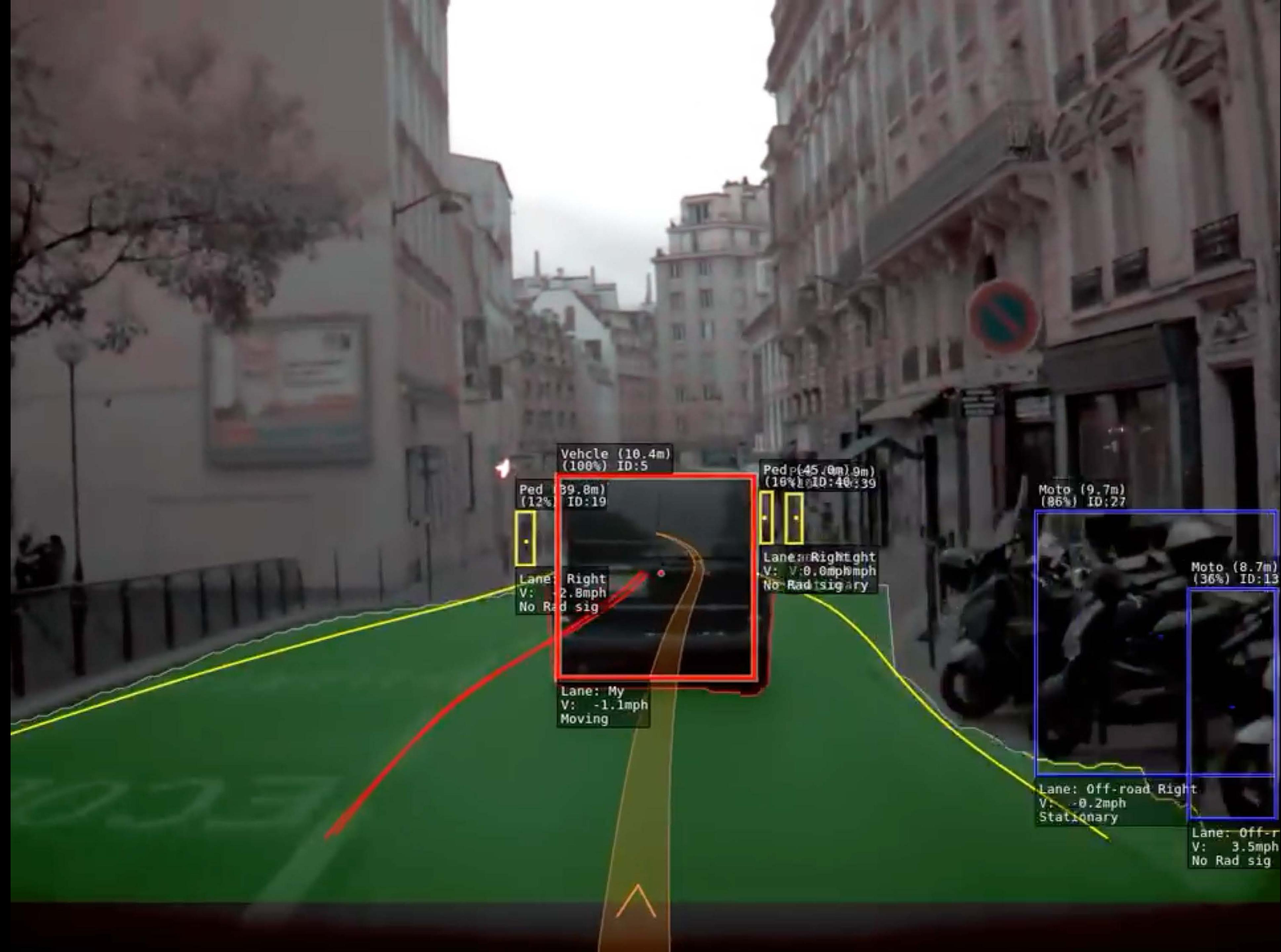
How do I convey my intent to the passenger and everyone else ?





Perception in AV Stack







Where
am I?

Detailed three-dimensional maps that highlight information such as road profiles, curbs and sidewalks, lane markers, crosswalks, traffic lights, stop signs, and other road features.



Scan constantly for objects around the vehicle—pedestrians, cyclists, vehicles, road work, obstructions—and continuously read traffic controls, from traffic light color and railroad crossing gates to temporary stop signs.

**What's
around me?**



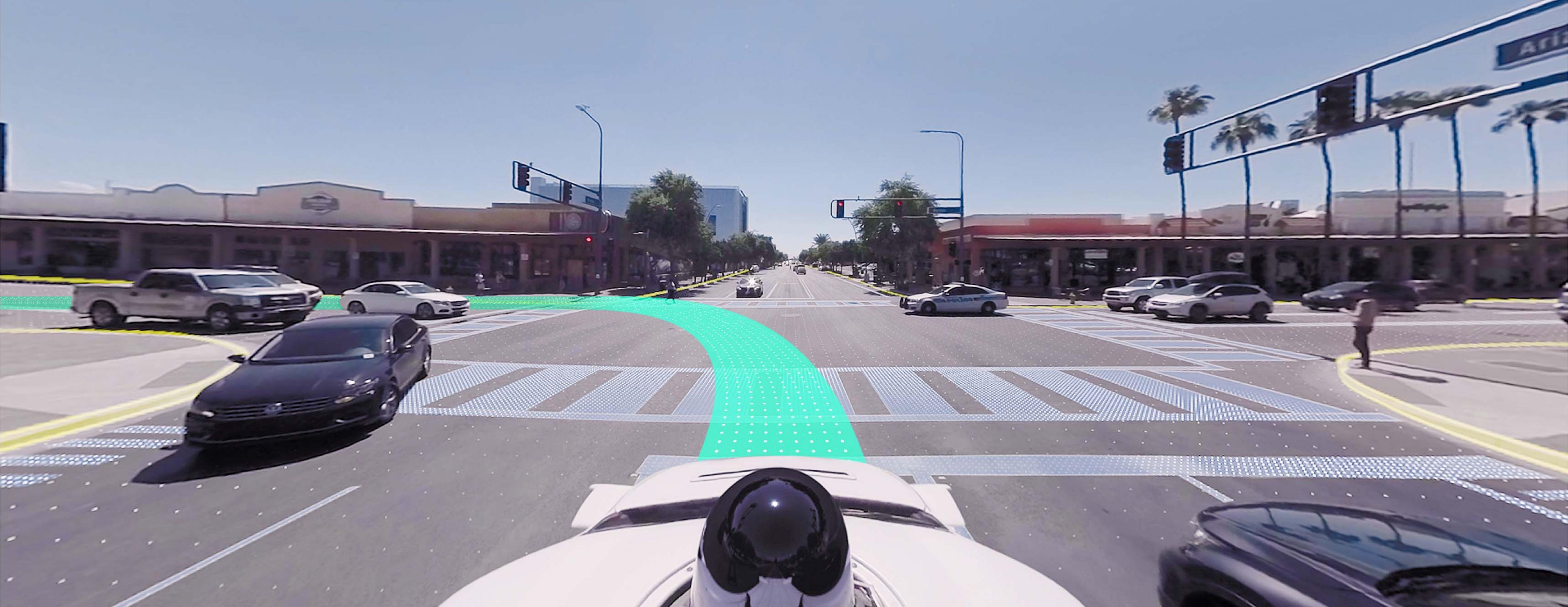
Predict the movements of everything around you
based on their speed and trajectory

What will
happen next?



Determine the exact trajectory, speed, lane, and steering maneuvers needed to progress along the route safely

What should I do?

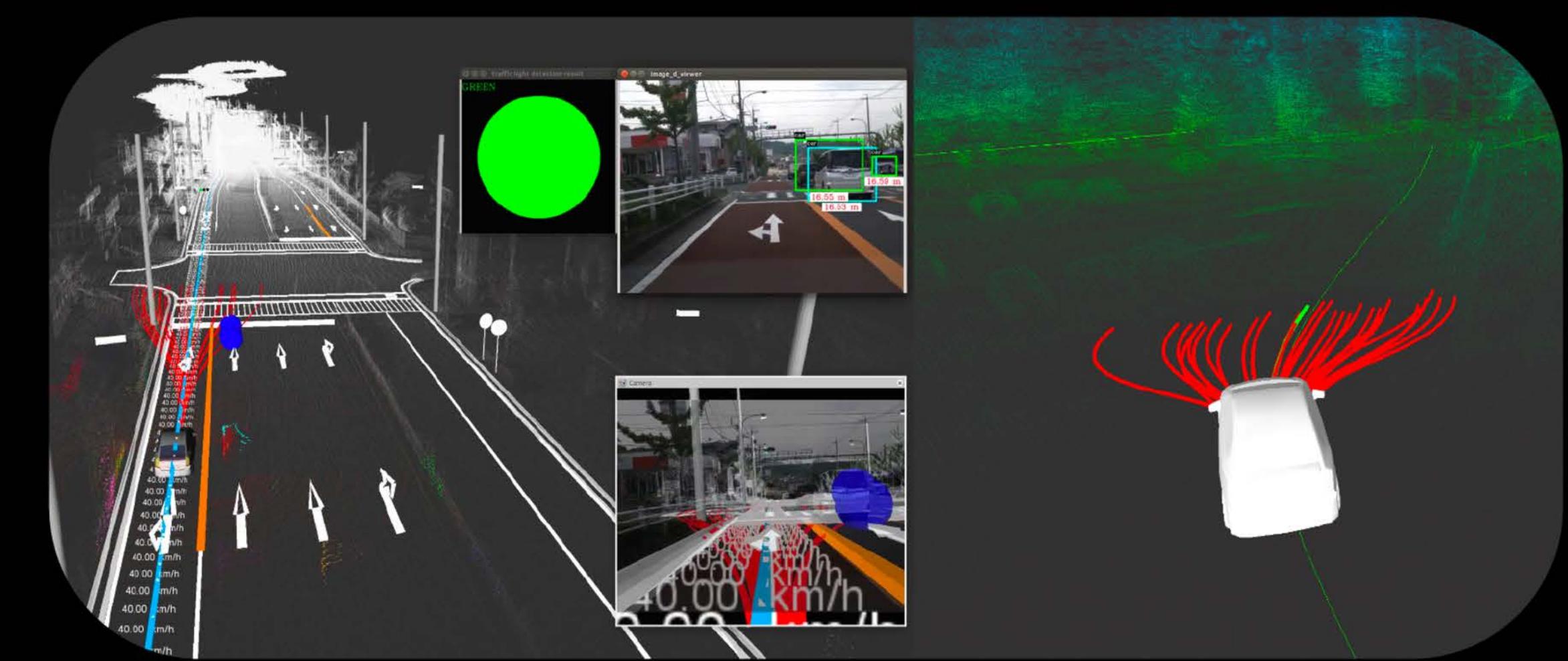
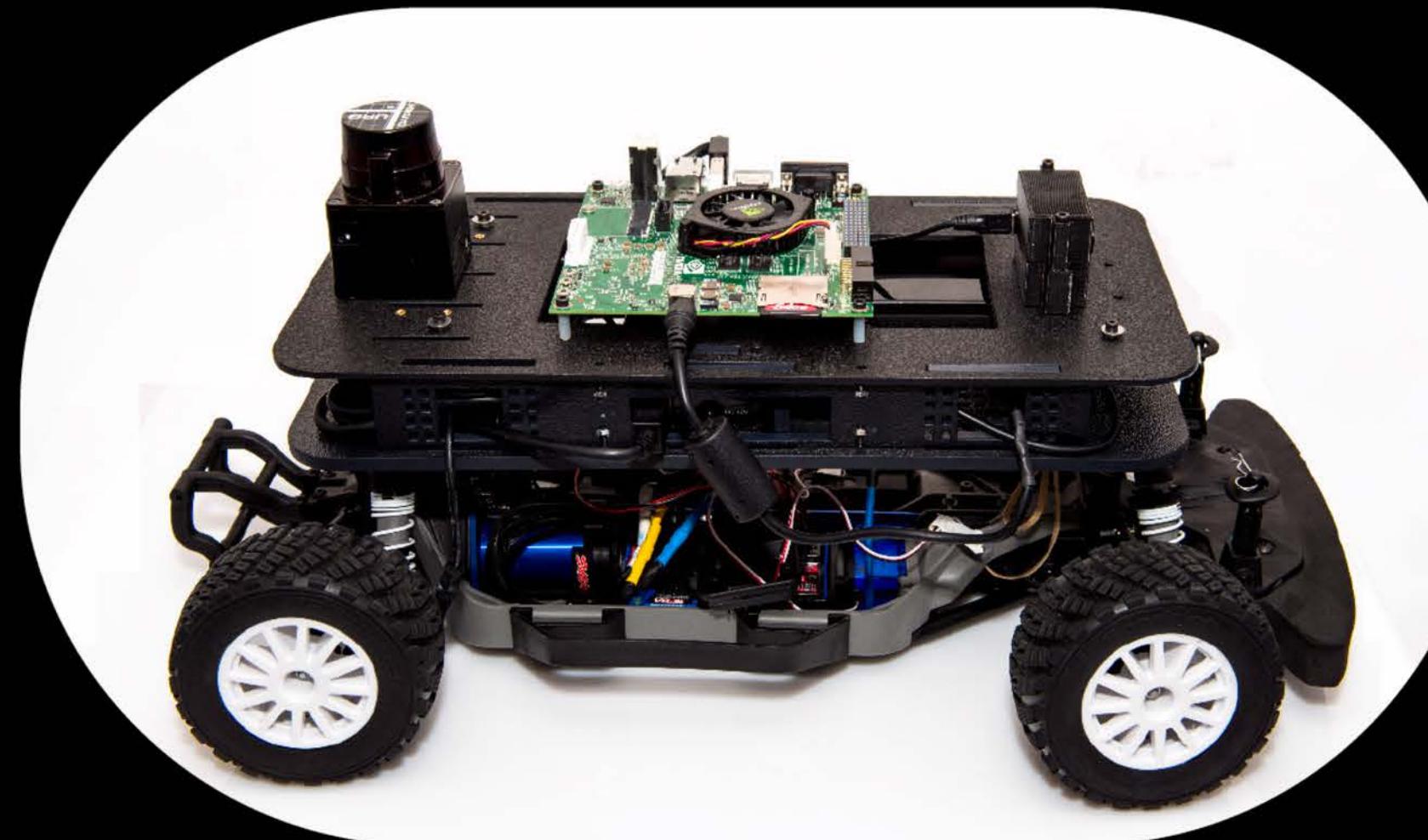


F1/10

Autonomous Racing

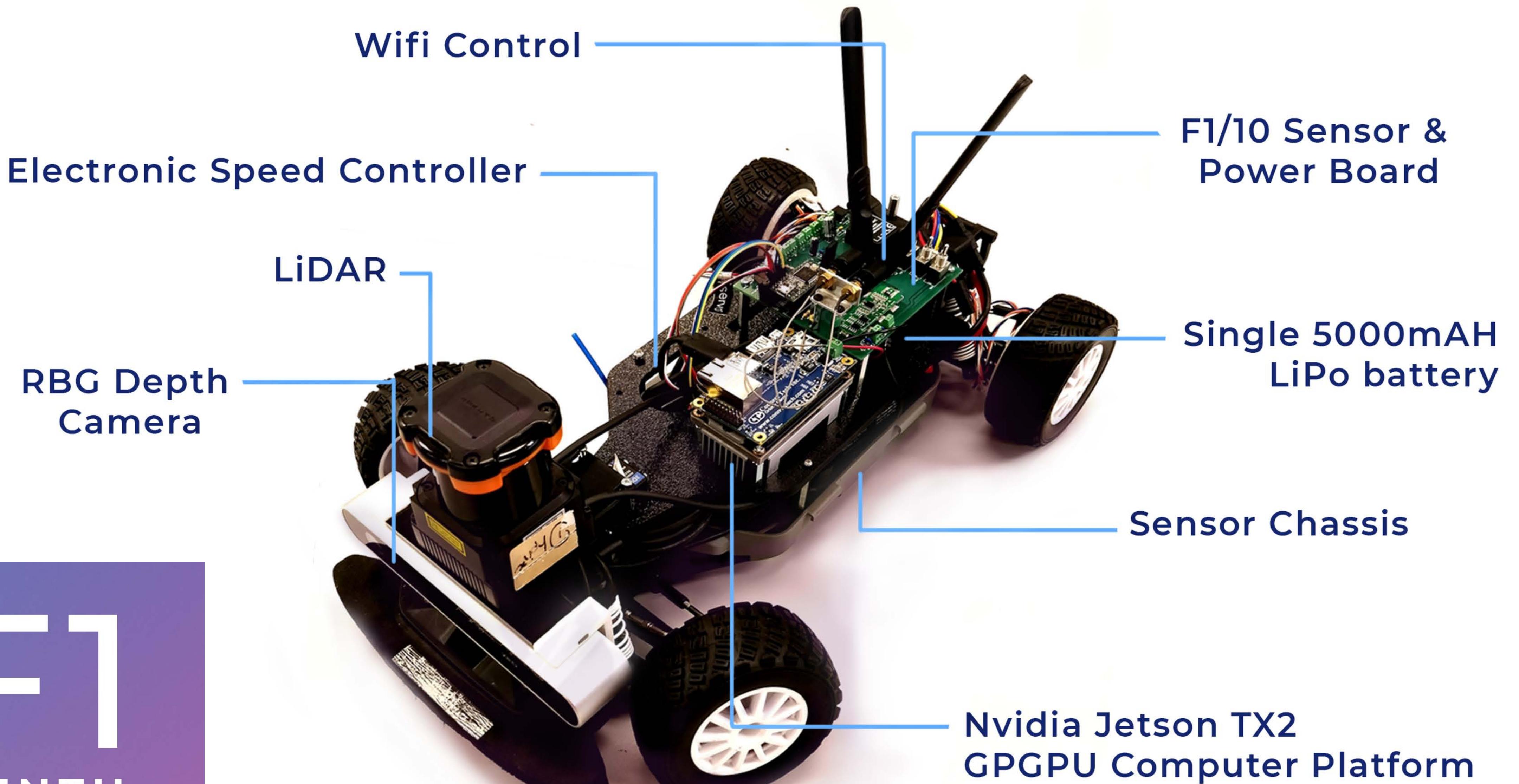
1/10 the scale. 10 times the fun!

Build. Drive. Race.



