Self-Driving Vehicular Project

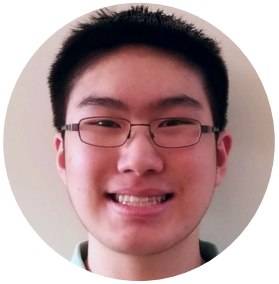
Team Members:
Adarsh Narayanan, Joshua Menezes, Christopher Sawicki, Tommy Chu, Brandon Cheng, Ruben Alias, Aleicia Zhu, Ranvith Adulla, Arya Chhabra, Suhani Sengupta

This work was supported in part by the NSF REU program and the donation from nVERSES CAPITAL

Introducing the Team

Adarsh Narayanan
Rutgers ECE UG

Christopher Sawicki
Cornell MechE UG



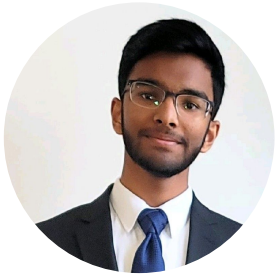
Brandon Cheng
Rutgers ECE UG

Aleicia Zhu
High School

Suhani Sengupta
High School

Joshua Menezes
Rutgers ECE UG

Tommy Chu
Rutgers ECE UG



Ruben Alias
Rutgers ECE UG

Ranvith Adulla
High School

Arya Chhabra
High School

Goal:

Build two model cars that are able to drive through the miniature smart city

RASCAL

Robotic Autonomous Scale Car for Adaptive Learning

- Restricted to real car movement
- Use machine learning to drive autonomously

SCAMP

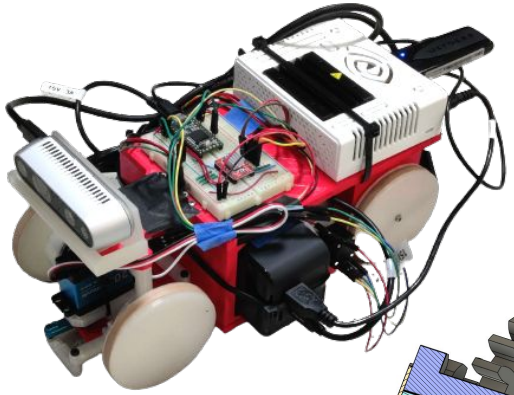
Self-guided Computer Assisted Mecanum Pathfinder

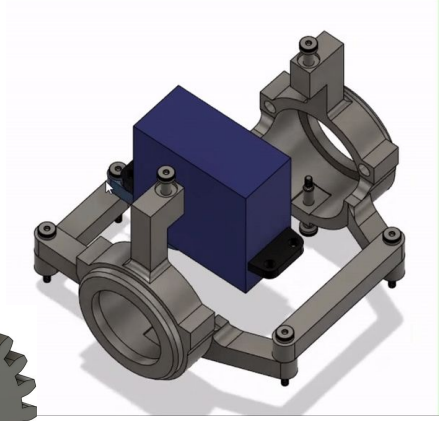
- Mimic a real car's path
- Simulate traffic for autonomous car

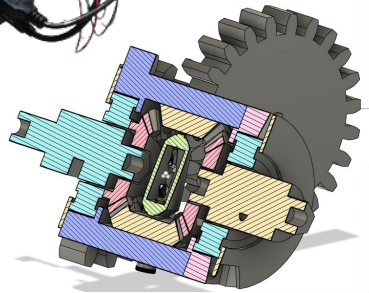
RASCAL

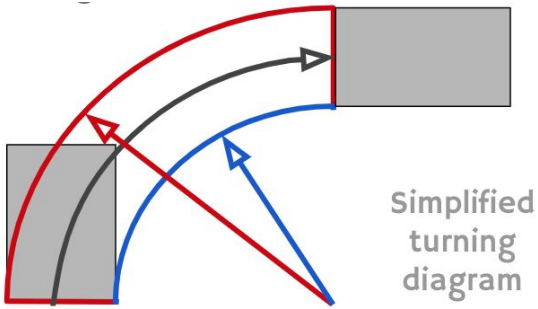
(Robotic Autonomous Scale Car for Adaptive Learning)

