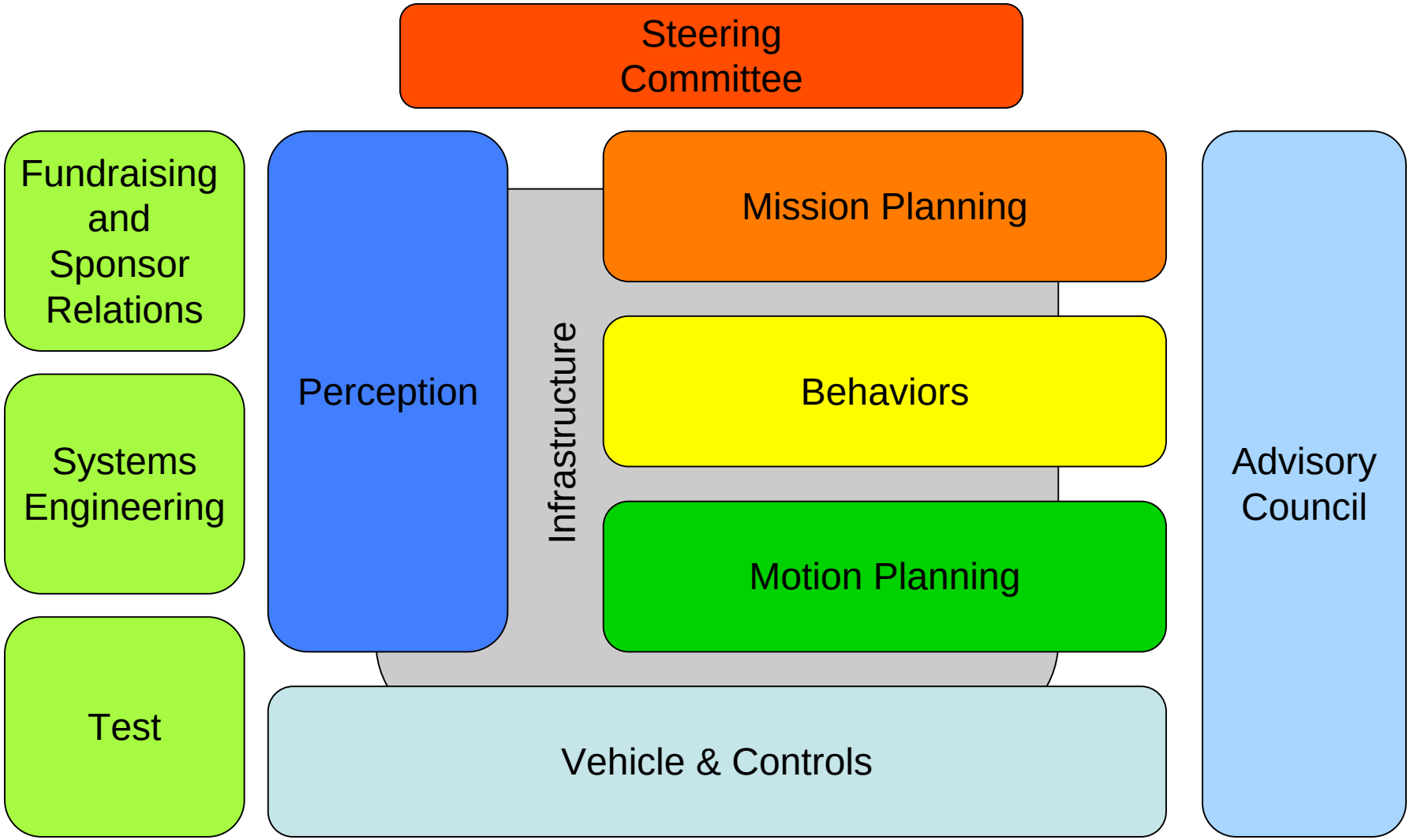
Tartan Racing and the DARPA Urban Challenge

TARTANRACING @ Carnegie Mellon

http://www.tartanracing.org

Joshua Anhalt, Hong Bae, Drew Bagnell, Christopher Baker, Robert Bittner, Thomas Brown, M. Clark, Michael Darms, Daniel Demitrish, John Dolan, Dave Duggins, Dave Ferguson, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Alonzo Kelly, Sascha Kolski, Bakhtiar Litkouhi, Nick Miller, Matt McNaughton, Jim Nickolaou, Kevin Peterson, Maxim Likhachev, Raj Rajkumar, Paul E. Rybski, Varsha Sadekar, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Joshua Struble, Jarrod Snider, Anthony Stentz, Michael Taylor, Chris Urmson, Red Whittaker, Ziv Wolkowicki, Wende Zhang, Jason Ziglar

Tartan Racing Organization



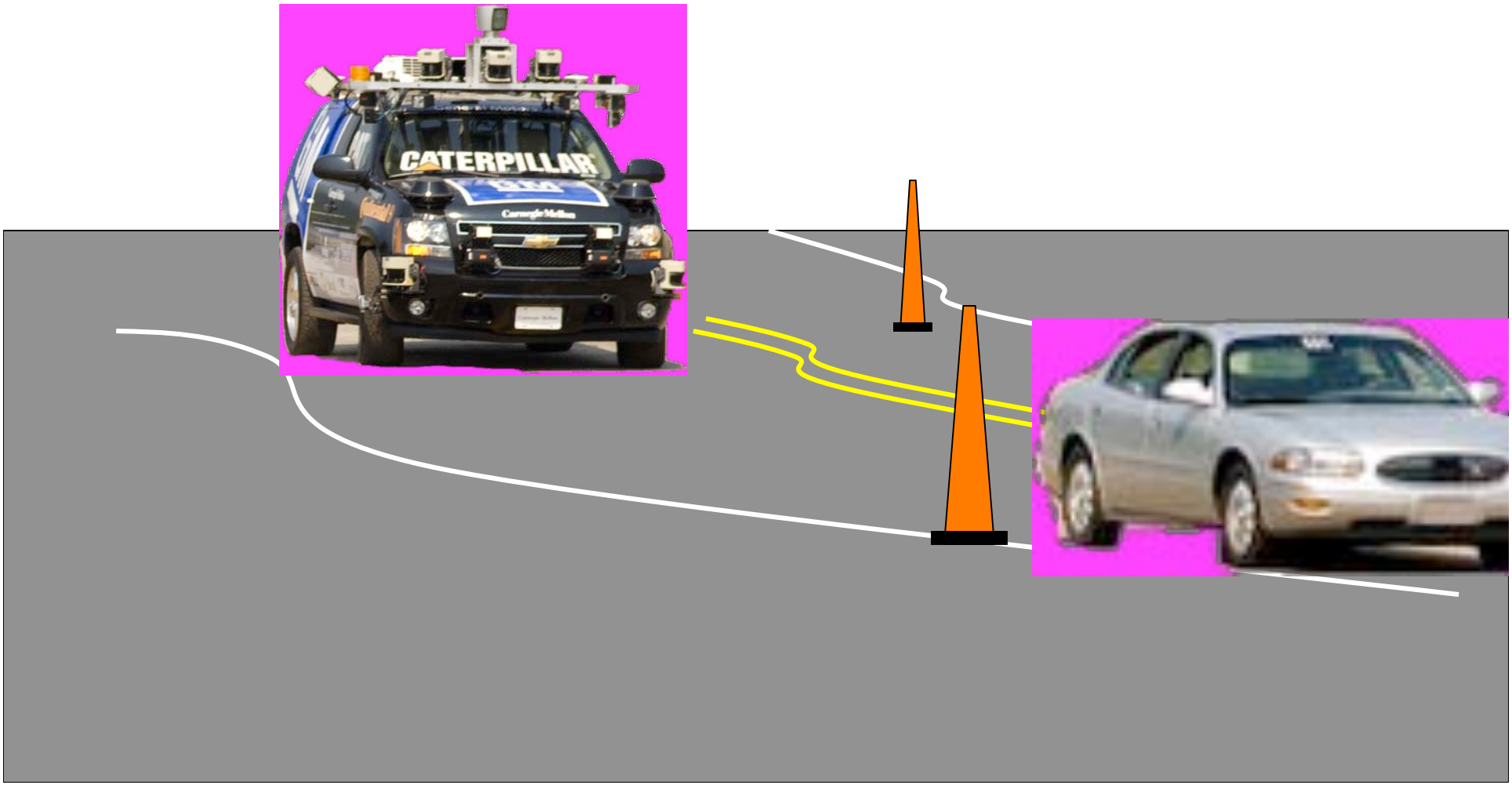
Perception for The Urban Challenge

Paul E. Rybski

The Robotics Institute

Carnegie Mellon University

Perception



Perception on Boss

Velodyne
multi-plane lidar
360°x26° FOV, 60m

Applanix
GPS/INS

Continental
ISF 172 lidar
14°, 150m

SICK Scanning Lidar
90/180° FOV, 40m

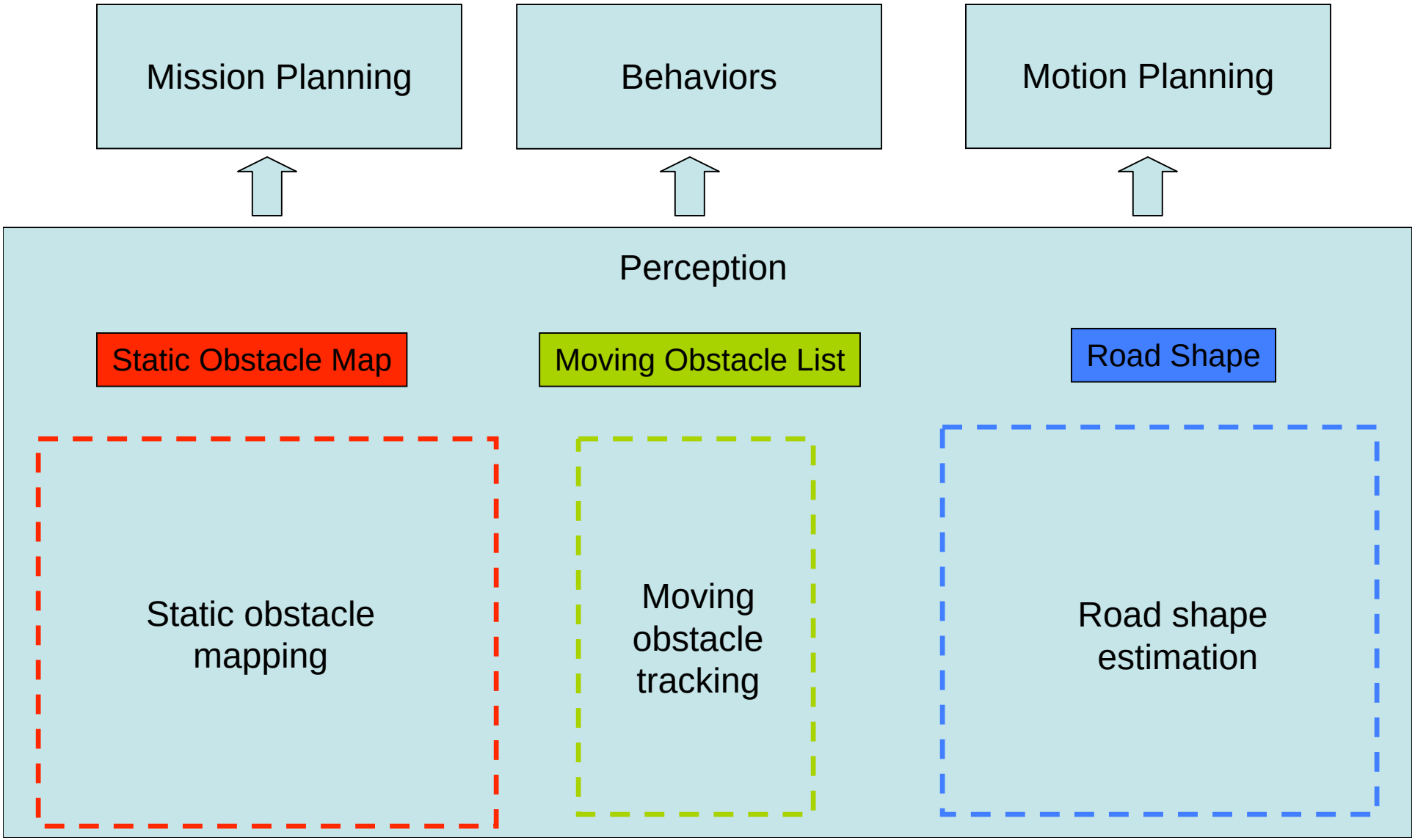
Object Tracking

IBEO
180° FOV,
multi-plane, multi-echo

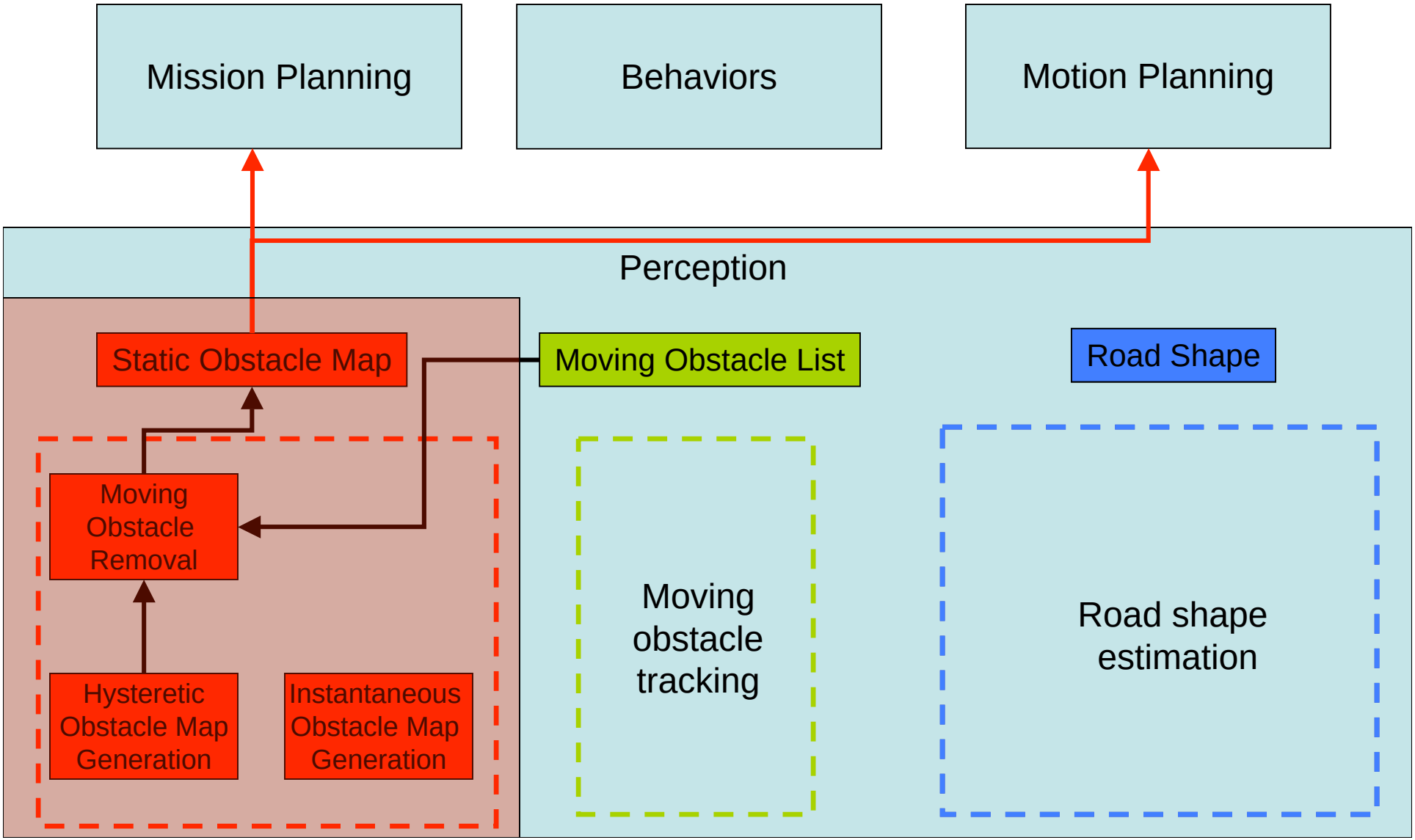
Continental
ARS 300 radar
60/17°, 60/200m