Perception. Planning. Control

Build



F1
TENTH

f1tenth.org

Similar dynamics, different parameters

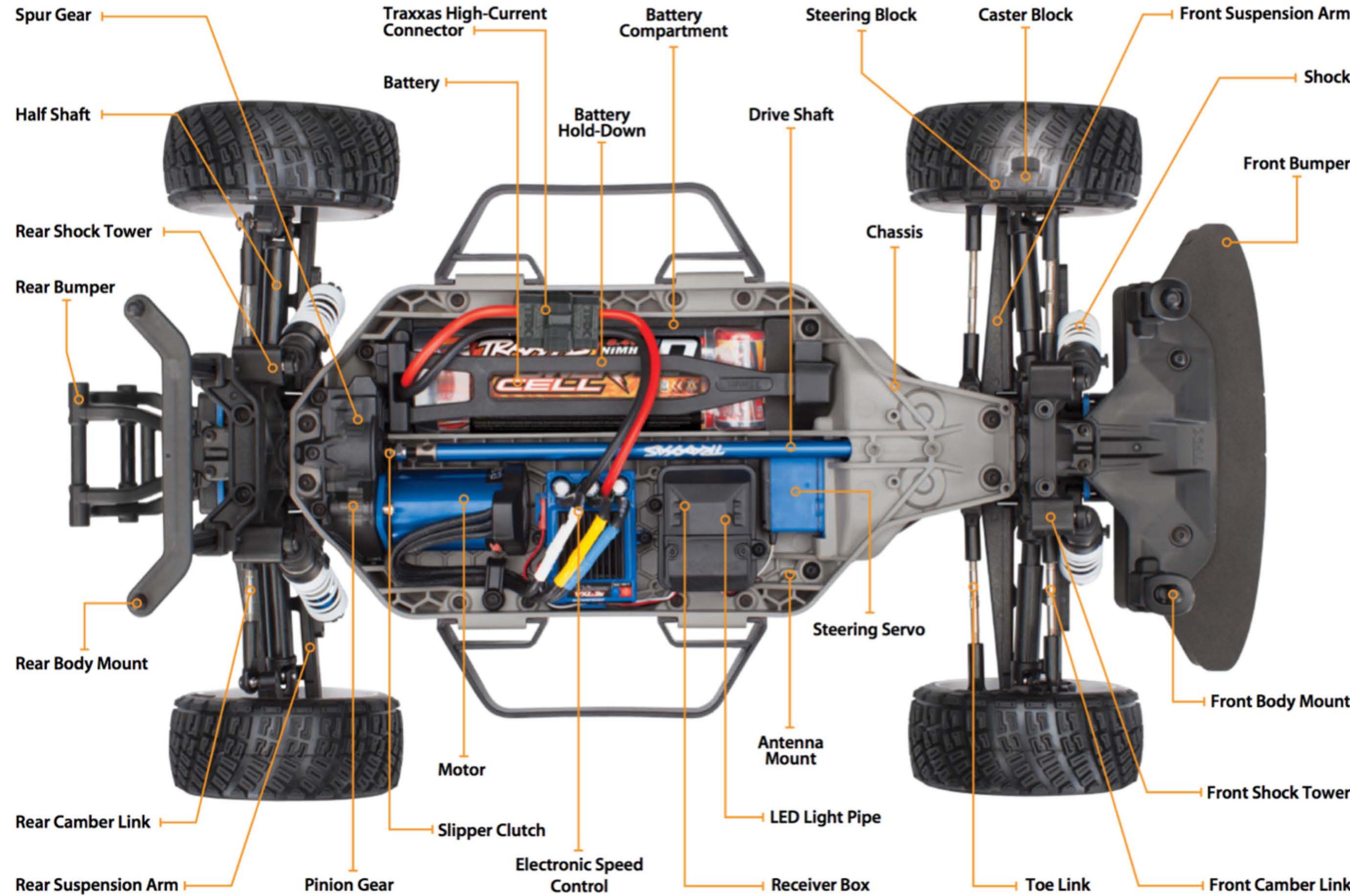
TRAXXAS XO-1 vs Tesla Model S



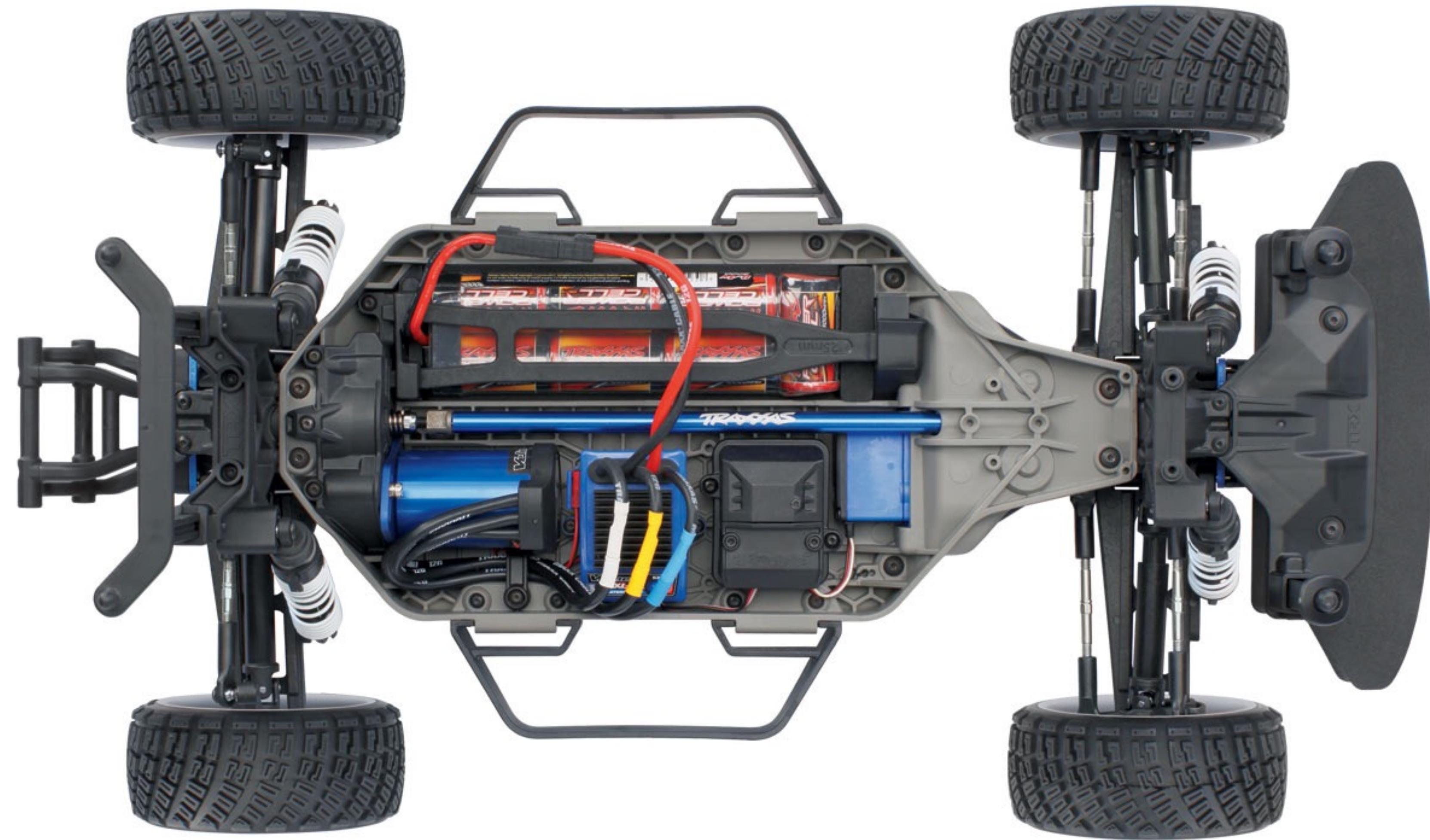
Similar sensors,
different scale



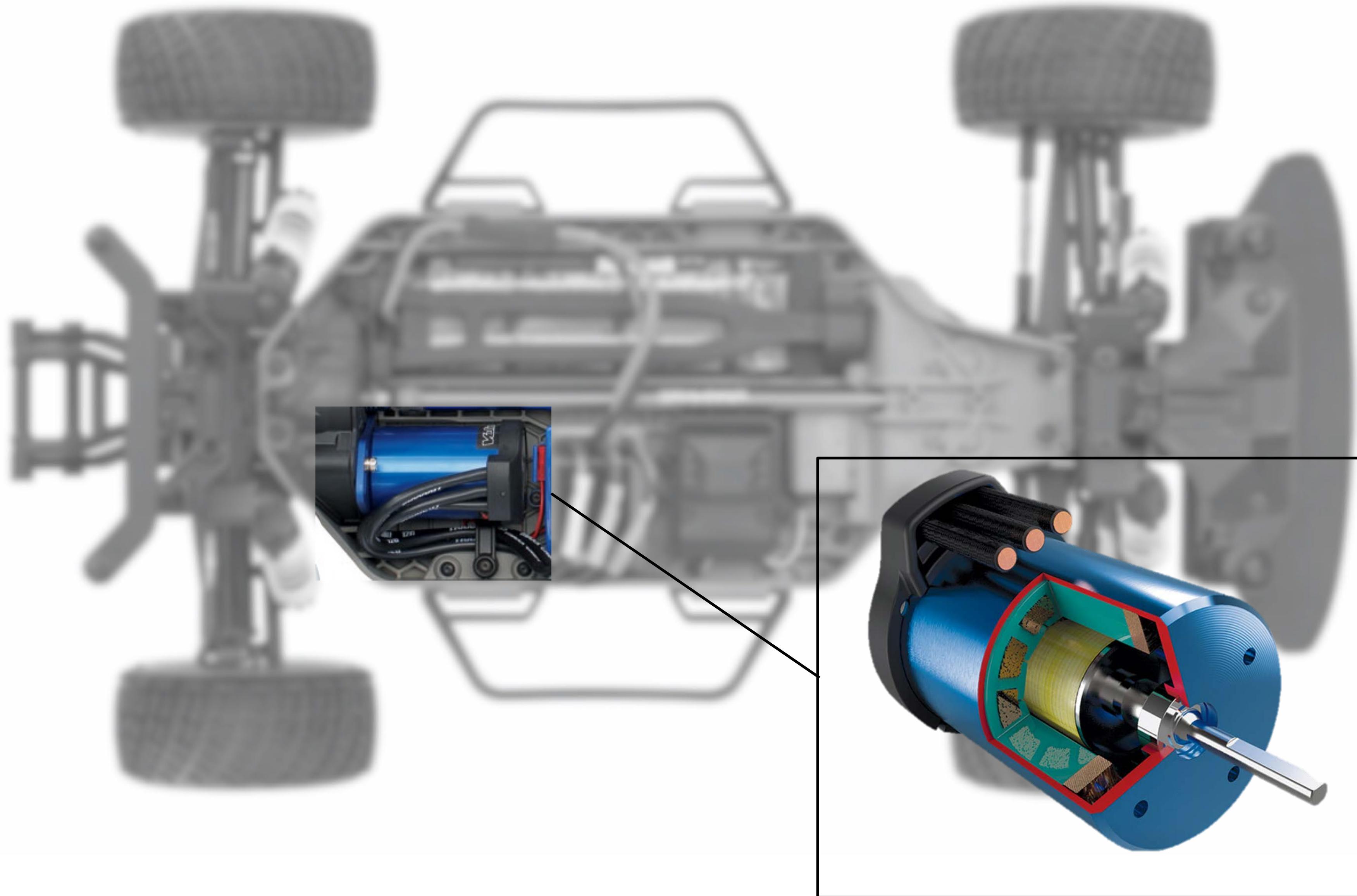
Traxxas 1/10 scale RC race car



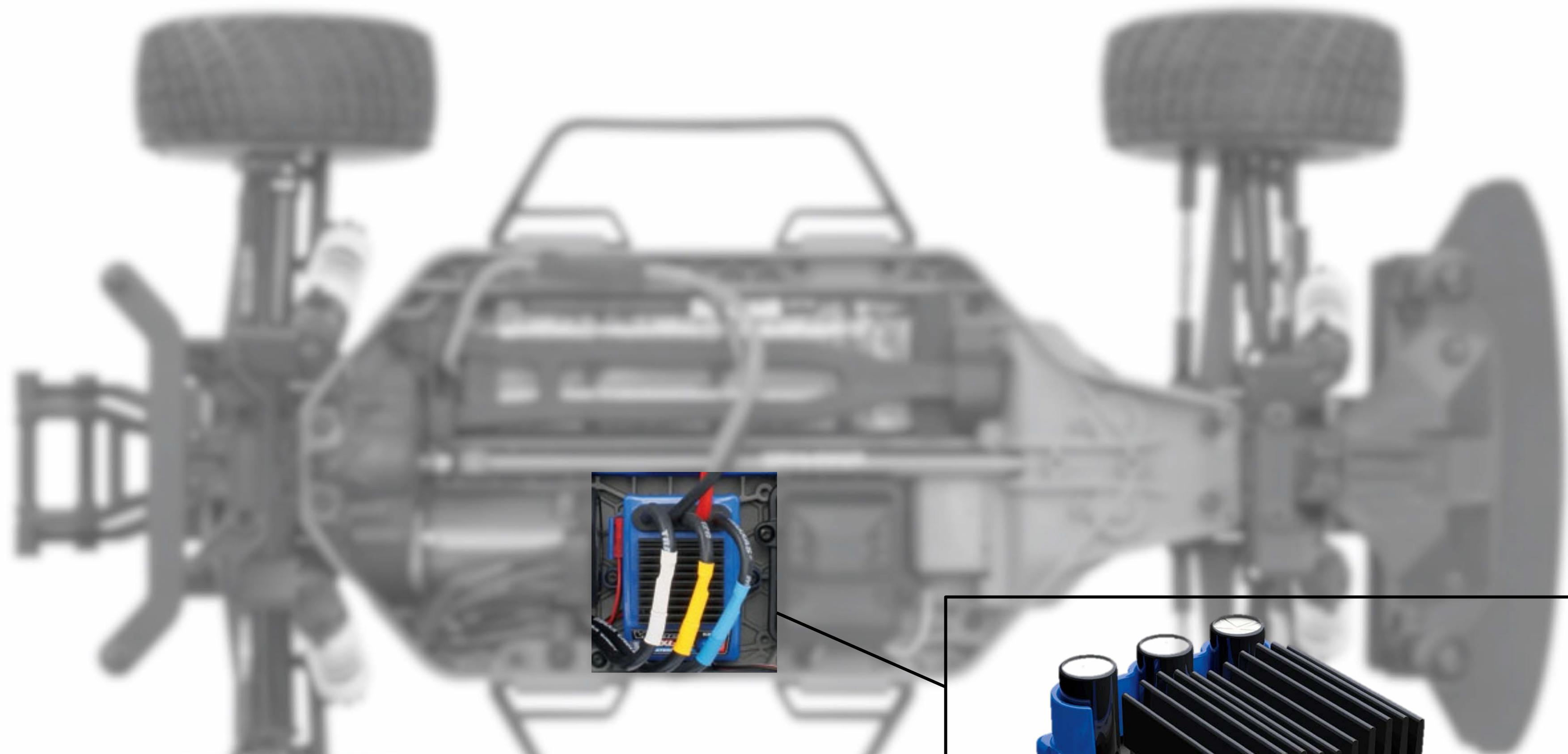
Traxxas 1/10 scale RC race car



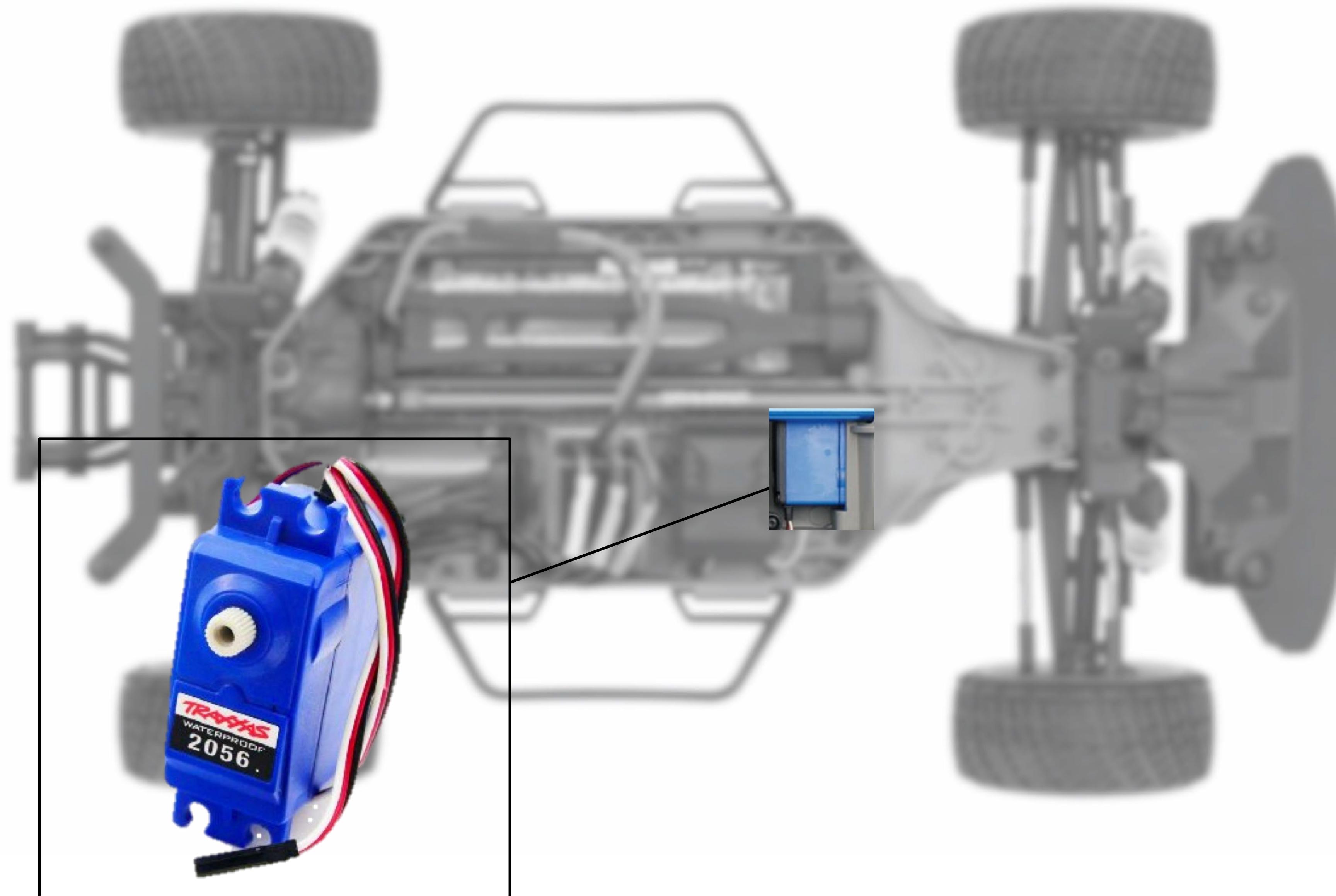
Brushless DC motor



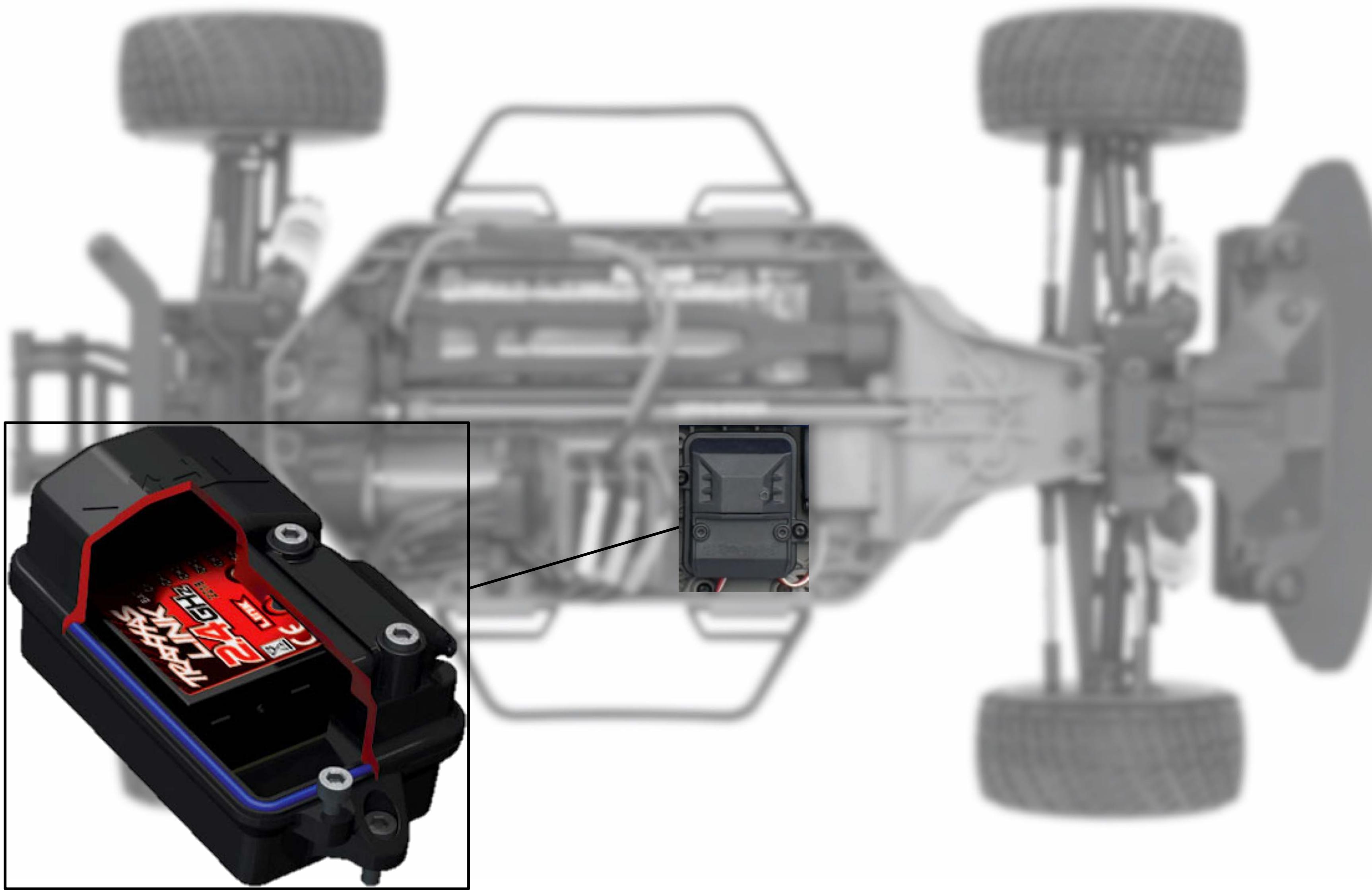
Electronic Speed Control (ESC)



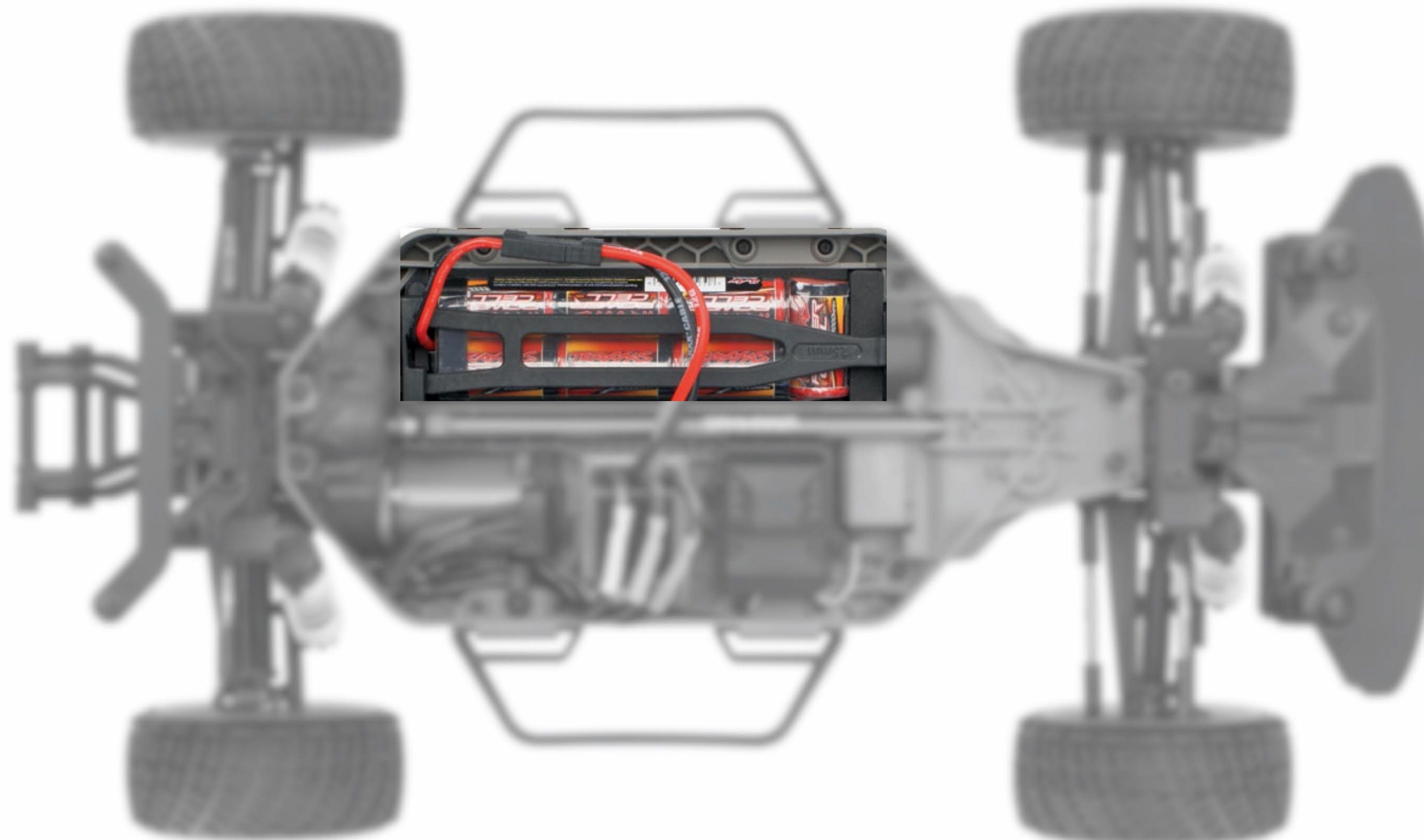
Servo motor for steering



Radio receiver



Battery pack



Sensor Integration



LiDAR



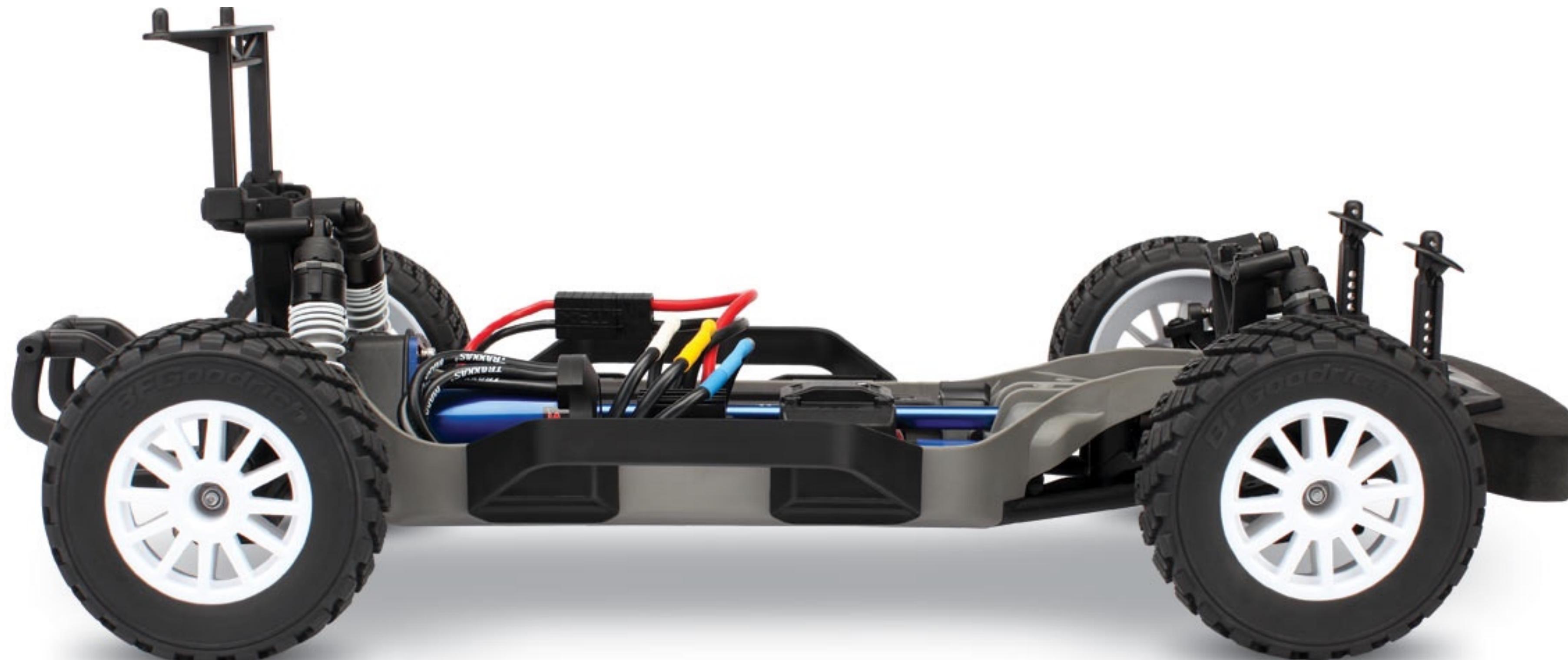
Camera



IMU



IR Depth Cameras



Sensor Integration



LiDAR



Camera



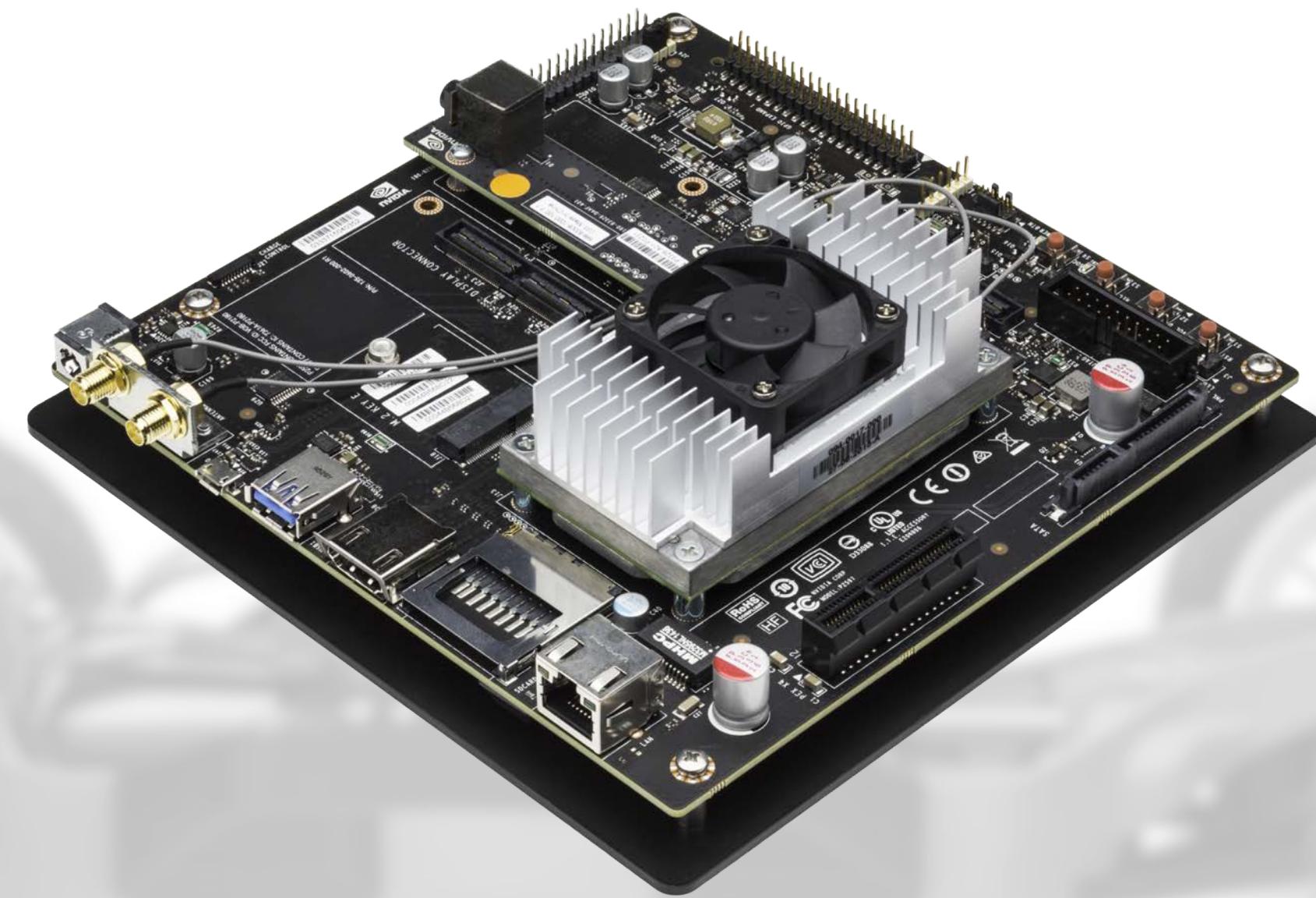
IMU



IR Depth Cameras



Wi-fi Telemetry

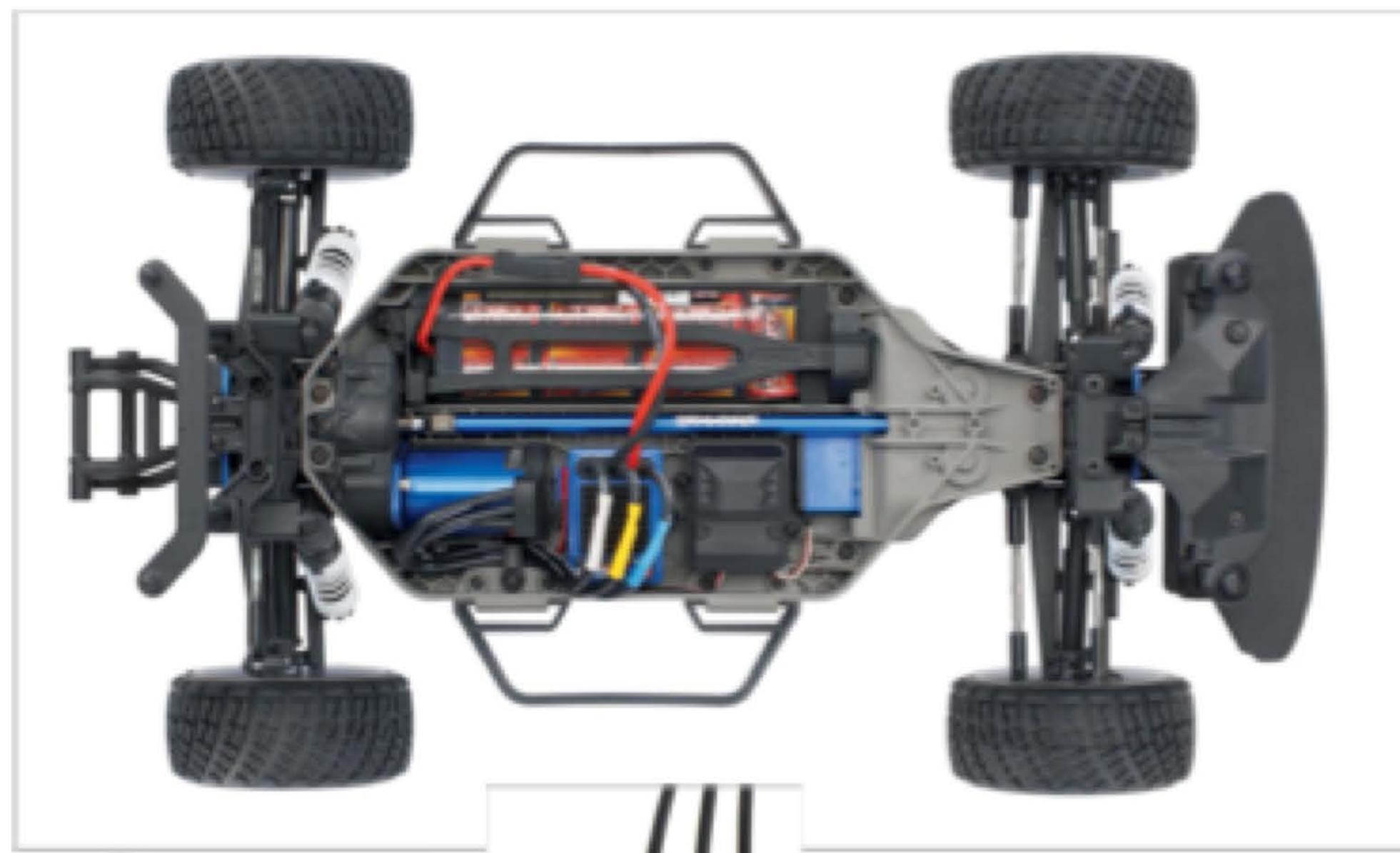


Onboard Computer



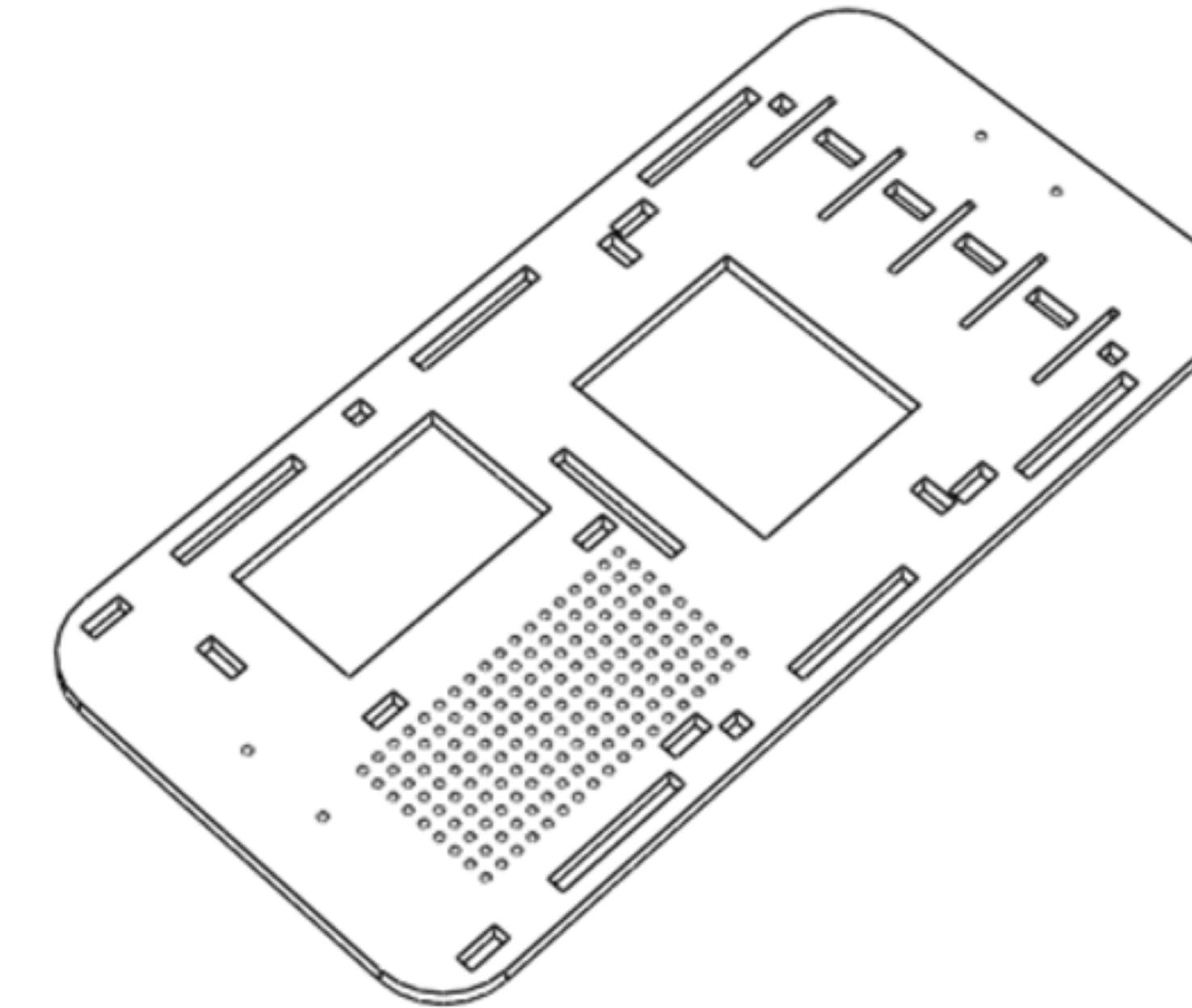
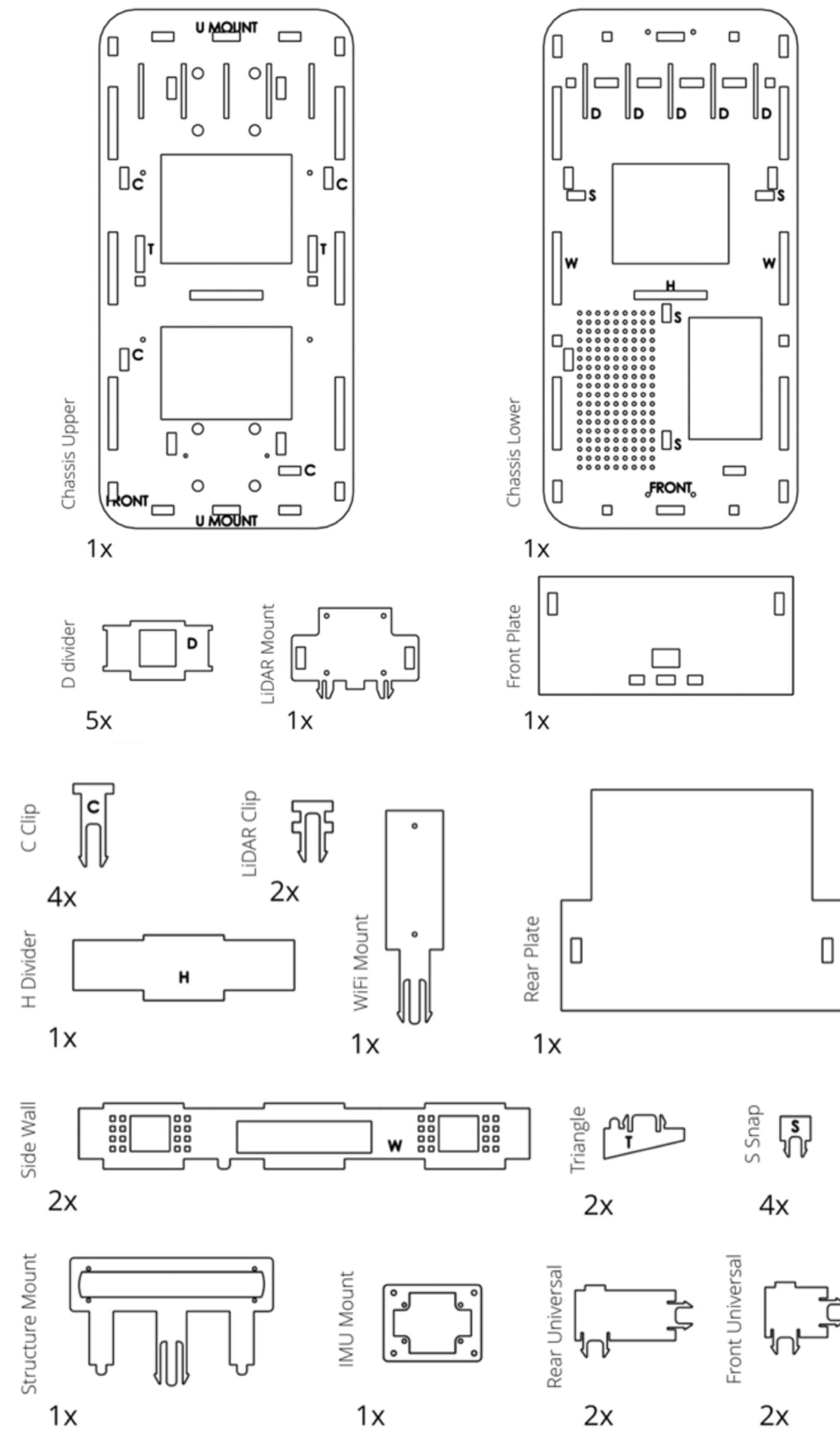
Motor Controller

Fastware



Approximately \$2,700

With simple IKEA style instructions:



F1/10 Race Car Assembly [Time Lapse]

Electronic Speed Control

Problem:

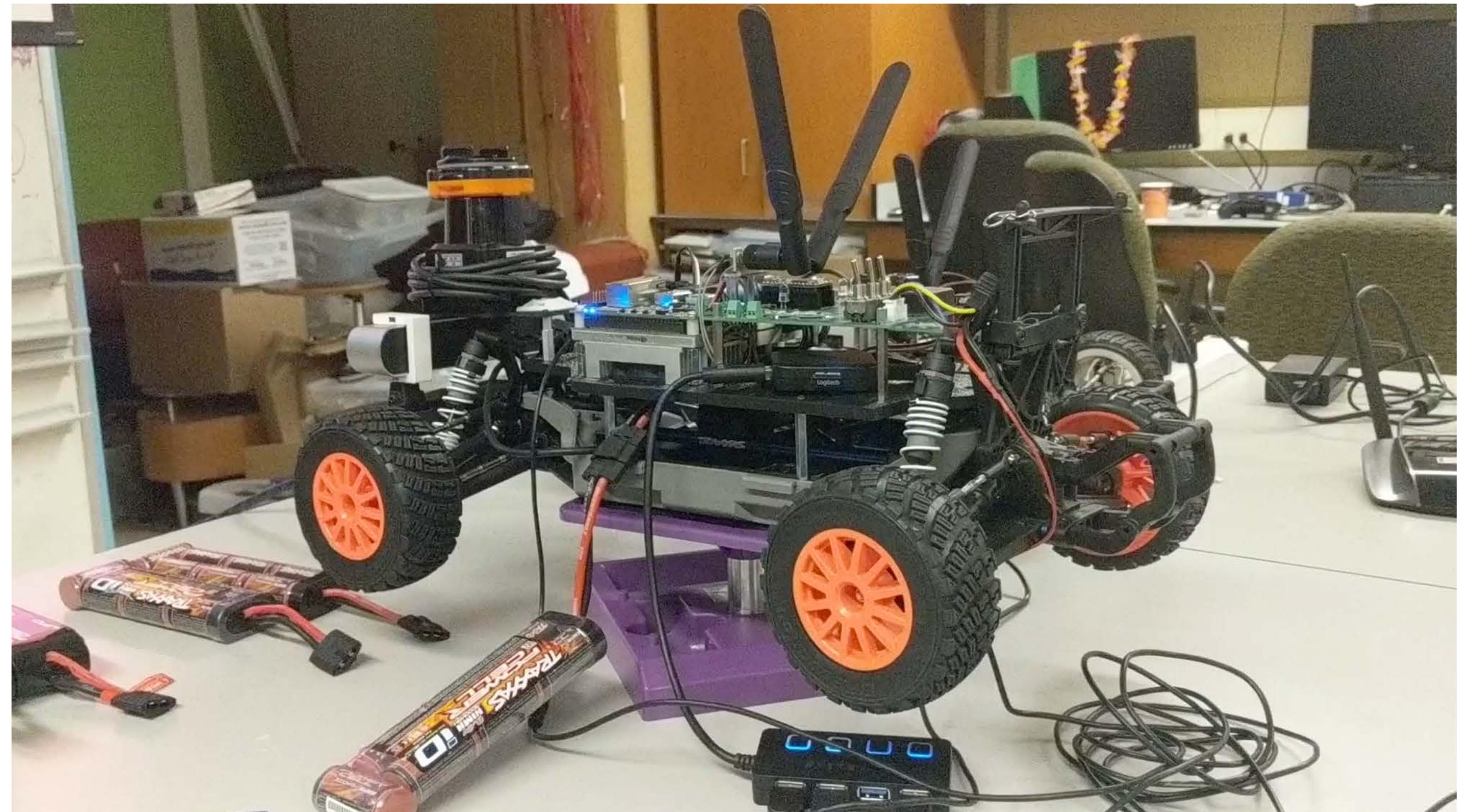
Actuate vehicle via commands in physical units

Understand:

PID, motor control, etc.

Implement:

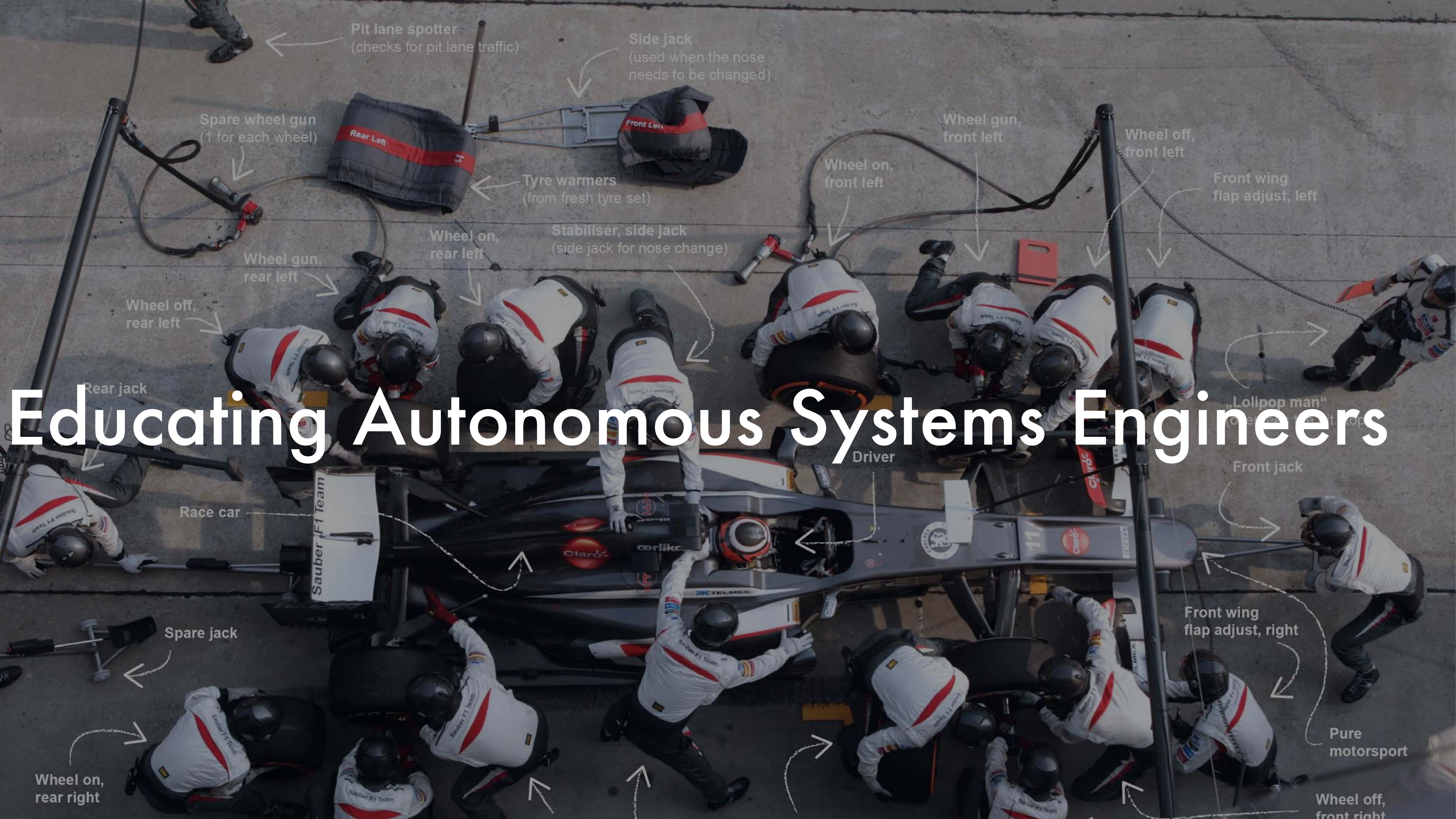
Motor controller, encoder, PID.