- Ackerman Steering
- Differential gear system
- Imitate a real car's motion

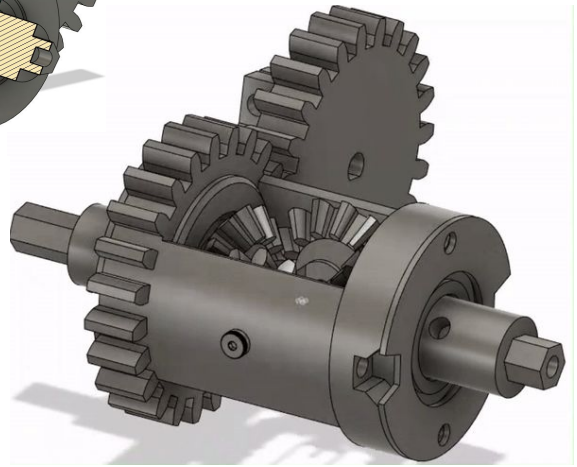




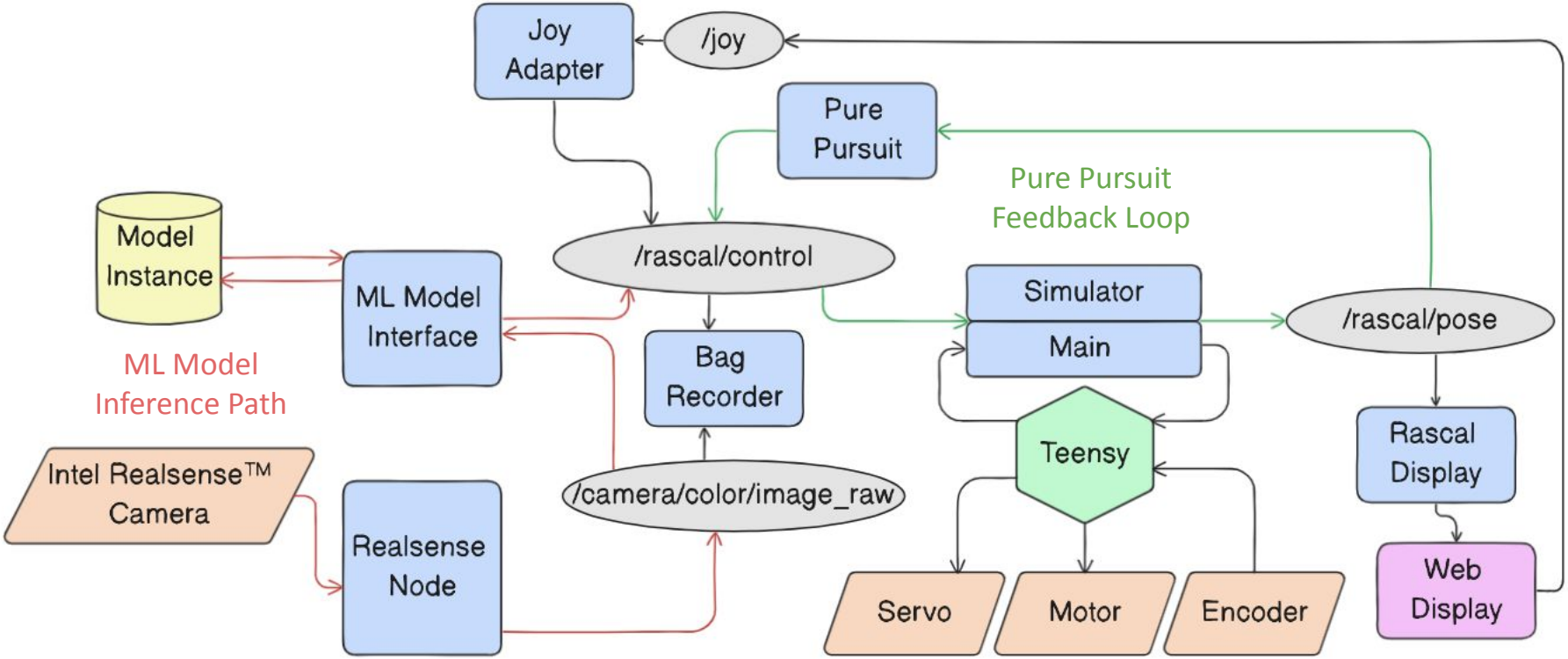




Simplified turning diagram



Software Architecture



Web Display

- Provides visual interface
- ROS node runs Flask server
- Add commands and points from any ROS node