~16 Sensors total

Perception Architecture

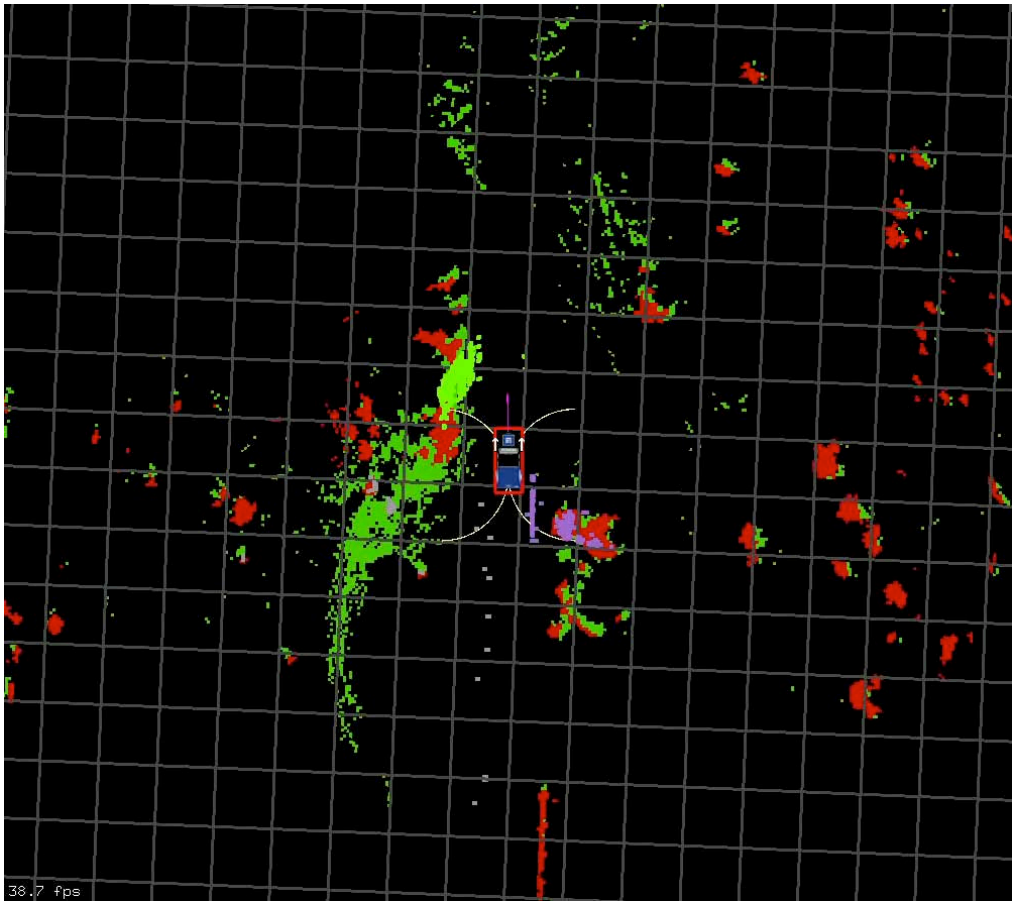


Perception Architecture



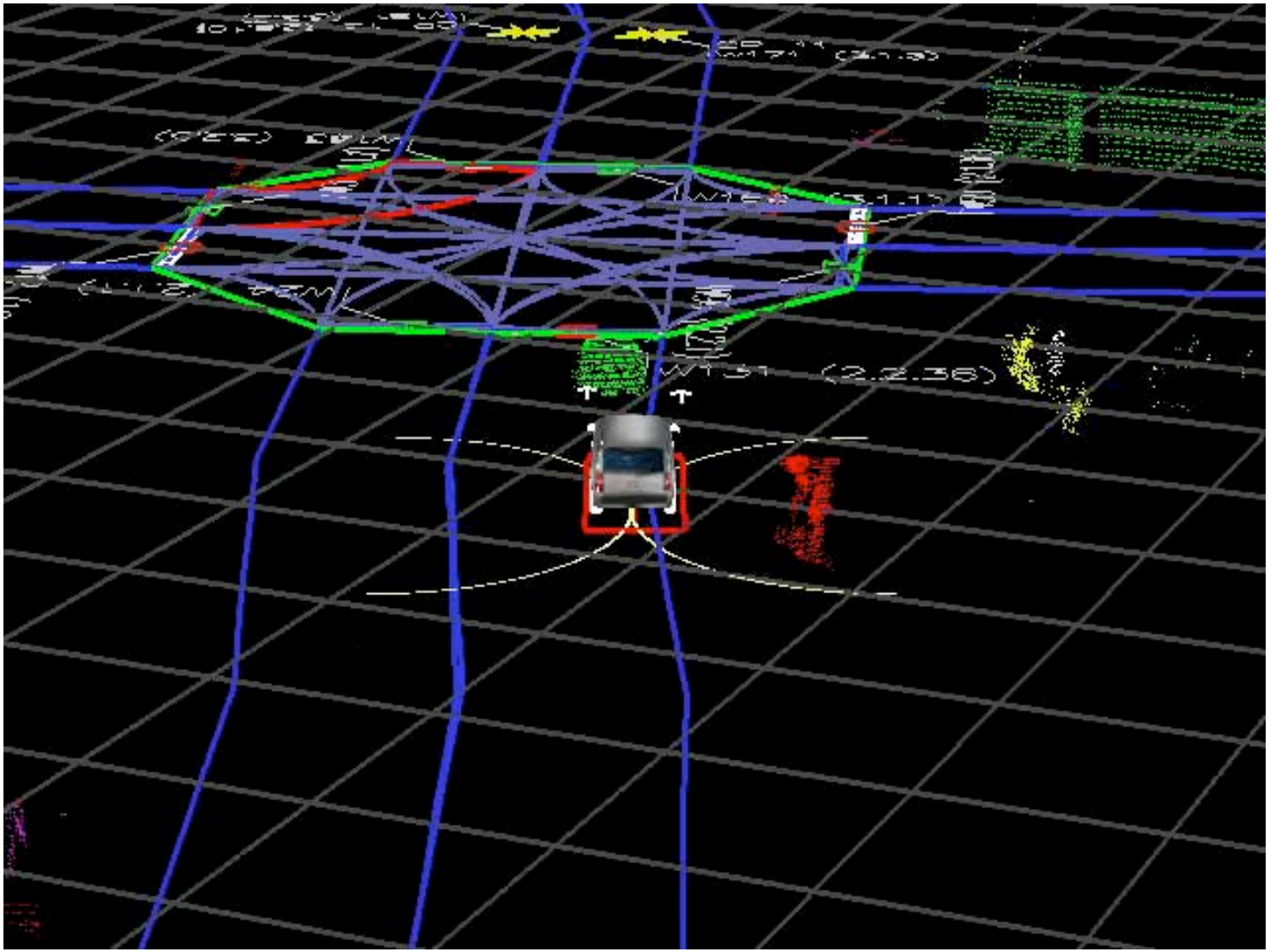
Static Obstacle Mapping

- 2D grid-based obstacle representation
  - Grid scrolled as vehicle moves
- Sensors generating static maps
  - Velodyne segmenter
  - IBEO mapper
  - SICK obstacle map
  - Curb point mapper
- Multiple grids fused into a single map
  - Different update rates



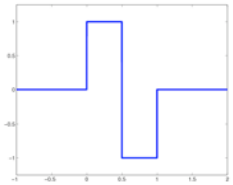
TartanRacing @ Carnegie Mellon

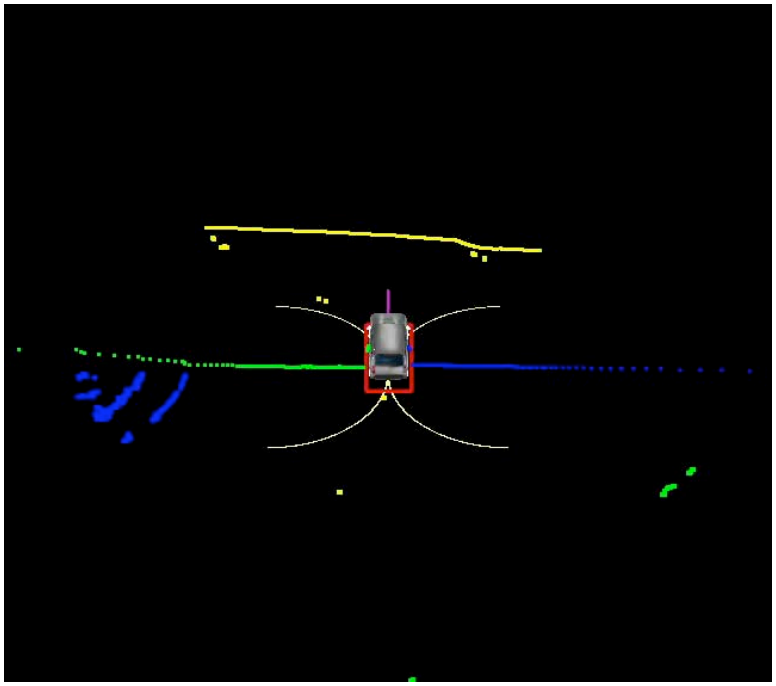
Velodyne Segmenter

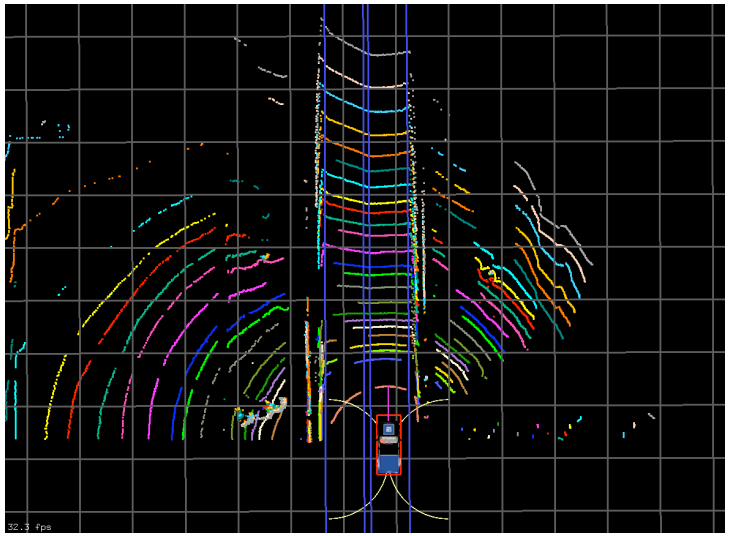


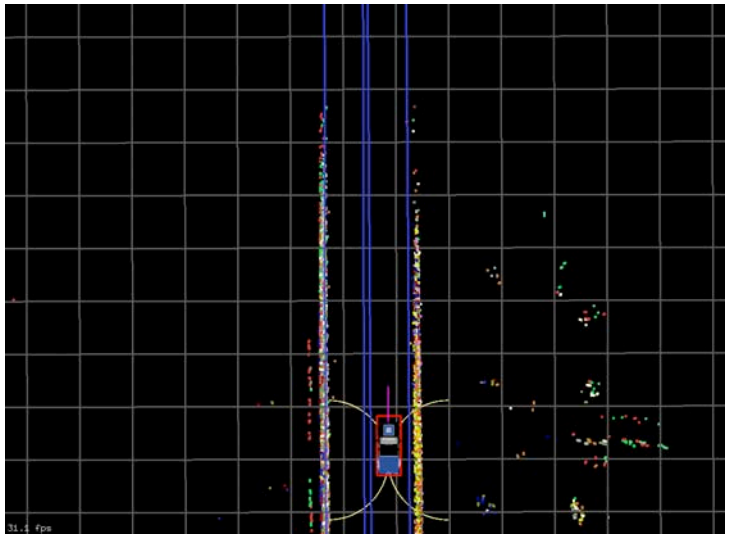
Wavelet Curb Detector

- Looking for geometric features on the side of the road
- Haar wavelet transform used to identify edges along lidar scans at multiple resolutions

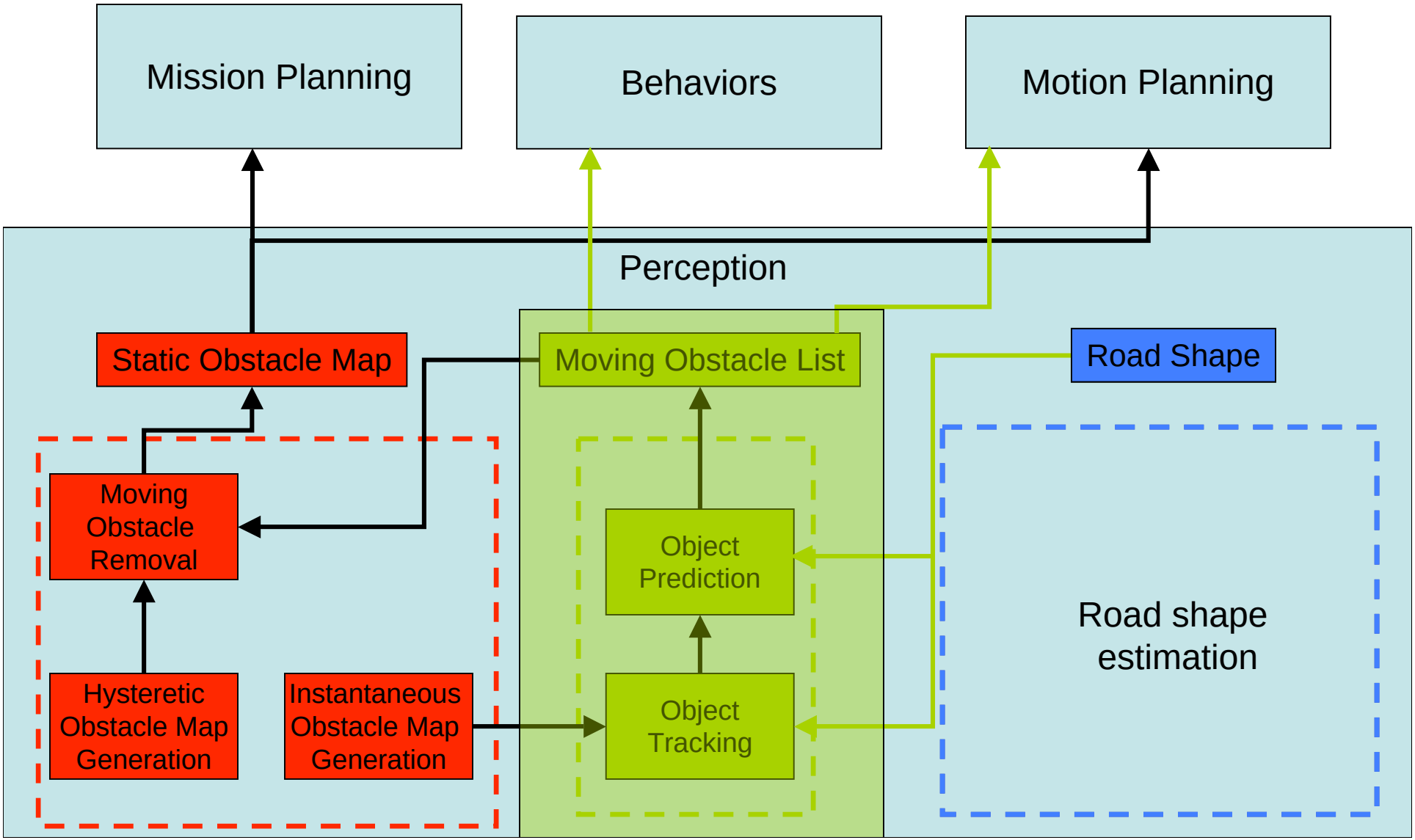








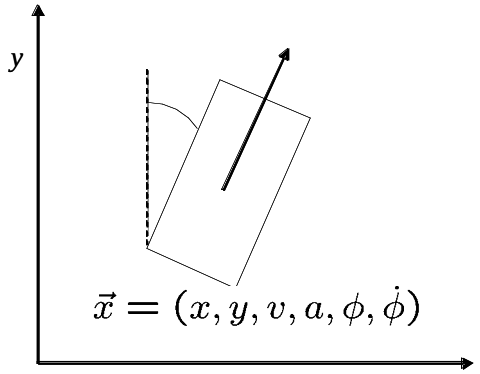
Perception Architecture



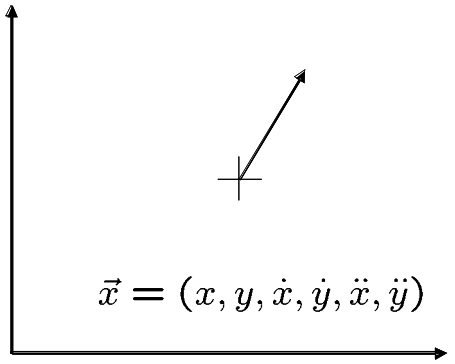
Detecting/Tracking Moving Obstacles

- Multi-sensor, multi-object tracking architecture
- Sensors
  - 3x SICK lidar
  - 2x IBEO lidar
  - 2x ISF lidar
  - 5x ARS 300 radar
  - Velodyne

- Extended Kalman Filter
  - Simple bicycle model
    - Used when shape from lidar is available
  - Point model
    - Used for radar and long-range lidar returns
- Validate targets using Velodyne segmenter