Drive

Perception. Planning. Control

Educating Autonomous Systems Engineers





Mapping and Localization

Control and Navigation

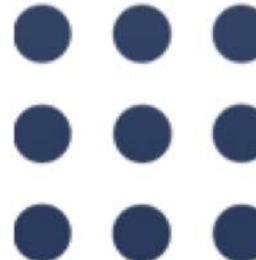
Perception and Sensing Engineer



Future racing teams will look like this

ROS: Robot Operating System

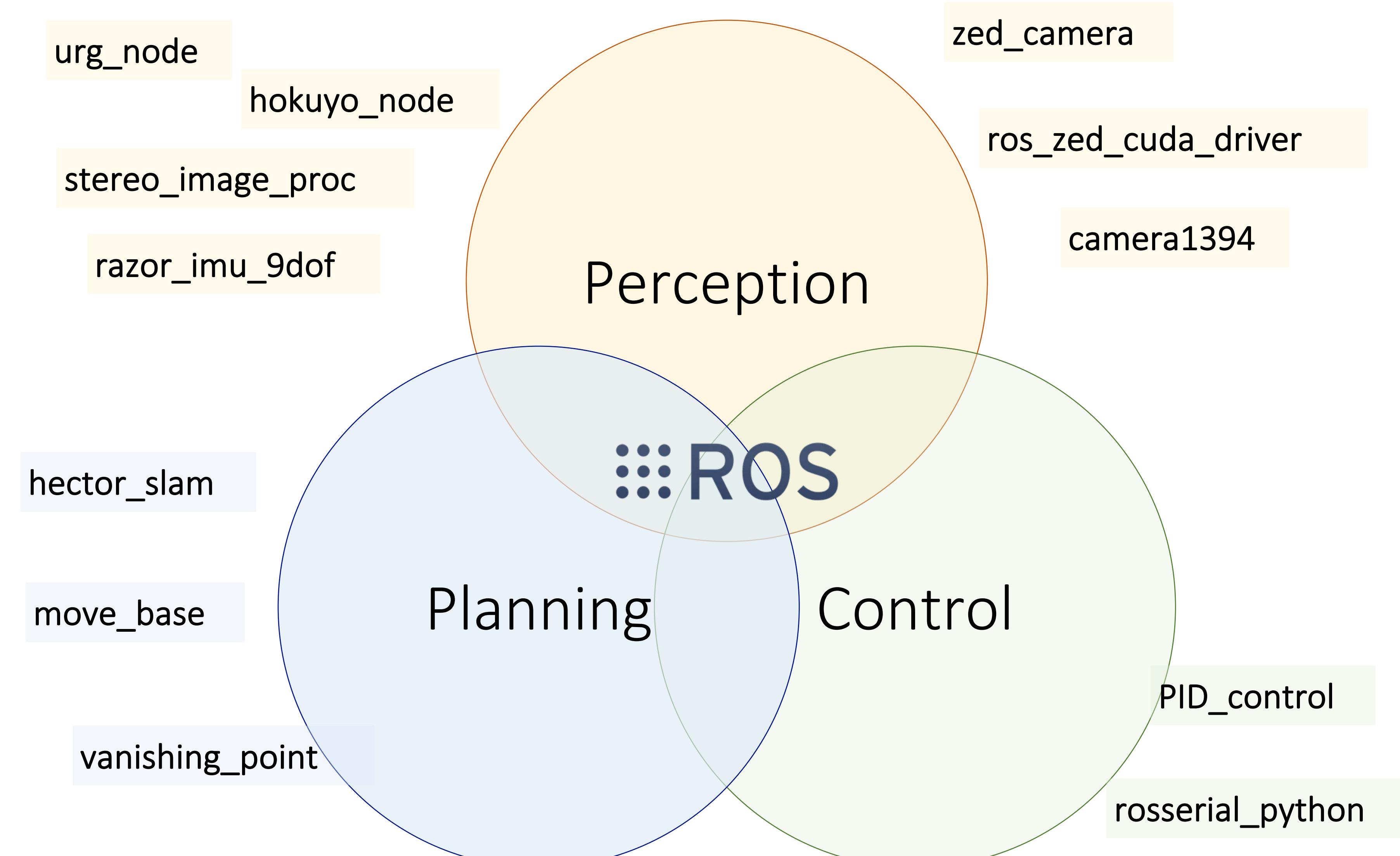
The screenshot shows the official ROS website. At the top, there's a navigation bar with links for "About", "Why ROS?", "Getting Started", "Get Involved", and "Blog". The main content area features a large image of a white PR2 robot. To the left of the robot is a dark box containing the title "What is ROS?" and a brief description of what ROS is. Below this is a "Read More" button. To the right of the robot is a block of Python code demonstrating ROS message publishing. At the bottom of the page, there's a horizontal navigation bar with several small circular icons.

 **ROS.org**



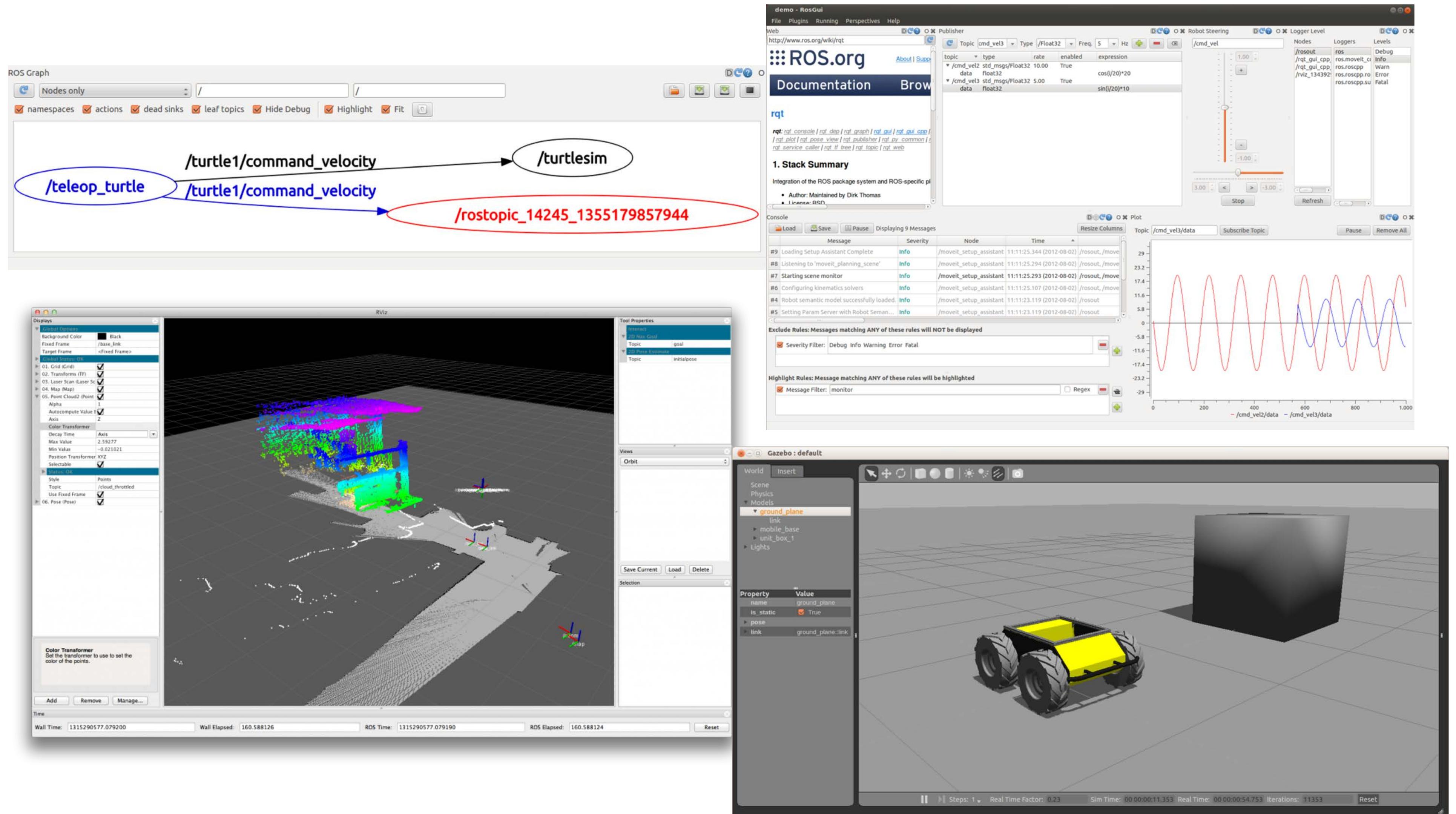
Open Source Robotics Foundation

ROS Capabilities



ROS Tools

Visualization, debugging and diagnostics, logging, and simulation



ROS: Topics

Topics are channels over which nodes exchange messages.

They are for **streaming data**

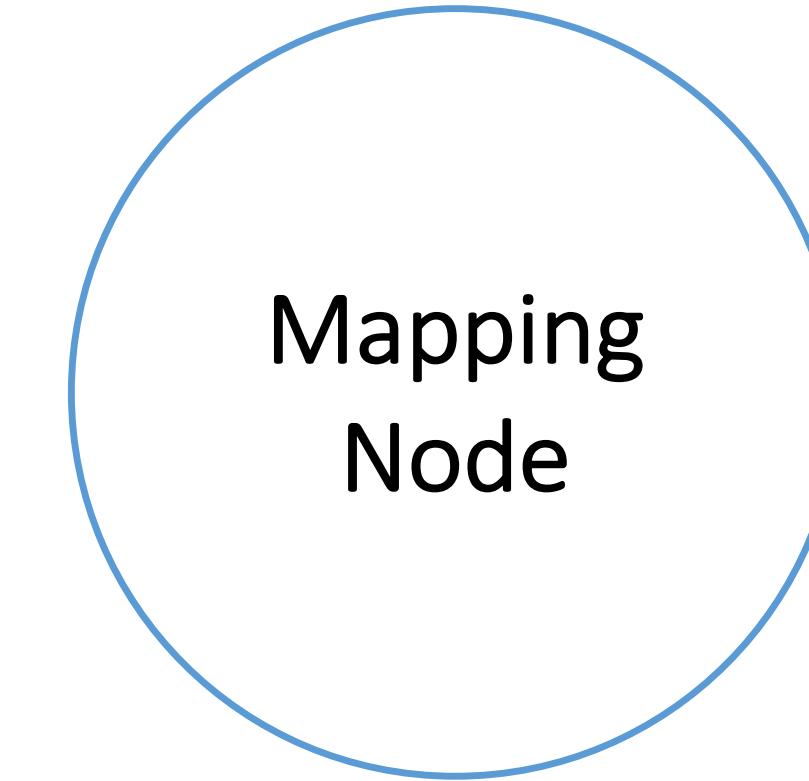


`hokuyo_node`

Publishes on topic: Scan



Scan [Topic]



Subscribes to topic: Scan



The entire course is organized as a long Formula 1 race weekend

Lab exercises



Practice Session 1:
Understanding ROS | Setting up the Car



Practice Session 2:
Perception/Sensing | Driving Straight



Practice Session 3:
Driving in a loop | Visual Odometry

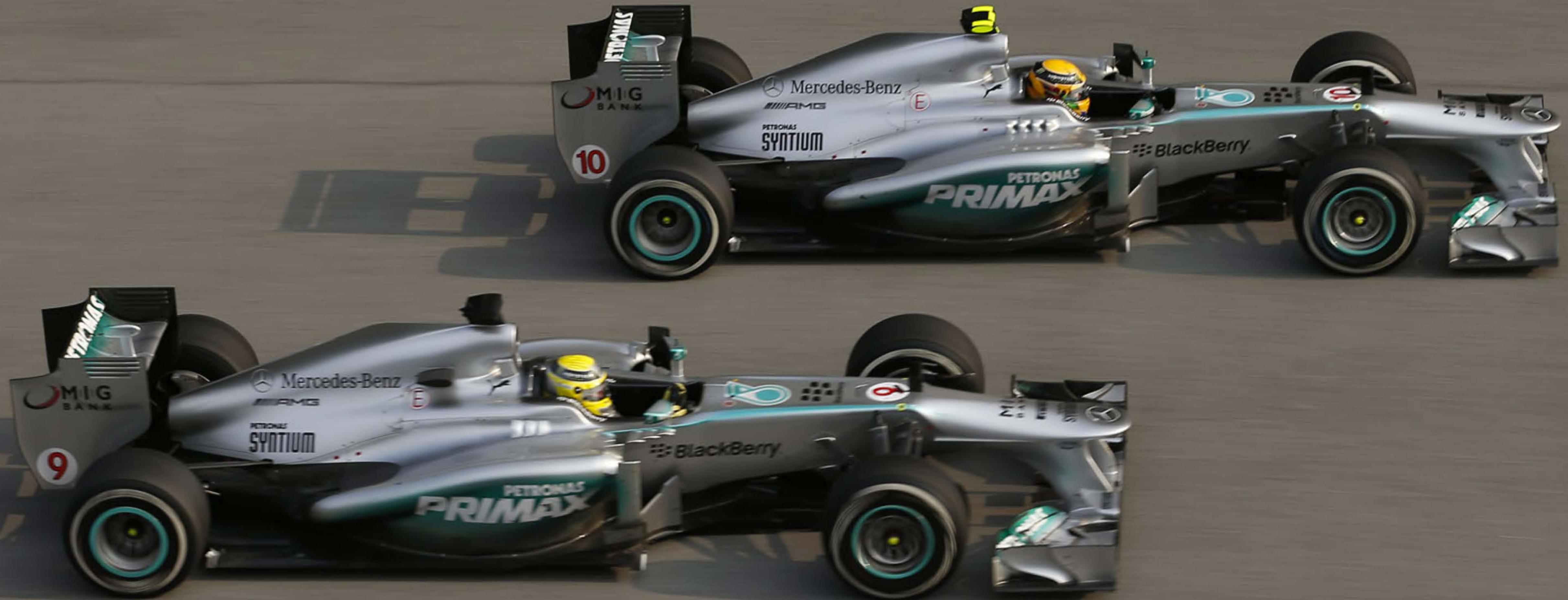


Practice Session 3:
Race | Time trials | Head-to-head racing



Race

All teams have similar cars, with the same sensors



Battle of algorithms

```
optimizer = tf.train.GradientDescentOptimizer(0.5)
```

```
train = optimizer.minimize(loss)
```

```
# Before starting, initialize the variables. We will 'run' this first.
```

```
init = tf.initialize_all_variables()
```

```
# Launch the graph.
```

```
sess = tf.Session()
```

```
sess.run(init)
```

```
# Fit the line.
```

```
for step in range(201):
```

```
    sess.run(train)
```

```
    if step % 20 == 0:
```



1st F1/10 Race: Oct 2016, Pittsburg, ES-Week



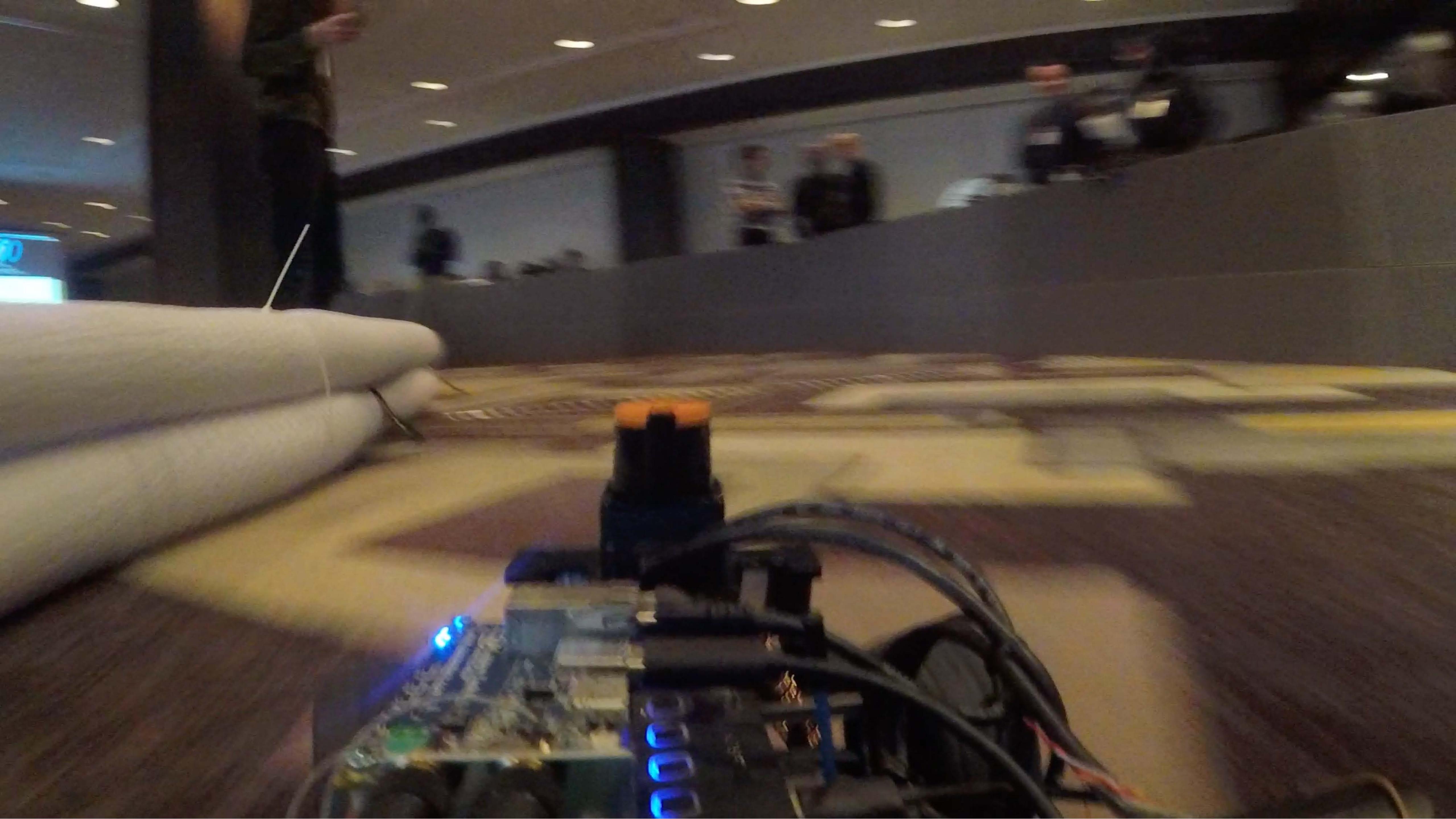
3rd F1/10 Race: Oct 2018, Torino, ES-Week



2nd F1/10 Race: Apr 2018, Porto, CPS-Week



4th F1/10 Race: Apr 2019, Montreal, CPS-Week



Head-to-head Autonomous Racing

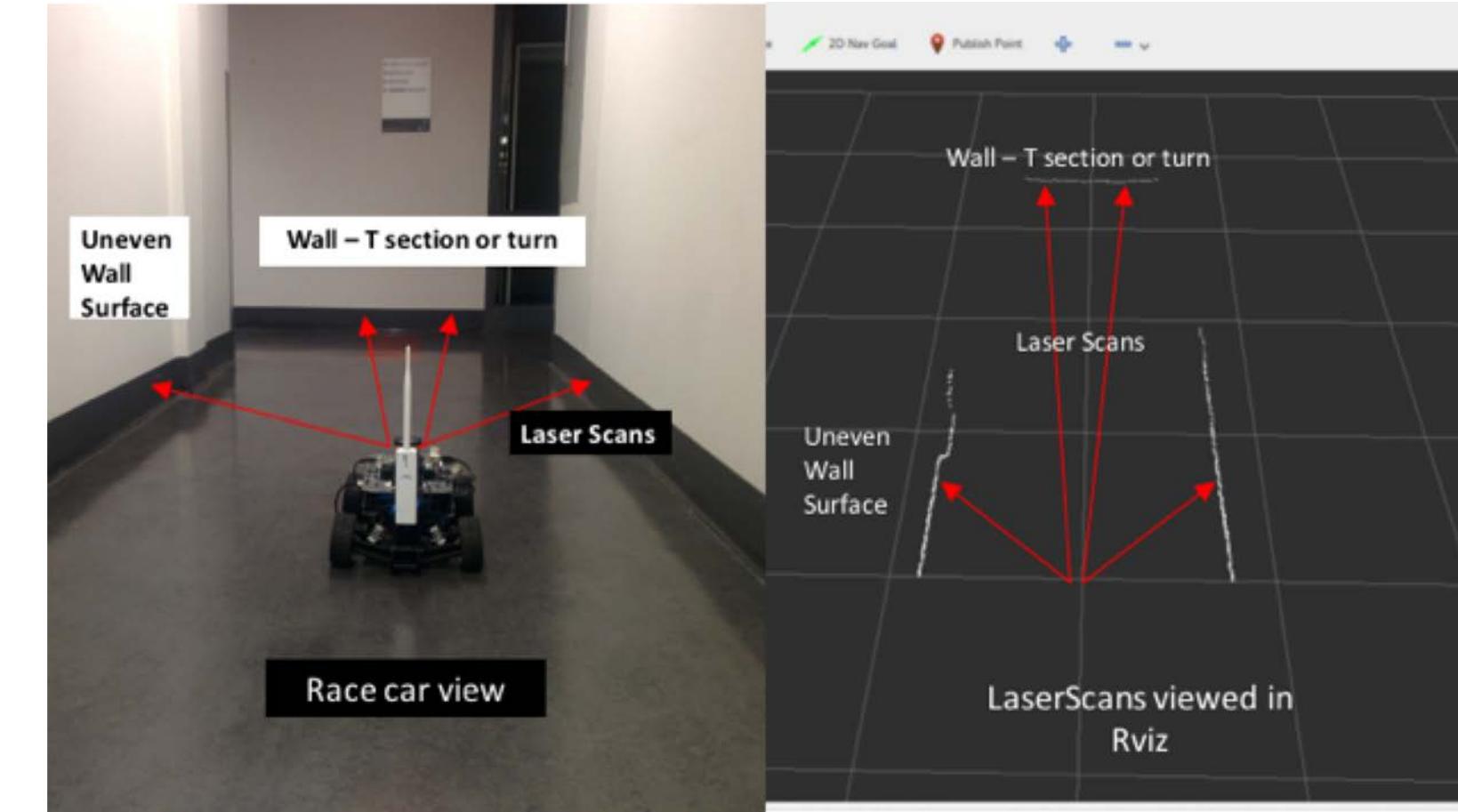
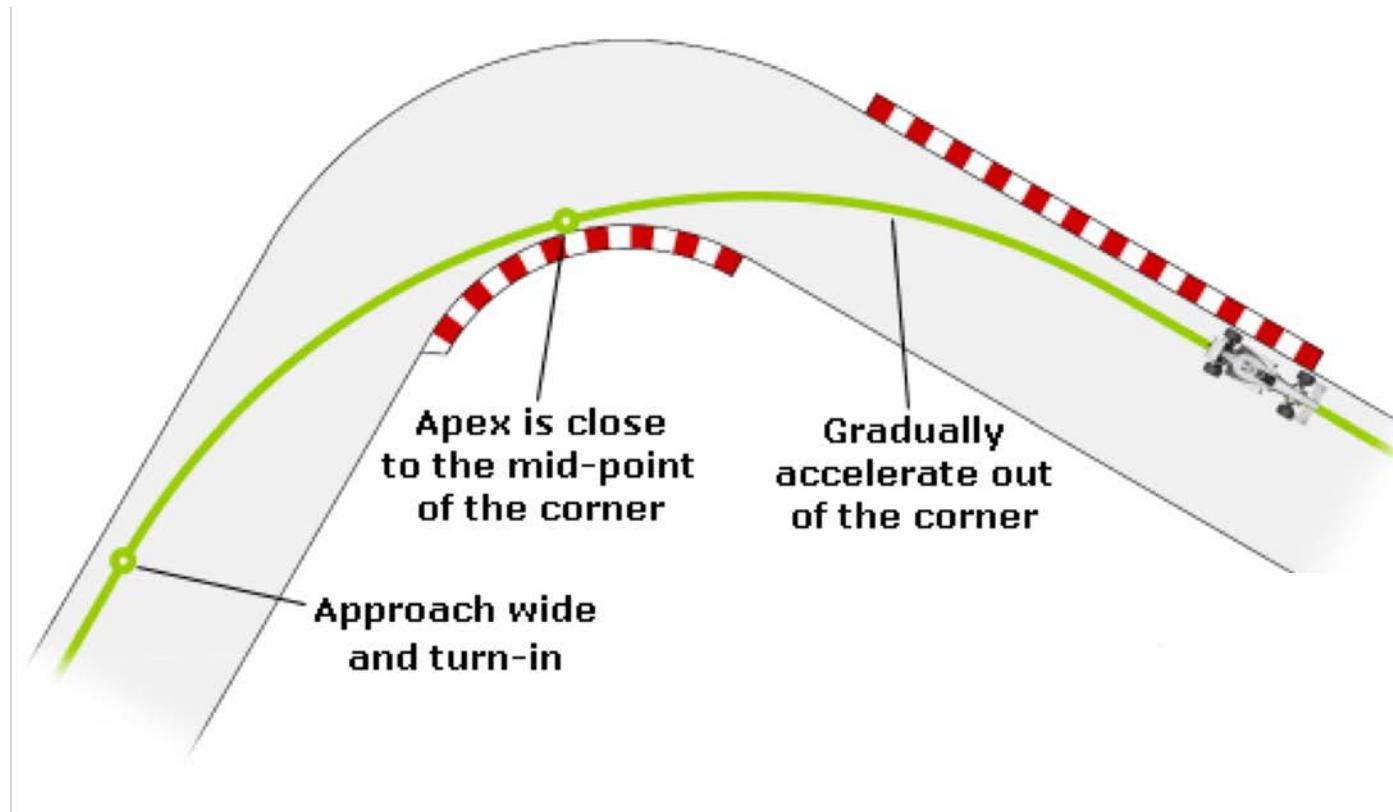
4th F1/10 International Autonomous Racing Competition

F1
TENTH
f1tenth.org

Perception. Planning. Control.

Perception. Planning. Control

Making sense of the surroundings

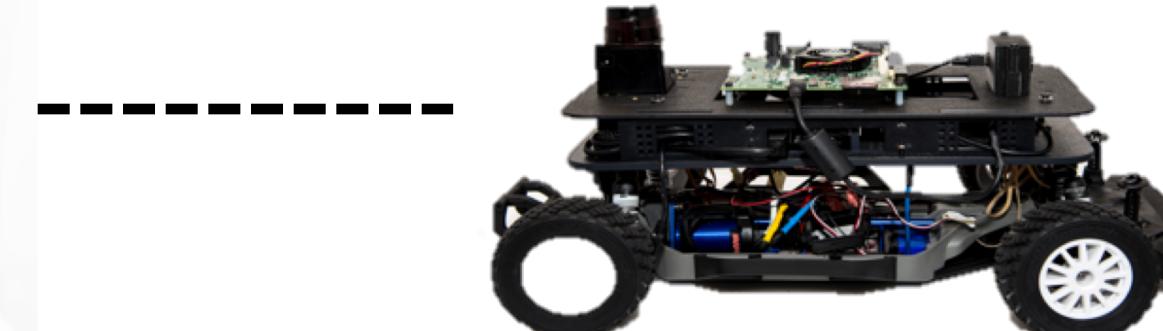
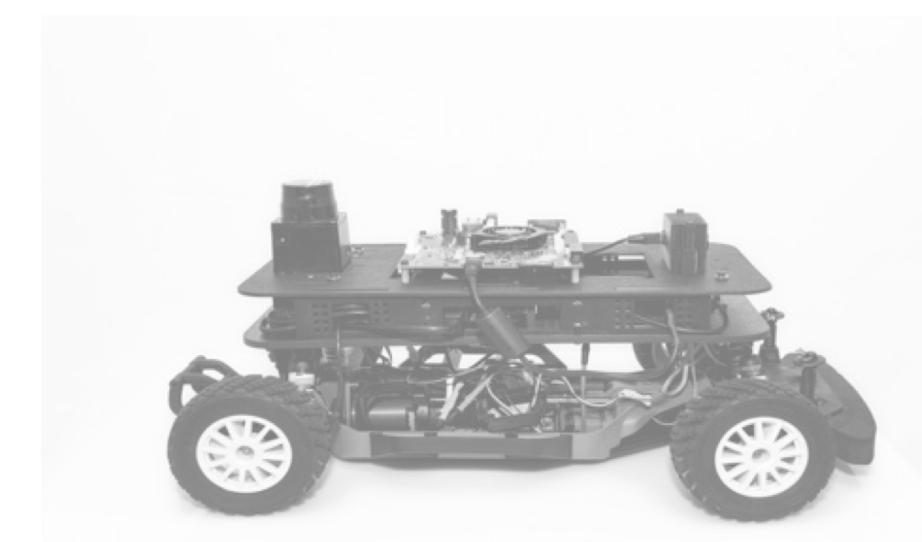
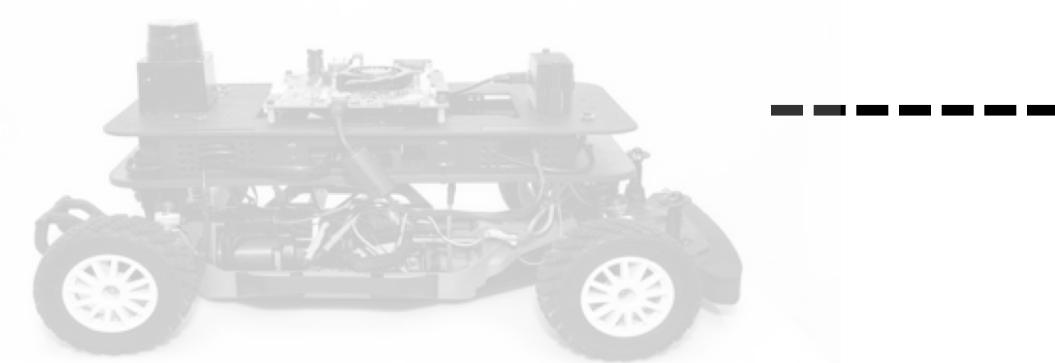
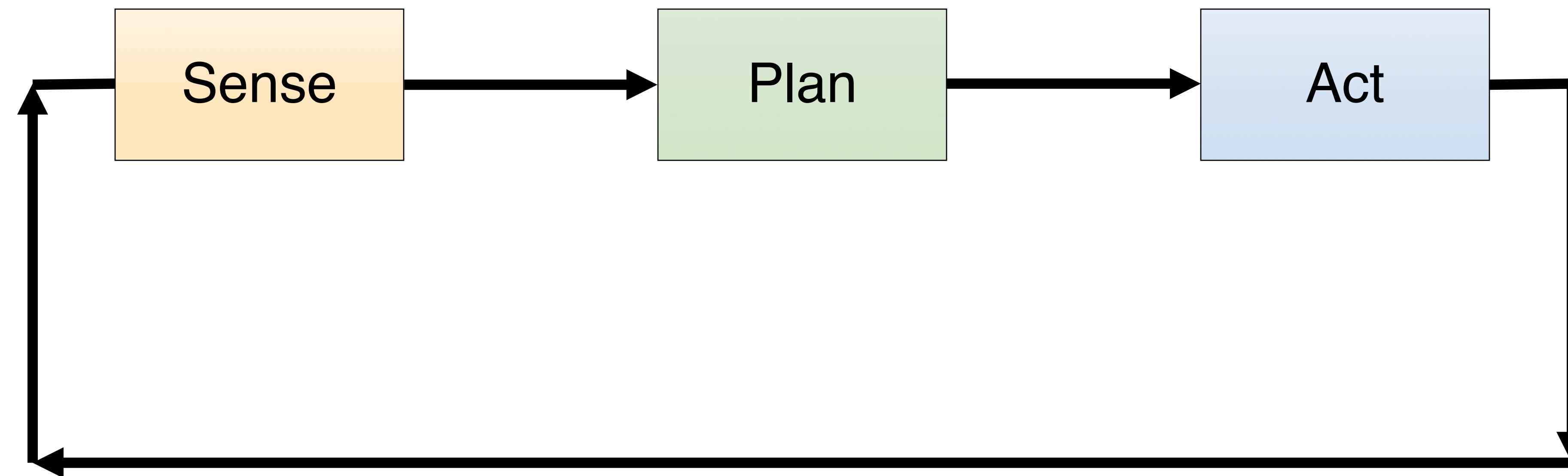