e.g. city outline,

commands for car parameters (speed & pos),

editing paths/pure-pursuit

Commands

get commands

Path/AddPoint

x: 0 y: 0 Run Command

Path/DeletePoint

index: 0 Run Command

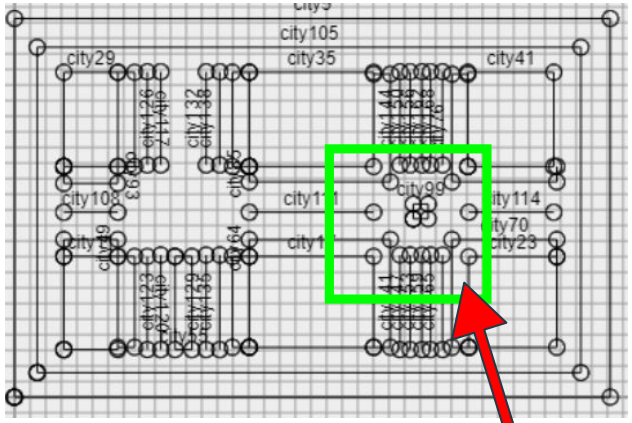
Path/GetLookahead

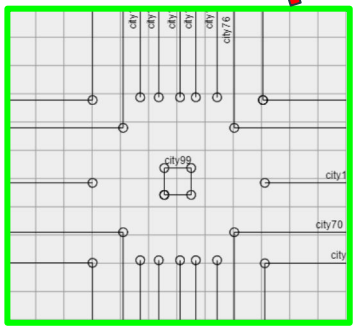
lookahead_dist: 1 start_index: 0 x: 0 y: 0 Run Command

Path/InsertPoint

index: 0 x: 0 y: 0 Run Command

Graphs

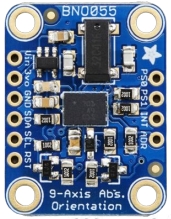


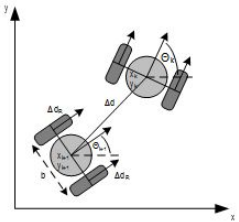


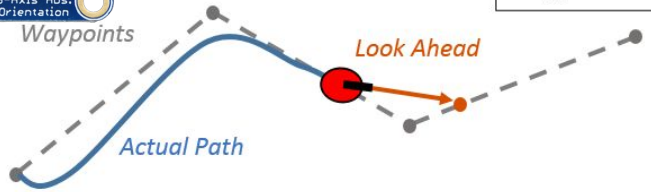
Intersection

Pure Pursuit

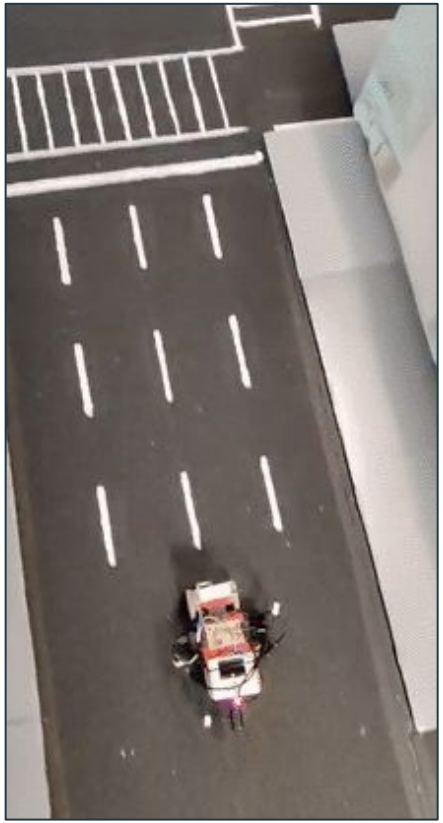
- Goal: Consistently follow a path
- Path following algorithm
  - Create path on web display
  - Car self-adjusts to stay on path
- Odometry to track car position