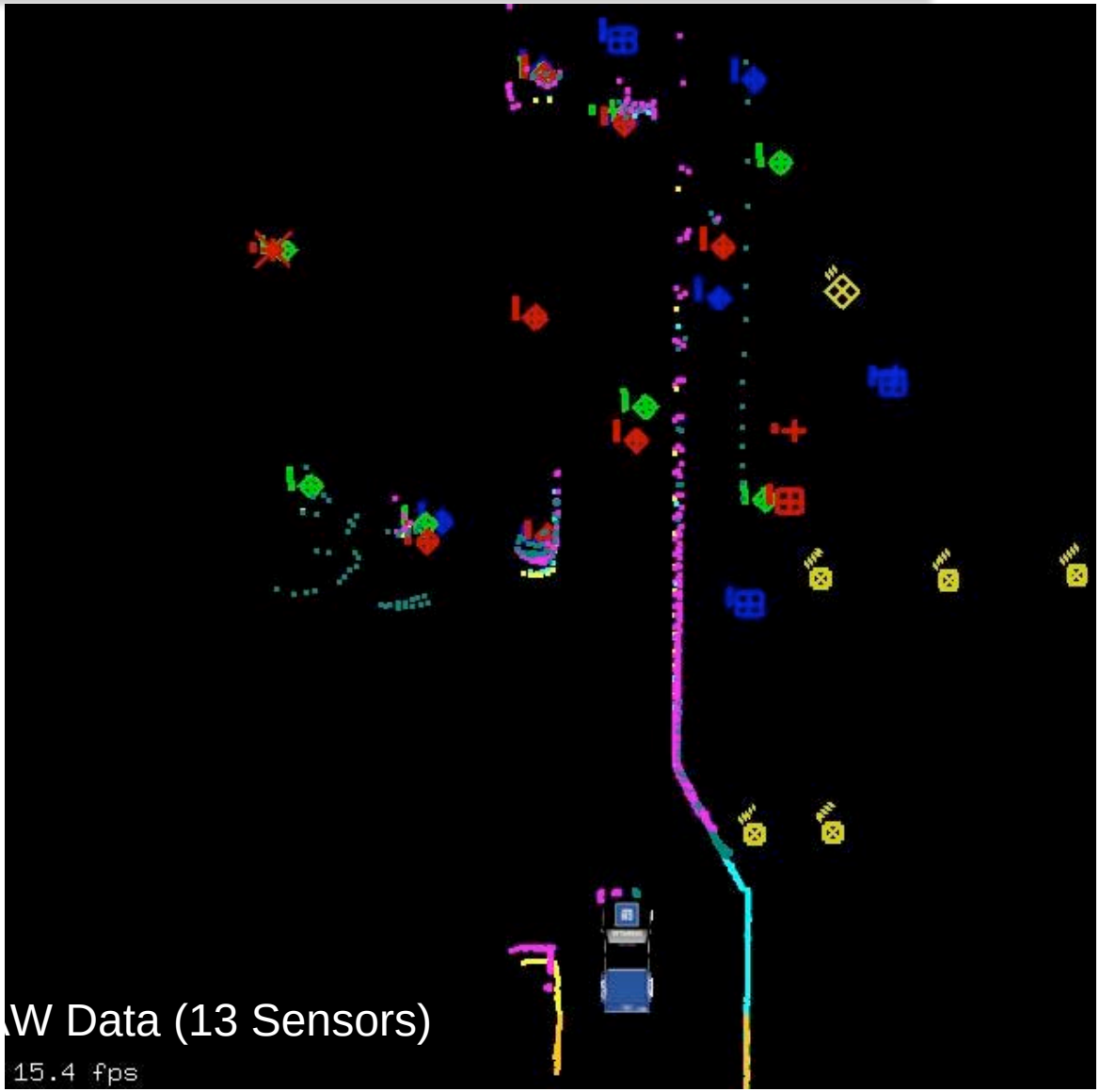
Simple bicycle model



Point model

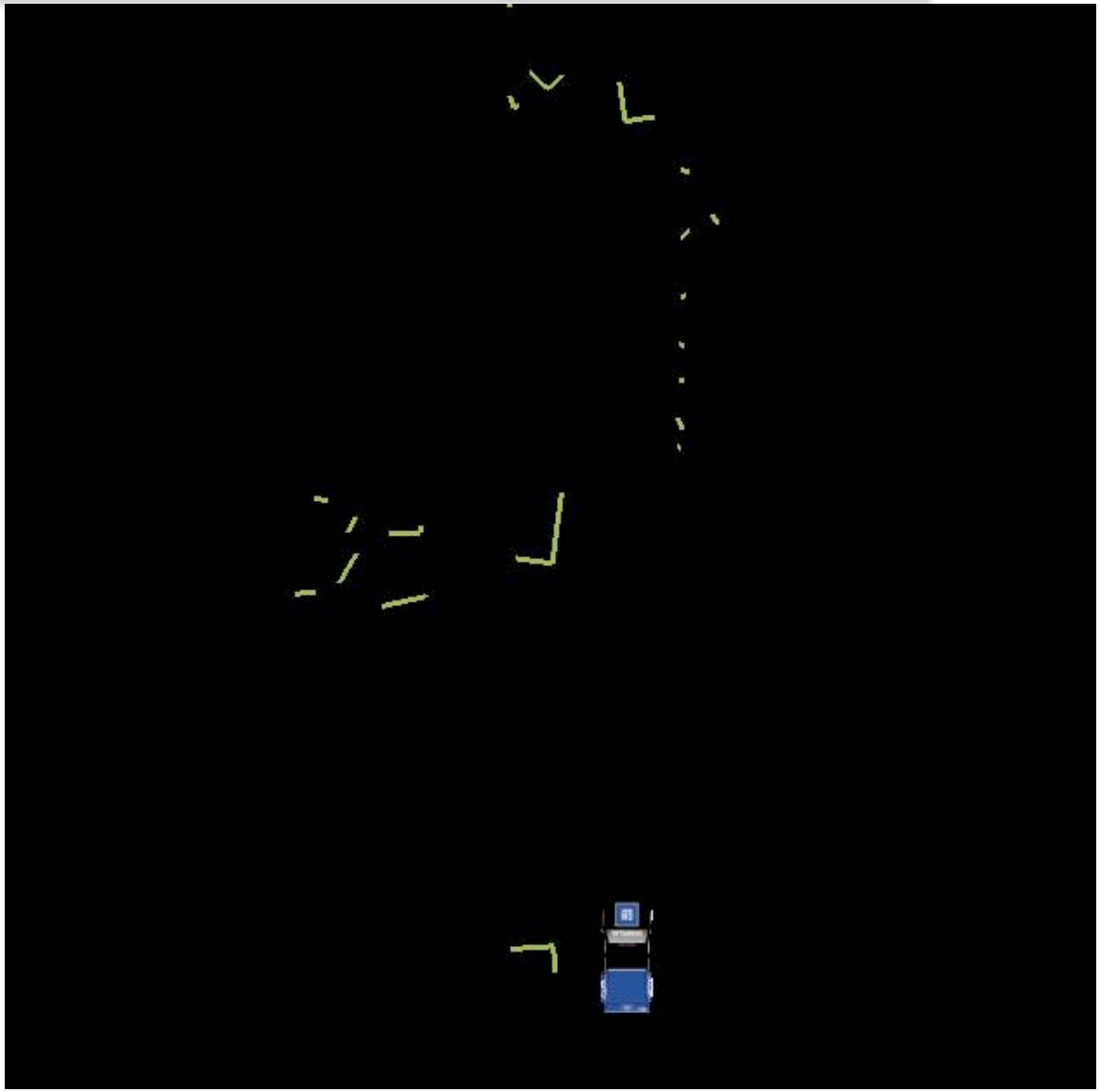
Vehicle Tracking

- Raw Data



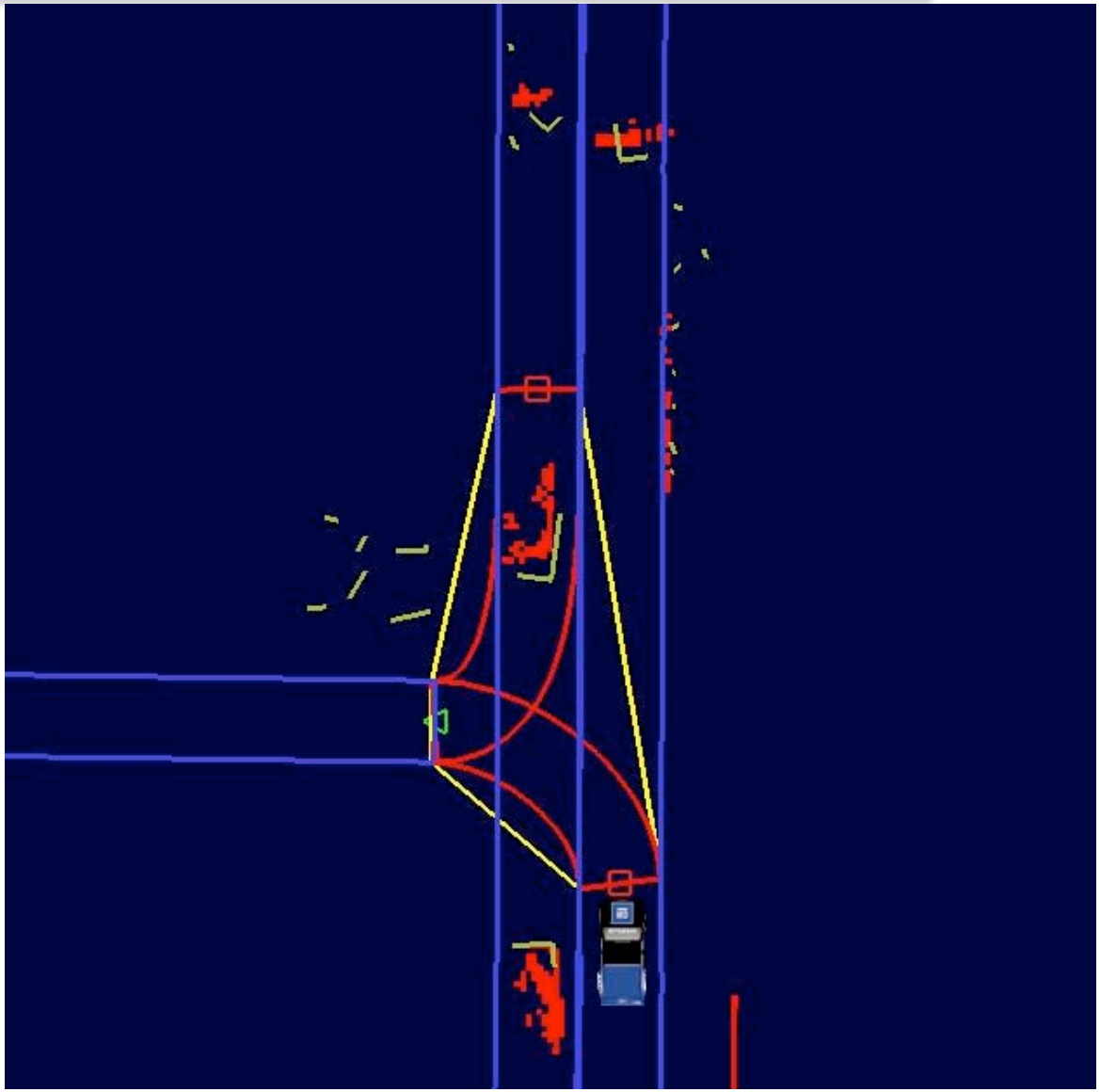
Vehicle Tracking

- Raw Data
- Feature Extraction



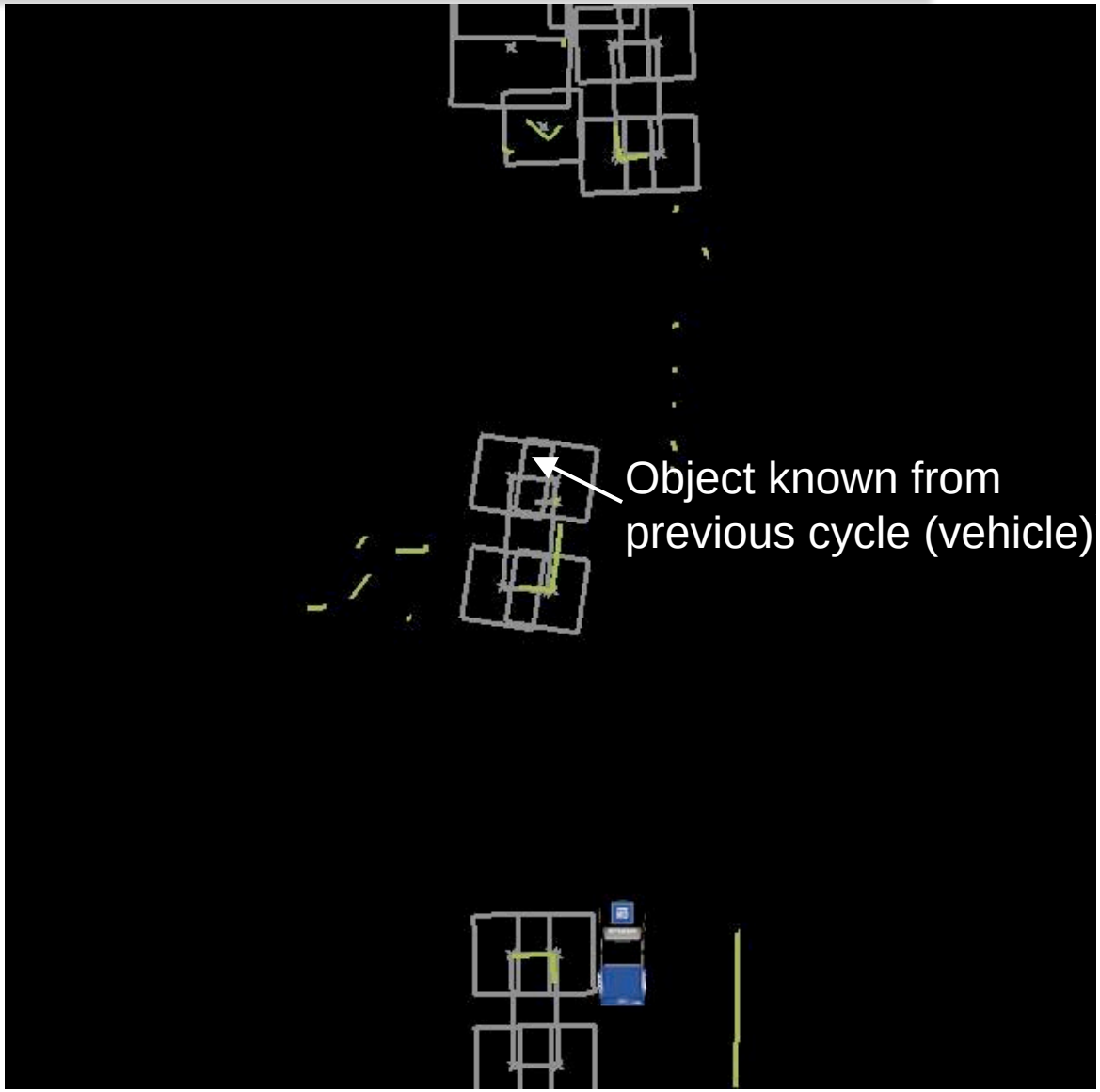
Vehicle Tracking

- Raw Data
- Feature Extraction
- Measurement Validation



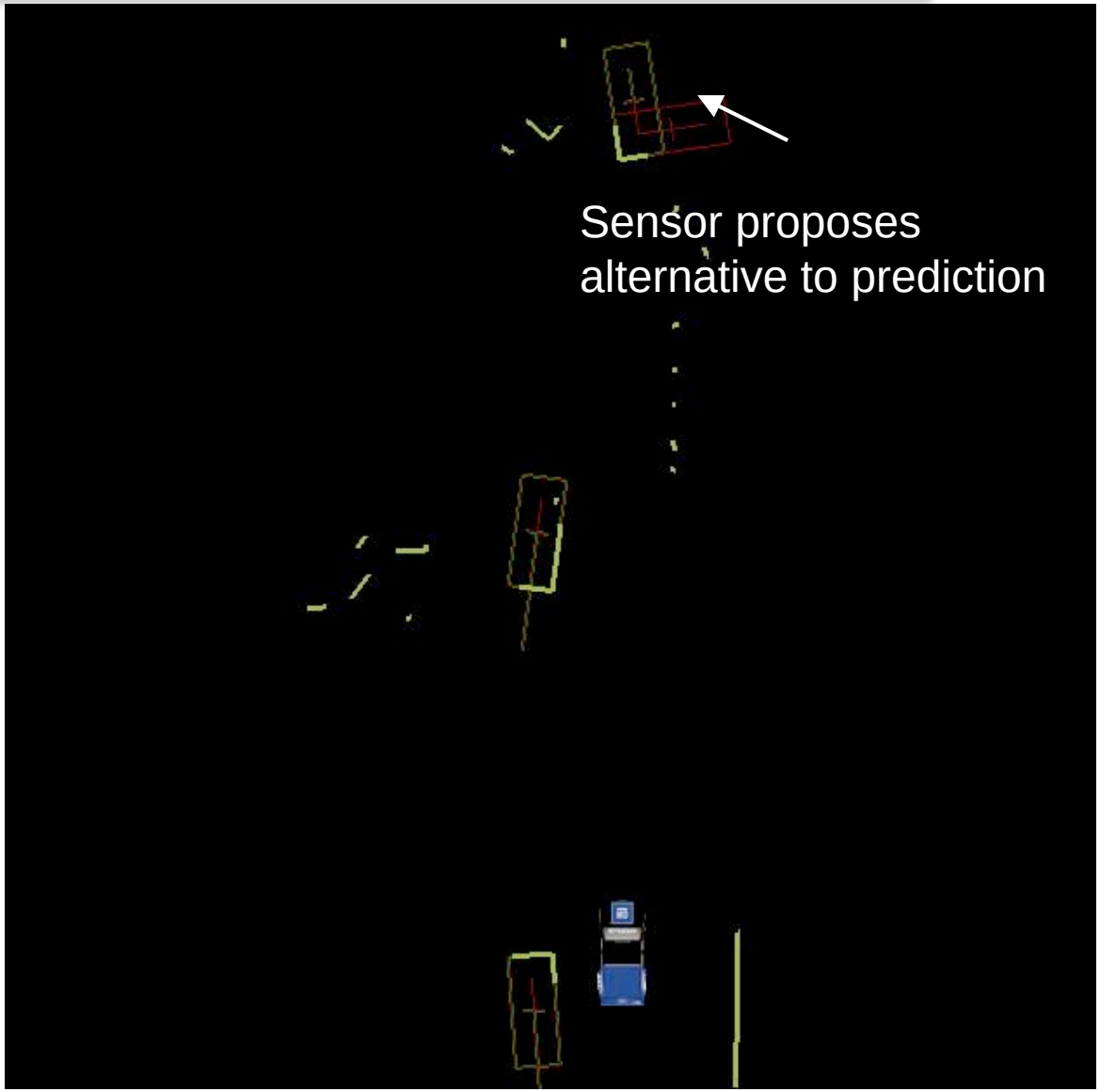
Vehicle Tracking

- Raw Data
- Feature Extraction
- Measurement Validation
- Data Association



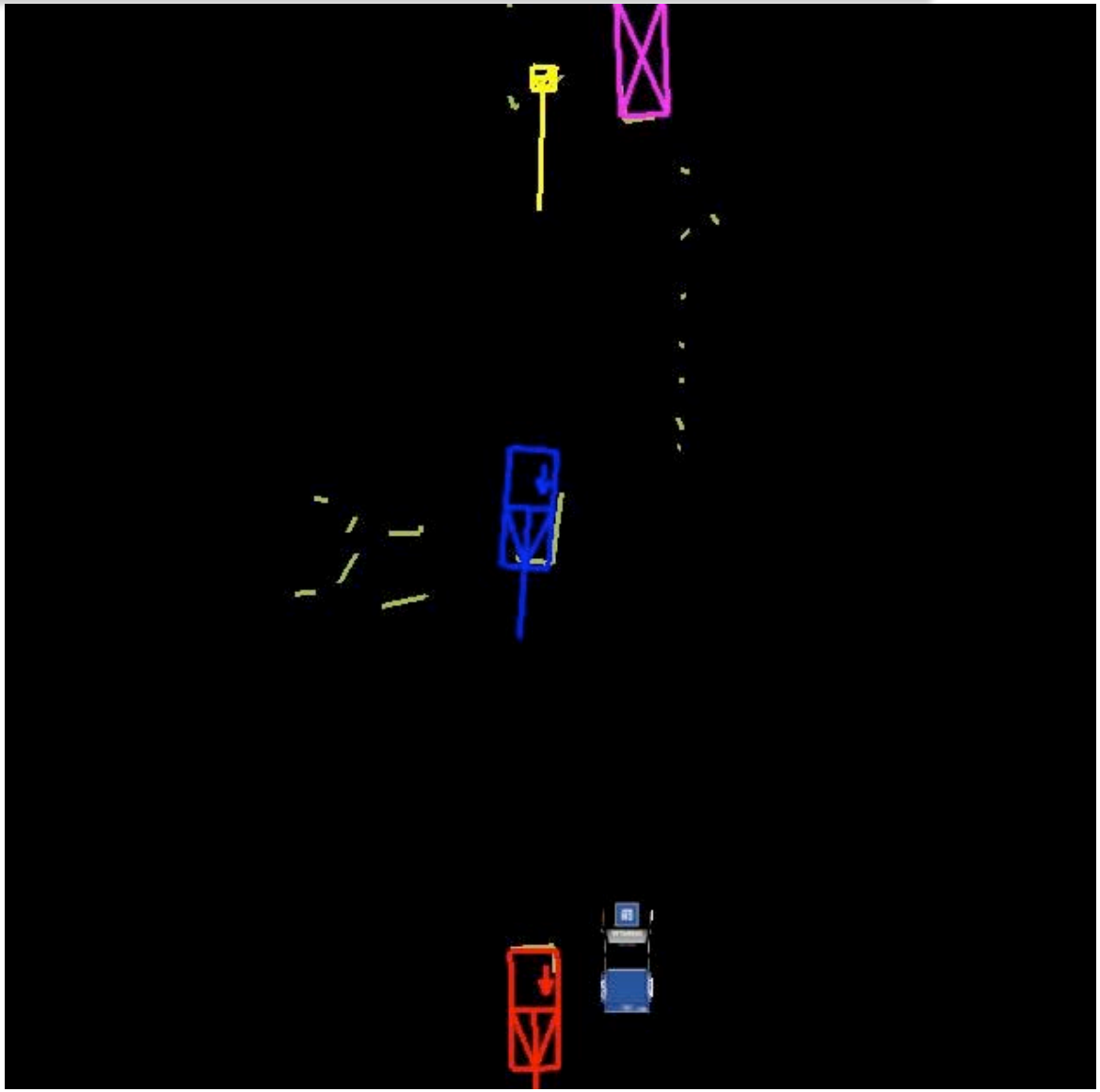
Vehicle Tracking

- Raw Data
- Feature Extraction
- Measurement Validation
- Data Association
- Proposals & Observation