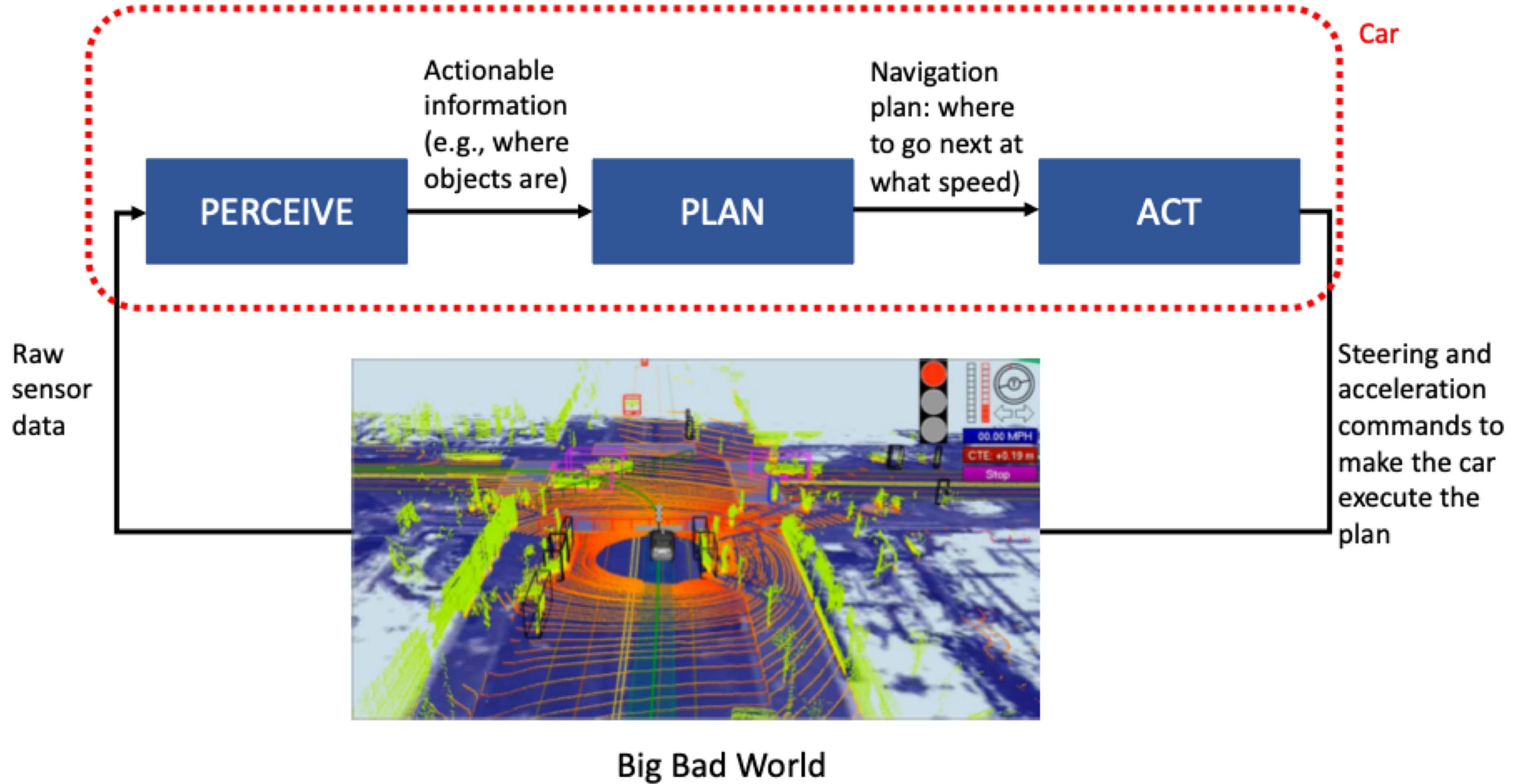
Planning the fastest racing line

Controlling the car to follow the planned path

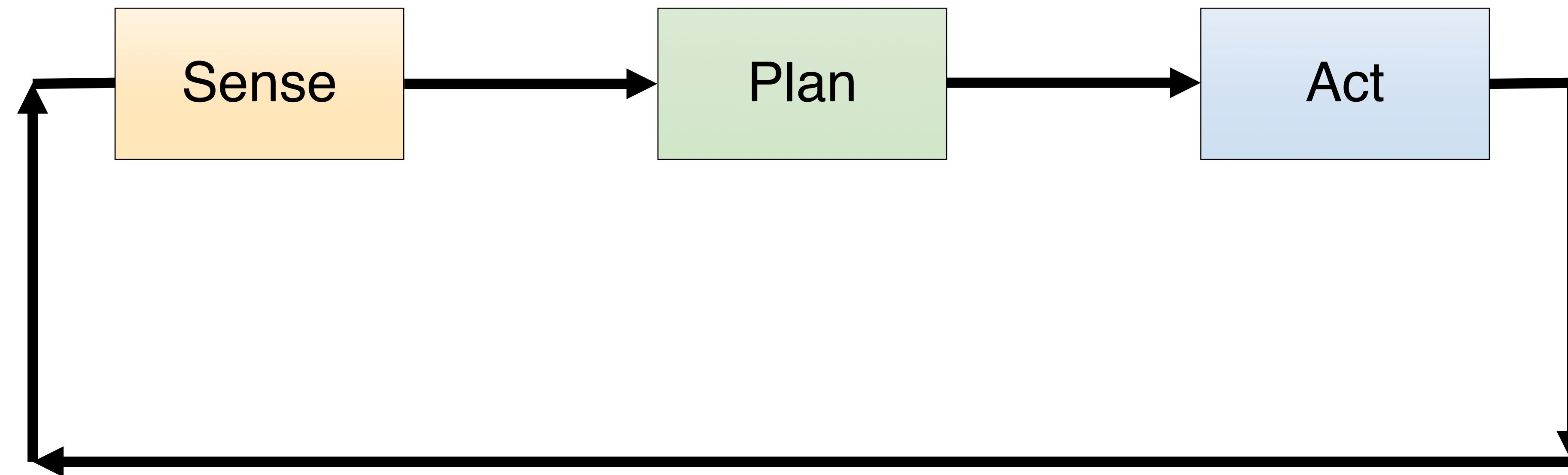


Closing the loop

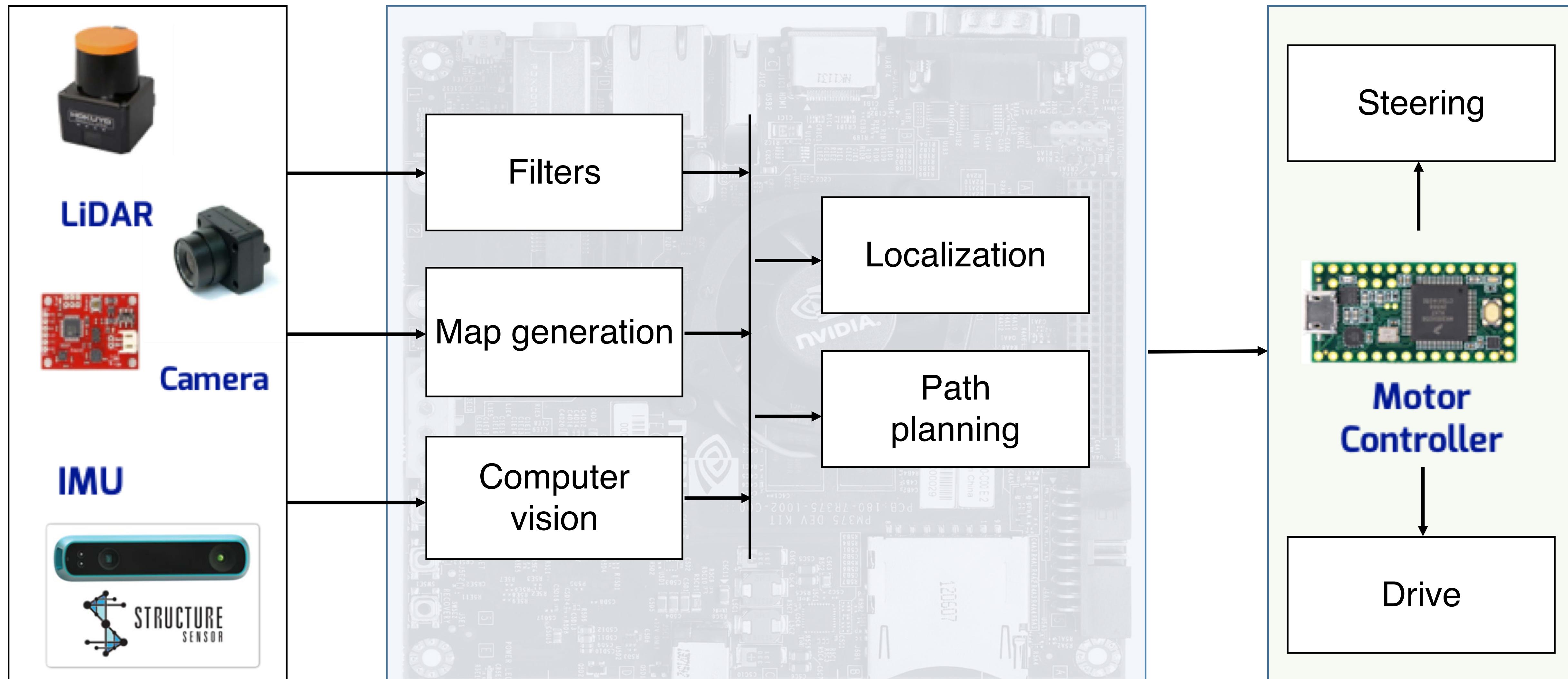




Closing the loop fast



System Architecture

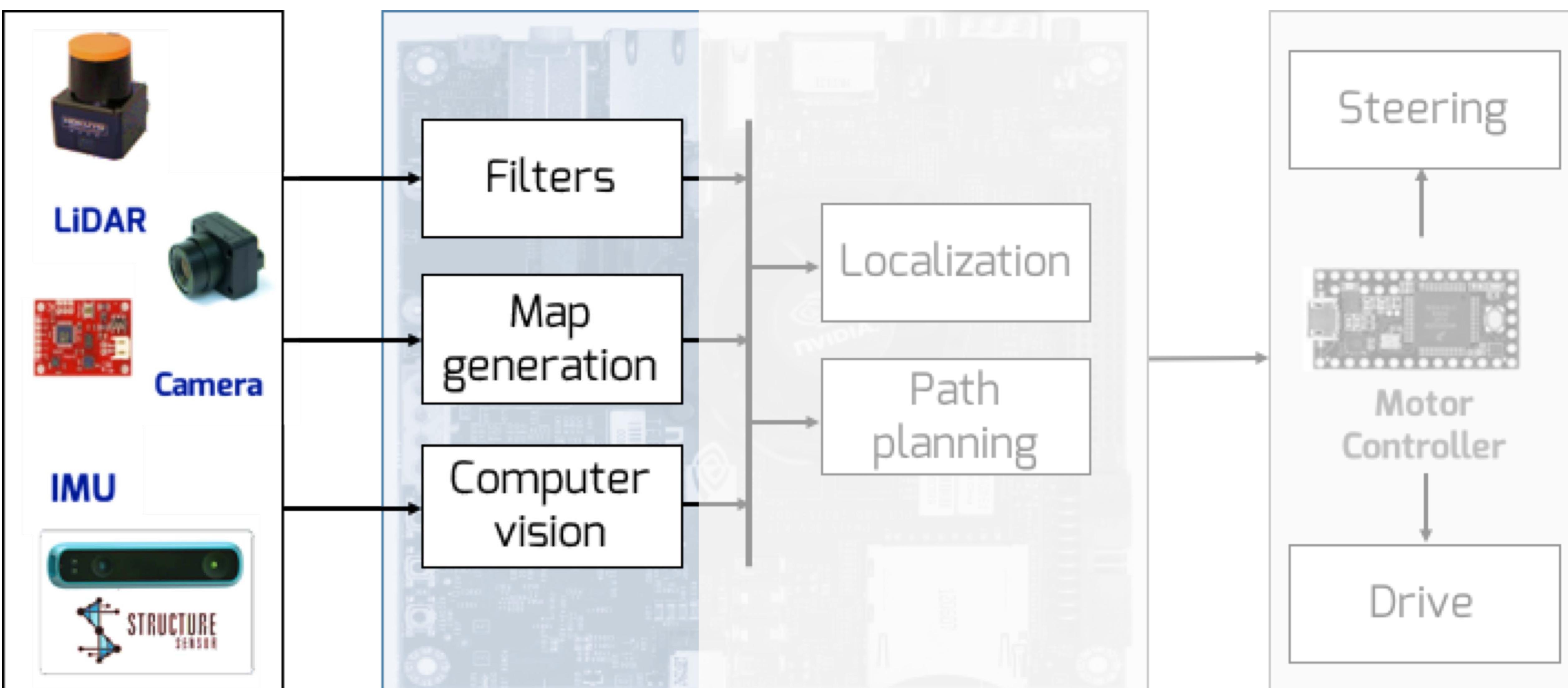


Perception

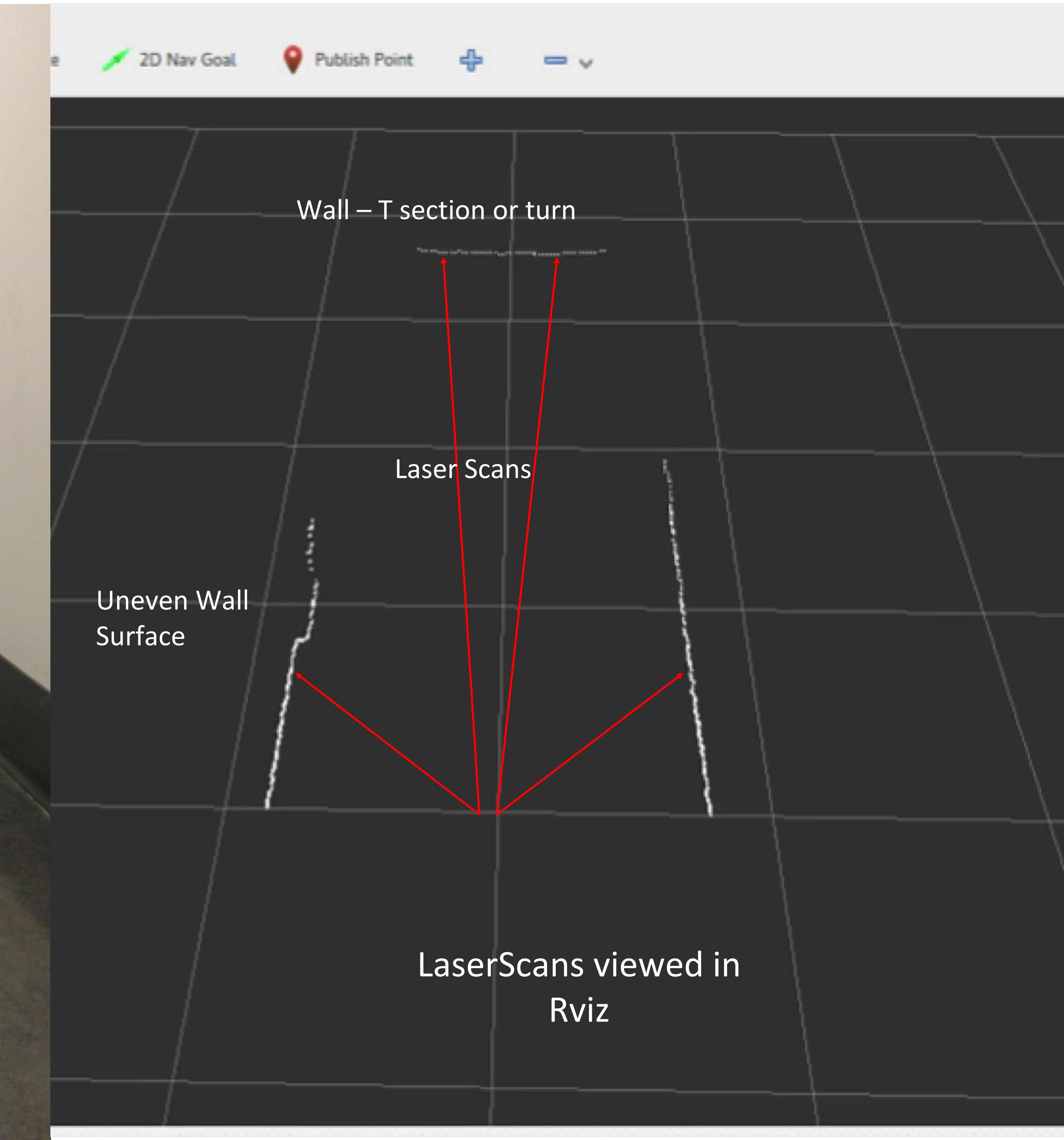
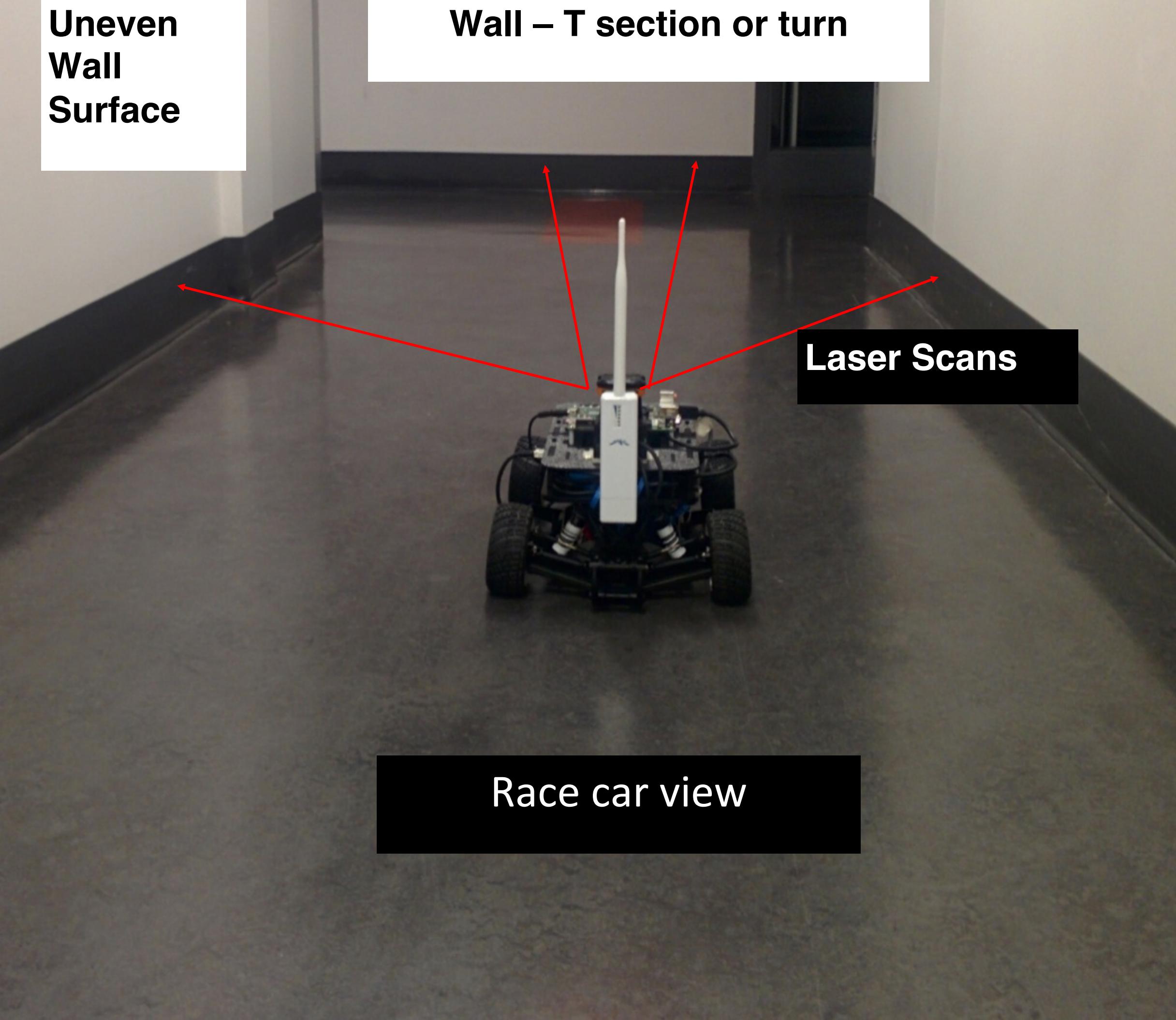
Planning

Control

Perception



Perception



Pose Representations and Transformations

Problem:

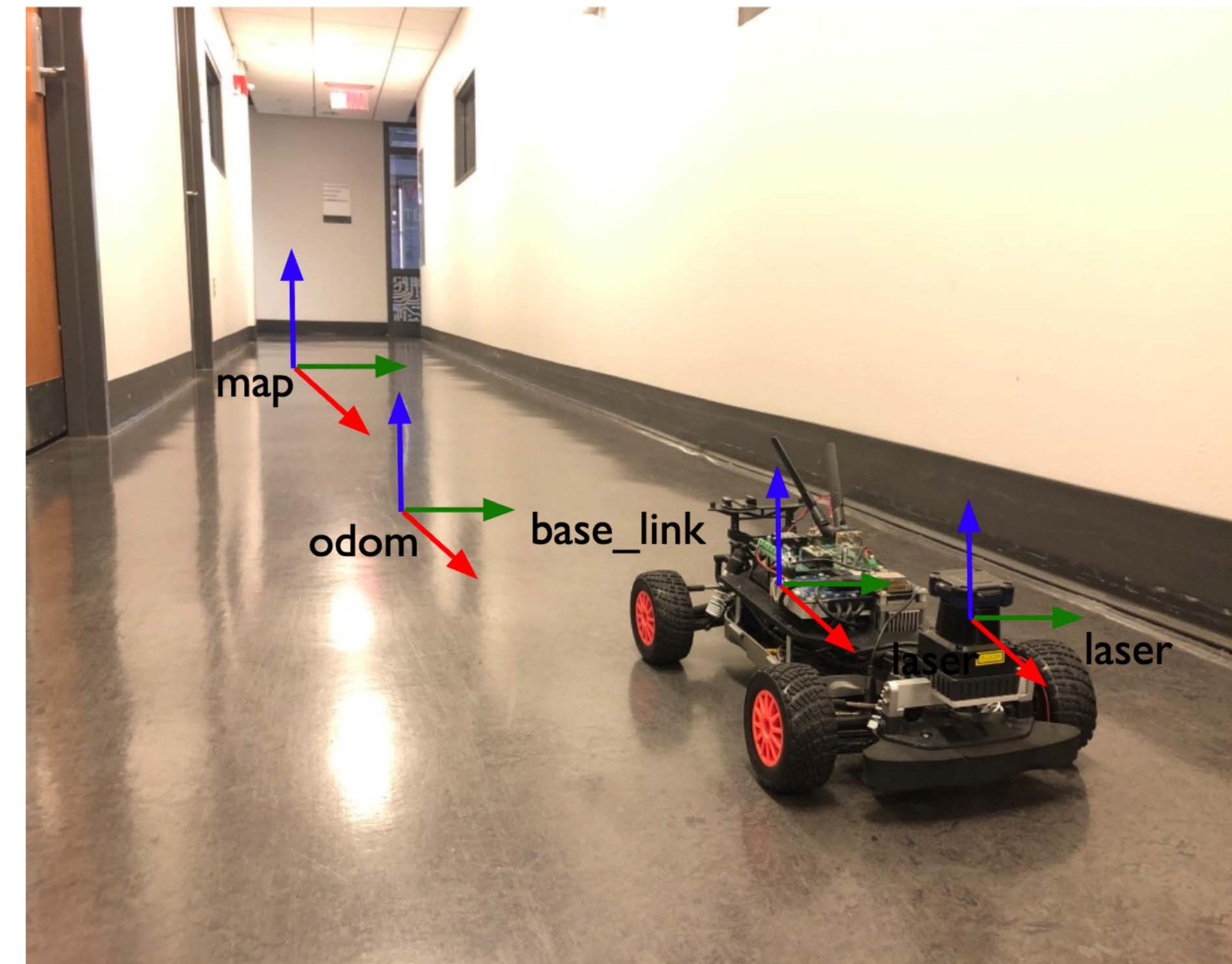
Sensors in different reference frames

Understand:

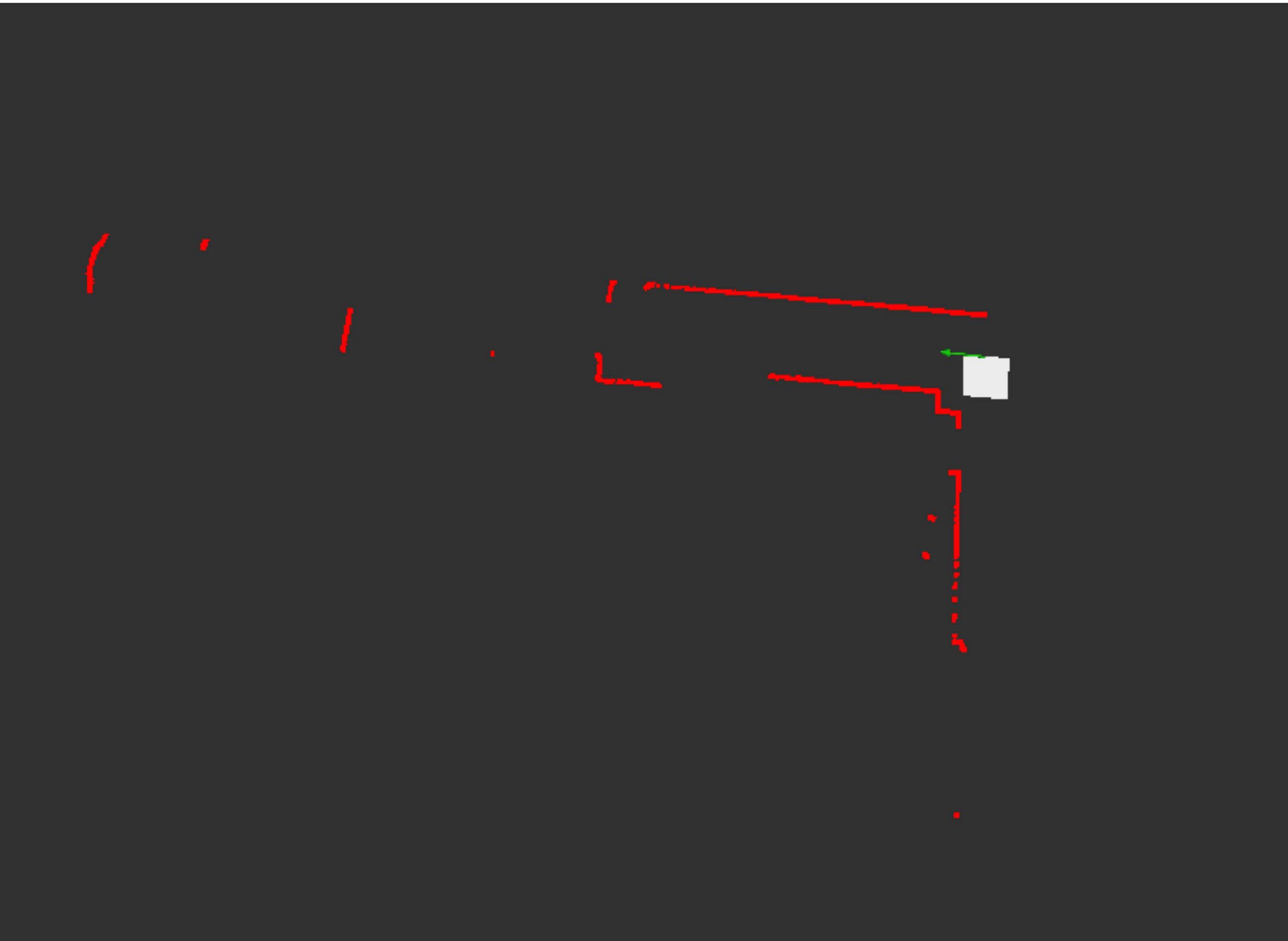
Rigid Body Transformations

Implement:

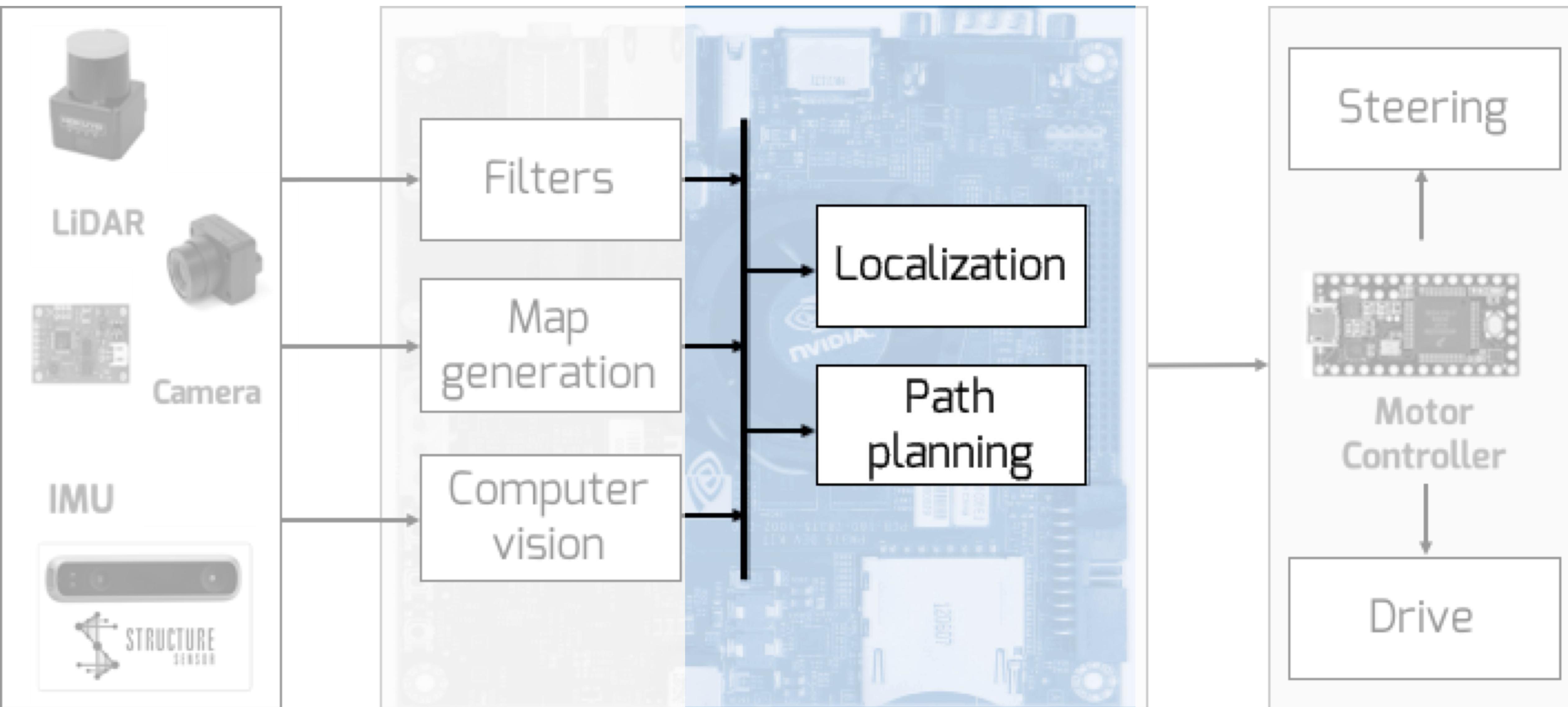
Tf tree, pose transformations in ROS



Registering a LIDAR Scan

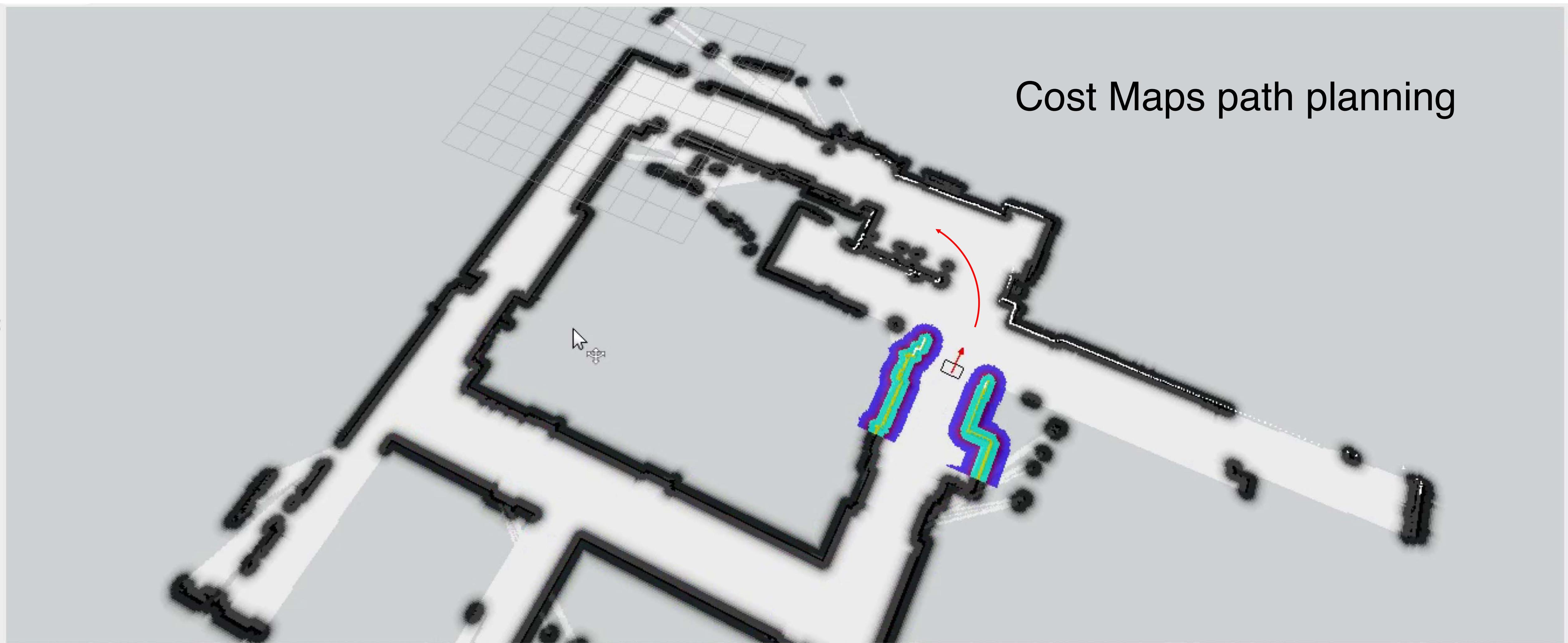


Planning



Planning

Cost Maps path planning



Time

ROS Time: 1453072851.57

ROS Elapsed: 30.29

Wall Time: 1454341978.42

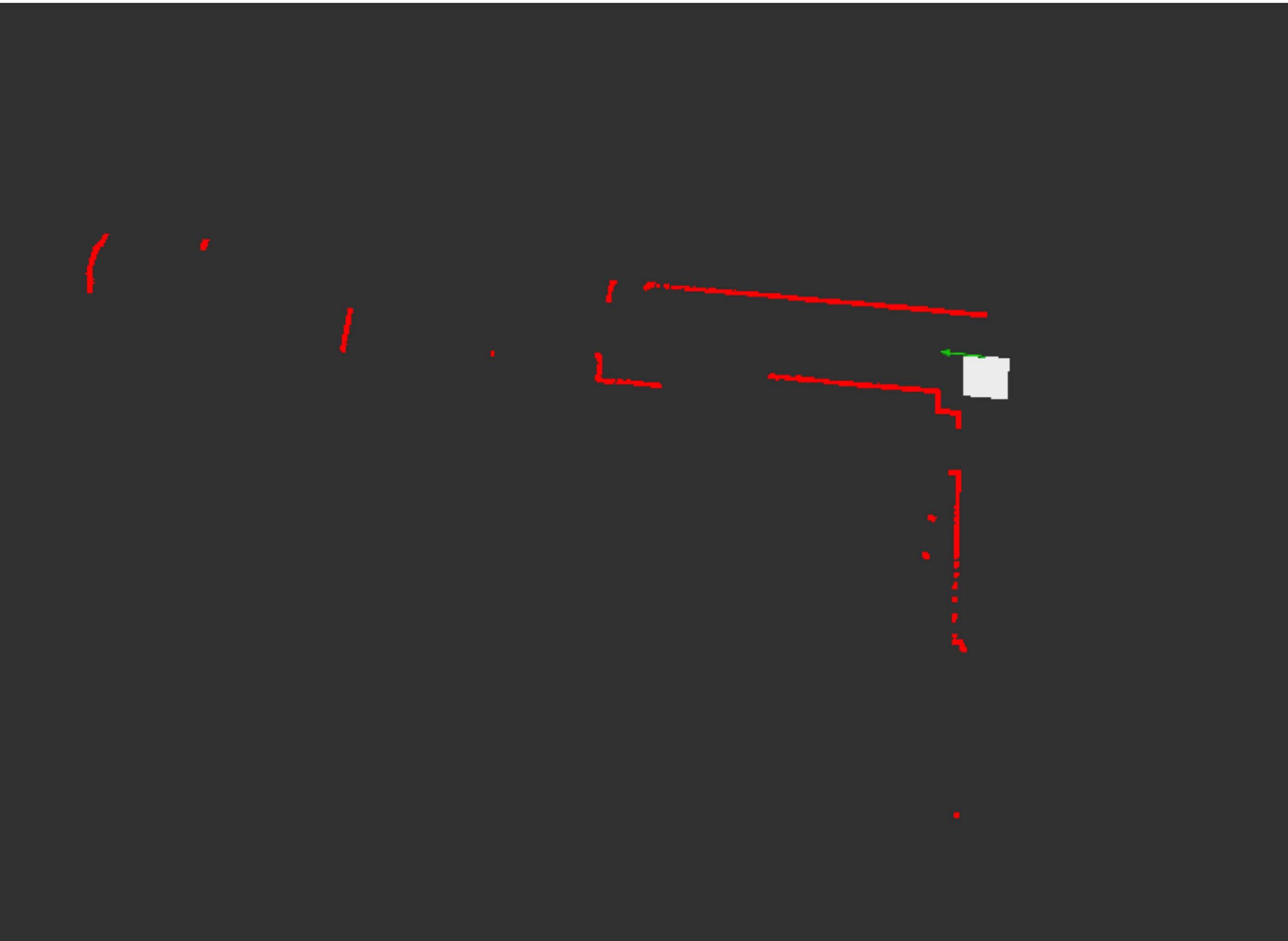
Wall Elapsed: 68.81

Experimental

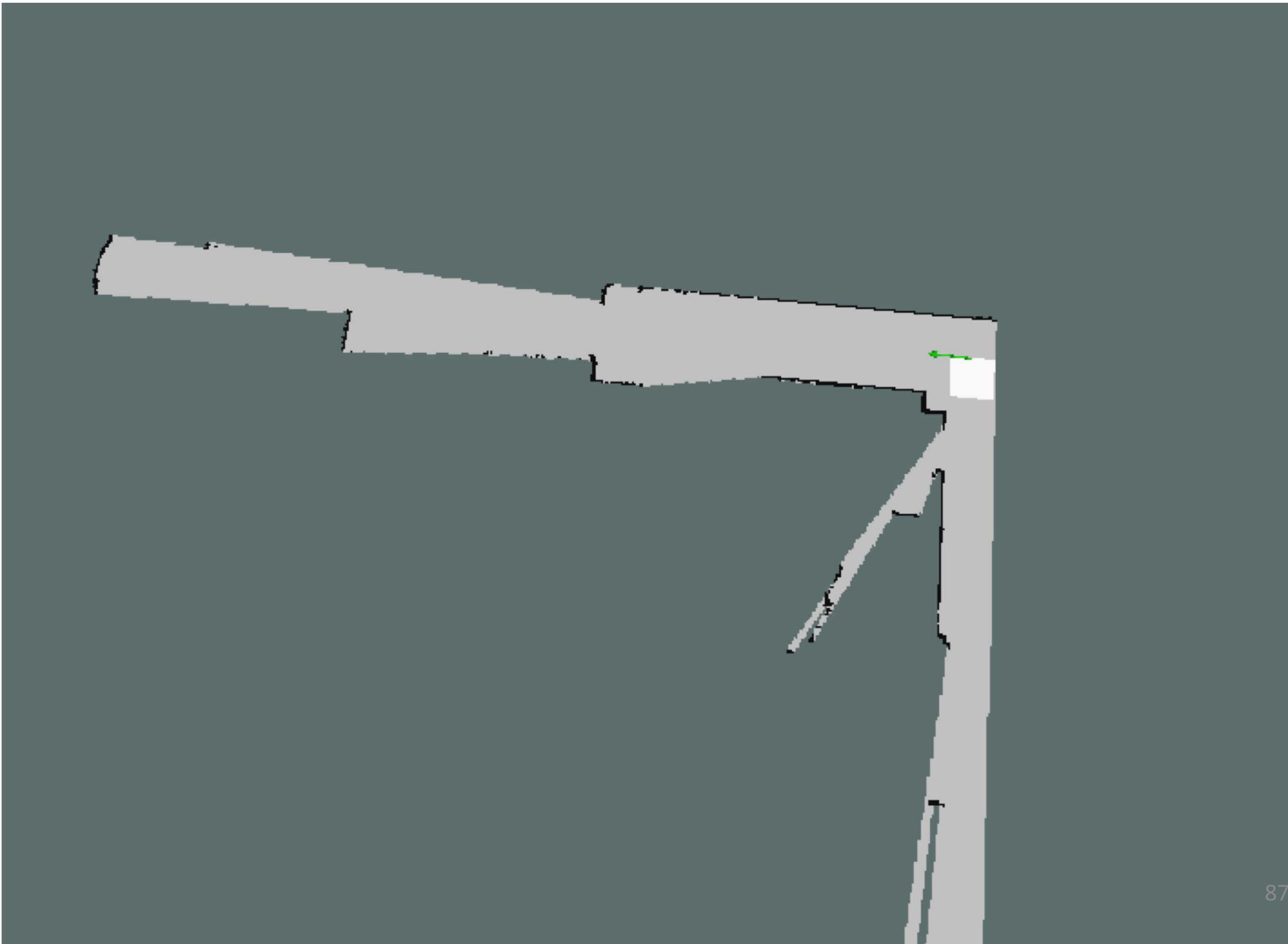
Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Move Z. Shift: More options.

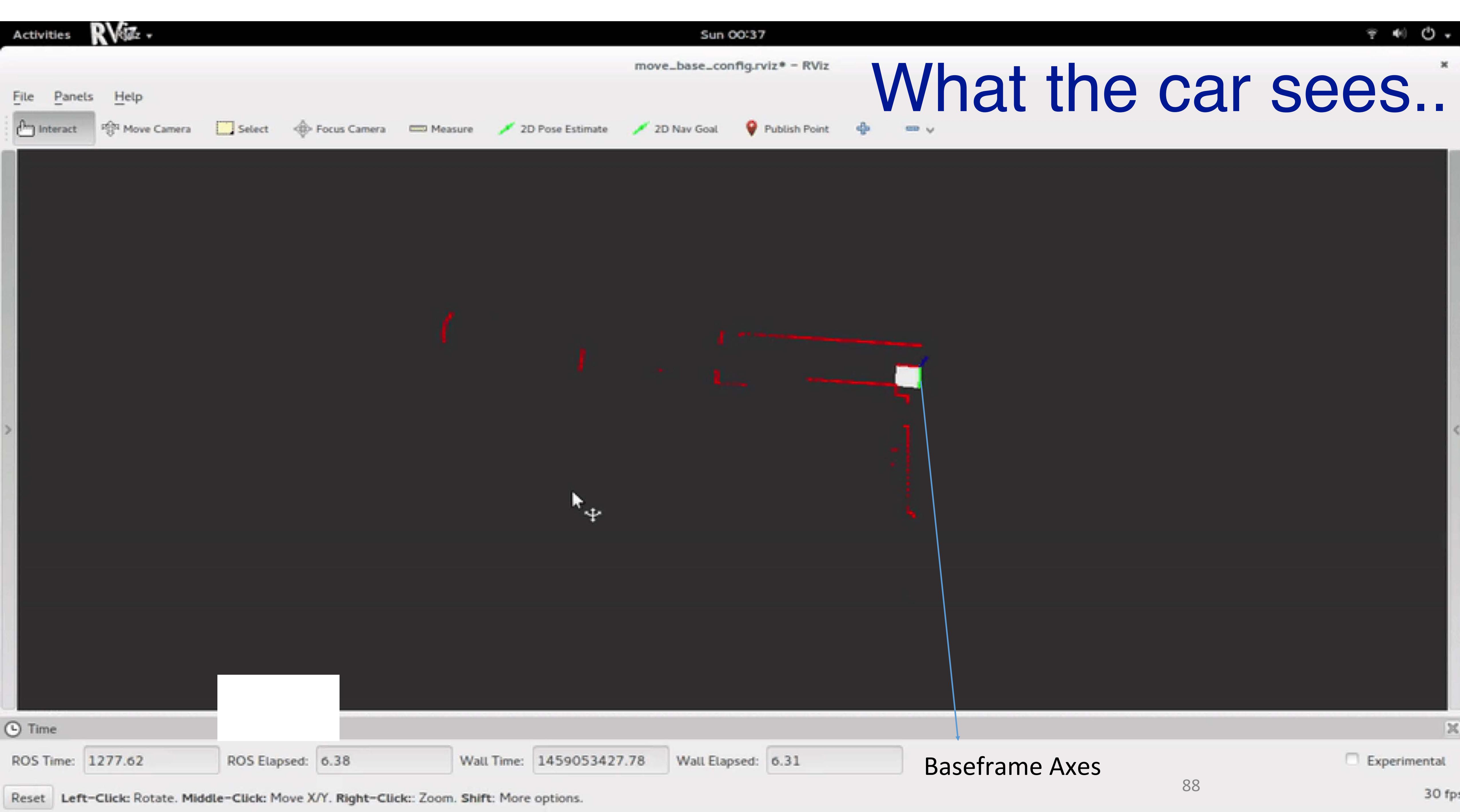
30 fps

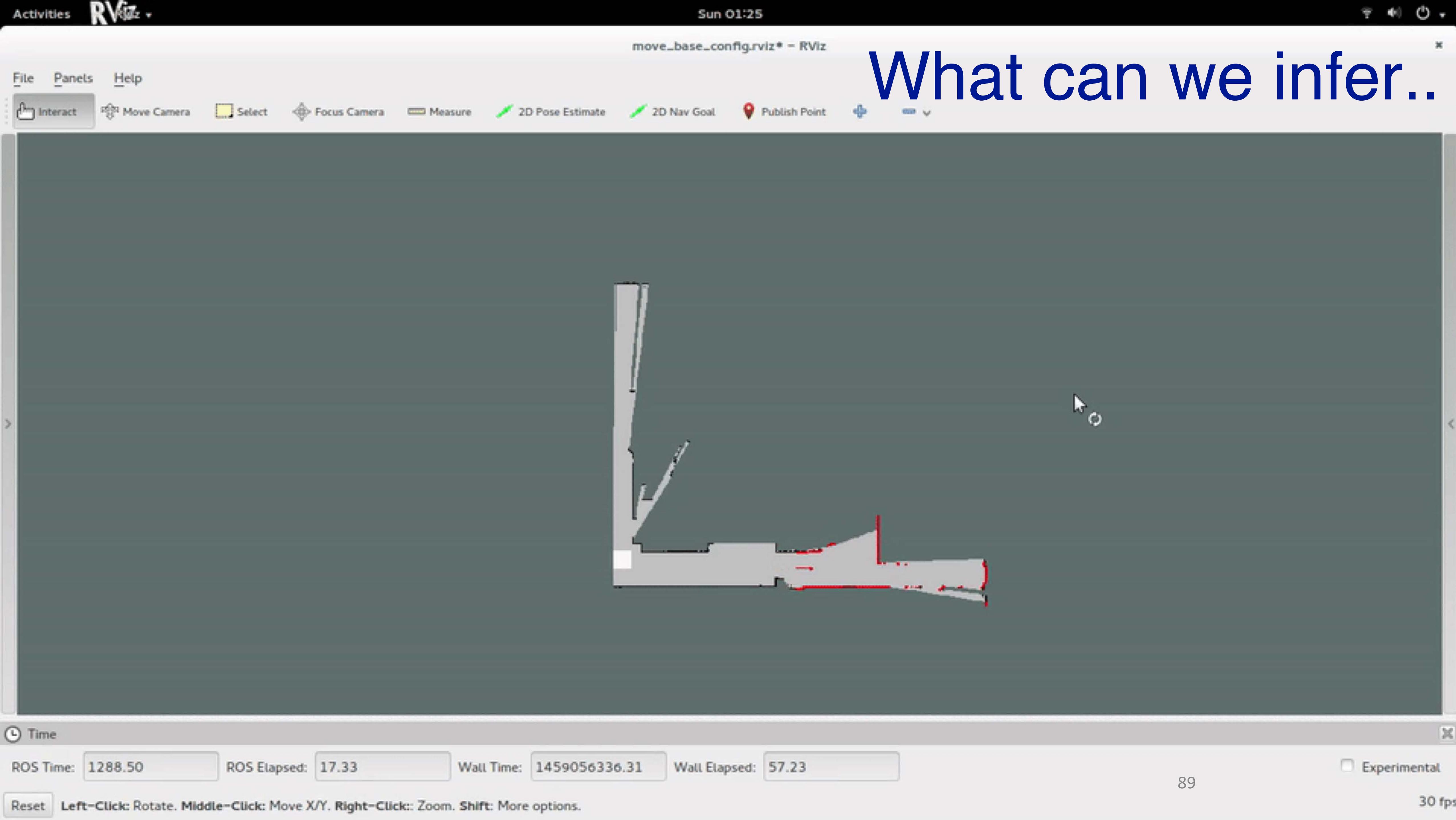
Registering a LIDAR Scan



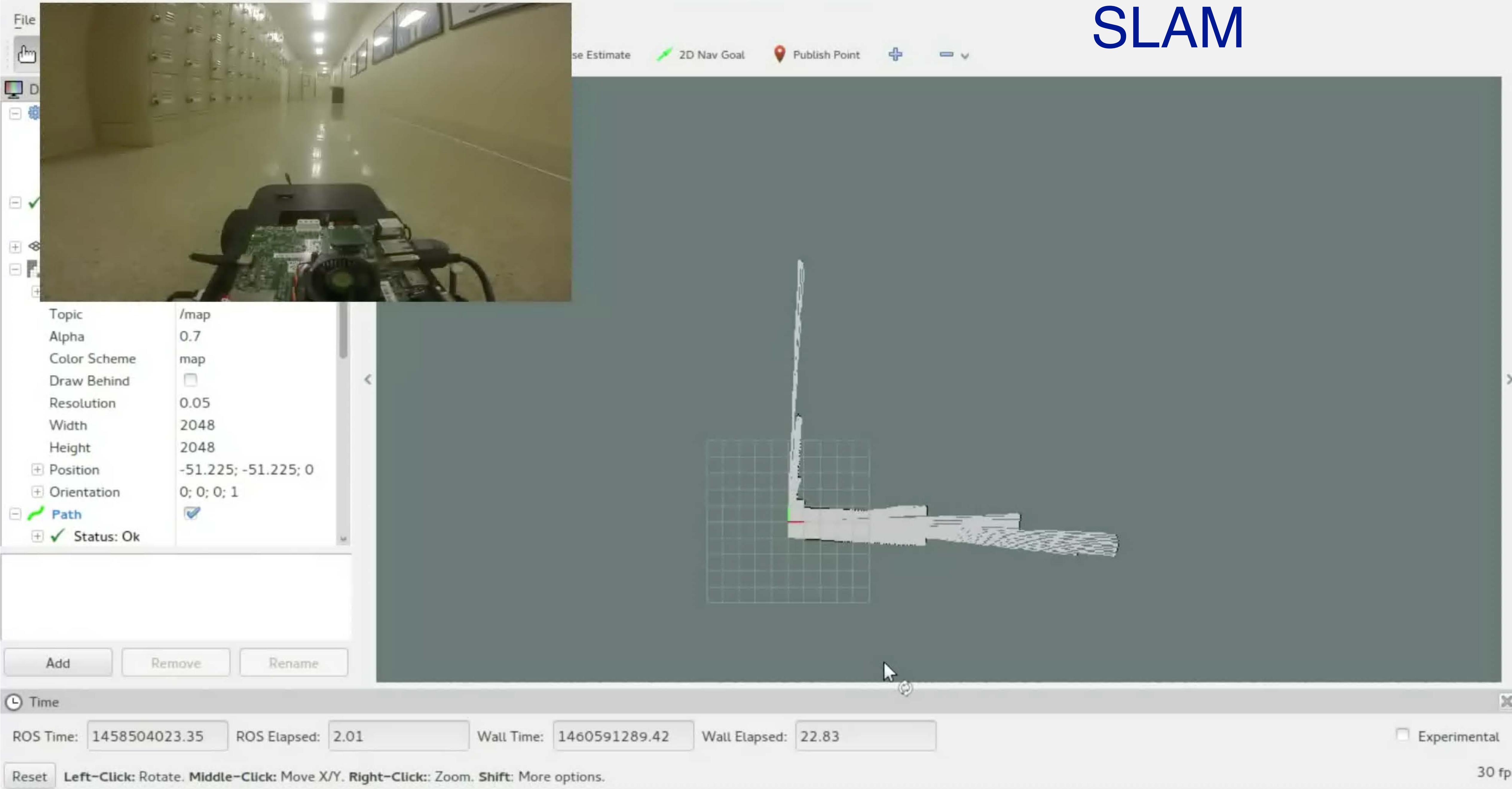
Registering a LIDAR Scan



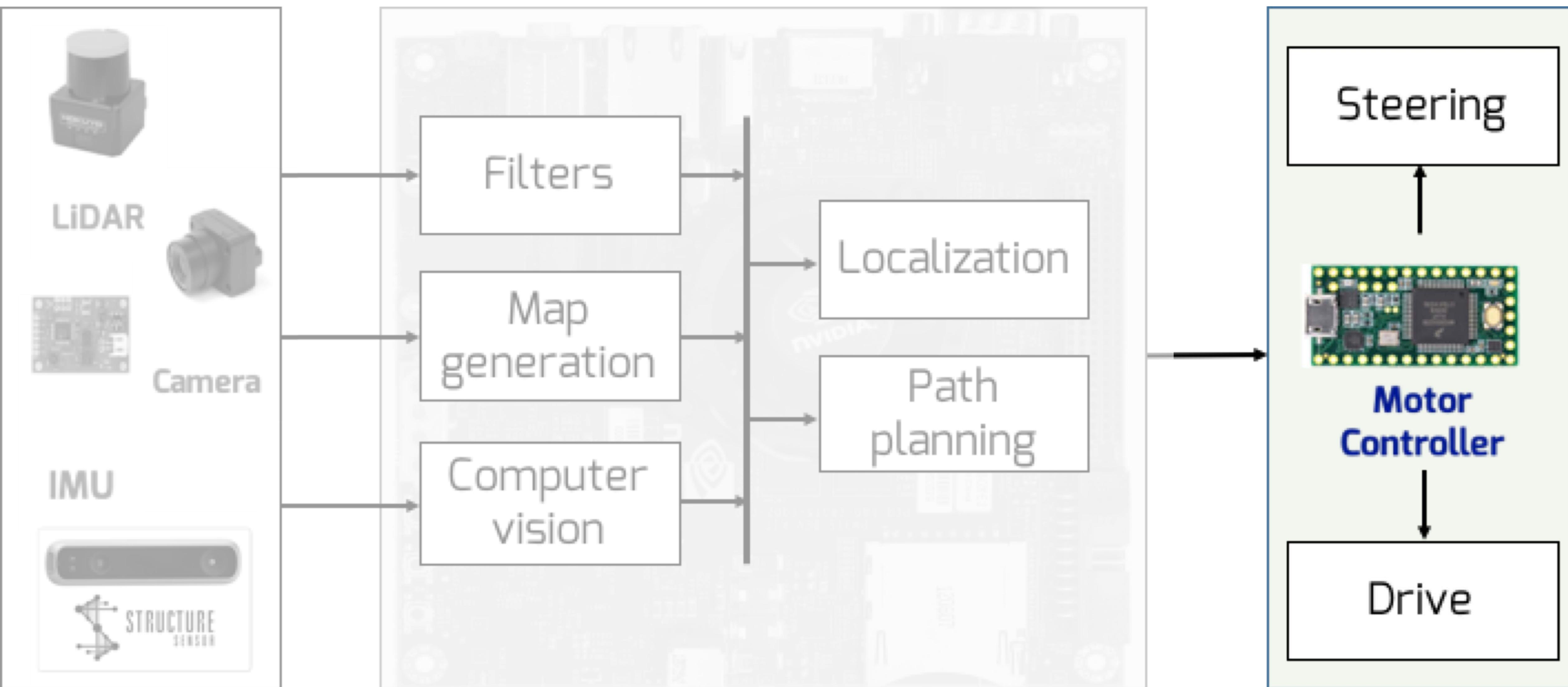




SLAM



Control



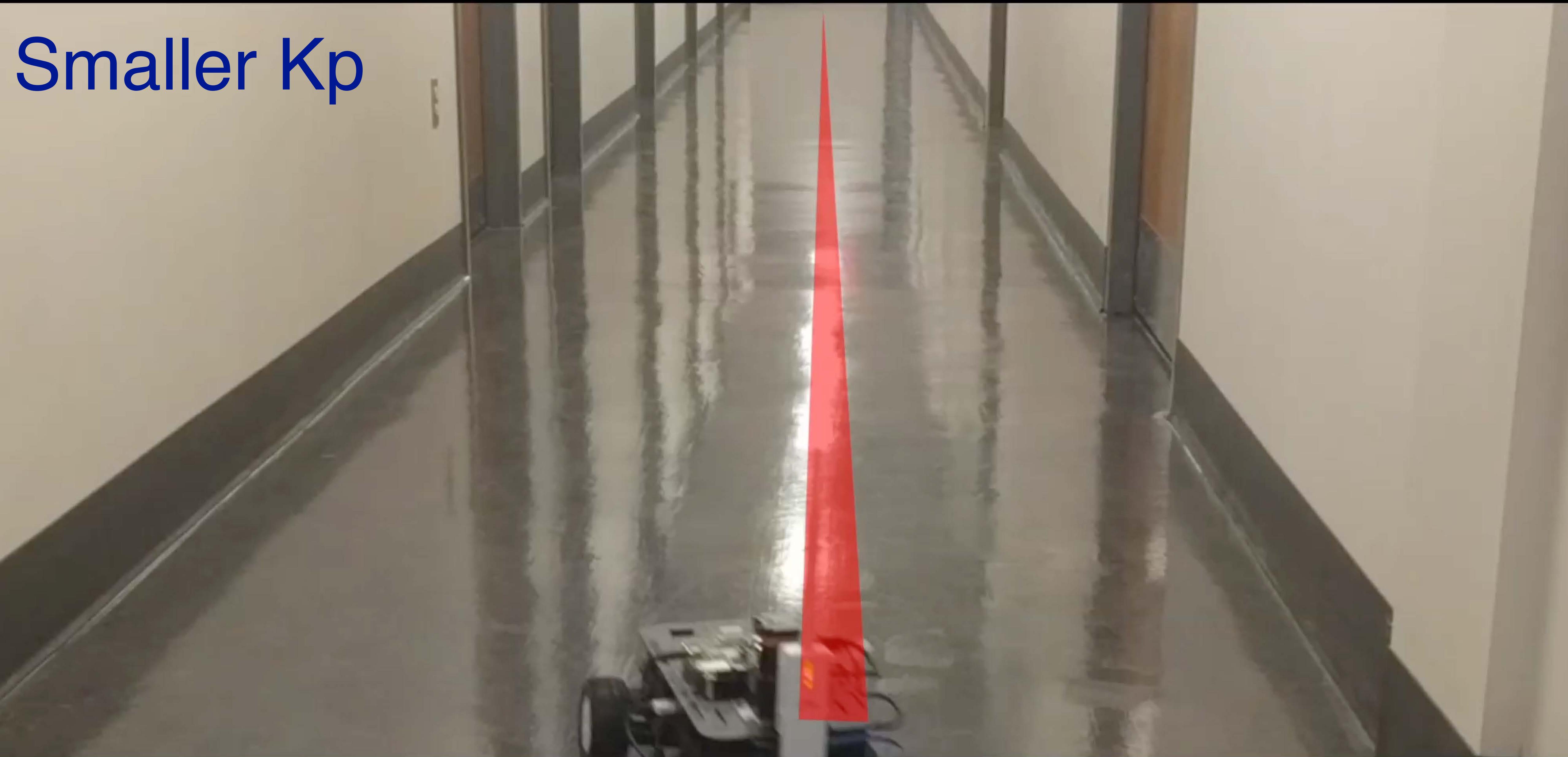
Control

Proportional, **I**ntegral, **D**erivative control

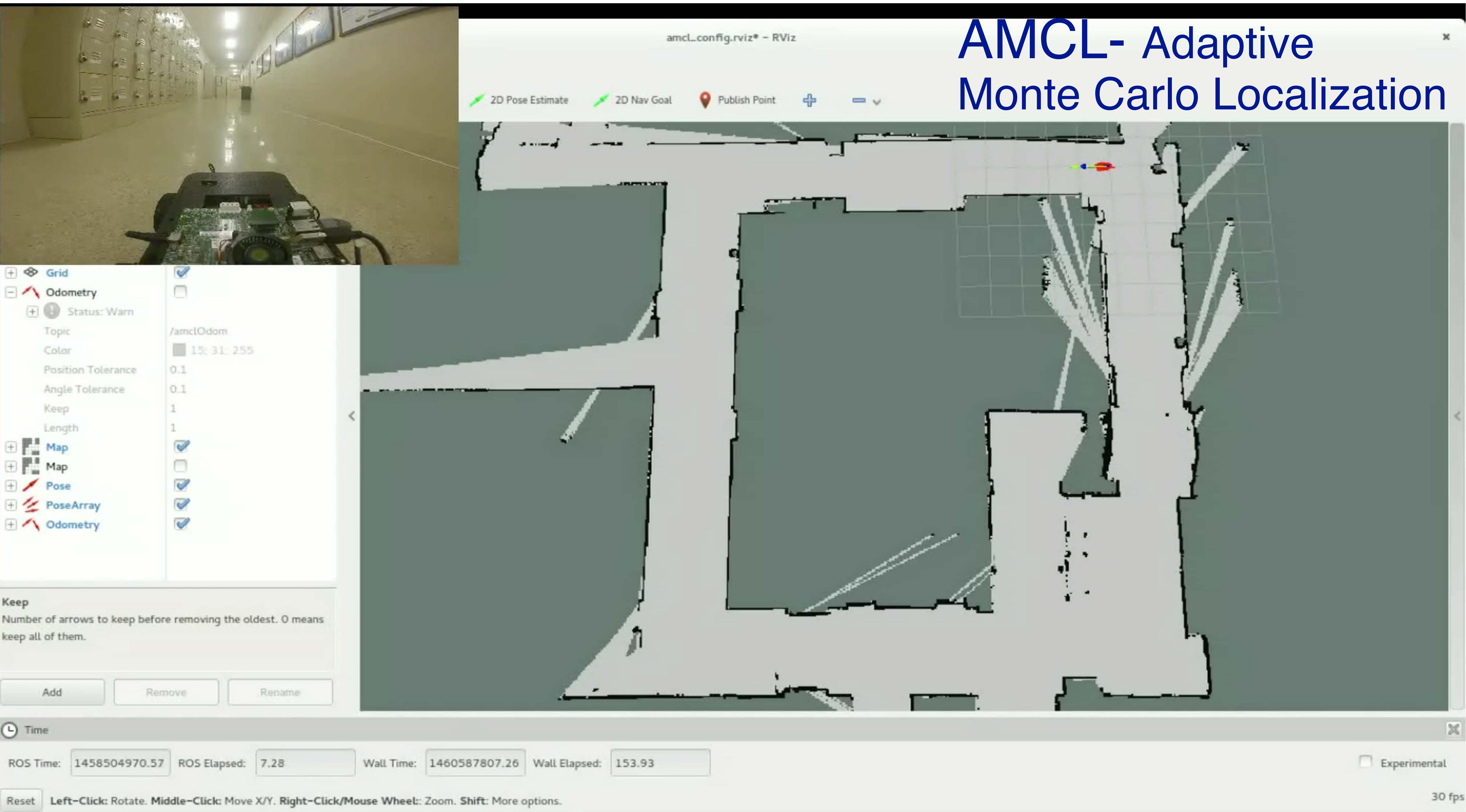
Nominal gains



Smaller K_p



AMCL- Adaptive Monte Carlo Localization



Wall Following

Problem:

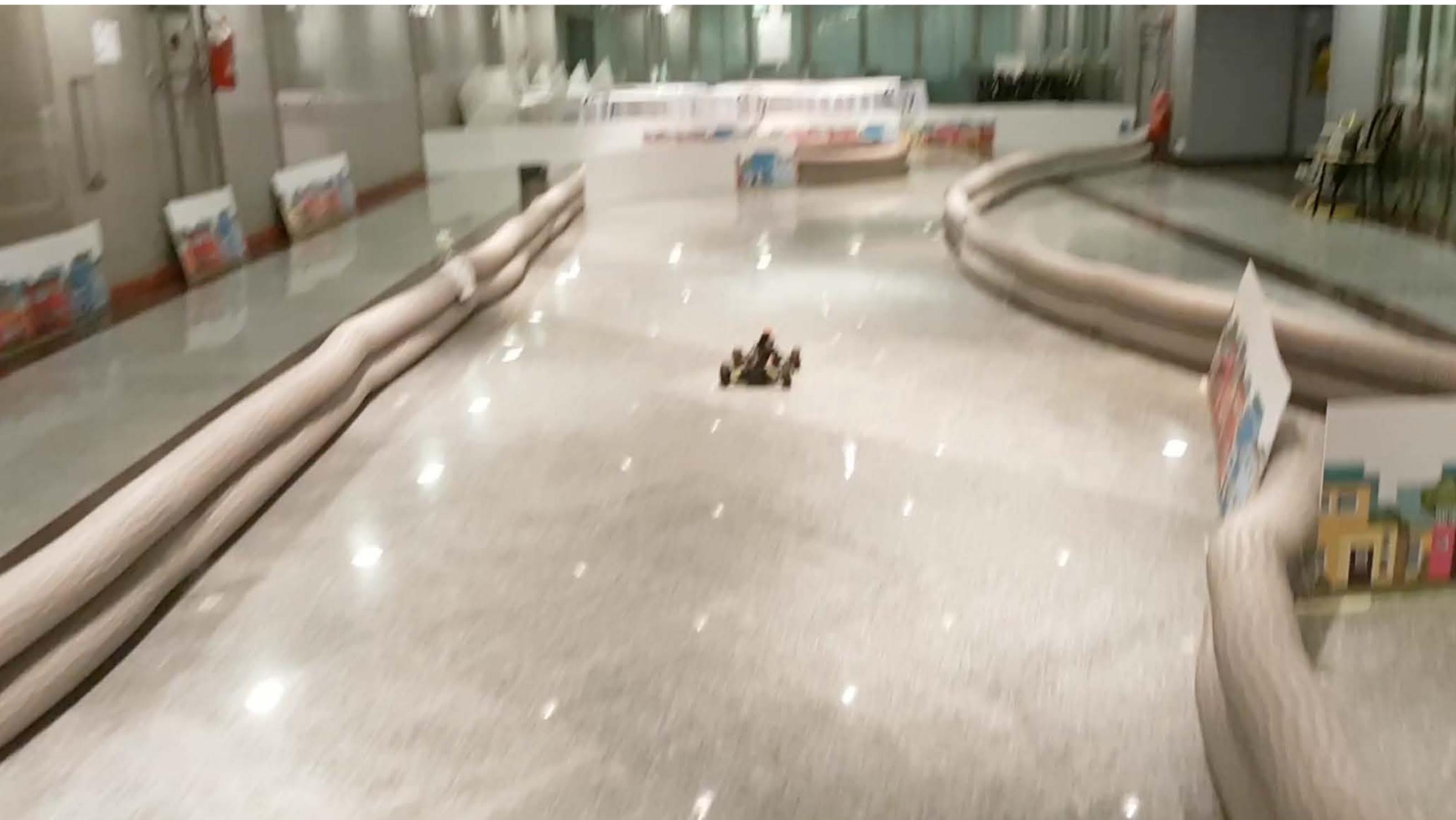
Drive around the track using a LIDAR

Understand:

LIDAR rays, PID, cross track error

Implement:

Wall following ROS controller



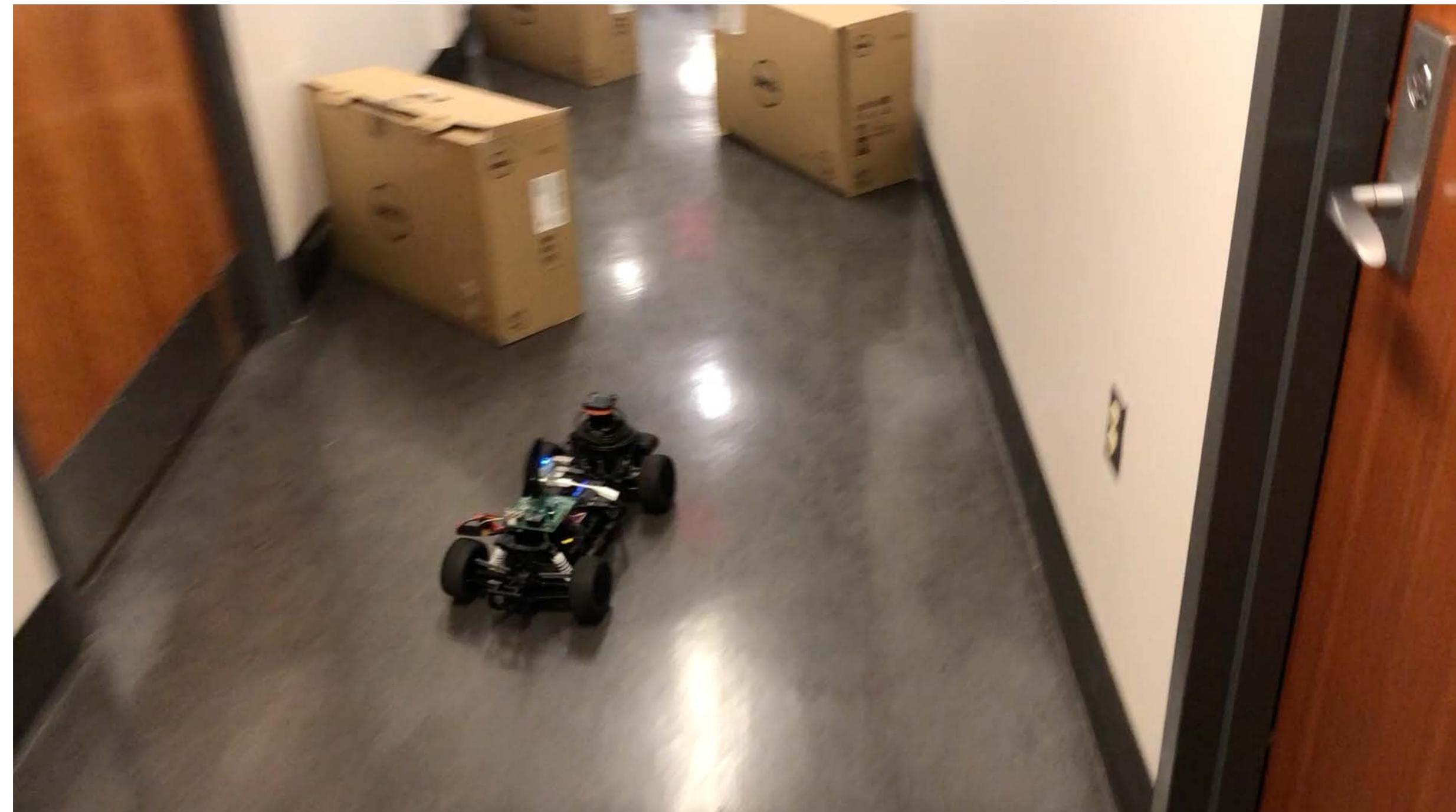
Obstacle Avoidance: Follow the gap

Reactive Navigation:

Use immediate sensor input to decide driving command

Planning for obstacle avoidance:

Use LIDAR for both static and dynamic obstacle avoidance



Race: Reactive Methods

Race Format:

Time-trial; single car on track

Penalties: Crashing

Baseline: Complete 5 laps without crashing

Example Video: CPS Week 2018

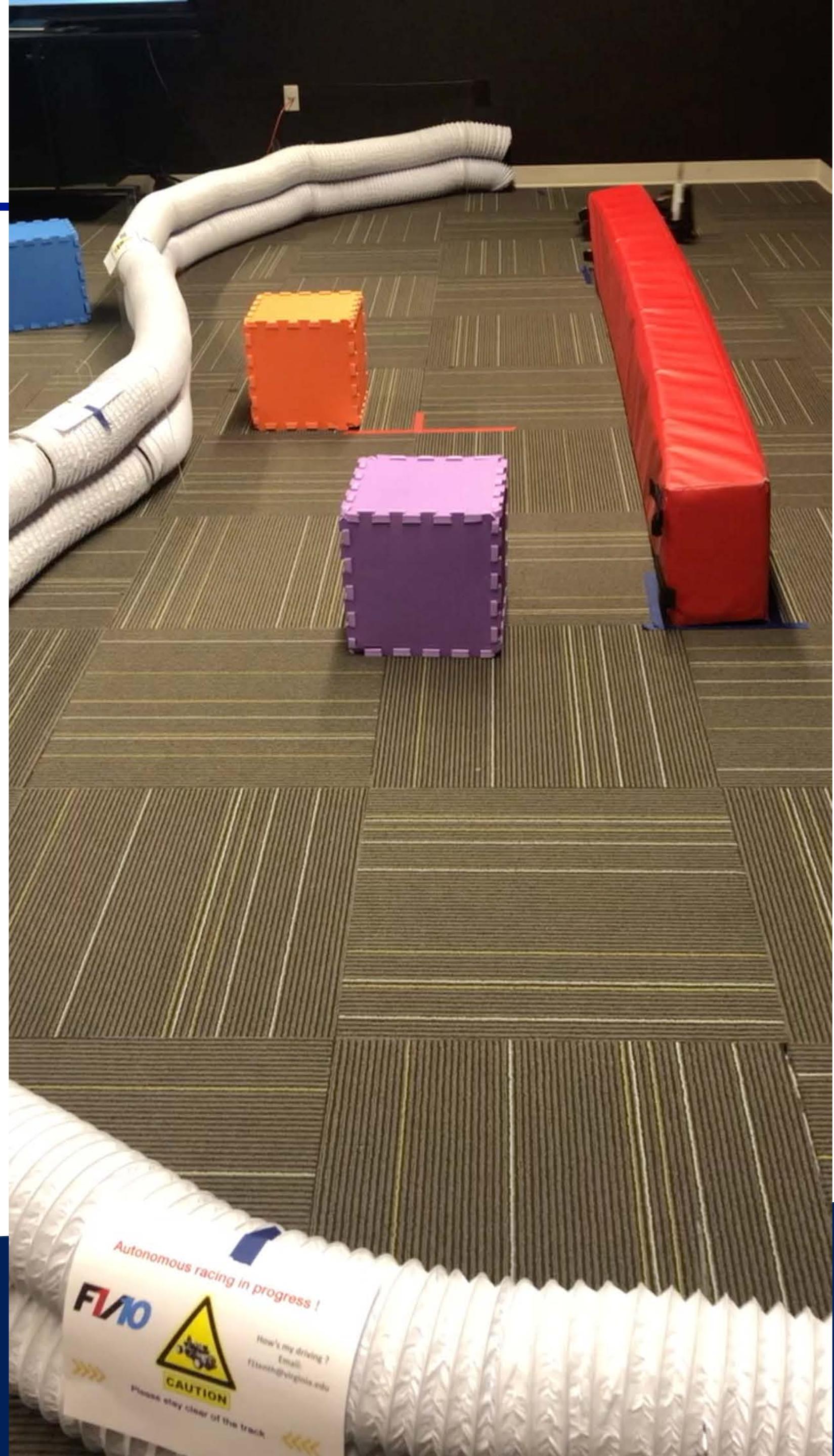


Motion Planning

Problem:

How to find a path between two nodes on a graph, how to navigate collision free through a cluttered space.

1. Search based motion planning
2. RRT
3. Time-Elastic Band Navigation
4. Potential Fields
5. Overtaking and race line planning

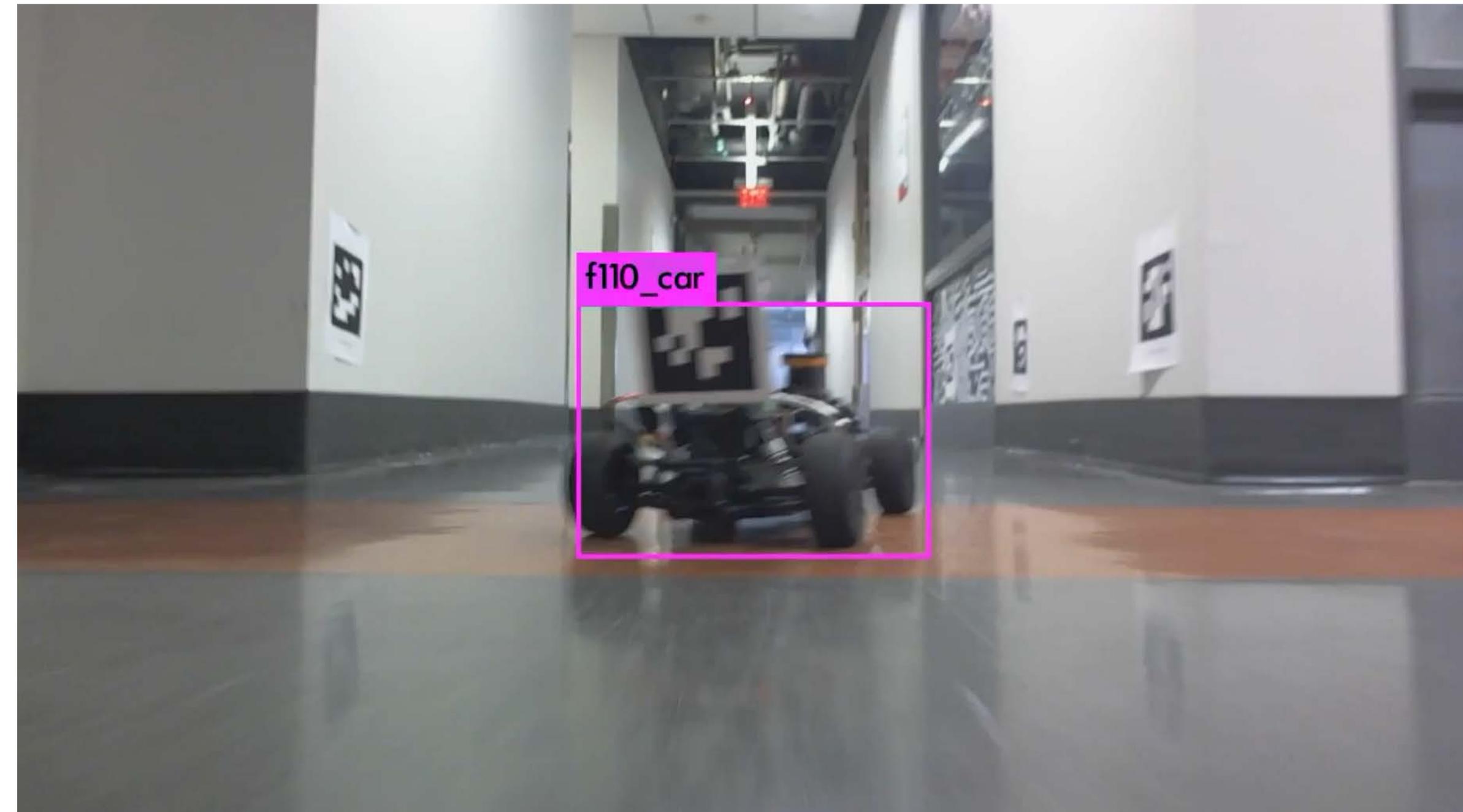


Vision: Detection and Pose Estimation

Problem:

Detect other F1/10 cars and estimate their pose and velocity

1. Deep Learning :YOLO
2. Geometrical vision

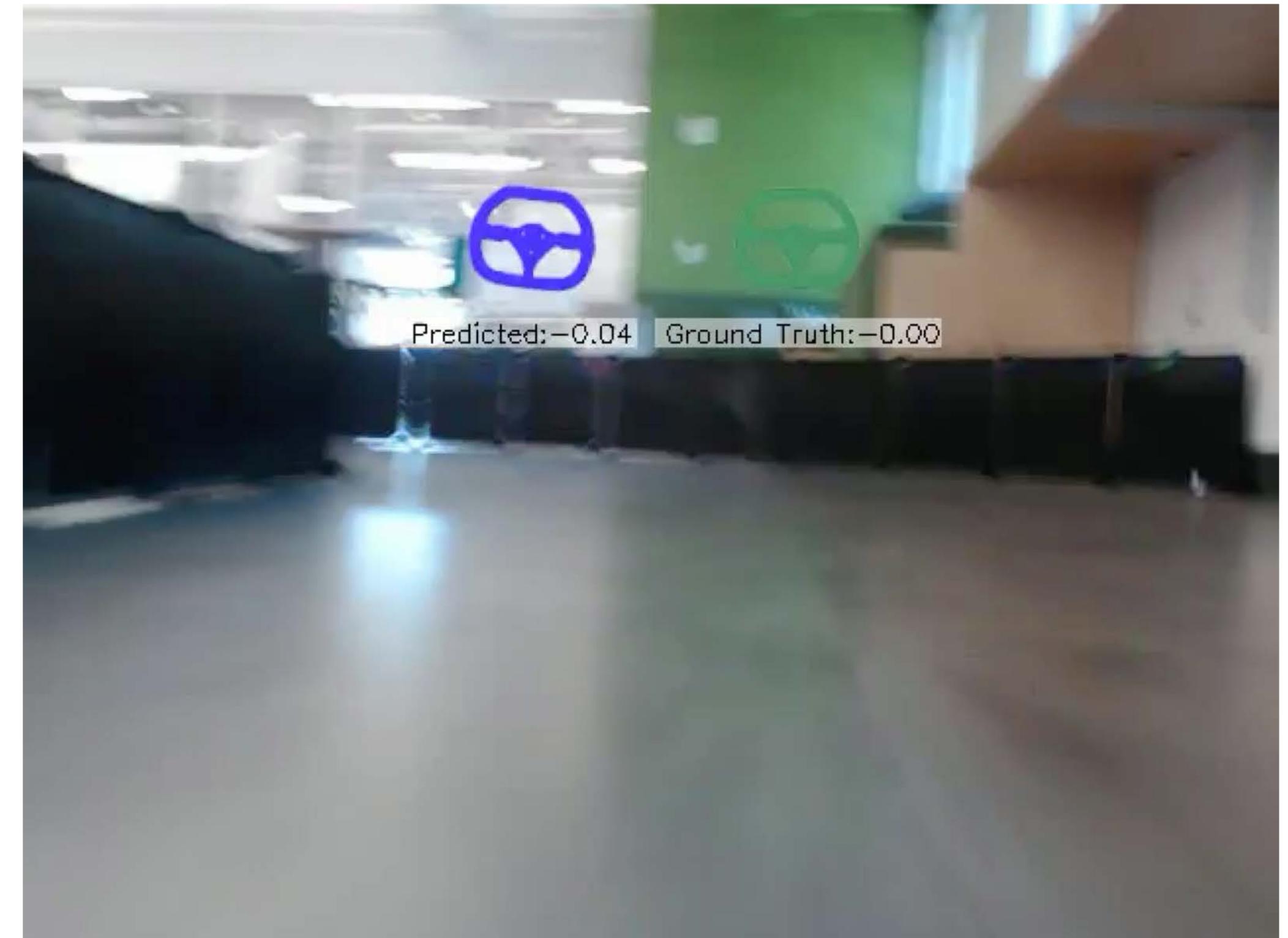


End-to-End Driving

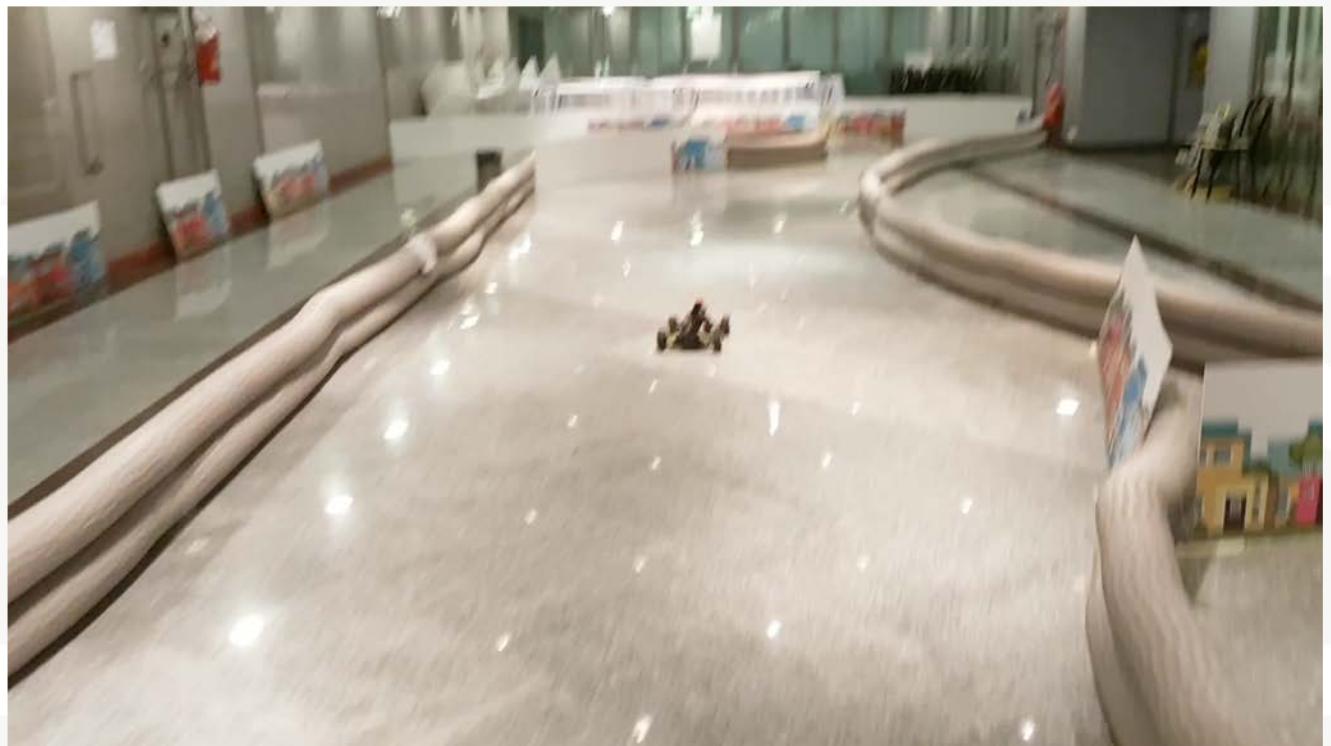
Problem:

End-to-End driving with imitation learning.

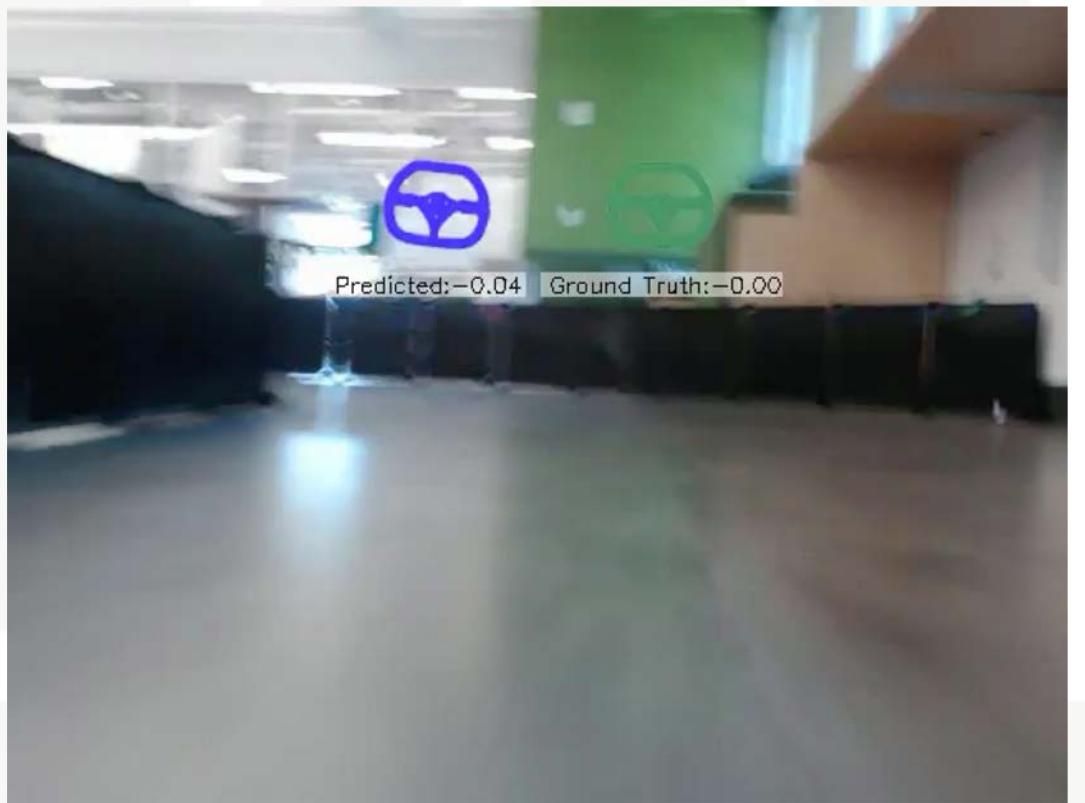
1. Train a network to map pixels to steering angle
2. Pixels to trajectory



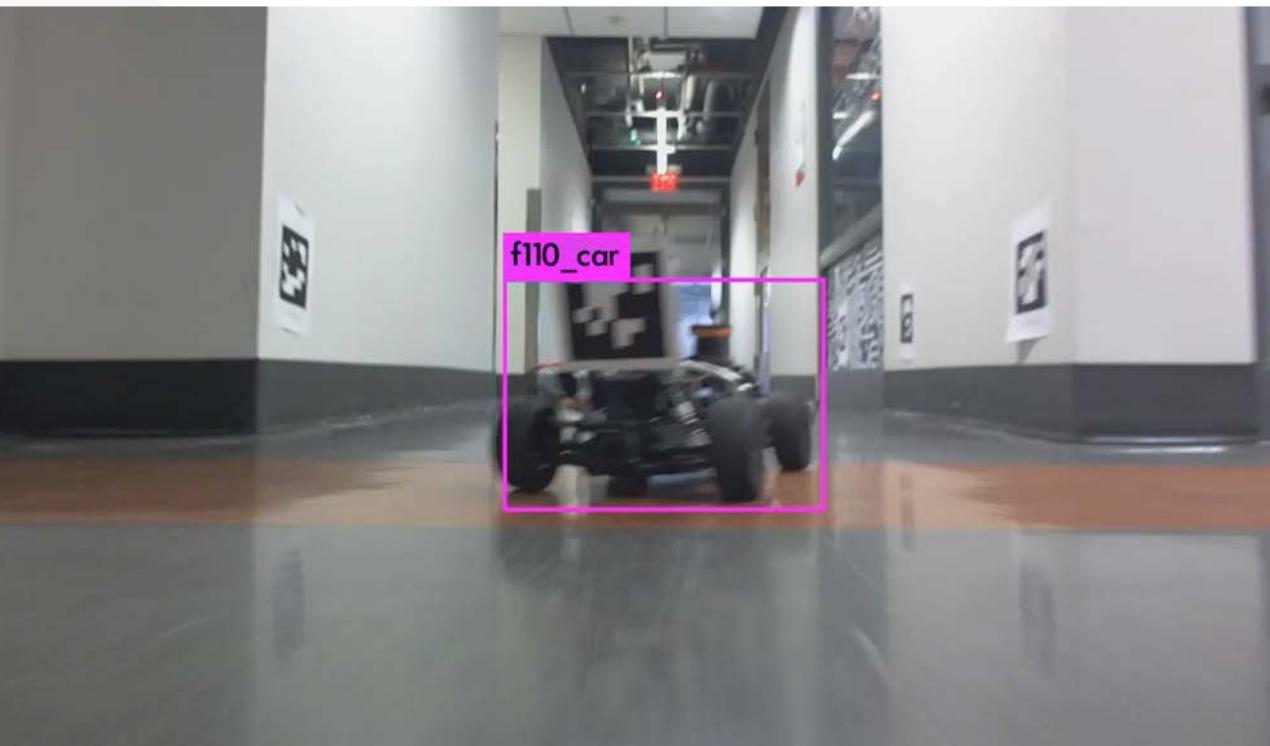
Examples



Wall Following



Deep Learning

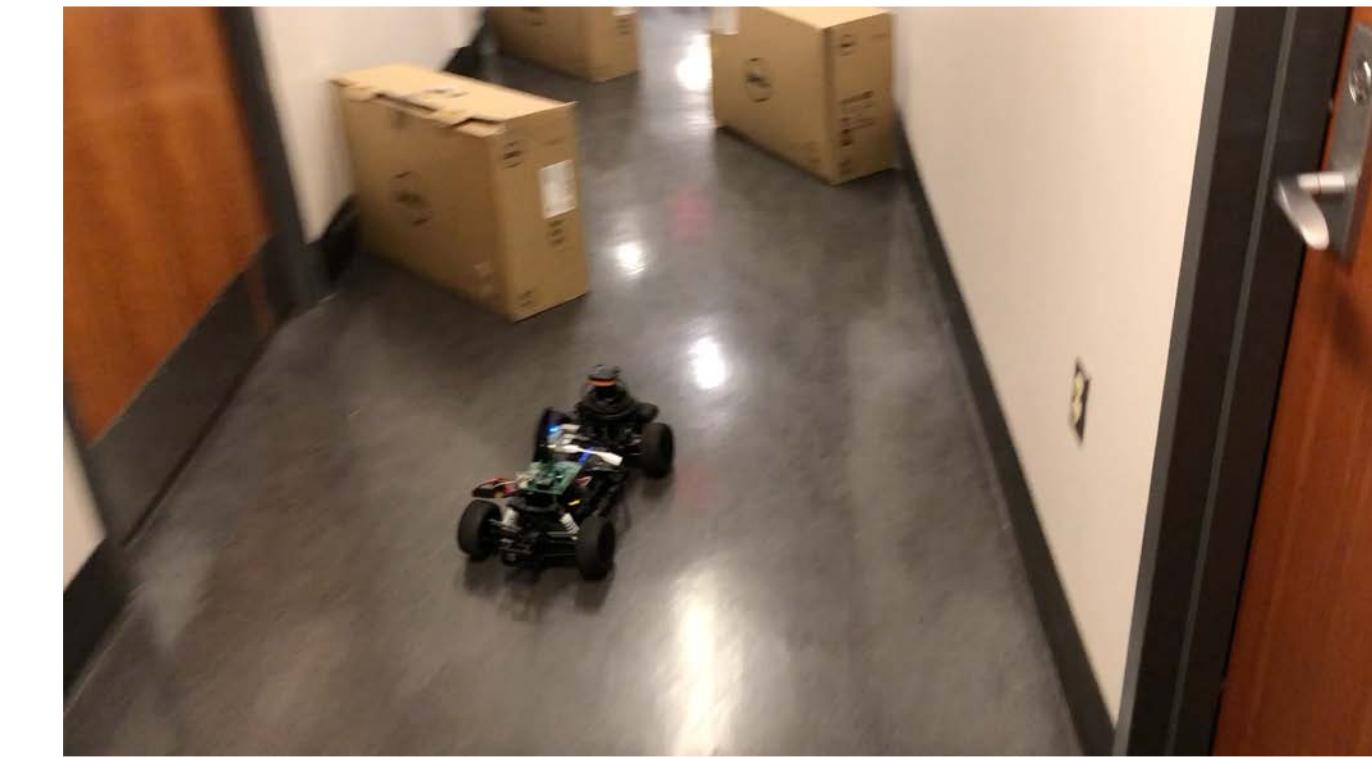


Object Detection and Tracking

Your
Project



Follow the gap



Obstacle Avoidance

ROS F1/10 Autonomous Racing Simulator

f1tenth.dev

Varundev SureshBabu

Prof. Madhur Behl

Computer Science | Link Lab | University of Virginia



Research

Localization and Mapping

Where am I ?

Scene Understanding

Where/who/what/why of everyone/everything else ?

Trajectory Planning and Control

Where should I go next ?

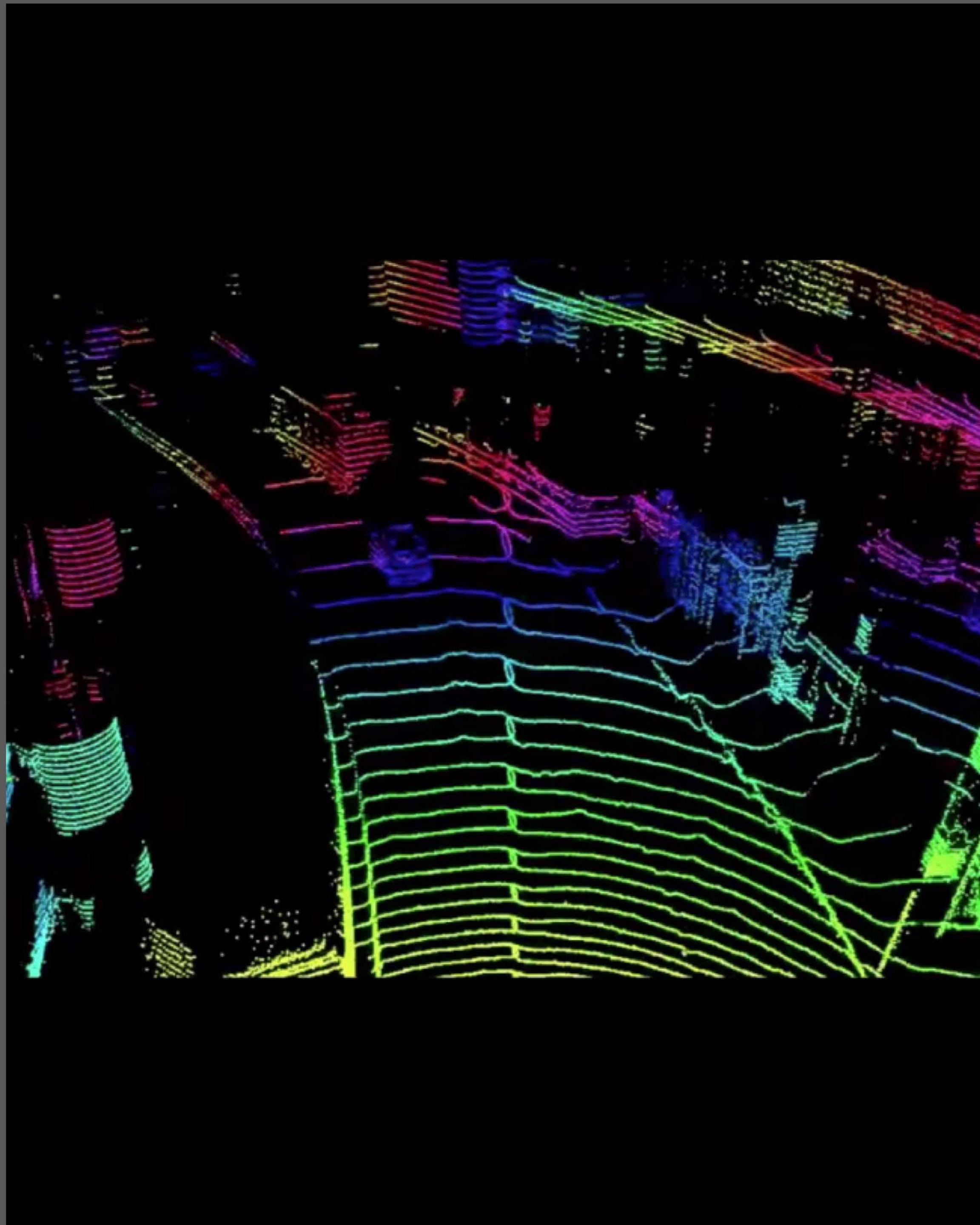
How do I steer and accelerate ?

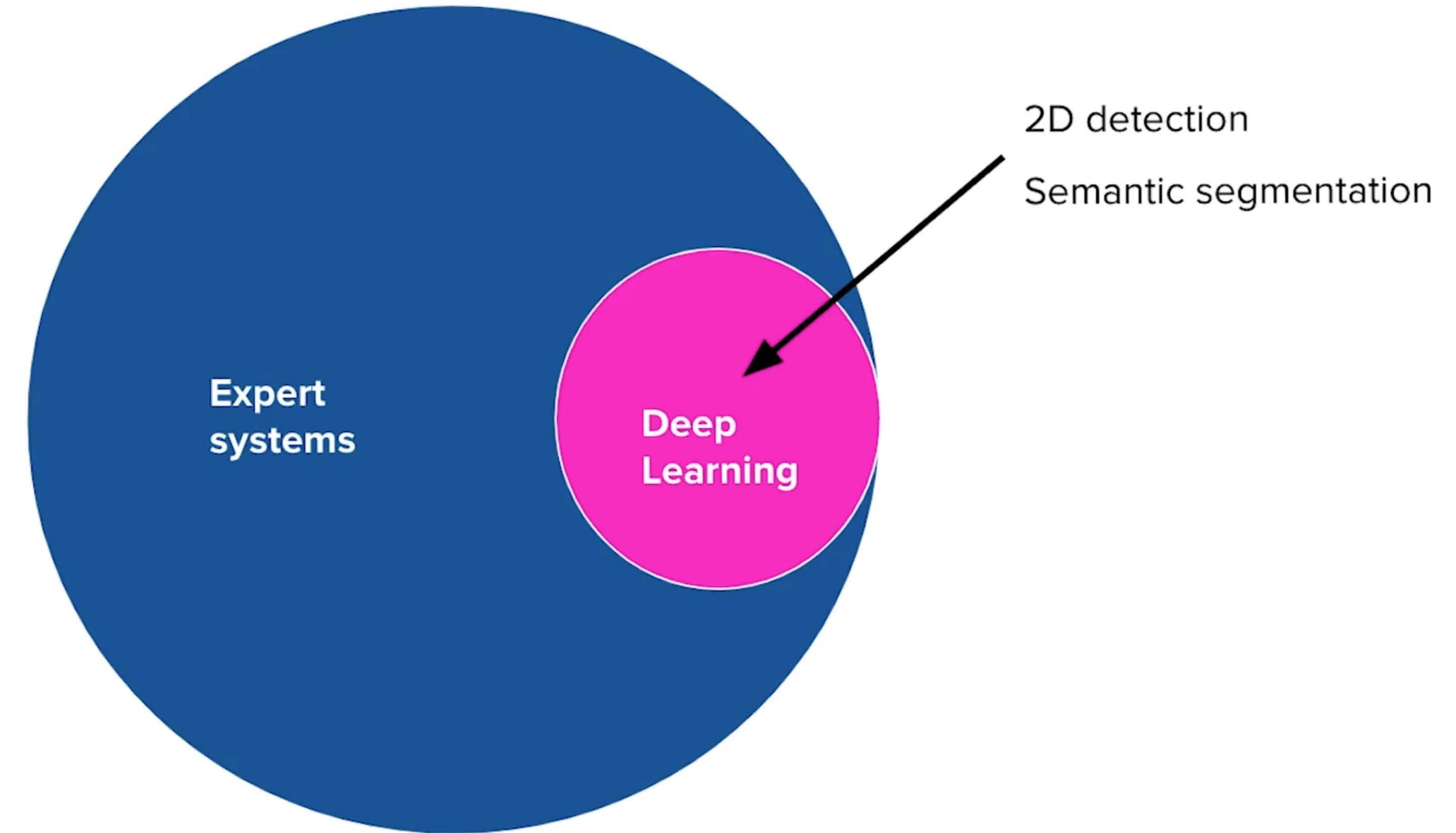
Human Interaction

How do I convey my intent to the passenger and everyone else ?