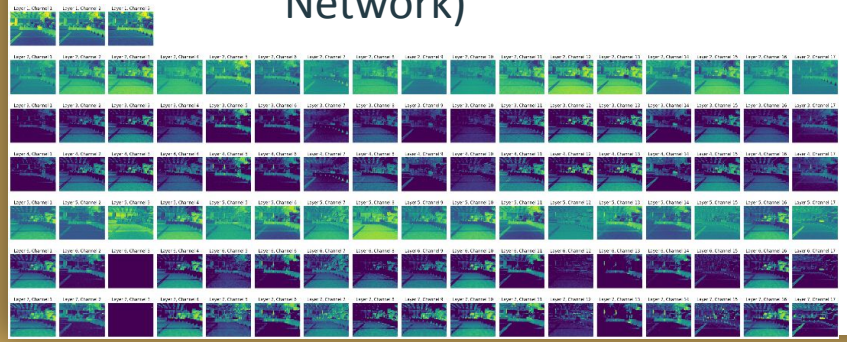


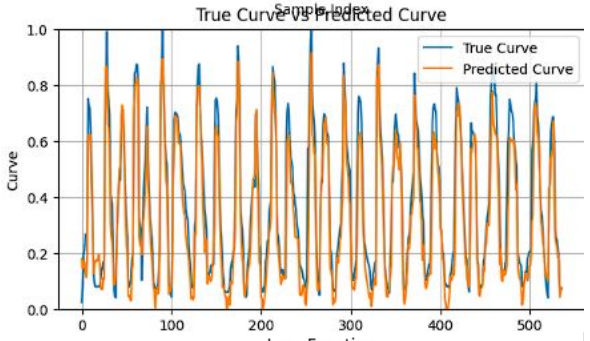
Results:

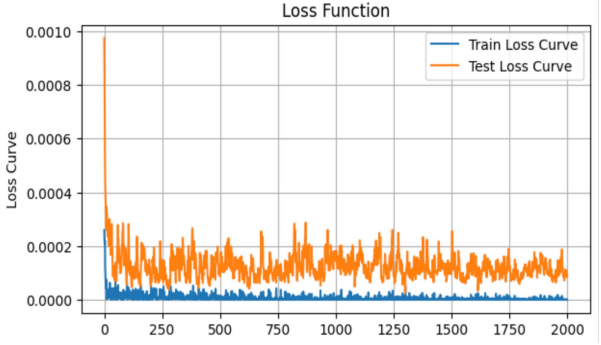


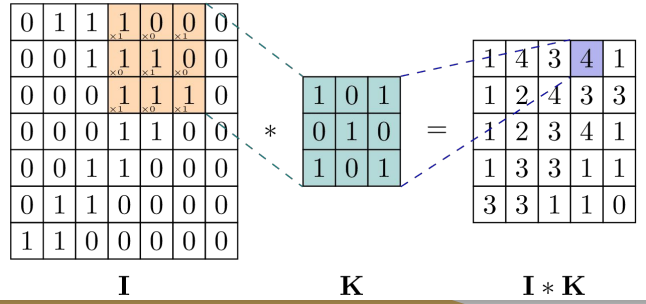
Machine Learning

- Goal: drive using just camera feed
- Does **not** require pre-programmed path
- Learns from example data collected by pure pursuit
- Pytorch
  - Imitation Learning Model
  - CNN (Convolutional Neural Network)