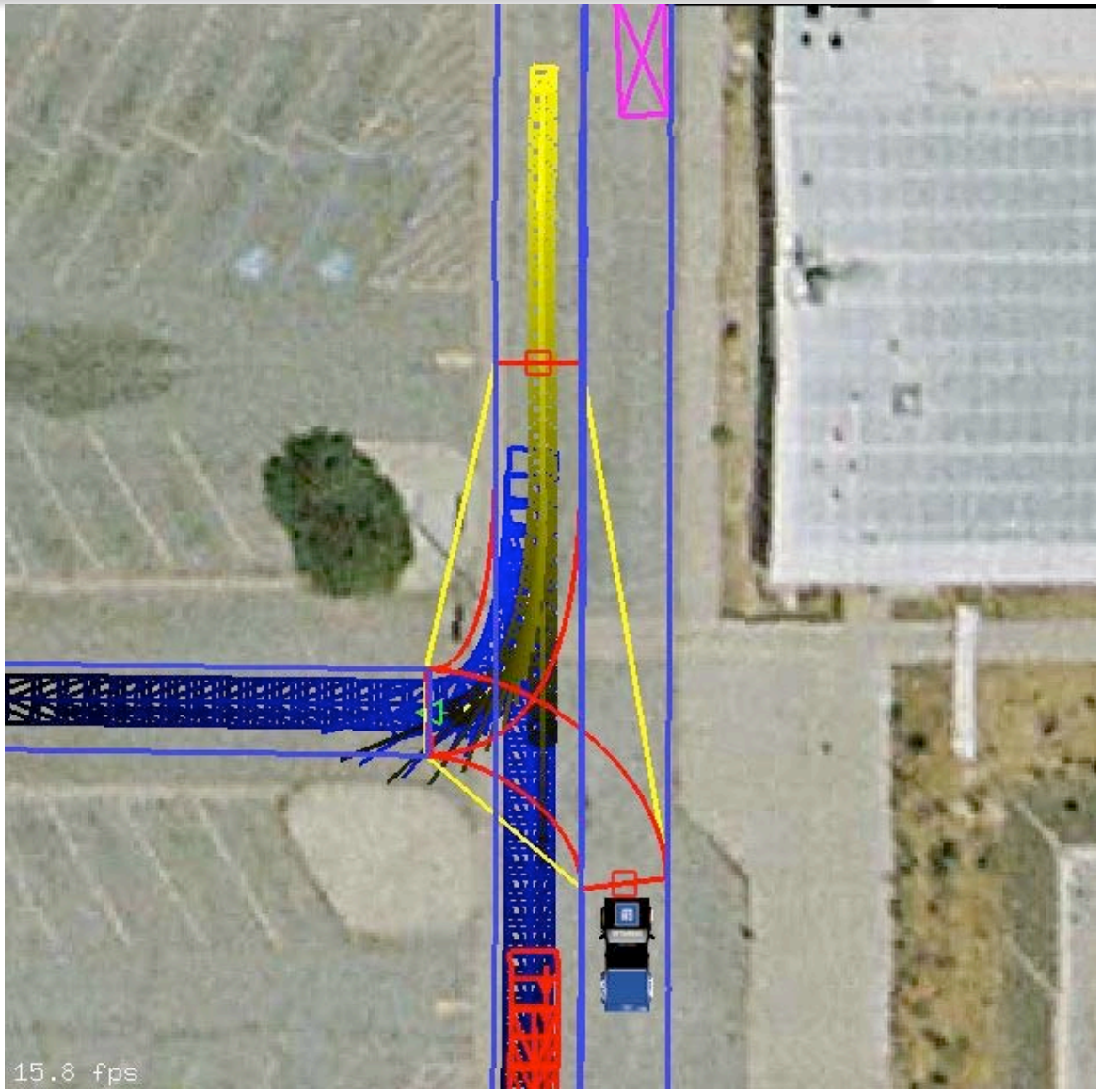
Vehicle Tracking

- Raw Data
- Feature Extraction
- Measurement Validation
- Data Association
- Proposals & Observation
- Model Voting
- Estimation
- Statistics

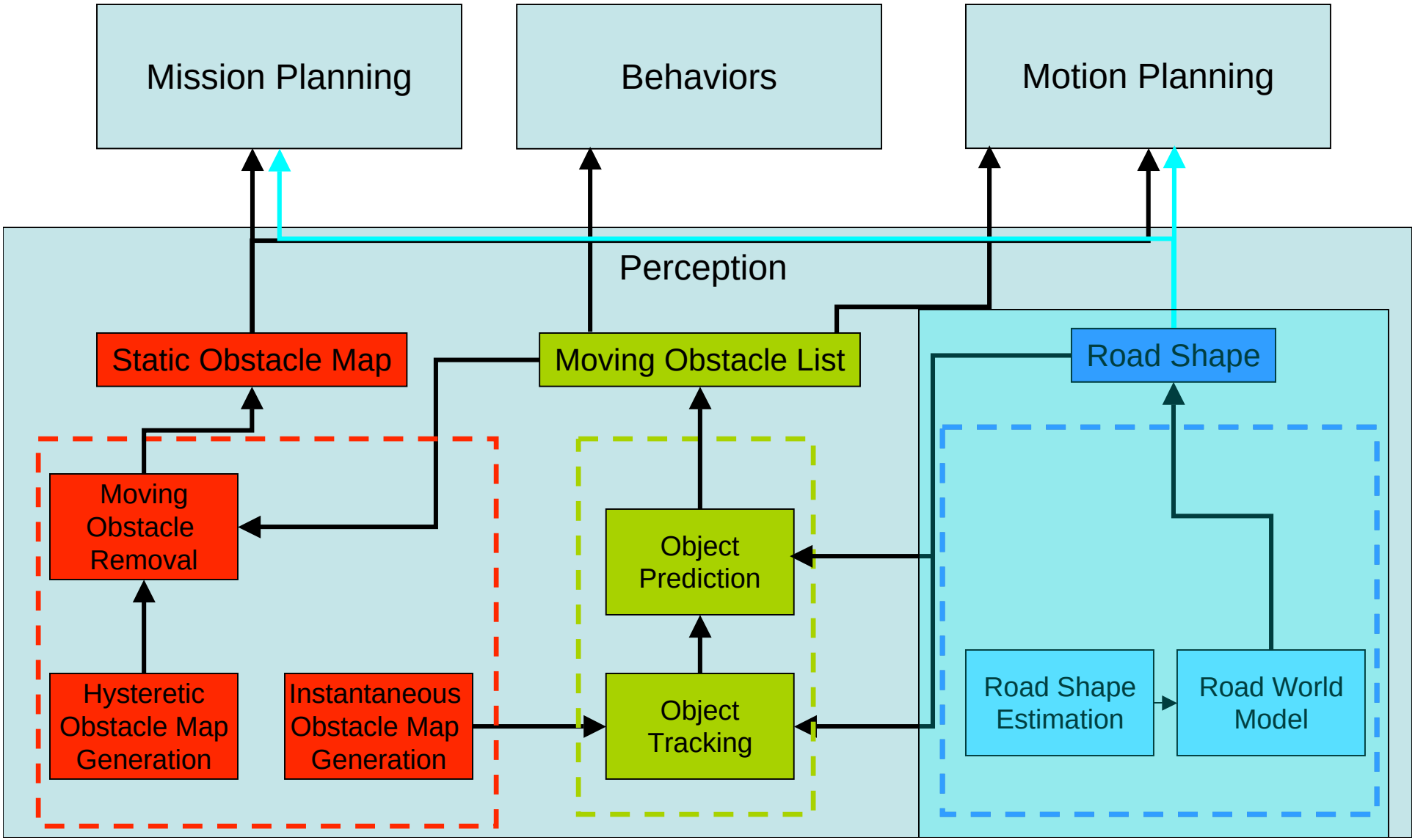


Vehicle Tracking

- Raw Data
- Feature Extraction
- Measurement Validation
- Data Association
- Proposals & Observation
- Model Voting
- Estimation
- Statistics
- Prediction

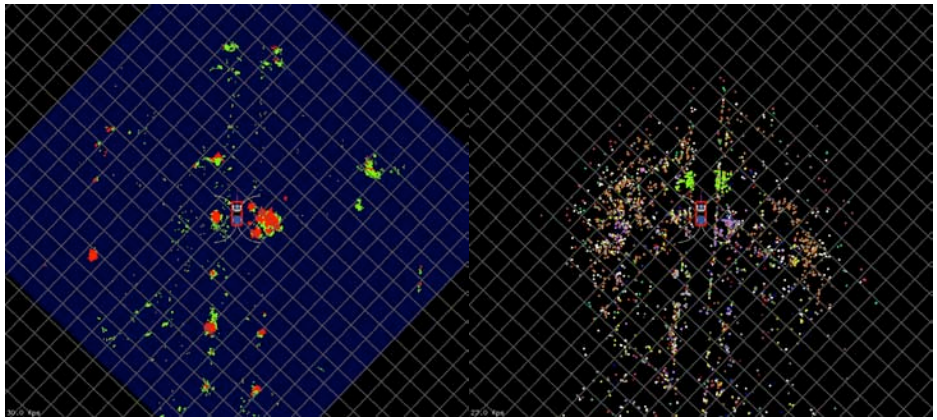


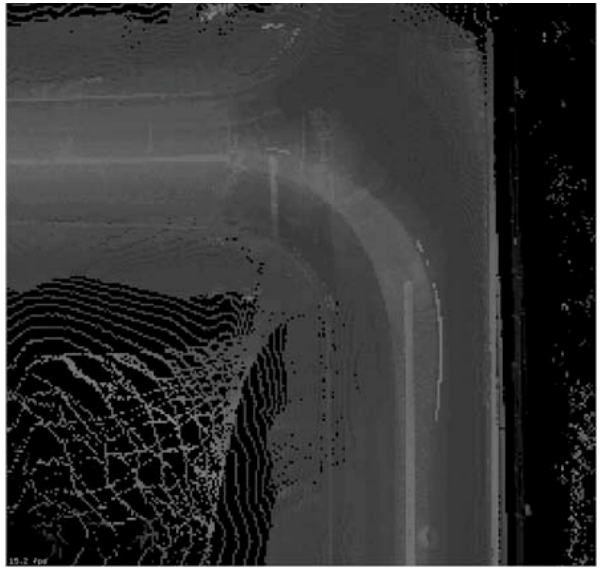
Perception Architecture



Road Shape Estimation

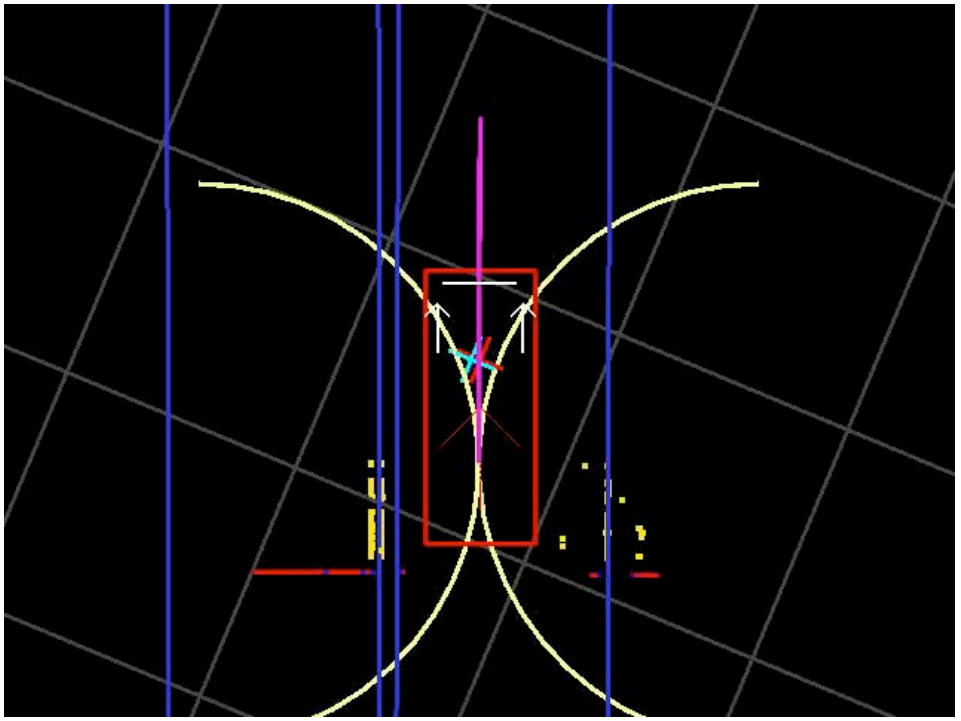
- Obstacles
  - Static obstacle map (Velodyne, SICK, IBEO)
- Curbs
  - Wavelet curb detector from lidar (Velodyne and SICK)
  - Geometry curb detector from lidar (SICK)
- Road lines
  - Lidar intensity (SICK)



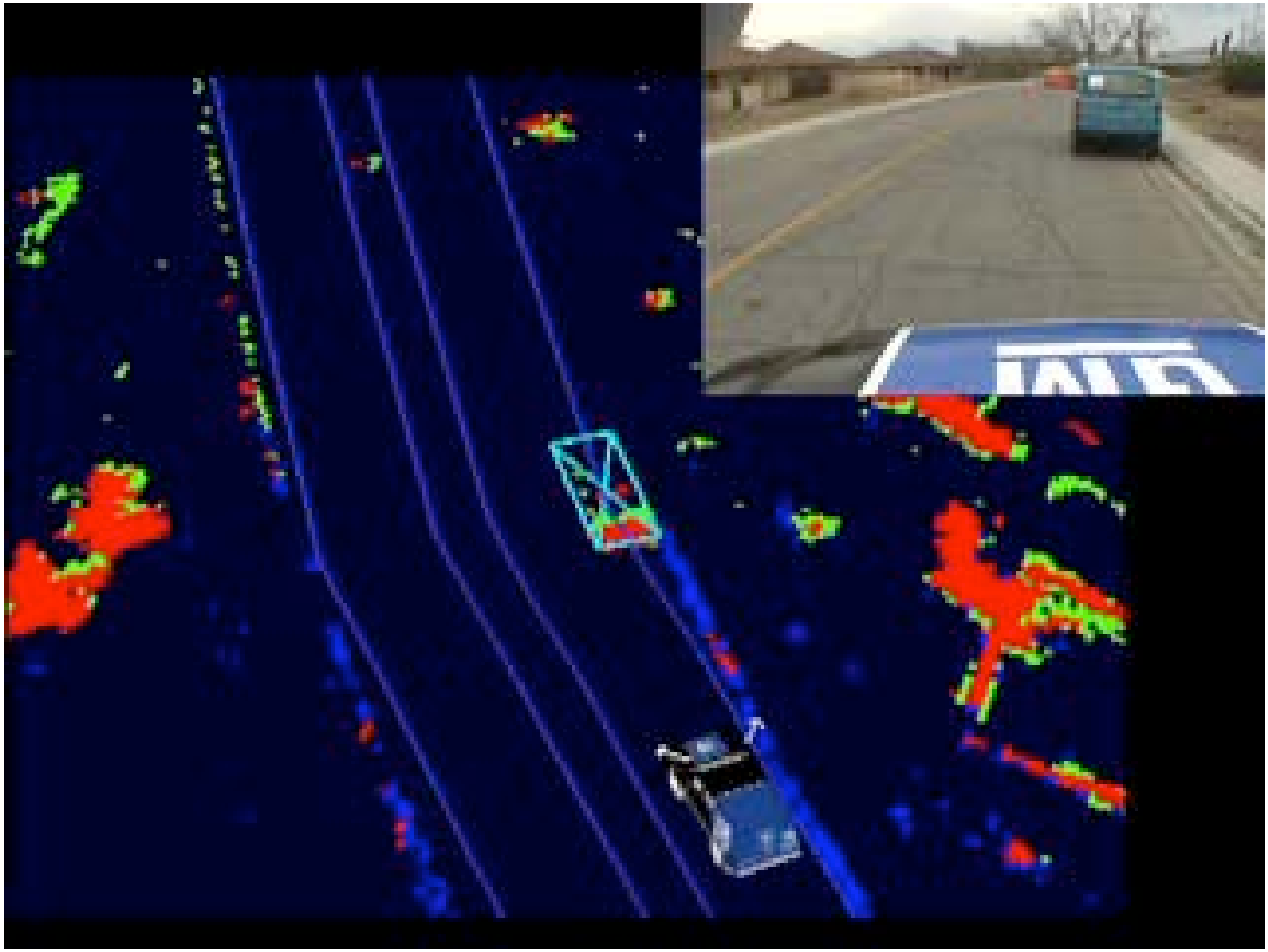


GPS Error Estimation

- Use intensity data from AFT mounted SICKs
- Localize robot with respect to surrounding lanes
- Only localize along one axis at a time