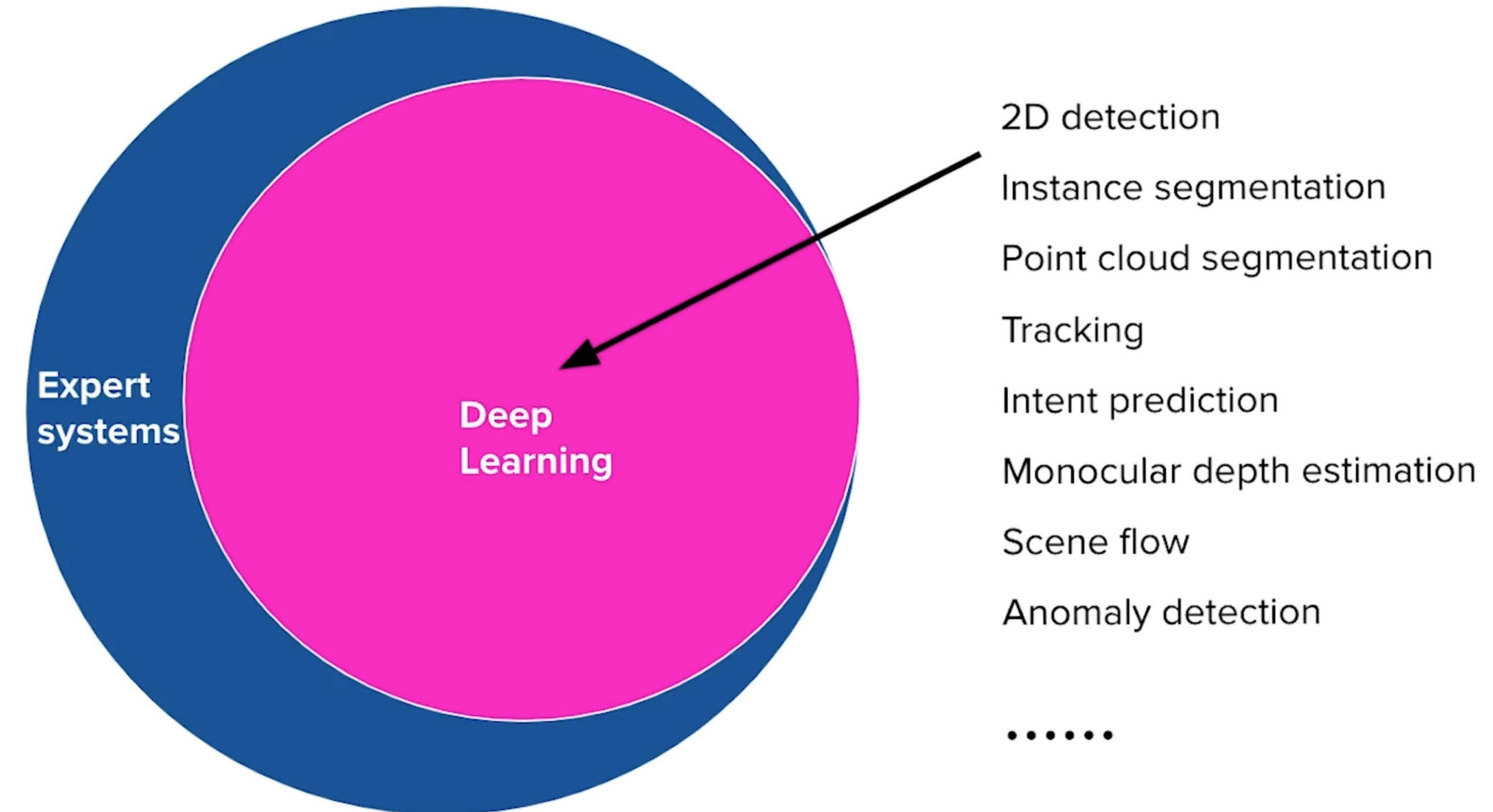
Deep Neural Networks

End-to-end learnable



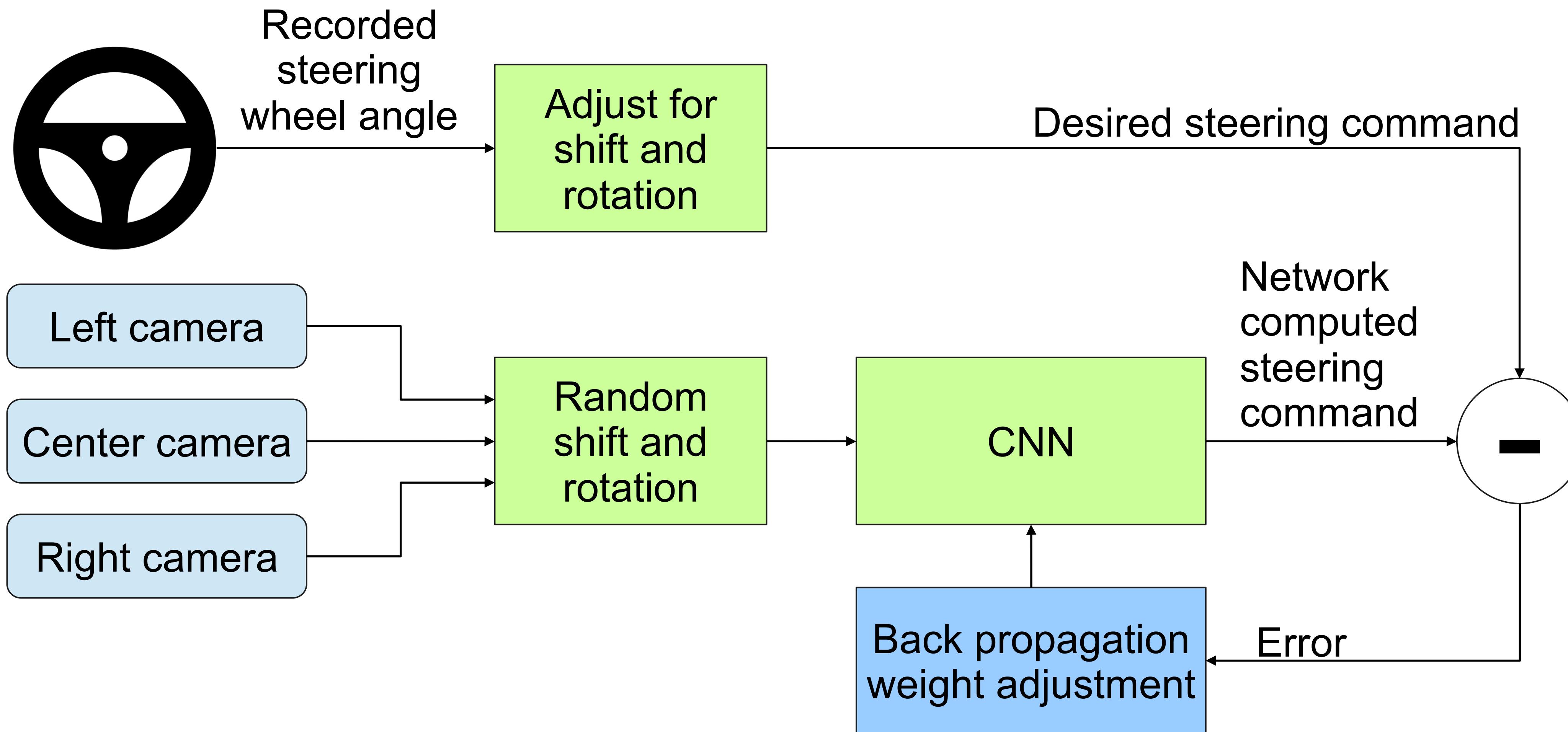


AV Perception in 2015

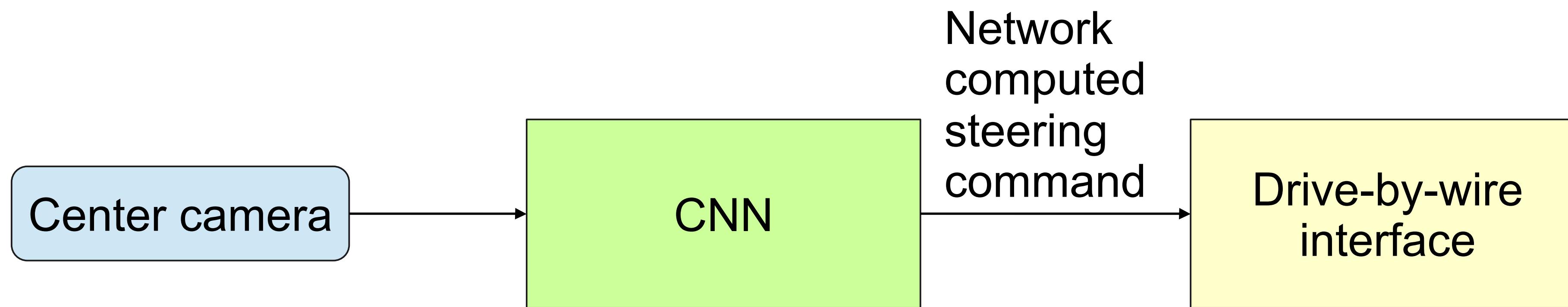


- 2D detection
- Instance segmentation
- Point cloud segmentation
- Tracking
- Intent prediction
- Monocular depth estimation
- Scene flow
- Anomaly detection
-

End-to-End Driving: PilotNET

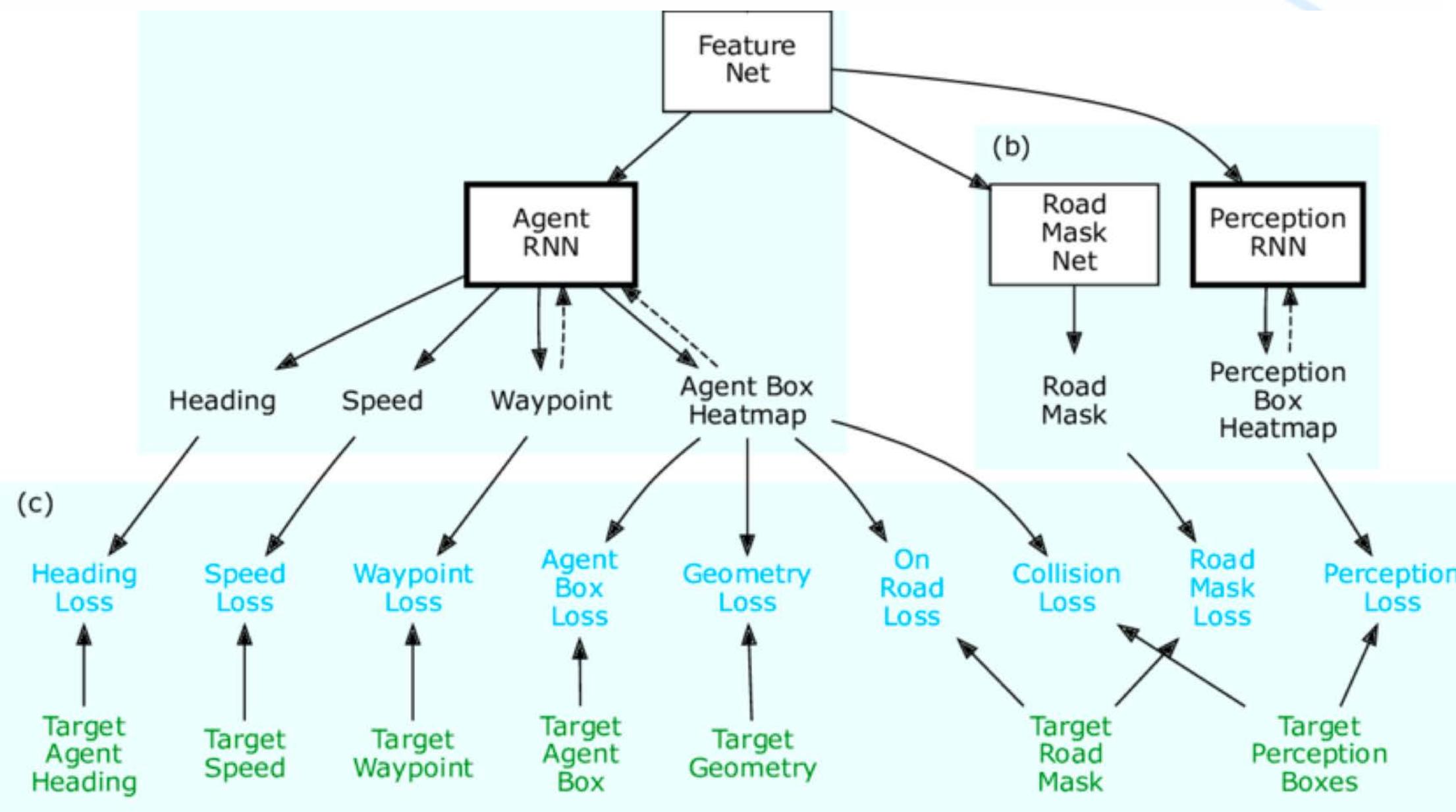
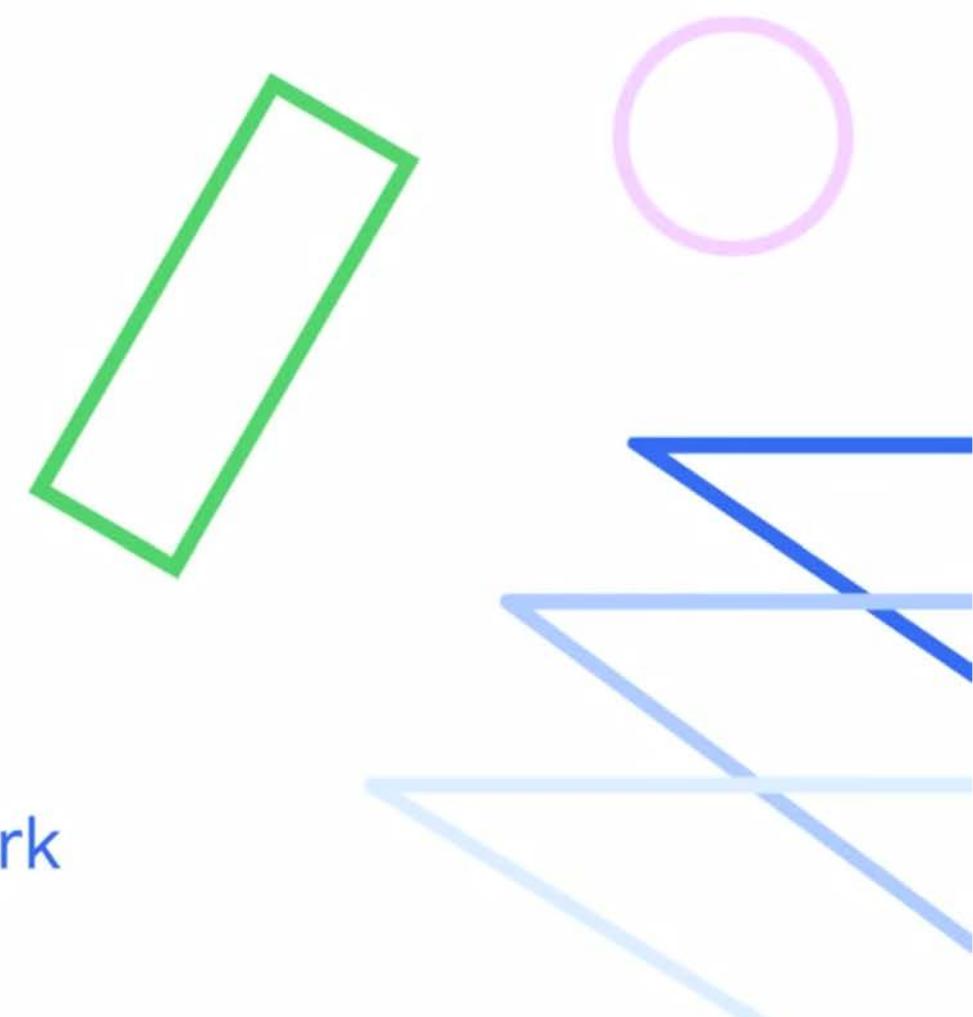


With a single front-facing camera

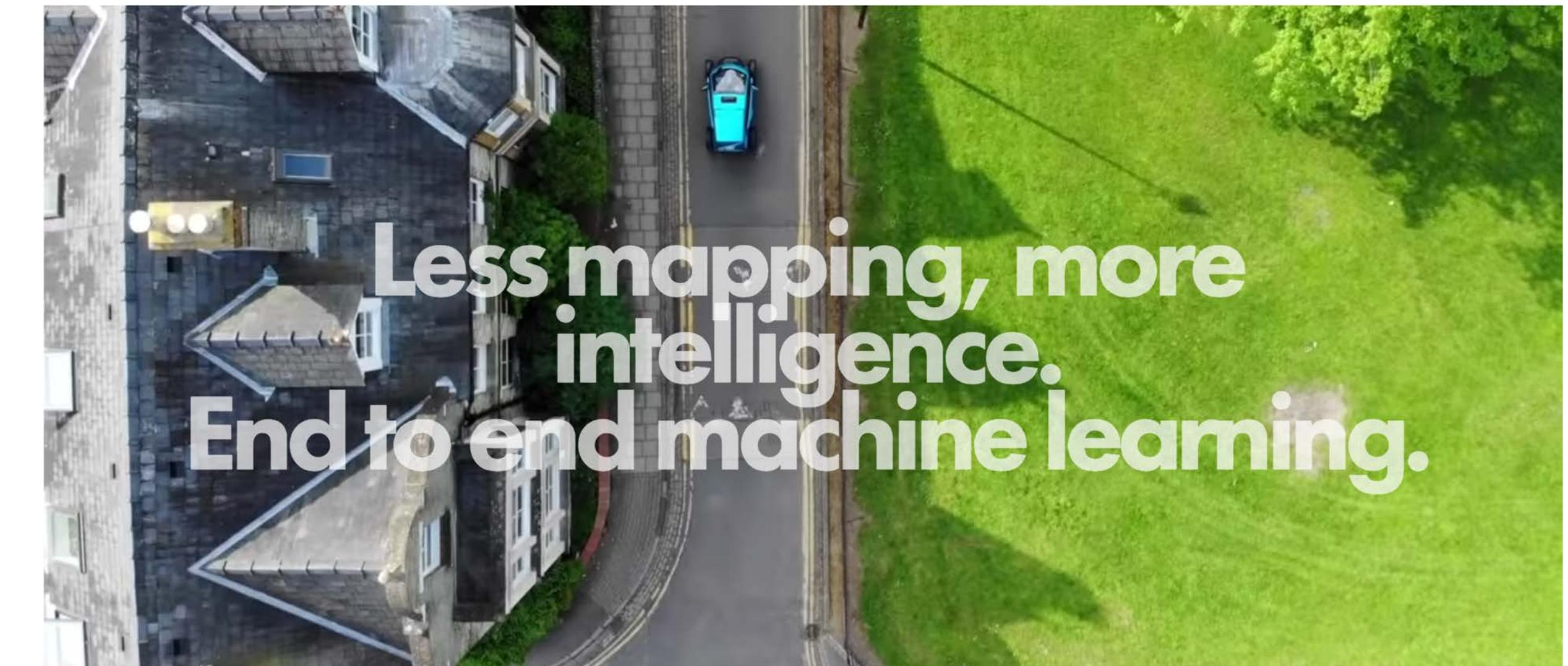


ChauffeurNet

A Deep Learned Driving Network

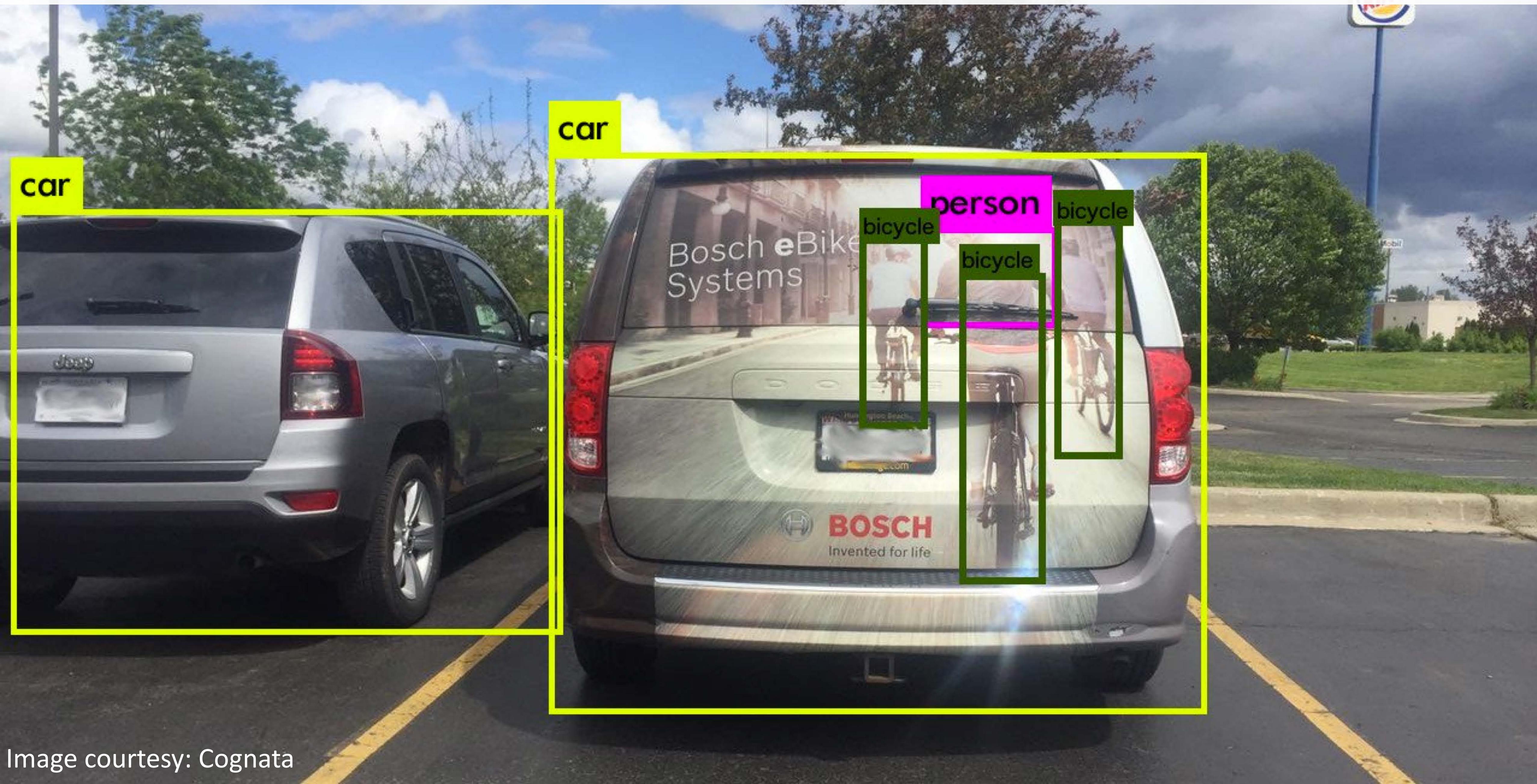


WAYMO



Machine intelligence is largely about training data.

When's a pedestrian not a pedestrian? When it's a decal.



One car ? or Multiple cars ?





Ramen Noodle place or Do Not Enter Sign ?









There is a bus right next to you!!





How can we ensure that an autonomous vehicle drives safely upon encountering an unusual traffic situation ?



DeepRacing AI

 UVA ENGINEERING
LINK LAB

BRUTE FORCE

Drive millions of miles in simulation or in the real world; too slow; infinite situations

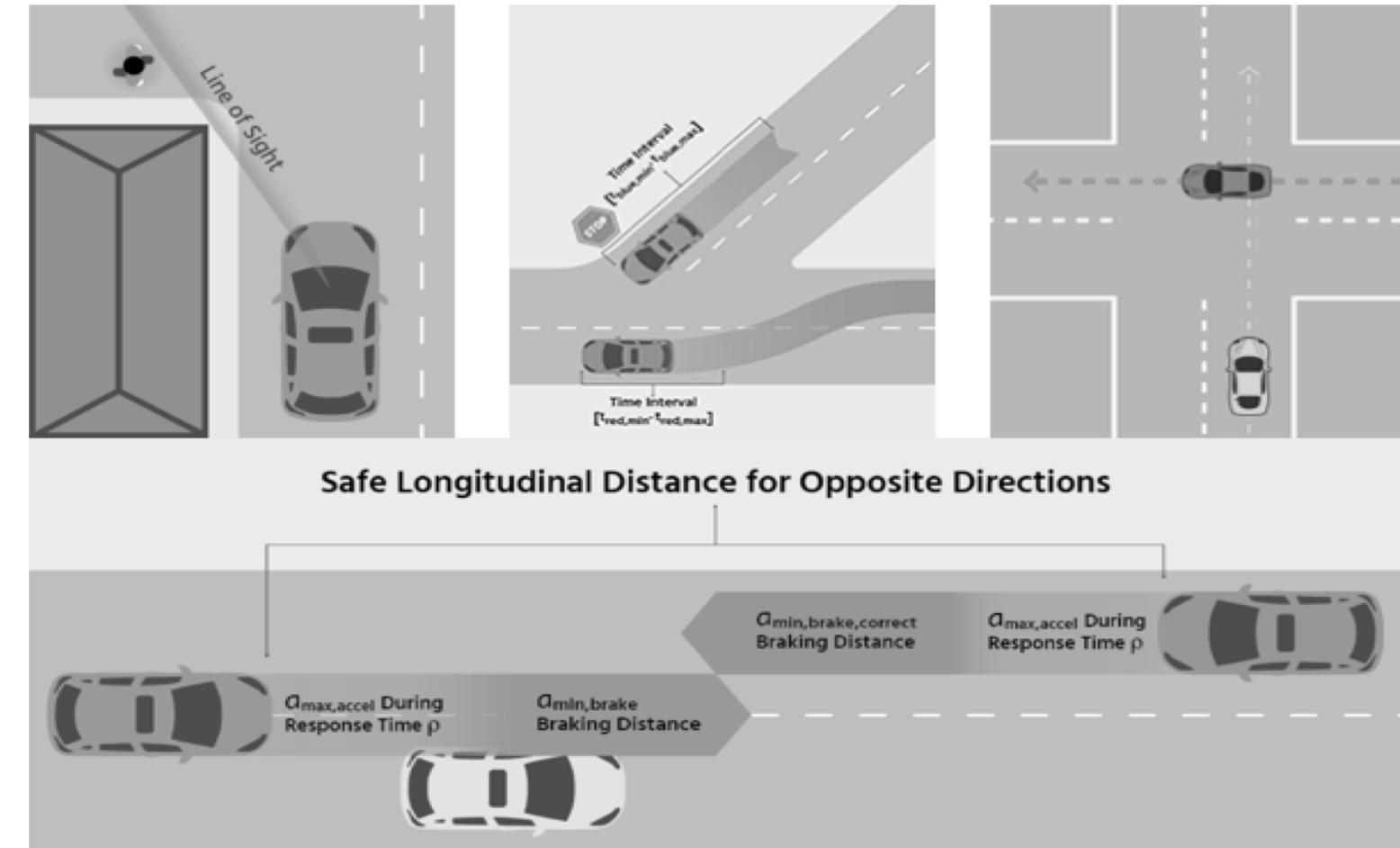


Current Attempts To Ensure Safety Of Autonomous Vehicles

How can we ensure that an autonomous vehicle drives safety upon encountering an unusual (edge) traffic situation ?

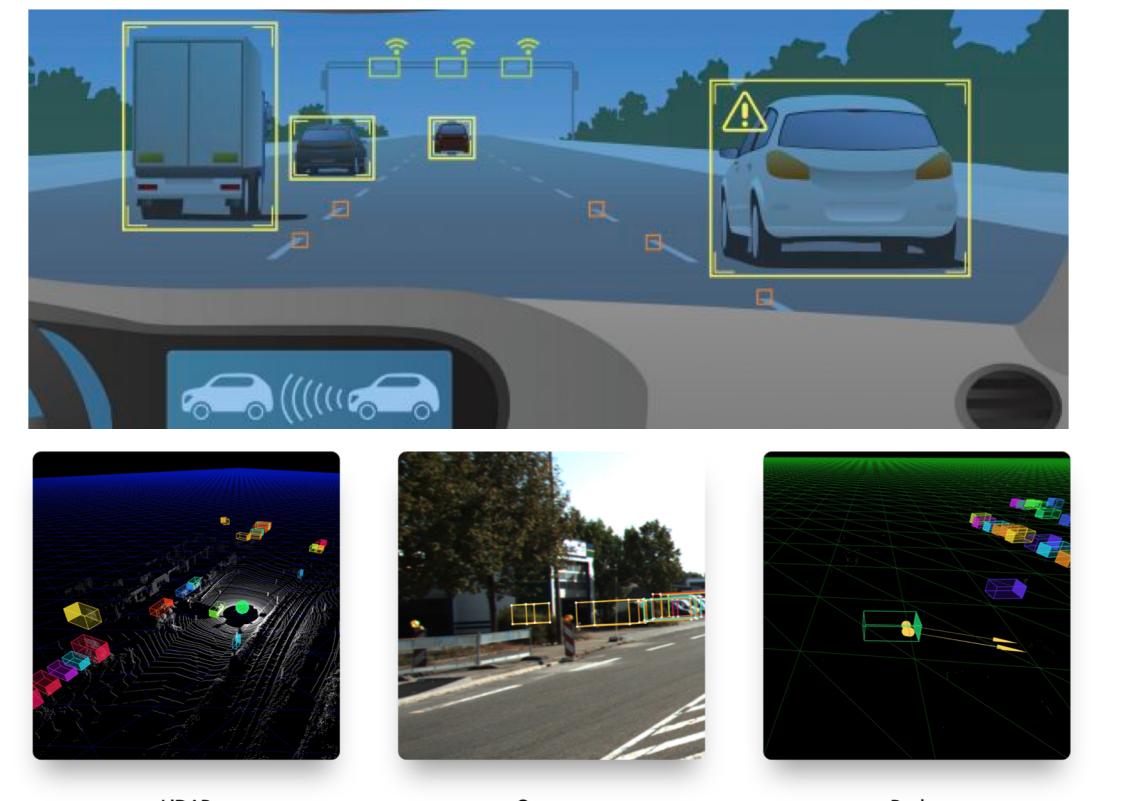
FORMAL METHODS

Reasoning about scenario specific safety; often too simplistic to generate guarantees



Safety Through Agility

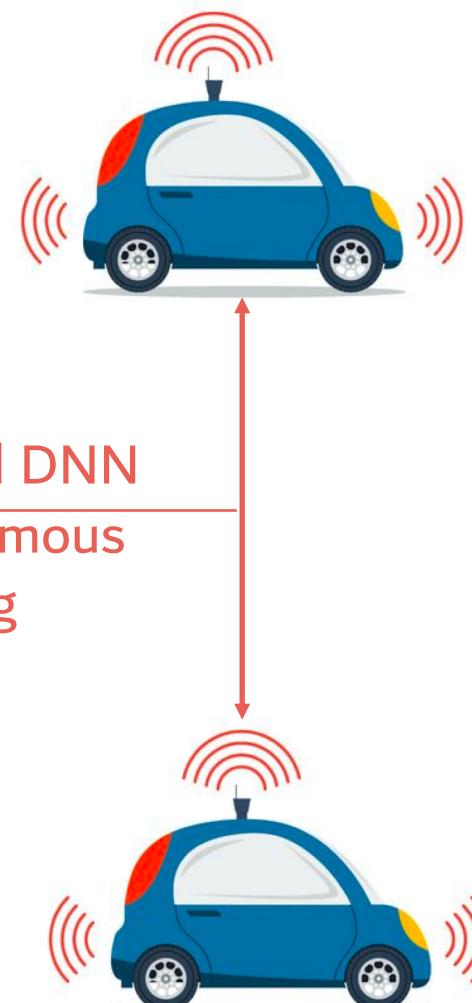
Autonomous vehicles get trained on mostly structured traffic data-sets.



AV: A

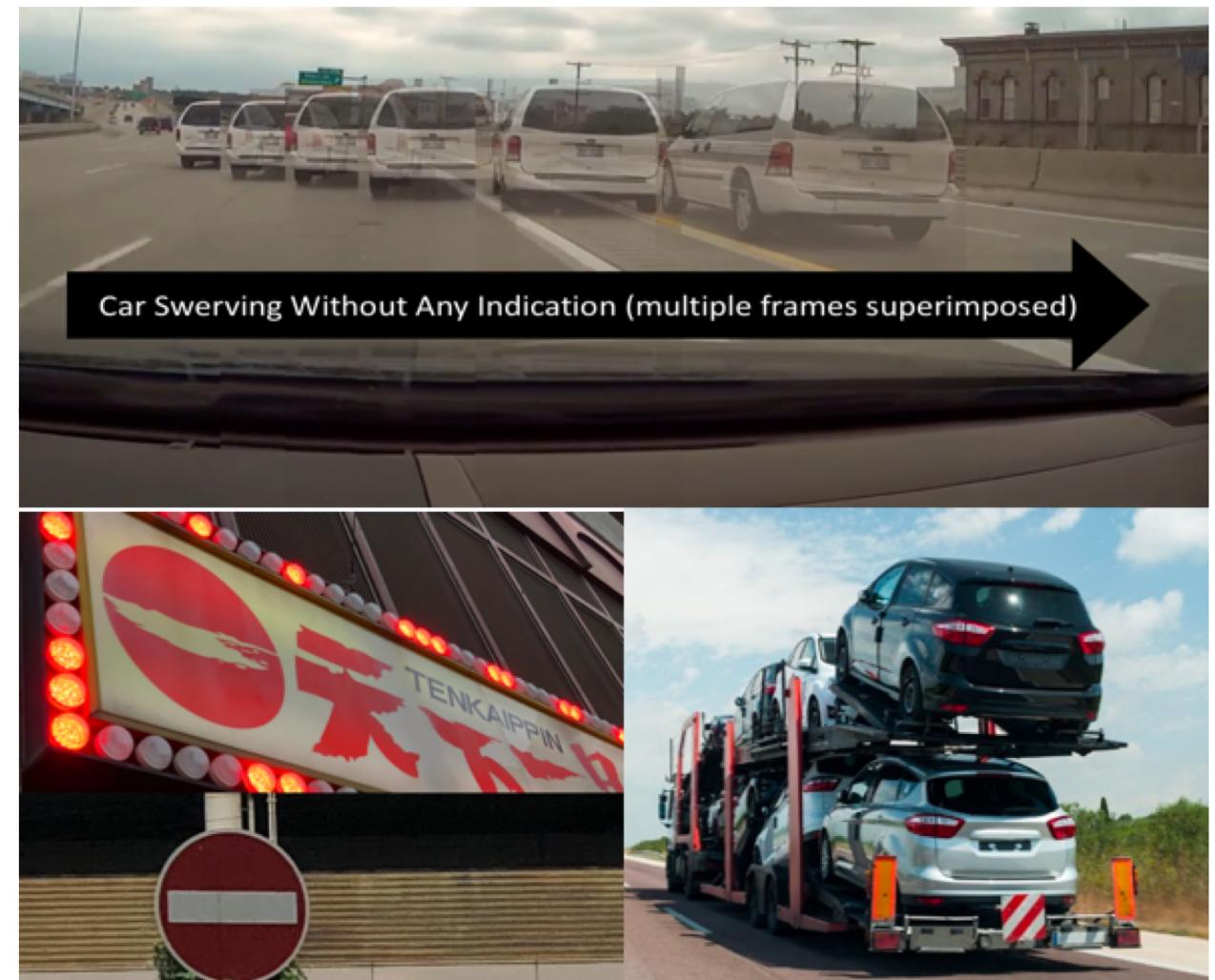
End-to-end DNN
for autonomous
driving

AV: B



DeepRacing
Agile Controller

AV: B is safer
overall.



Teaching autonomous vehicles to be agile by
autonomously
racing them against each other.