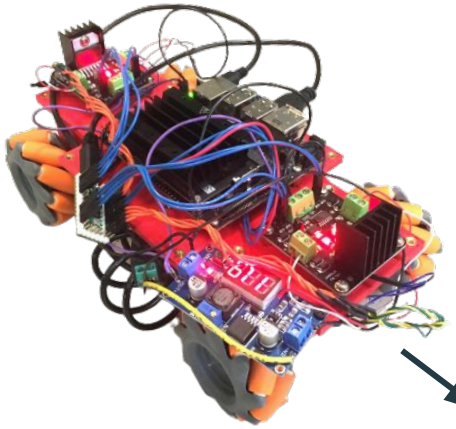




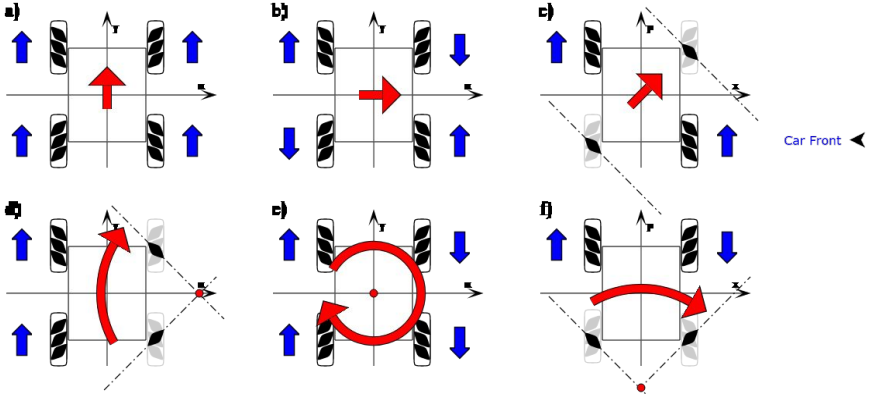


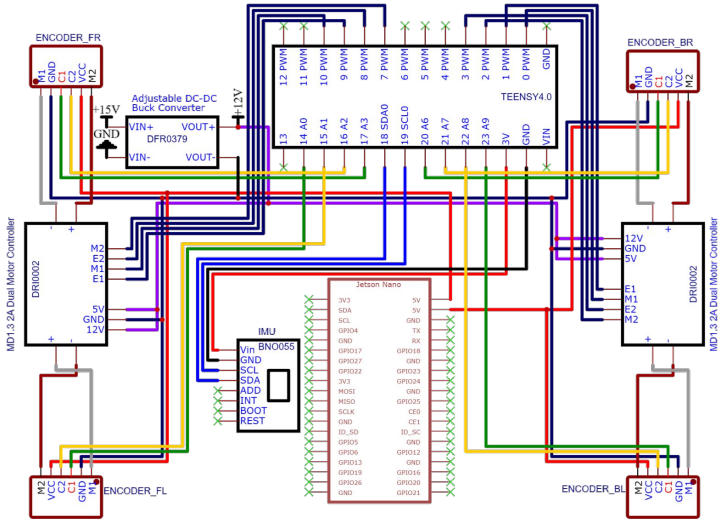


SCAMP
(Self-guided Computer Assisted Mecanum Pathfinder)



- Path following
- Mecanum wheels
- Documentation to replicate





SCAMP: Data Flow

Jetson Nano Computer

world (x, y)

Flask Server

goal (x, y)

Web server

goal (x, y, θ)

Serial

world (x, y)

Motor + Encoder

relative (x, y, θ)

Teensy 4.0 Microcontroller

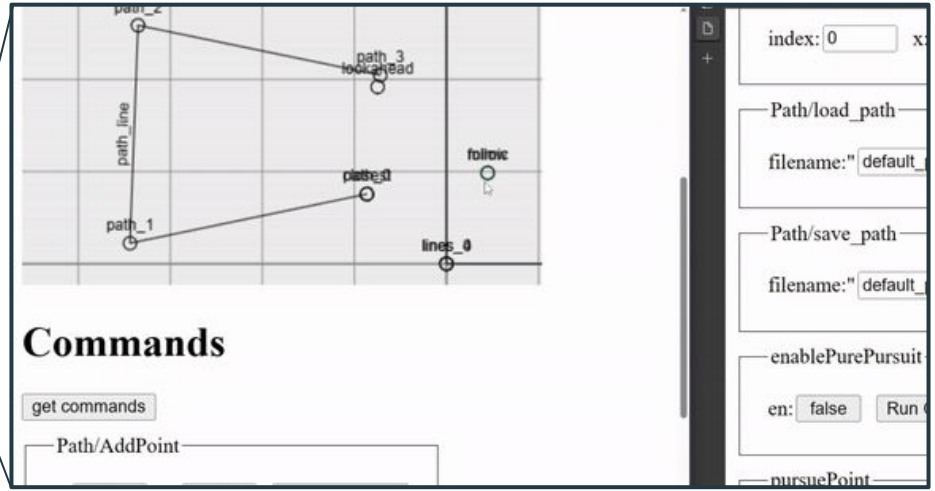
θ

Inertial Mass Unit (IMU)