Unified Perception on Boss



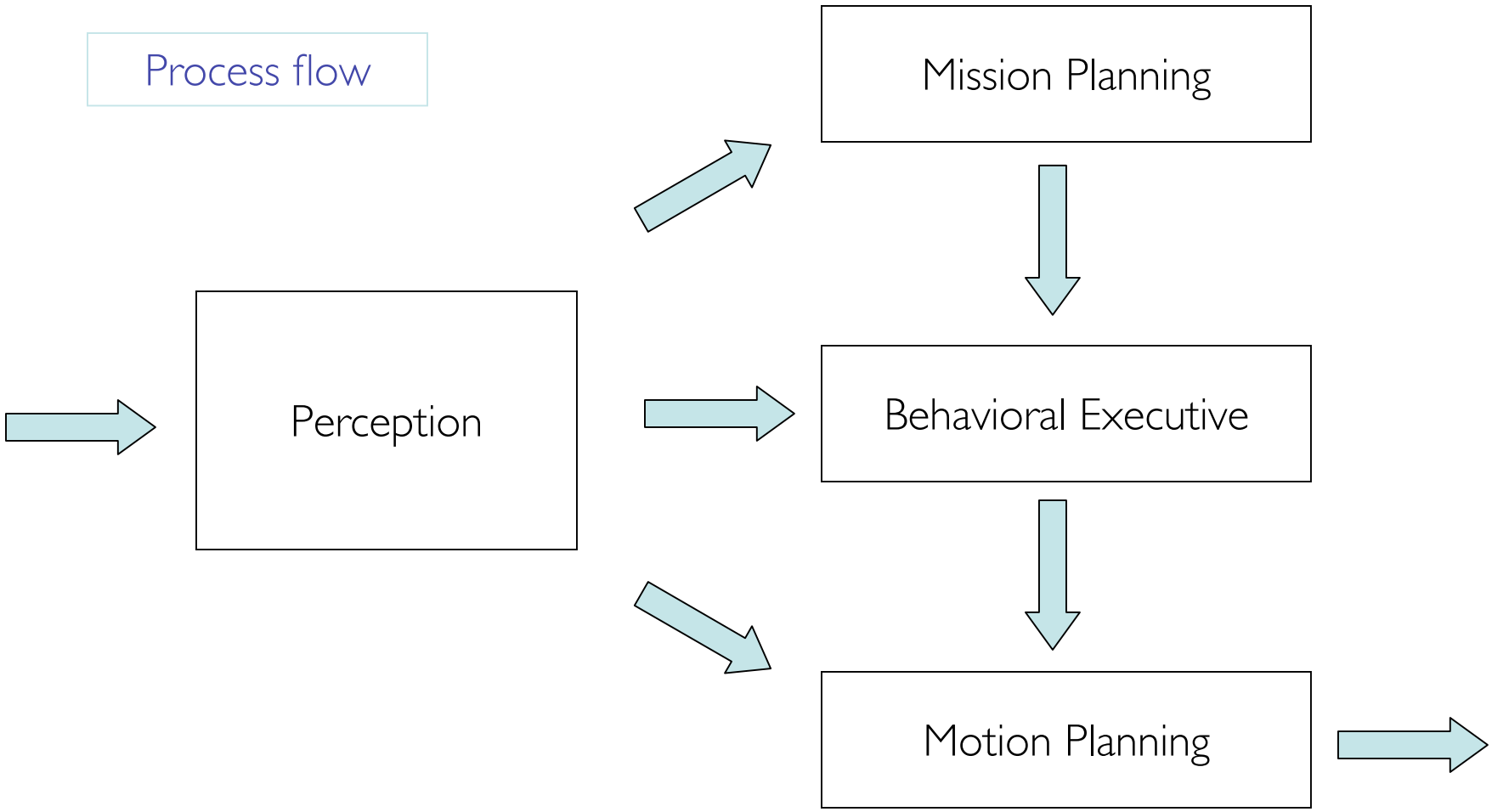
Motion Planning for The Urban Challenge

Dave Ferguson

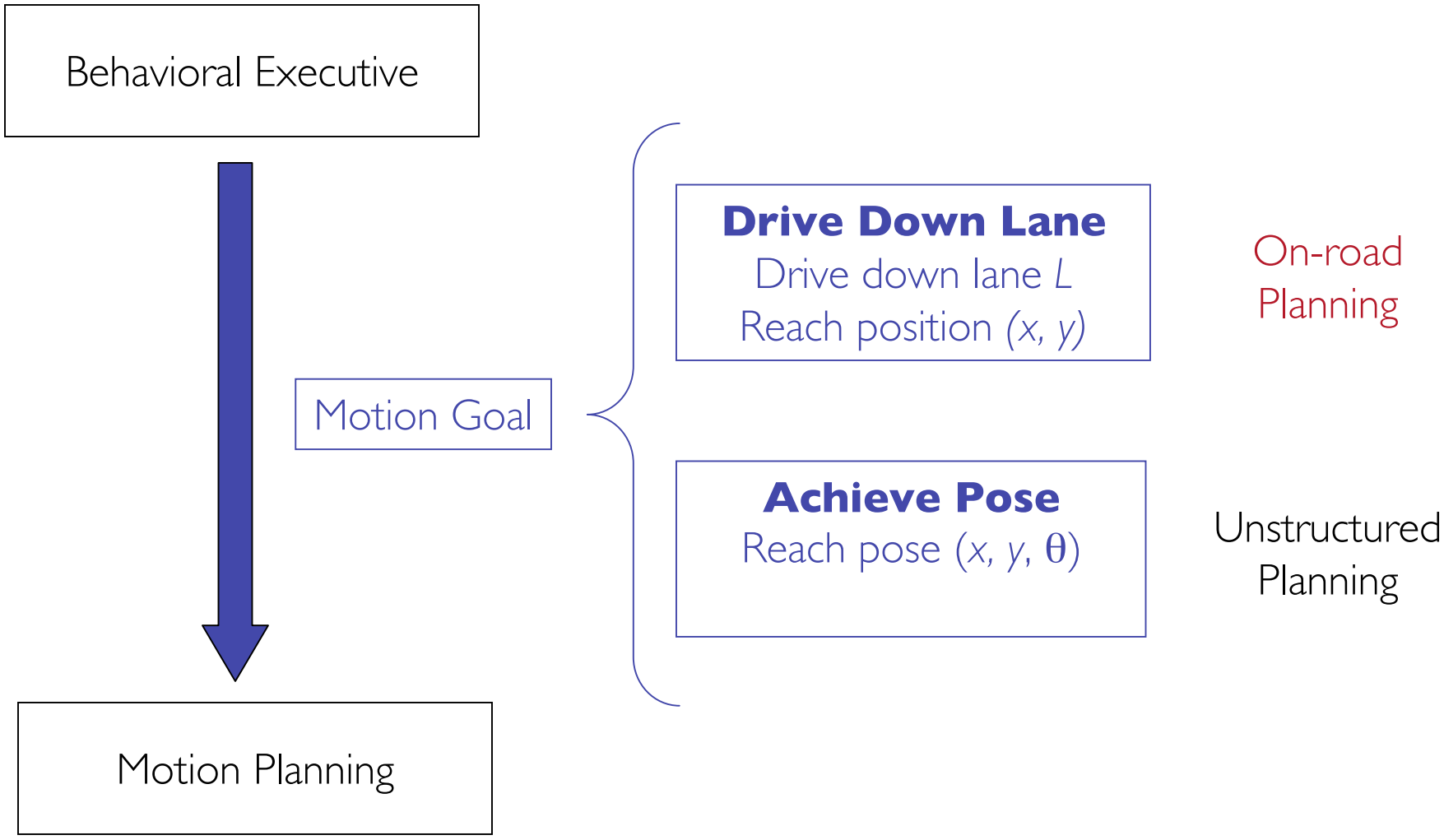
Intel Research Pittsburgh

System Architecture

Process flow



Motion Planning Interface



Motion Planning Overview

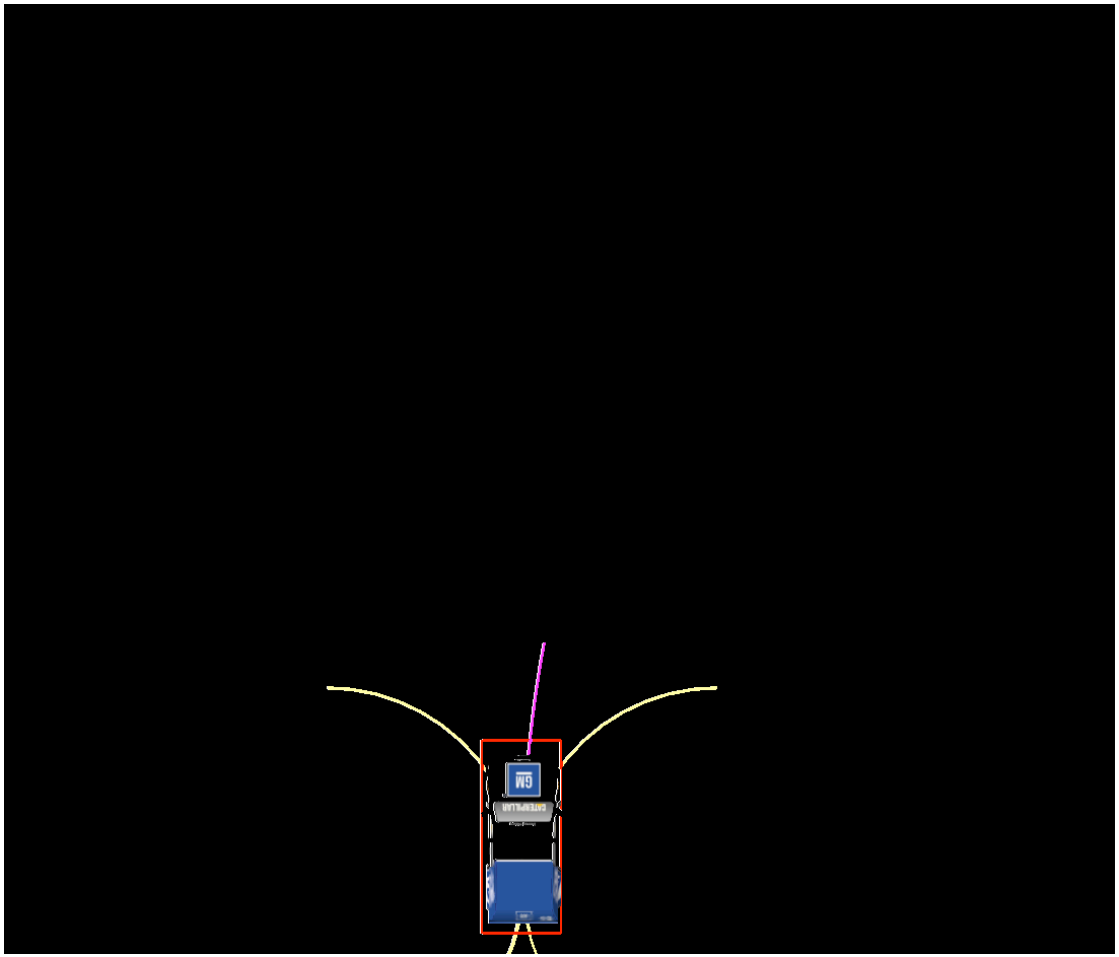
1. Generate path towards goal
2. Generate set of trajectories to track path
3. Evaluate set of trajectories
4. Select and execute best trajectory