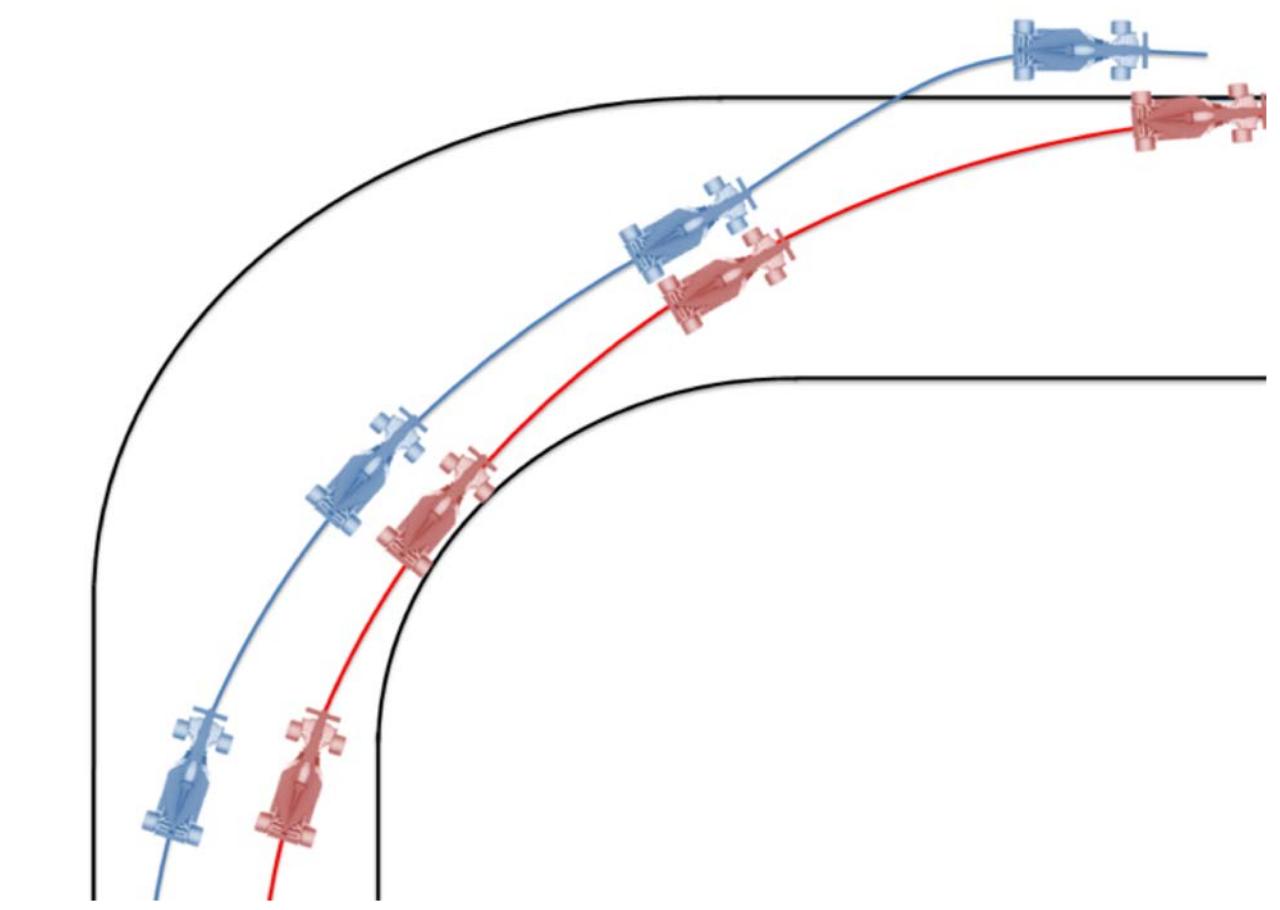
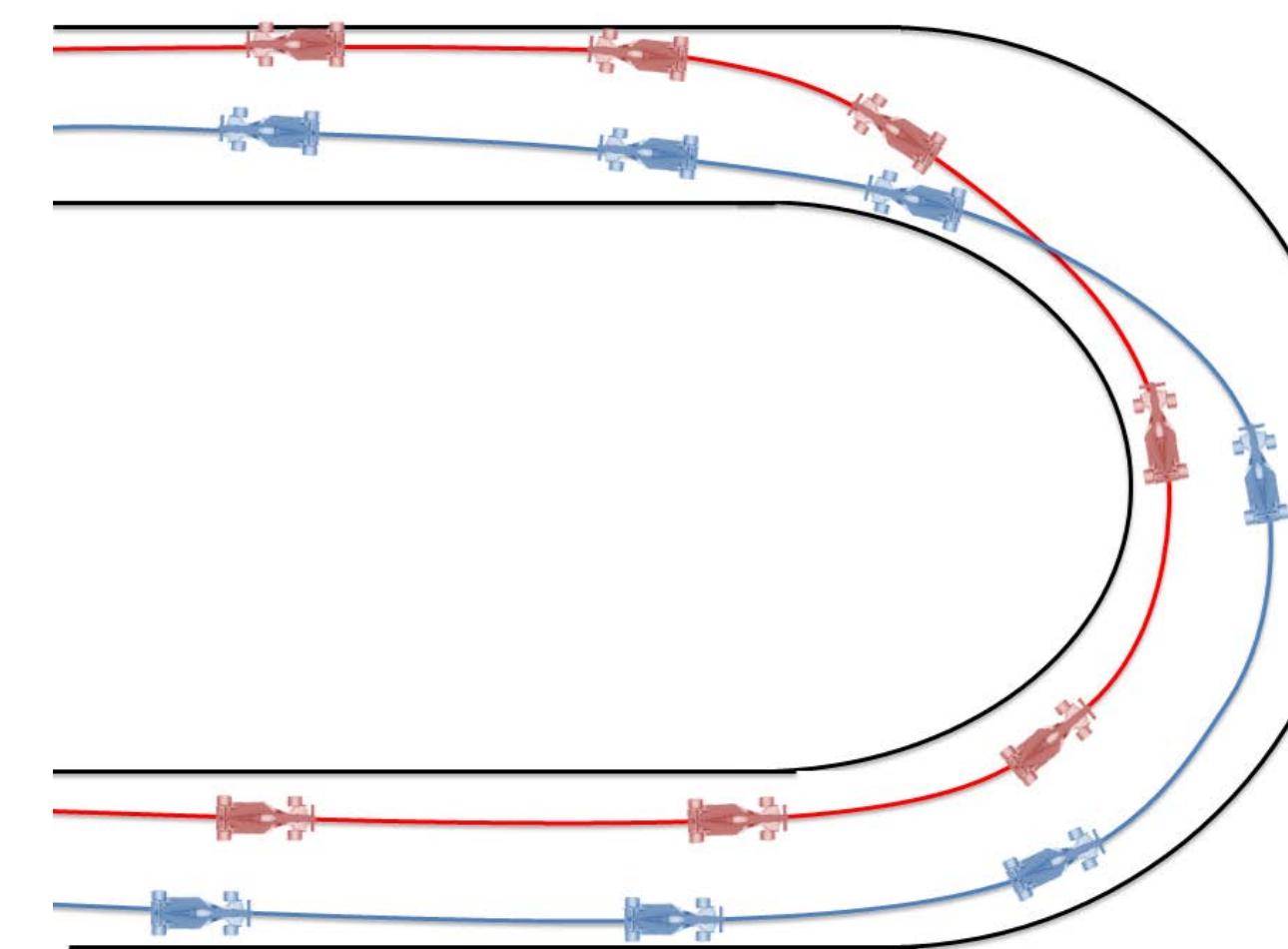
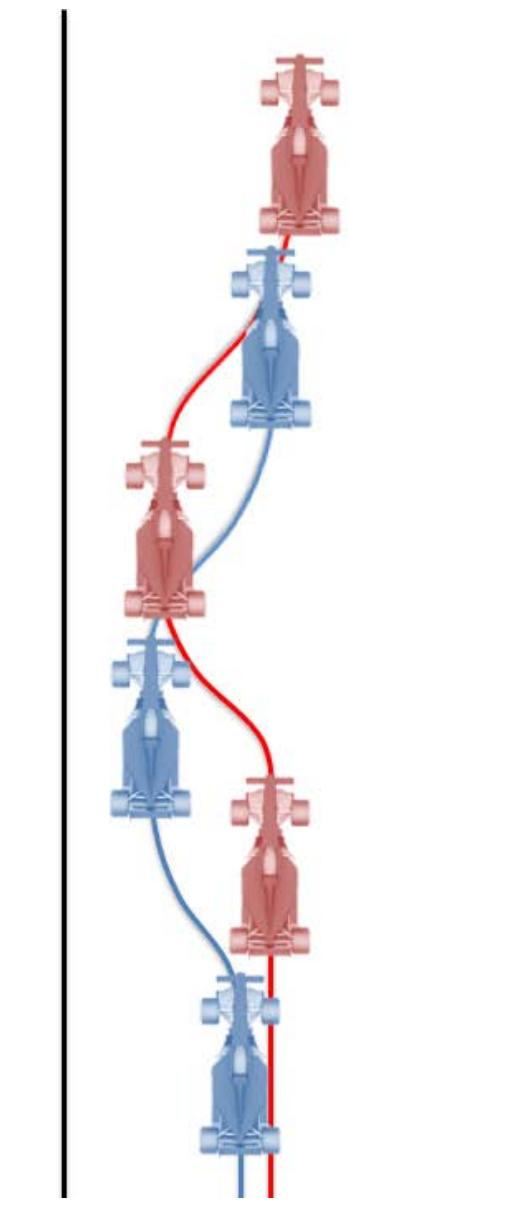
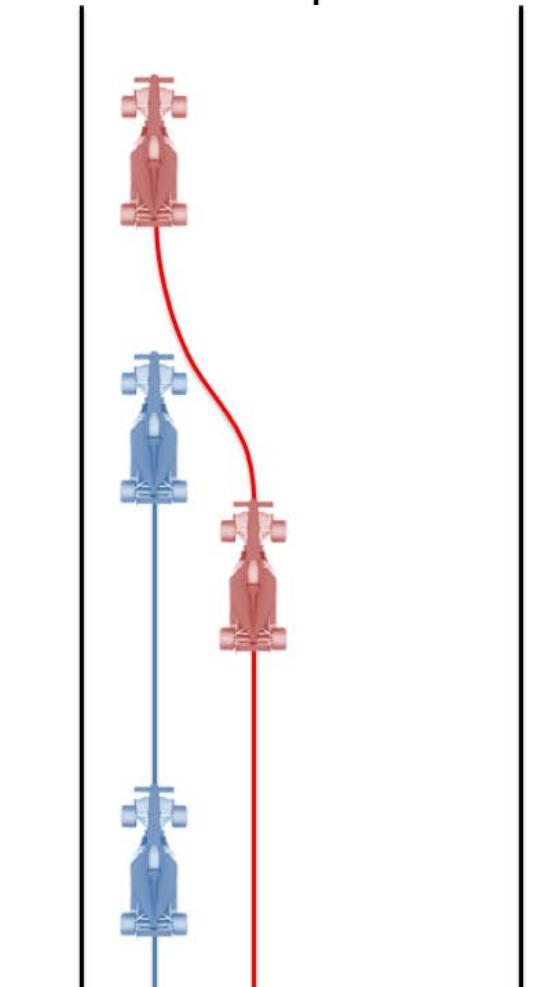
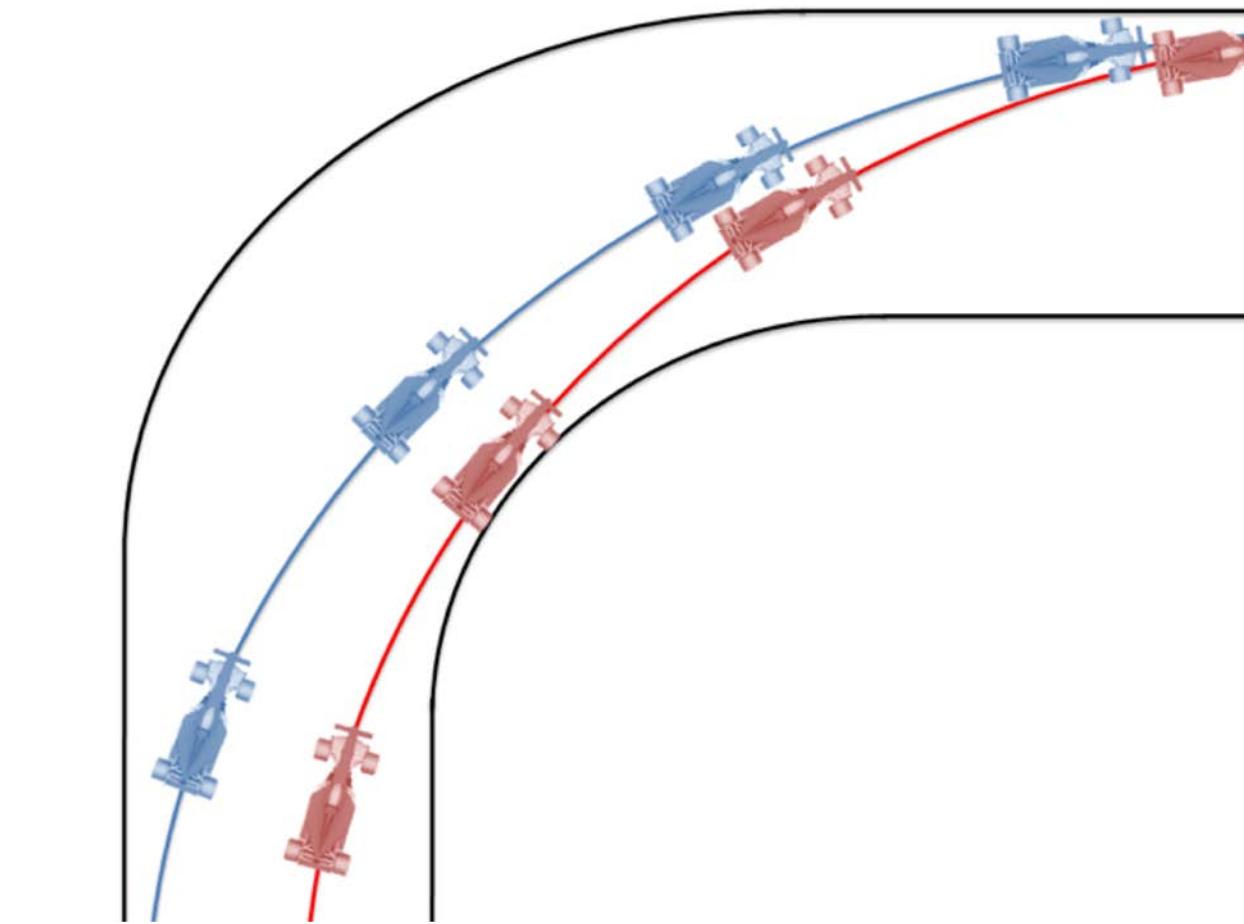
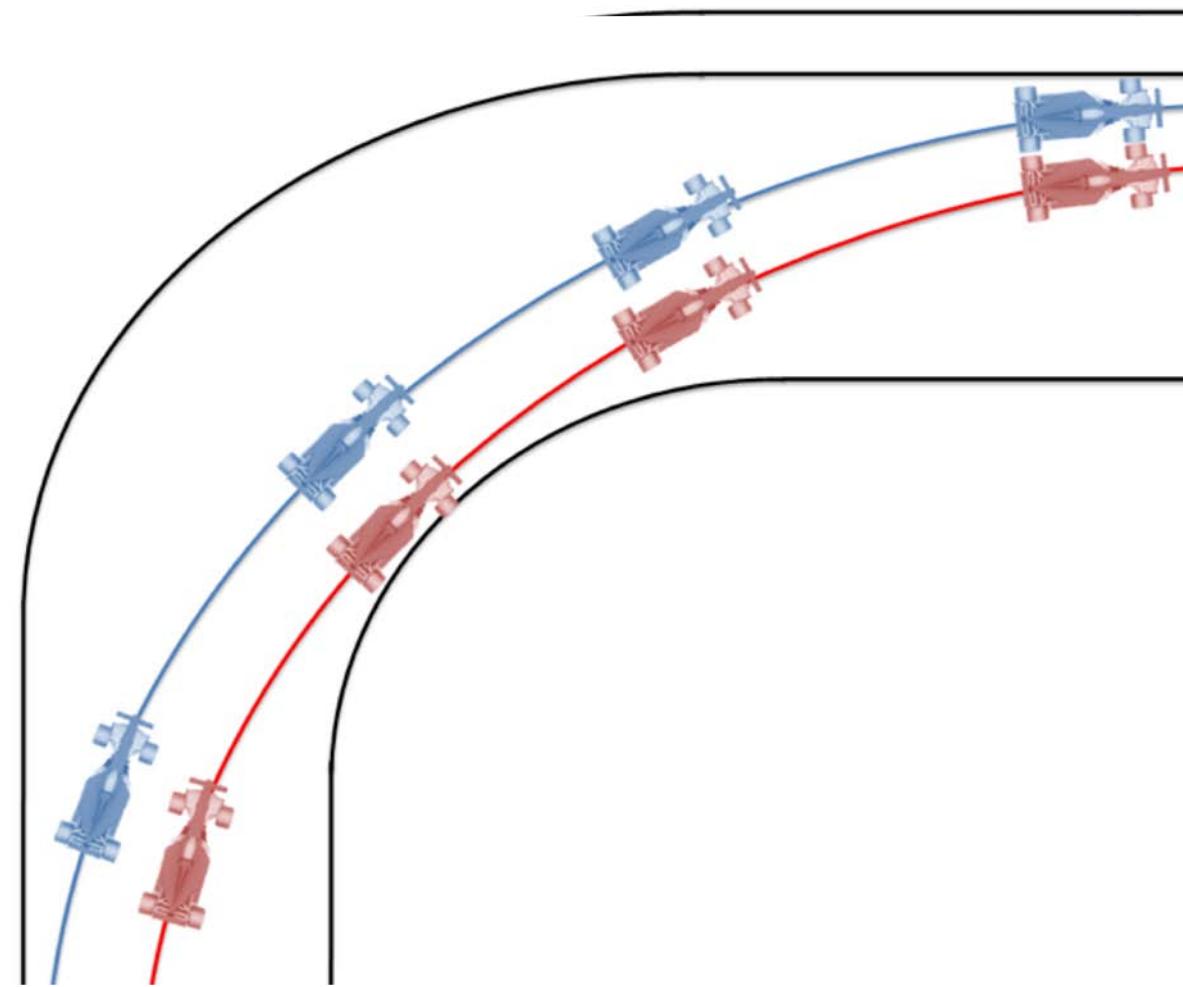
REPLAY



“If everything seems under control,
then you are not going fast enough”



Racing offers more frequent edge cases





1894... the world's first "horseless carriages" race,
Paris to Rouen.

1/12 LAPS

13B



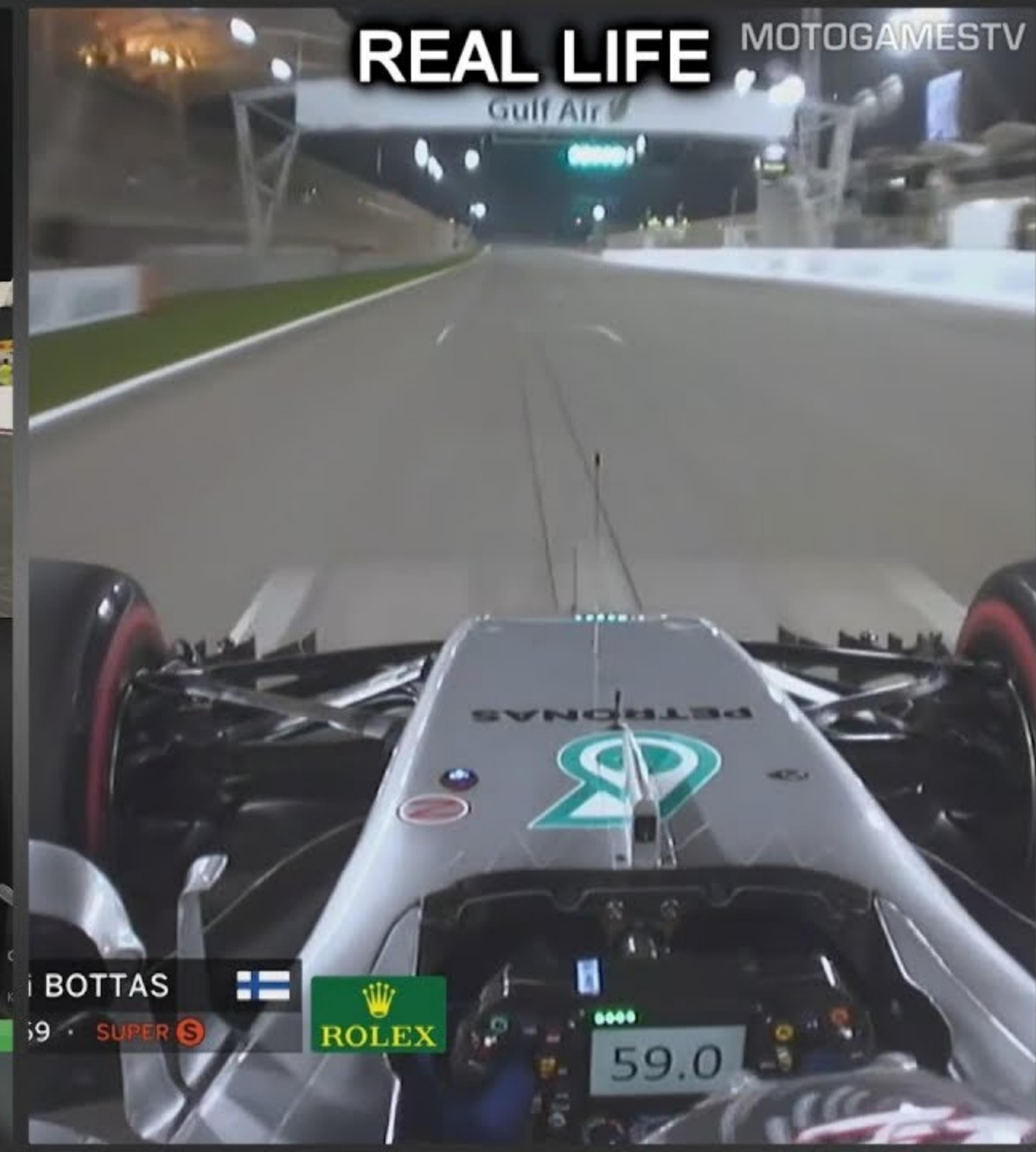
GT
FOCUS

OK FIA

GAME

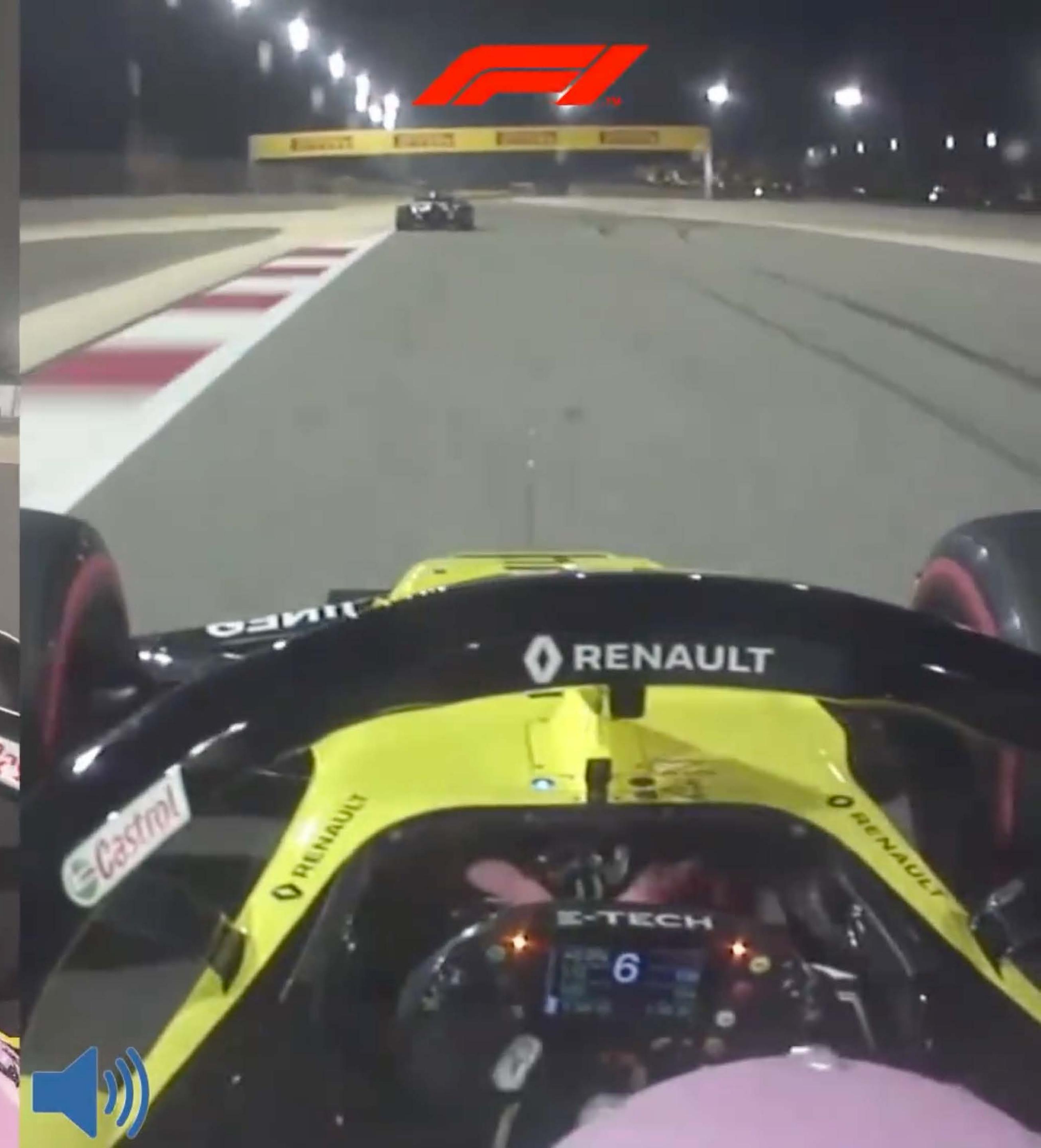


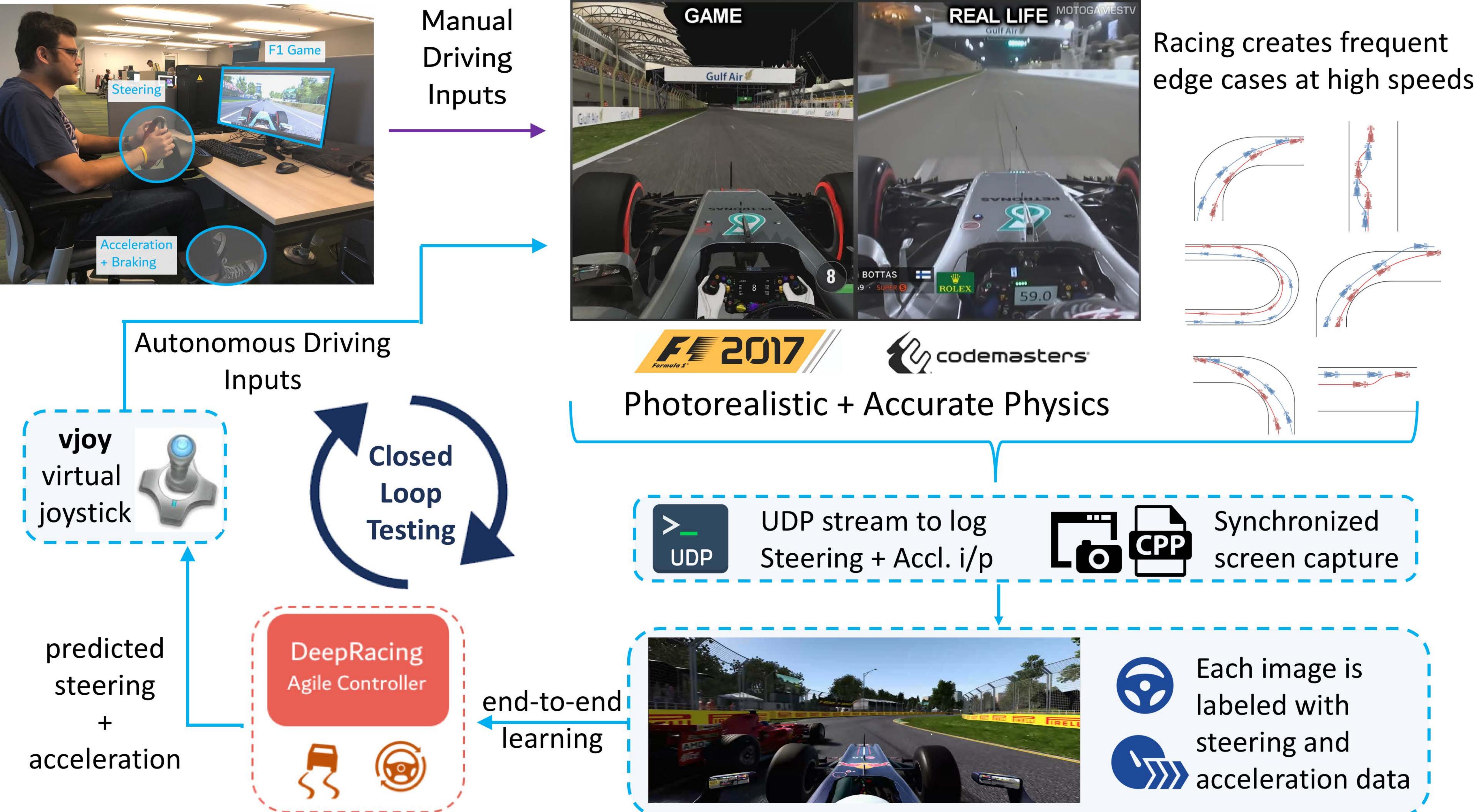
REAL LIFE

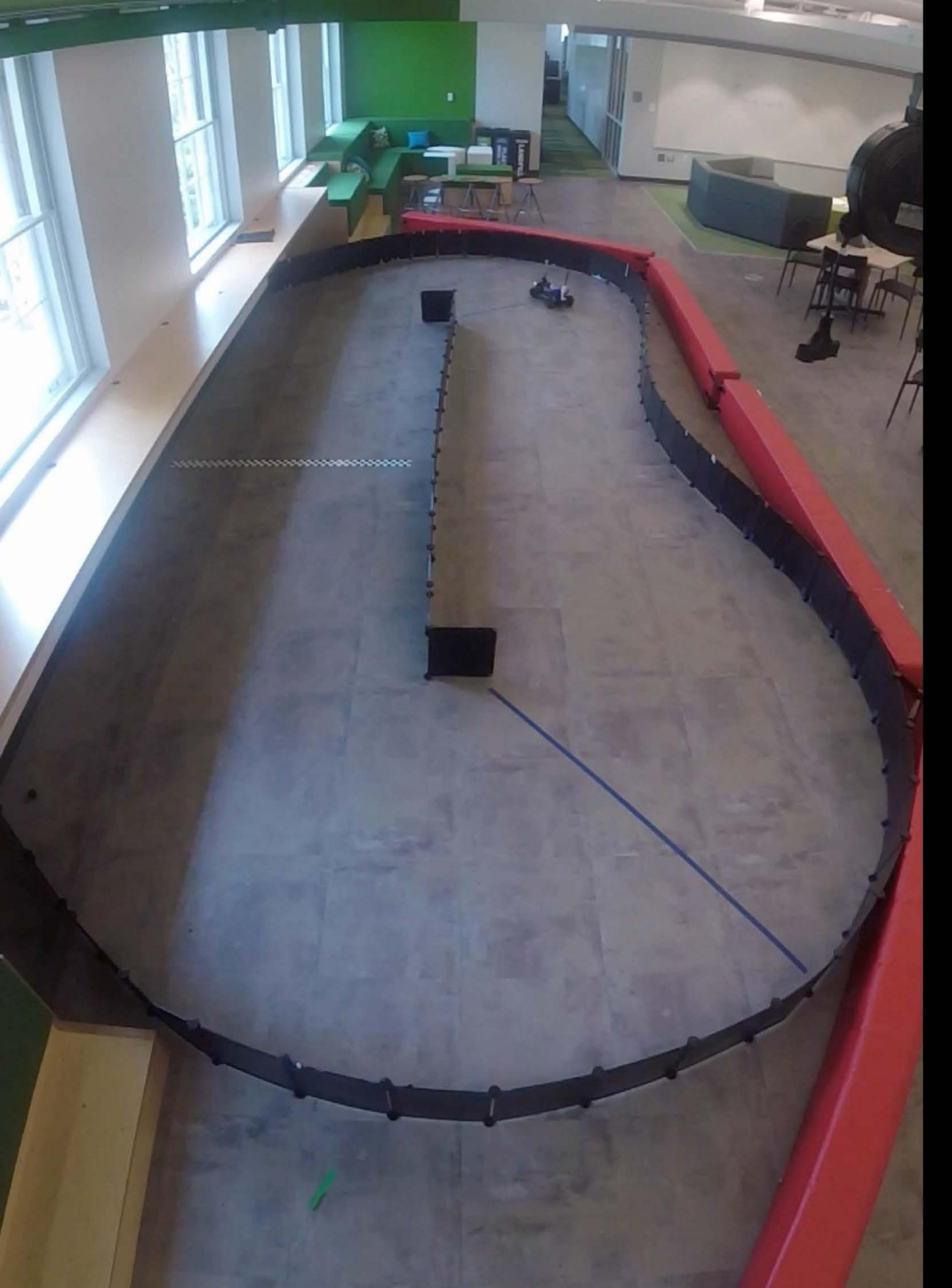


MOTOGAMESTV

F1 2019
THE OFFICIAL VIDEOGAME

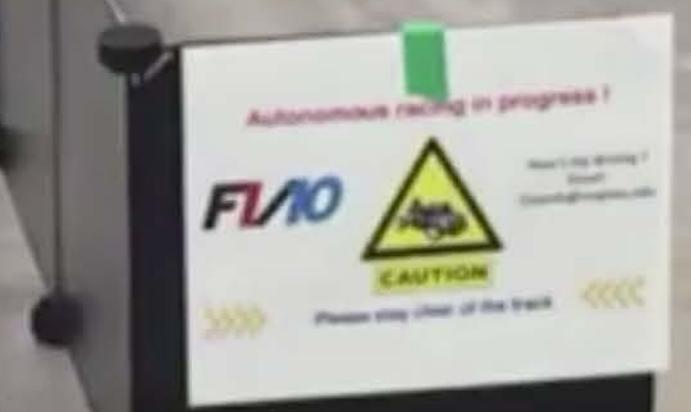






F1/10 FPV Driving





Autonomous racing in progress!

Fun!



From F1/10 To F1



Autonomous Indy Racing Challenge



Ca^rt^aalier
Autonomous Racing

HOOS Racing !!

BEFORE THE COURSE

I know programming

AFTER THE COURSE

ROS, knows I



madhur.behl@virginia.edu