Raw motor signal

Motor Driver

Motor power + motor direction



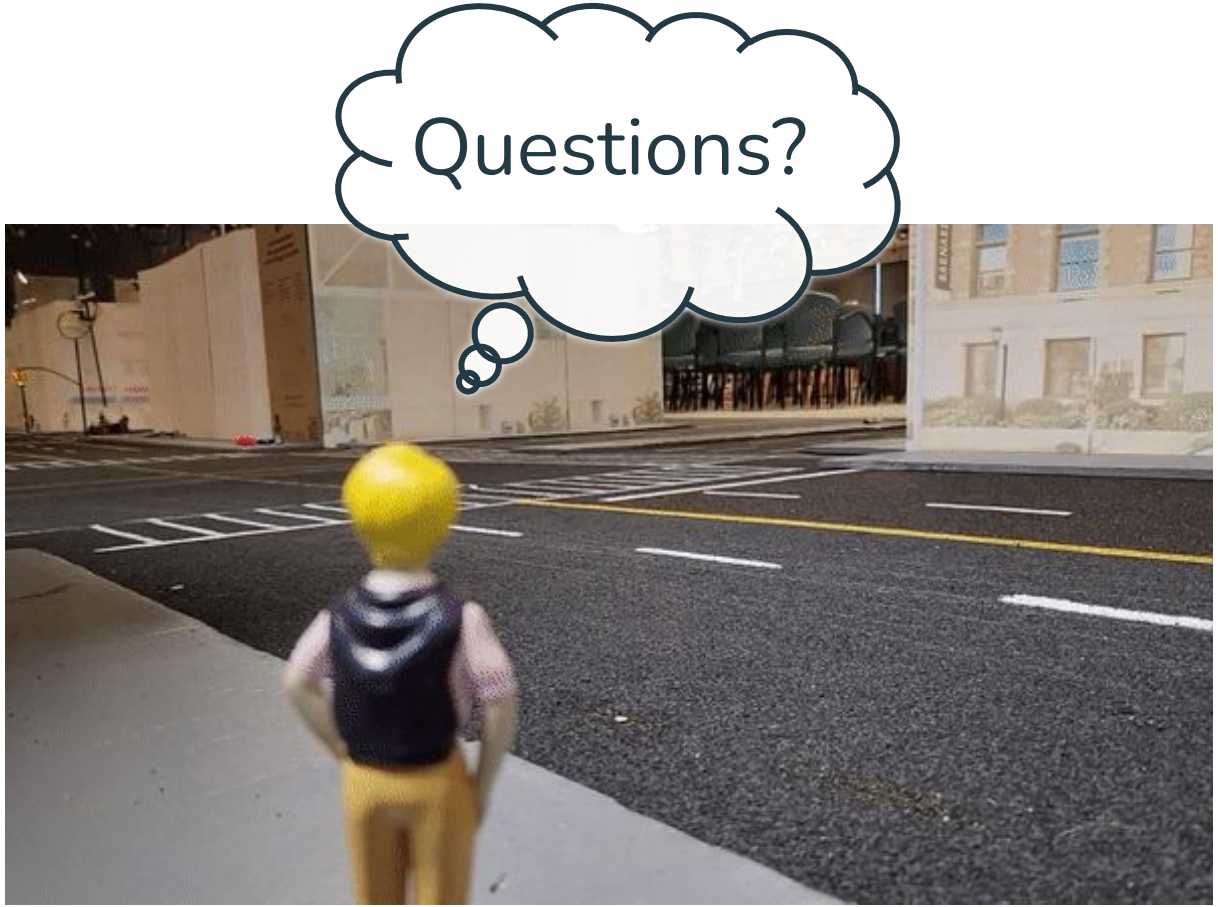
Future

RASCAL

- Simultaneous localization and mapping (SLAM)
- Integration with intersection cameras or larger field of view

SCAMP

- Self orientation (SLAM/Intersection Cameras)
- Path extraction from video
- Instantaneous speed


Questions?