Two general contexts

- On-road Planning
- Unstructured Planning

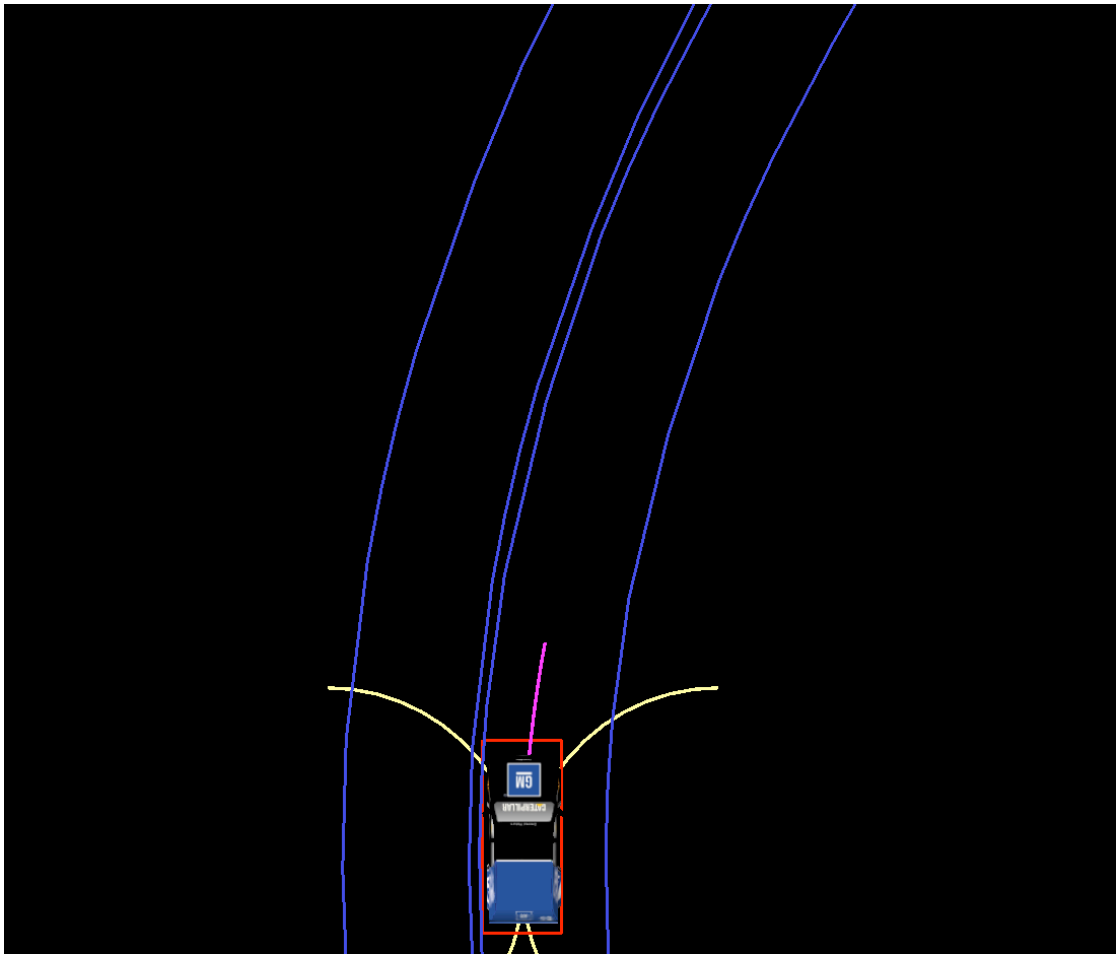
On-road Planning

- Task: Drive vehicle down desired lane



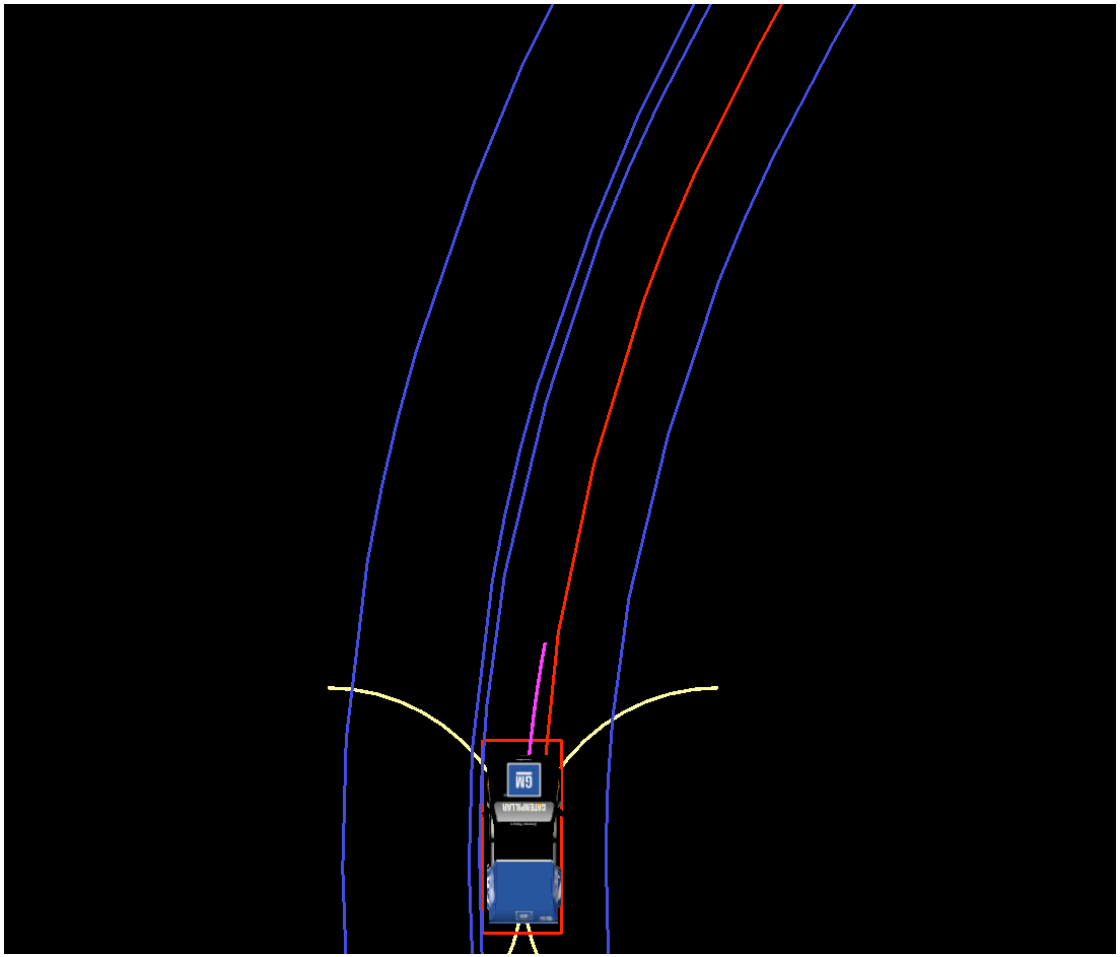
Path Generation

- Use center of lane as desired path



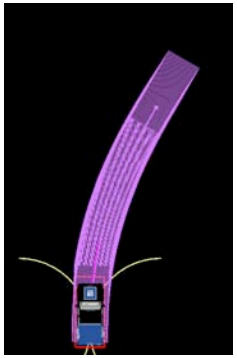
Path Generation

- Use center of lane as desired path

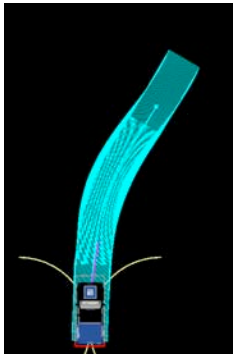


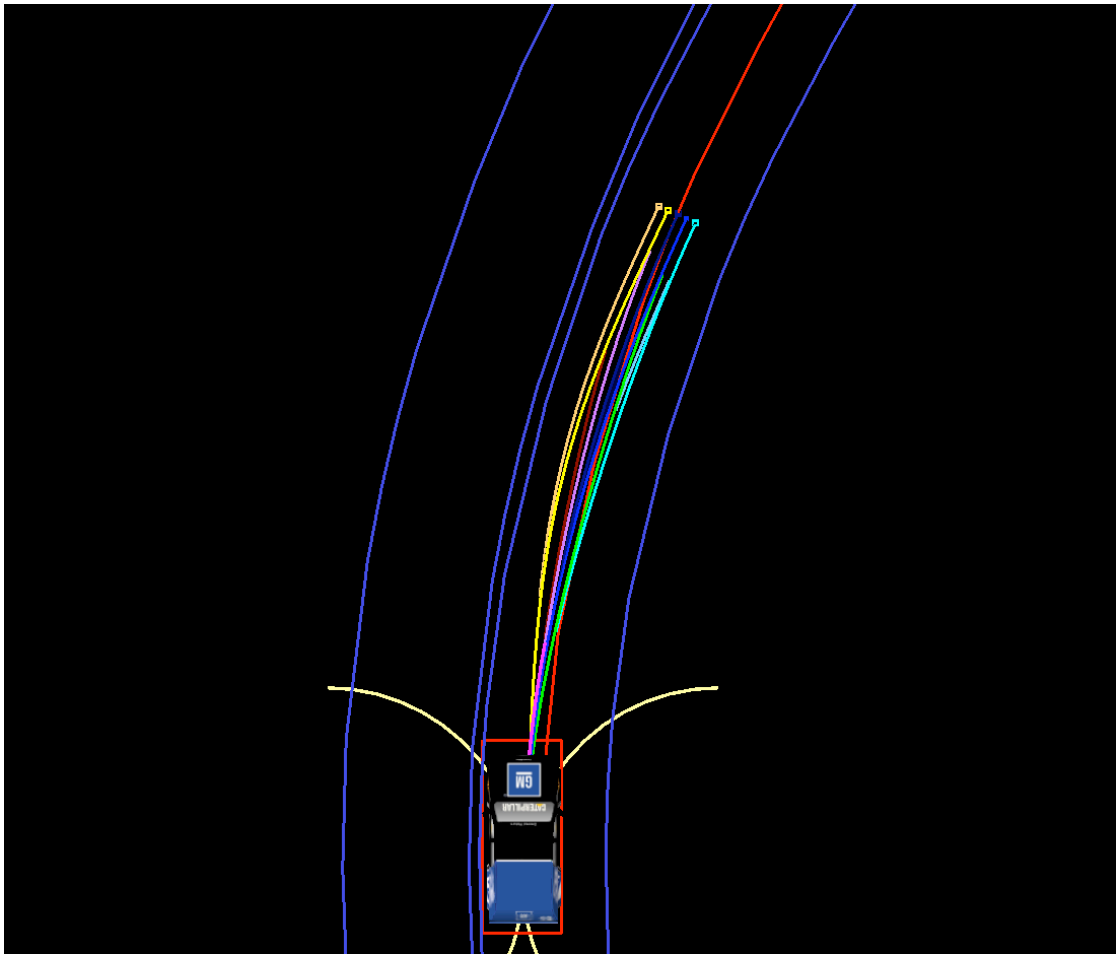
Trajectory Generation

- Generate set of trajectories along path



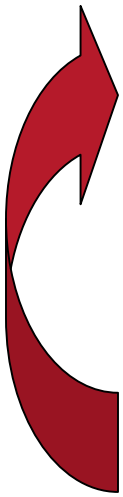
Smooth & Sharp Trajectories





How we generate each trajectory

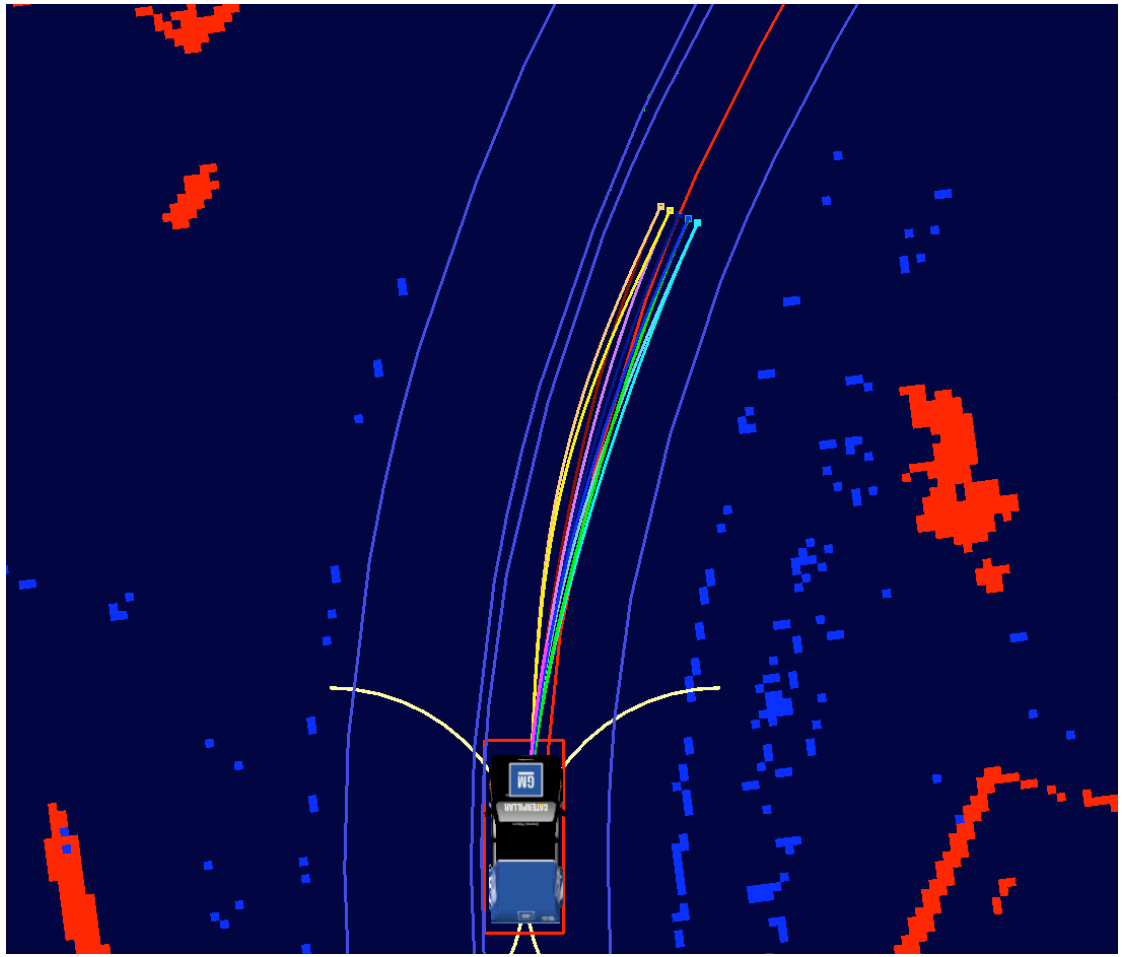
- Task: Compute controls to transition vehicle from state $\mathbf{x_I}$ to desired state $\mathbf{x_D}$



1. Grab initial set of controls parameters
2. Forwards-simulate controls using vehicle model
3. Compare resulting endpoint to desired endpoint
4. Update controls parameters to reduce endpoint error

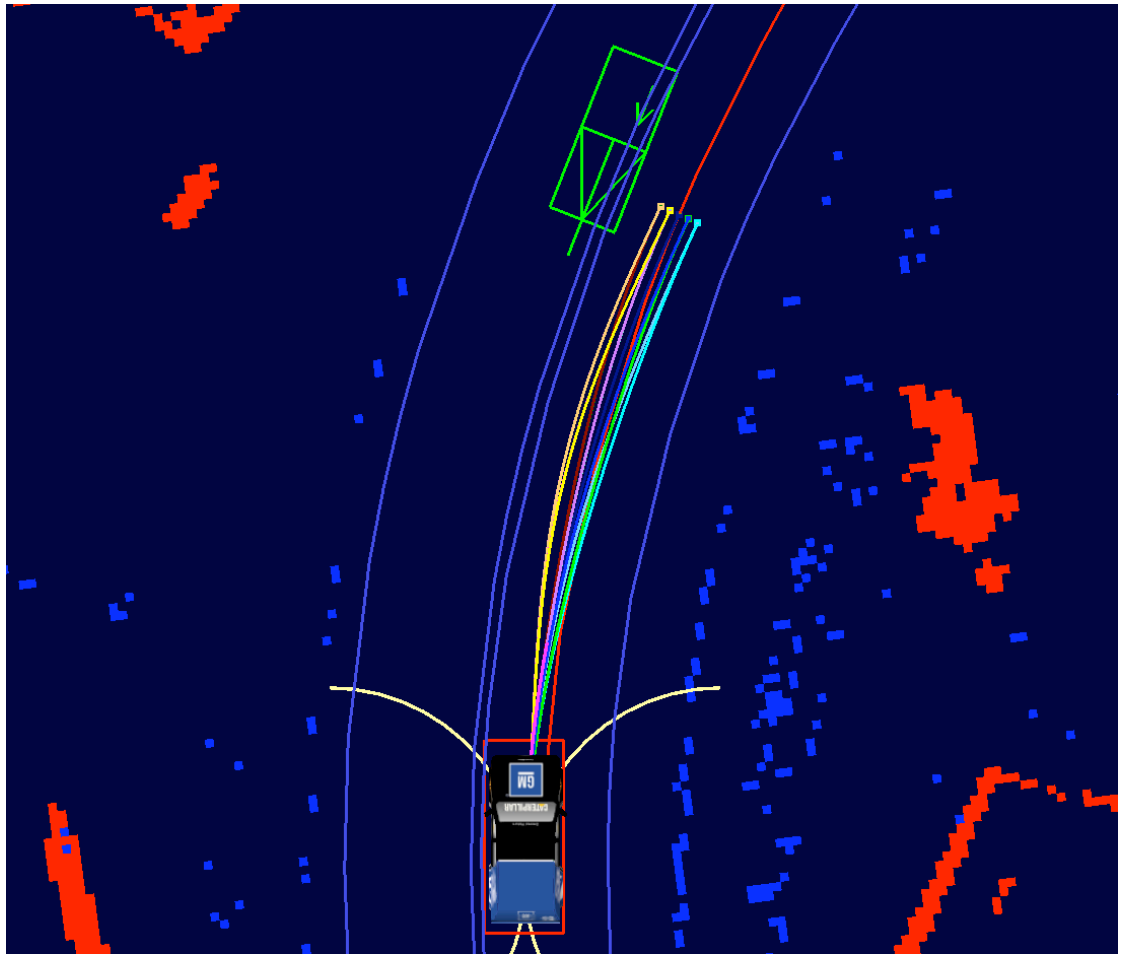
Trajectory Evaluation

- Check against path and static obstacles



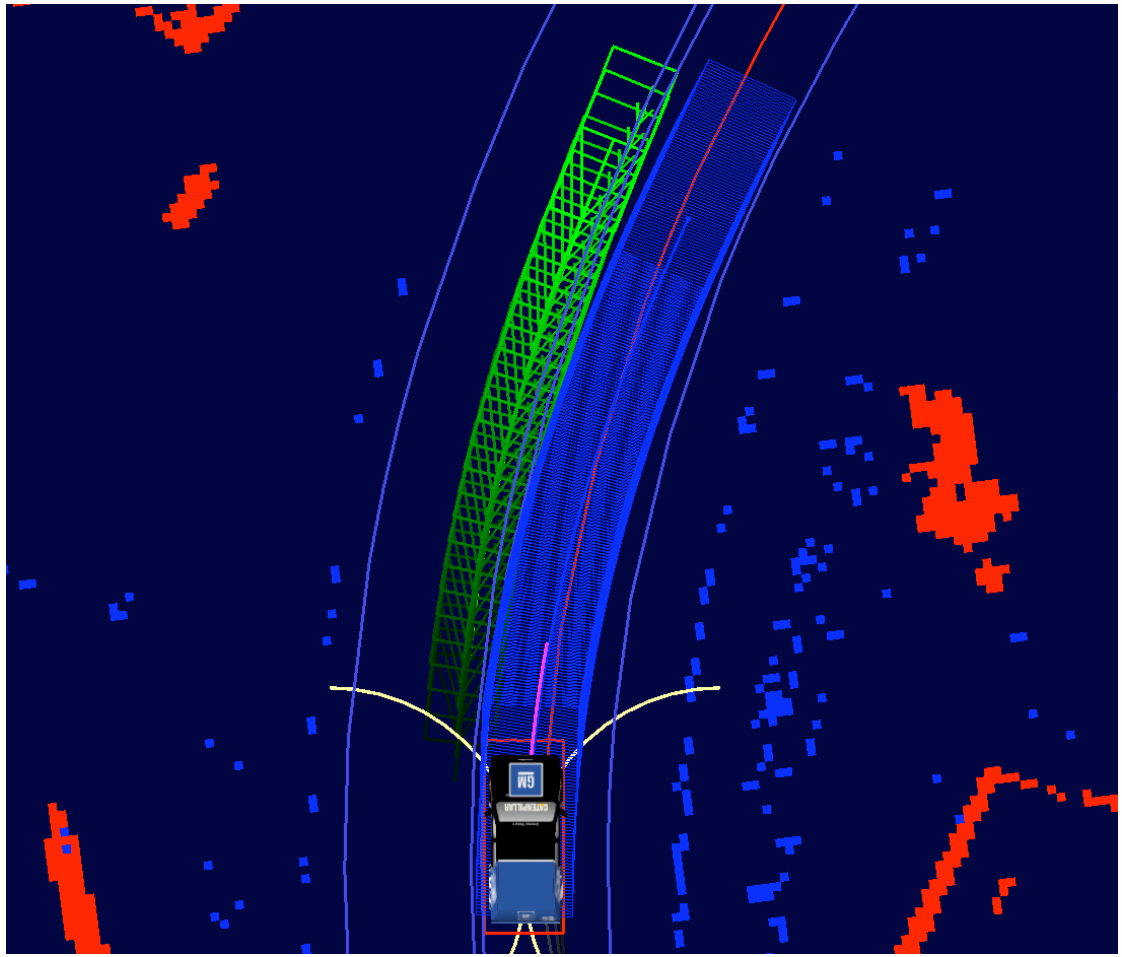
Trajectory Evaluation

- Check against dynamic obstacles



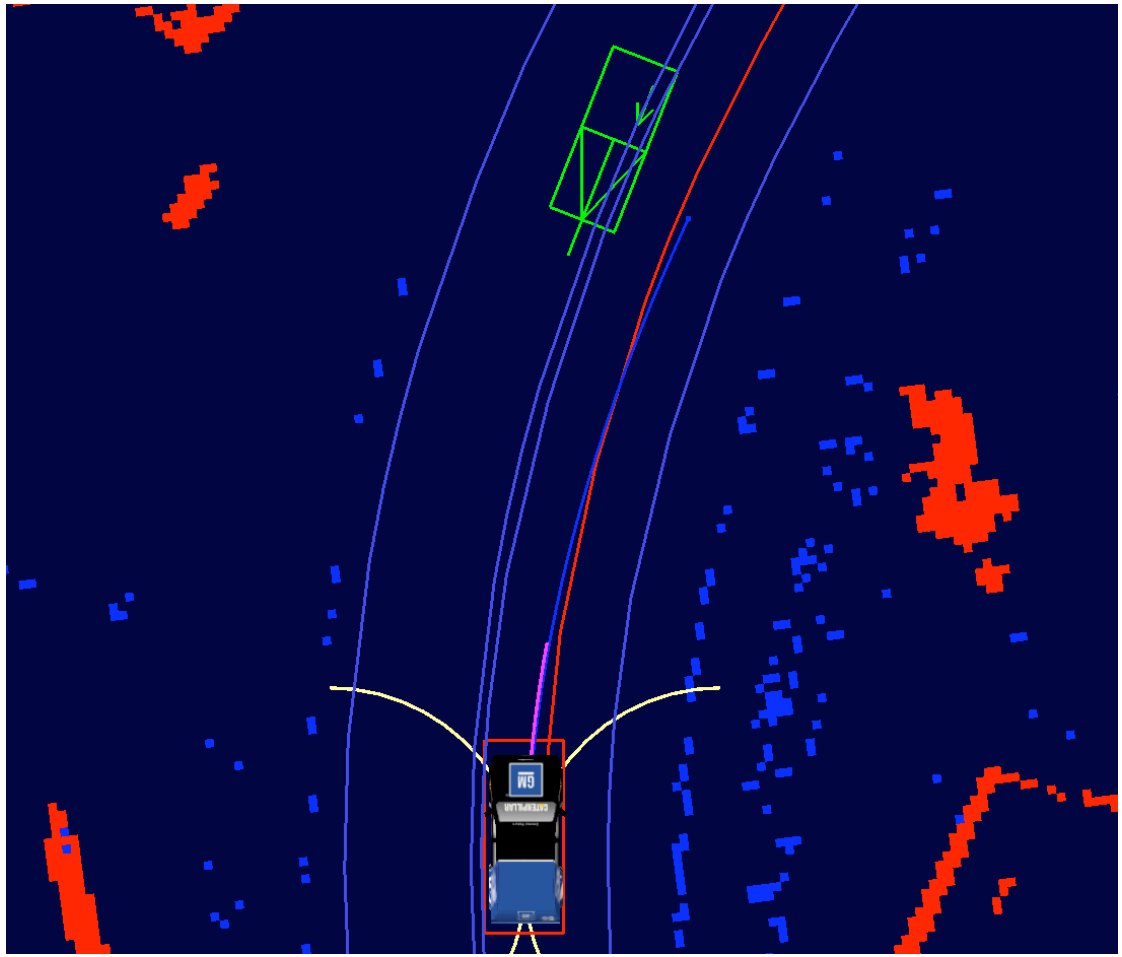
Trajectory Evaluation

- Check against dynamic obstacles

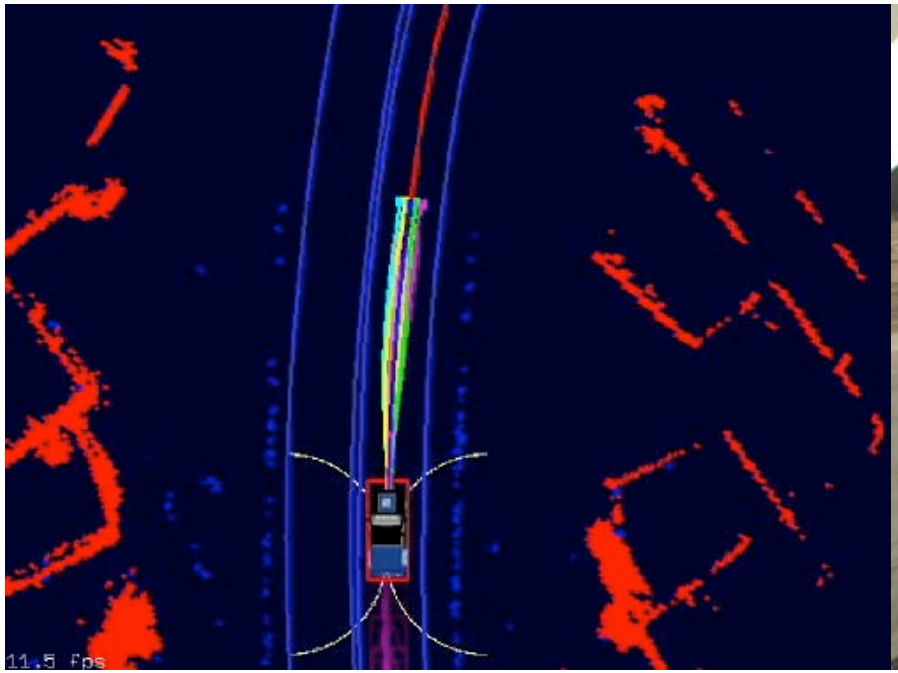


Selection and Execution

- Execute best trajectory



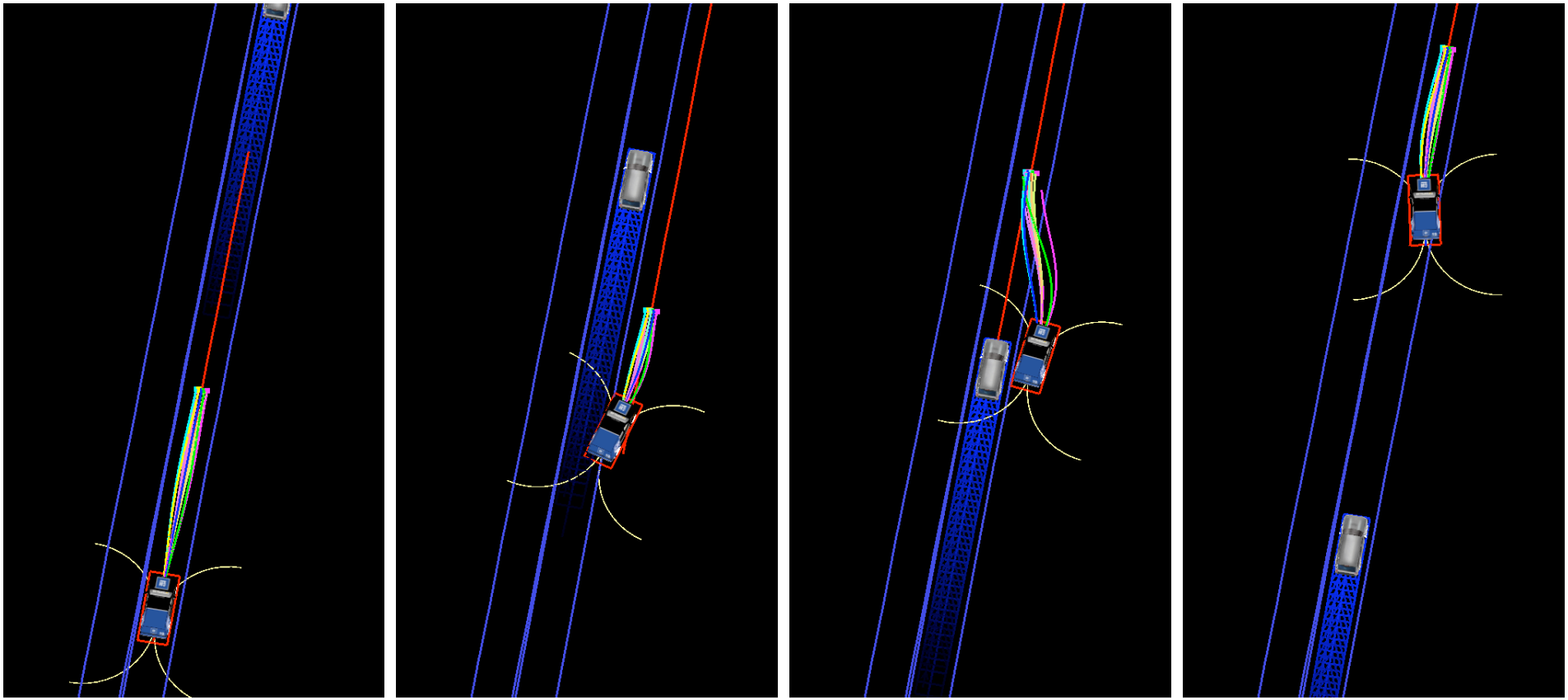
Examples



Lane Changes

TartanRacing @Carnegie Mellon

Defensive Driving

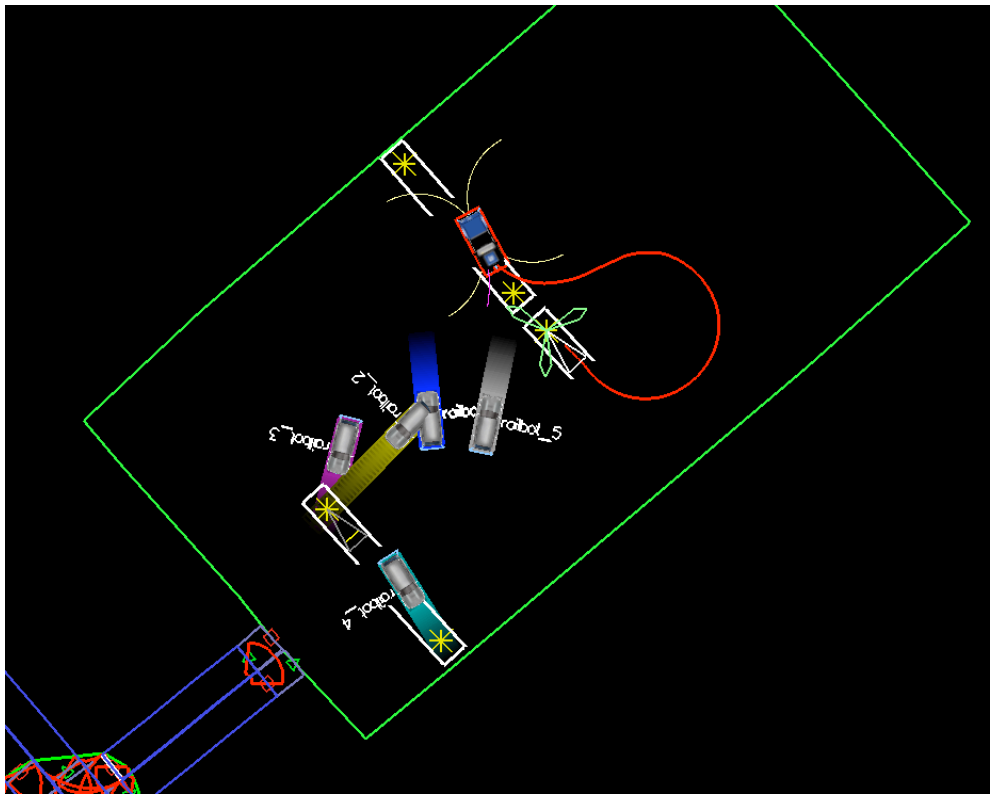


Unstructured Planning

- Task: Maneuver vehicle into desired pose without relying on any guiding structure

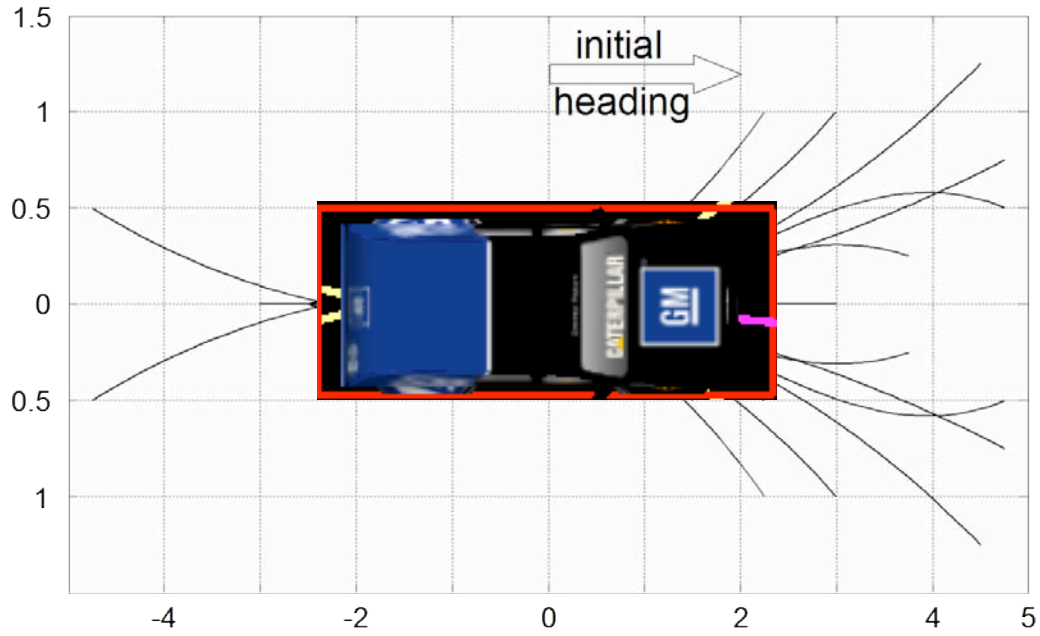
Challenges:

- Large distances
- Static & dynamic obstacles
- Relatively high speeds
- Complex maneuvers
- Non-holonomic vehicle



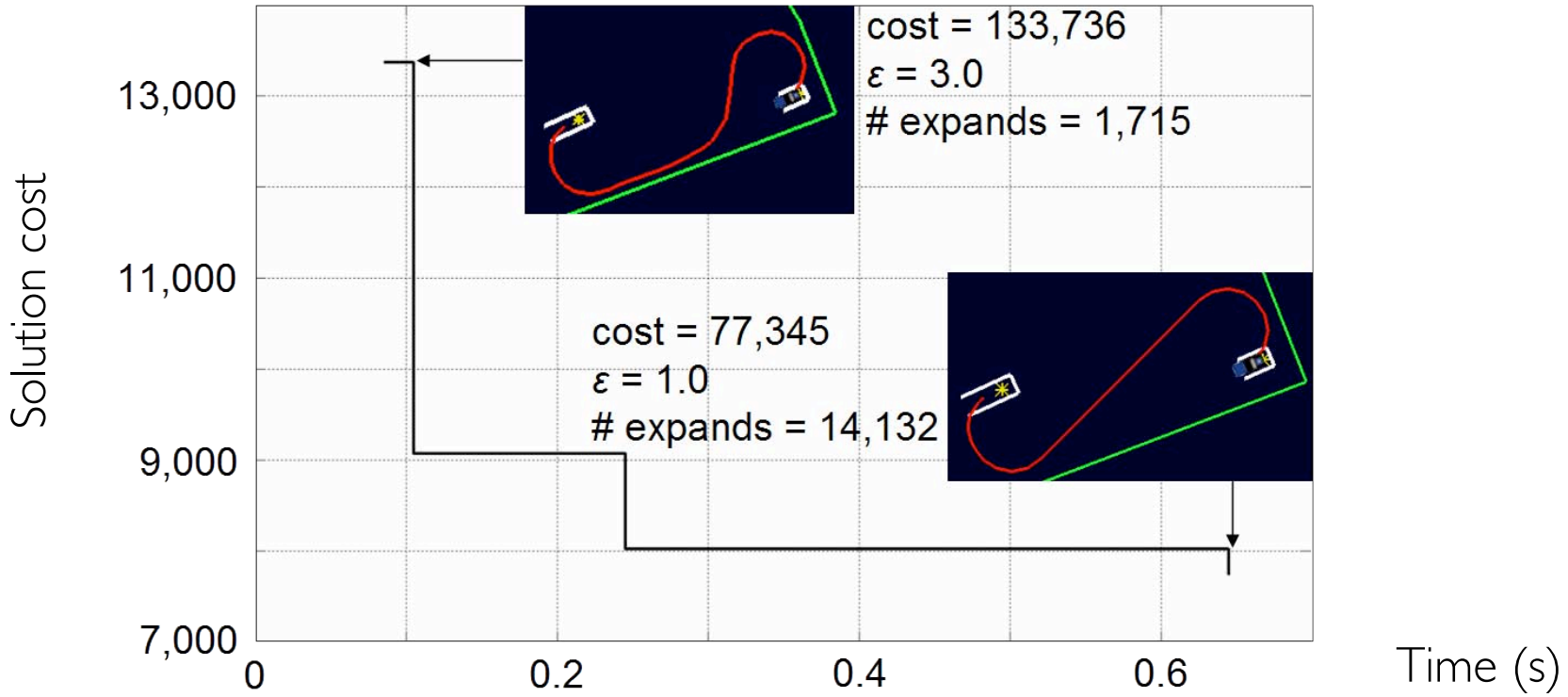
Lattice Planner

- Search over 4D state space: $(x,\ y,\ \theta,\ v)$
- For each state, consider fixed set of actions



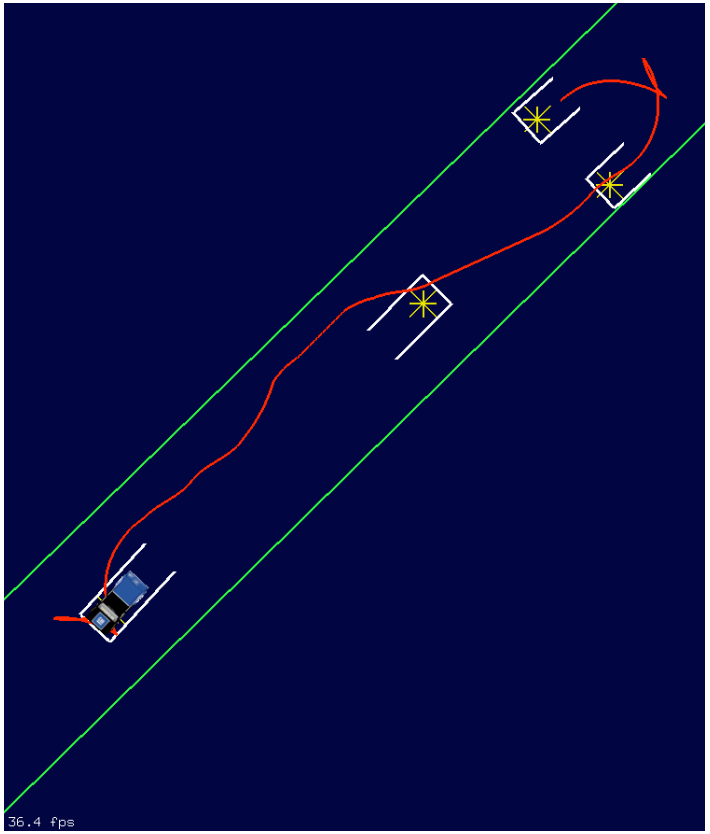
Search Algorithm

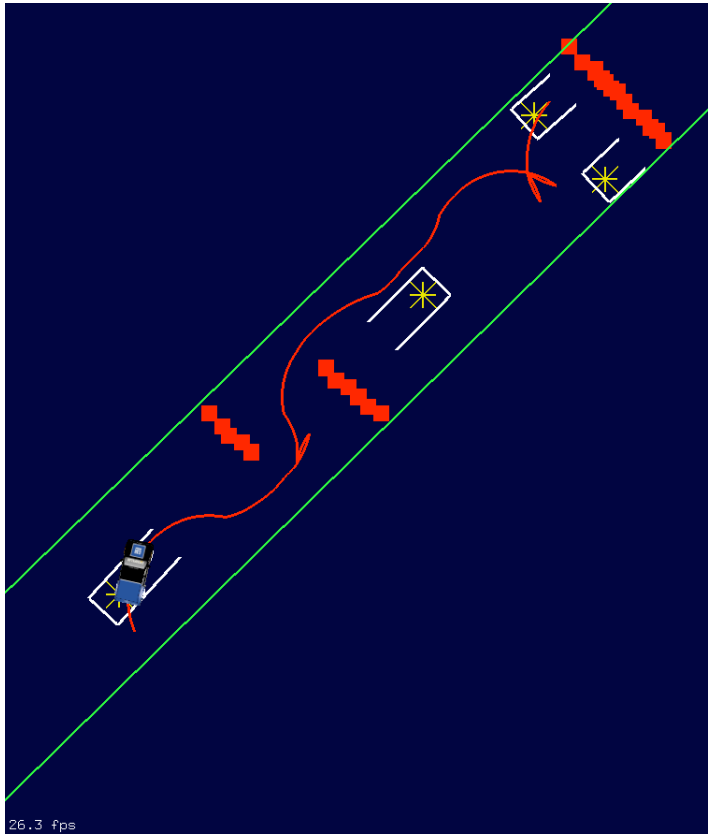
- Anytime D*
  - Initial, suboptimal solution generated quickly
  - Solution improved while time allows



Search Algorithm

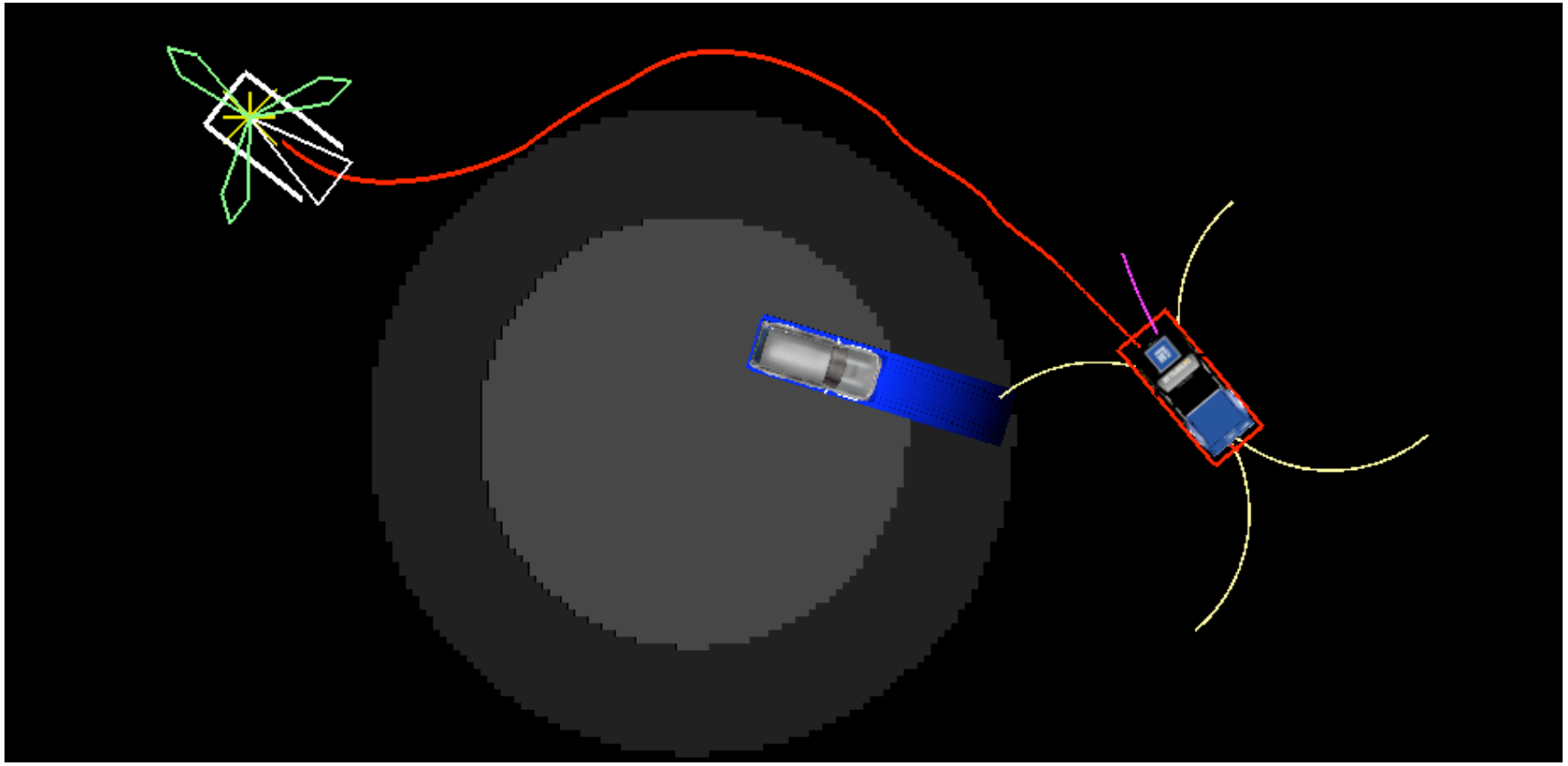
- Anytime D*
  - Solution repaired given new information





Avoiding Dynamic Obstacles

- Soft and hard constraints



Tracking Complex Paths

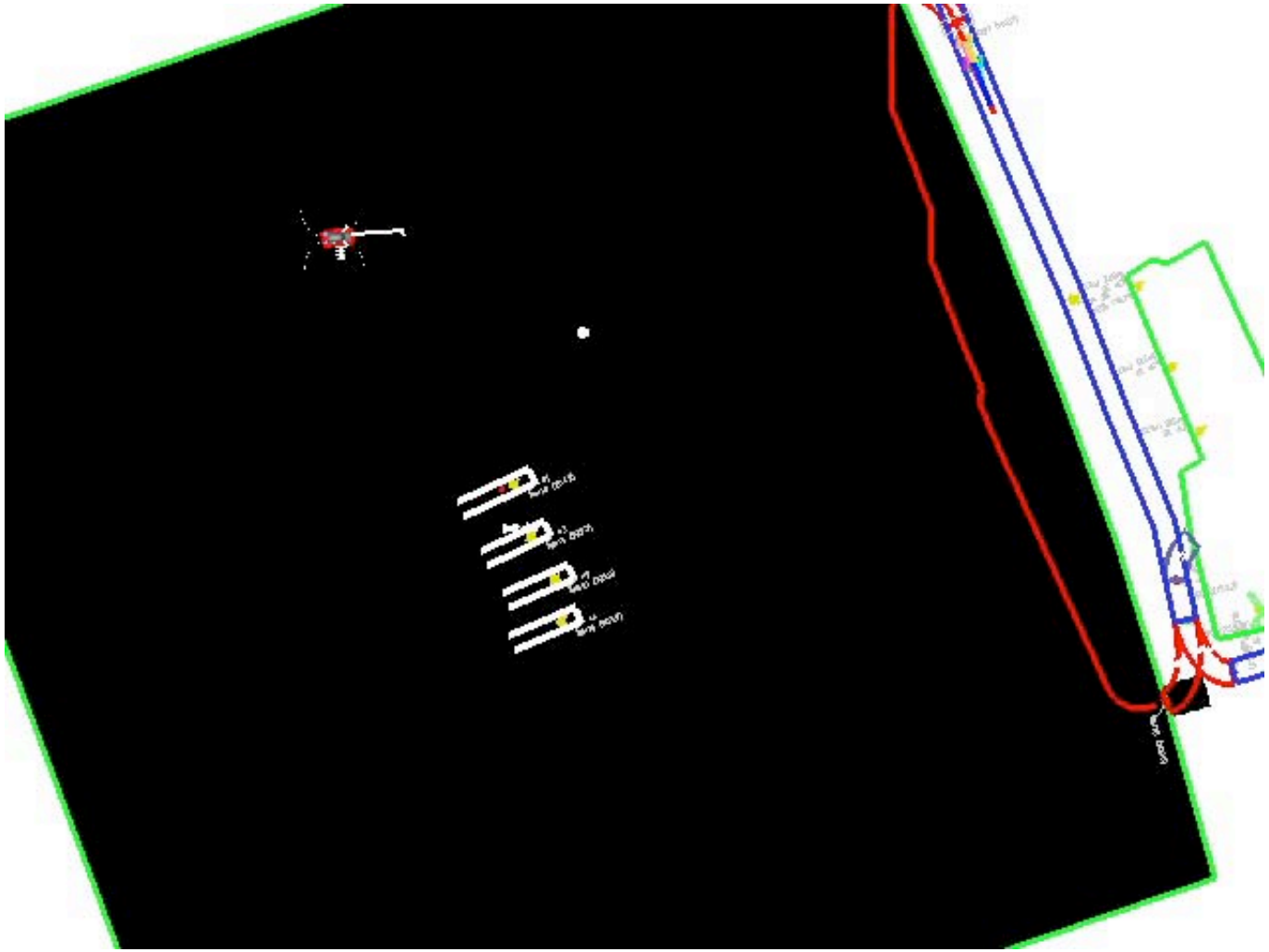
- Set of concatenated trajectories ending on path

Optimizations

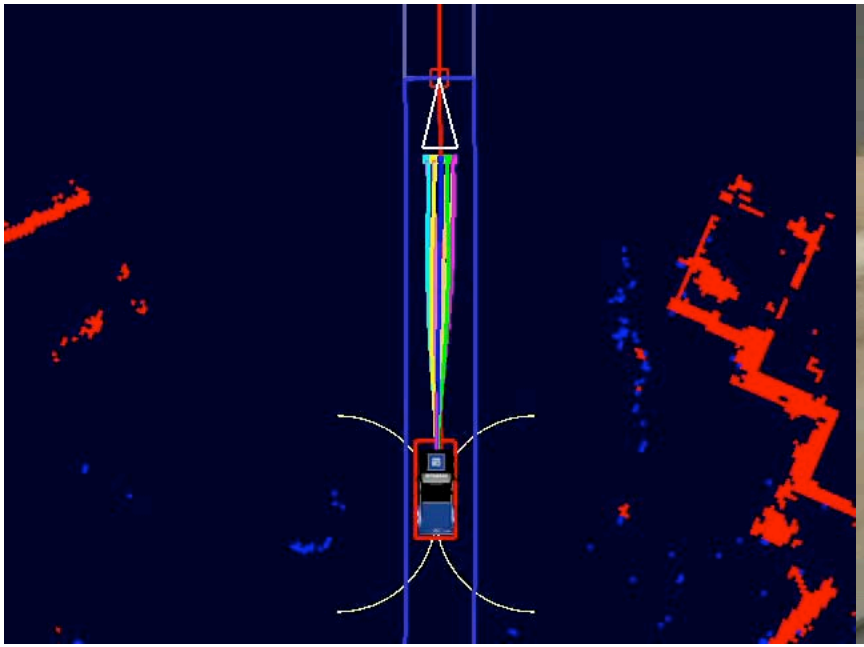
- Multi-resolution Lattice
  - Dense, high-resolution around robot & goal
  - Coarse, low-resolution elsewhere
- Combined Heuristic Function
  - Mechanism-constrained free-space heuristic
  - Environment-constrained 2D heuristic
- Reducing Convolution Efforts
  - Optimistic/pessimistic configuration-space maps
- Efficient Replanning
  - Hierarchical action representation
- **Preplanning**
- **Concurrent Planning**

In Action

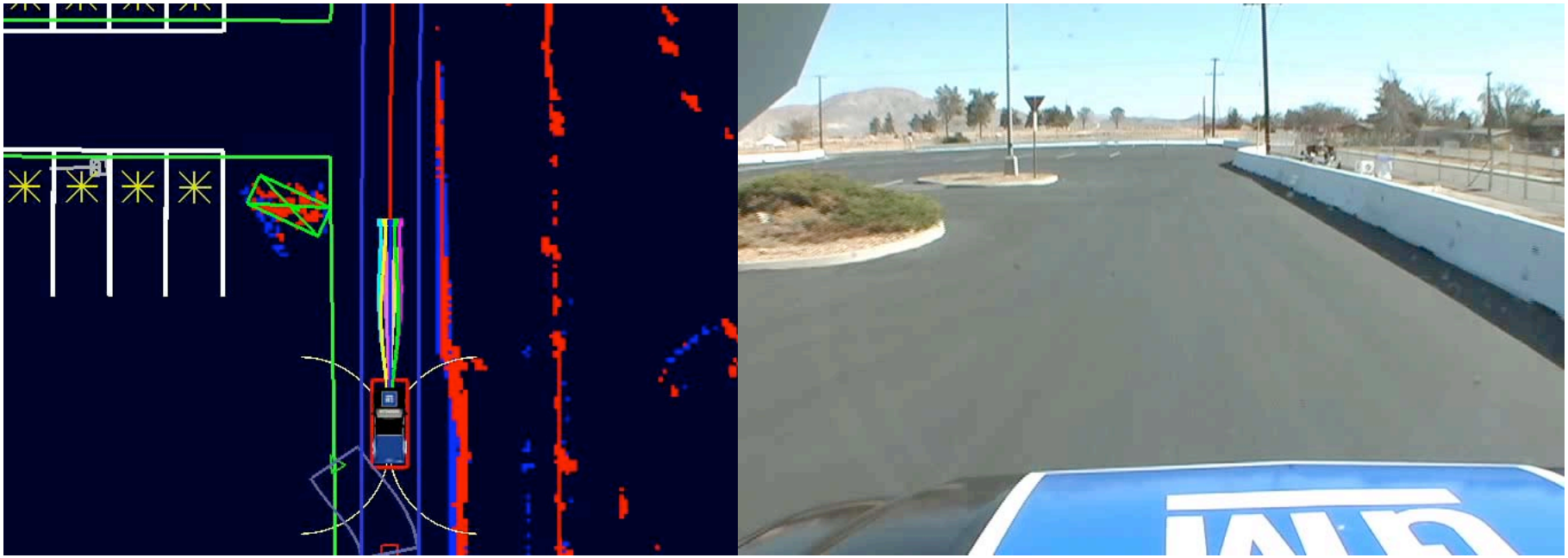
In Action



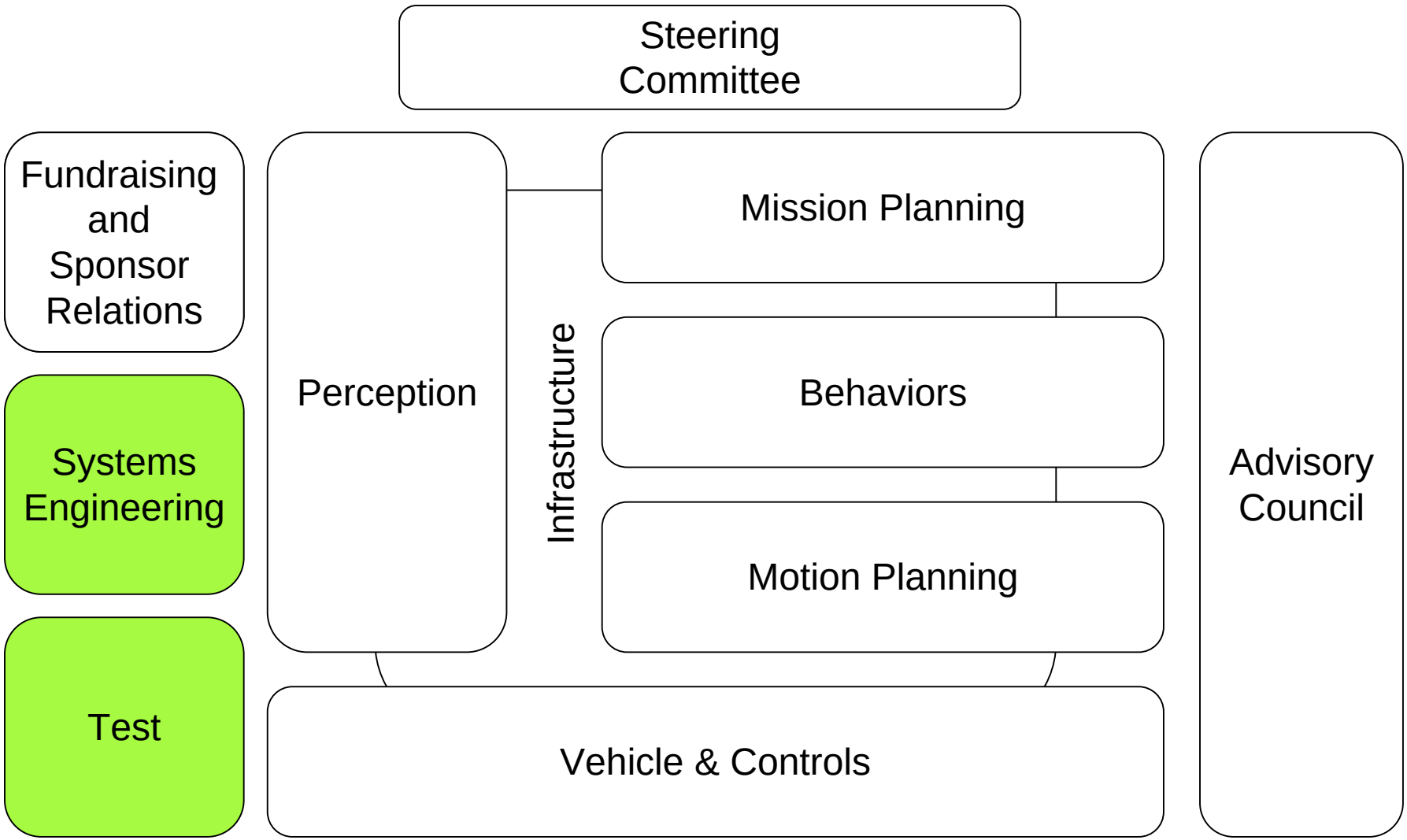
In Action



In Action



Organization



Systems Engineering

- Developed 400+ requirements
  - Derived from DARPA documents and internal design
  - Stuck with it
  - Tested, Verified or Cancelled every requirement
- Tracked progress in weekly system tests
- Tracked and addressed reliability defects to root cause
- Tracked software/development defects using bugzilla
  - Over 533 bugs closed between Sept 1 and Nov 3

Proven	DD-2230.1	Vehicle shift actuation shall be capable of moving between Drive and Reverse and vice versa in less than 10 seconds.	
Proven	DA-2250.4	Vehicle stops so front bumper is within 1 meter of the center of the stop line at intersection.	
Pending	INT-2252.4	Planning shall be capable of achieving a mission goal location with less than 50cm of error.	Does this req make use of GPS? CU – this is planning doing the right thing given the input from perception. I'm not sure if this is proven or not, so I guess not.
Pending	DD-2260.4	Vehicle shall be capable of localizing stop line relative to robot with <40cm error.	GPS acquisition coupled to percep not finalized
Pending	DD-2270.4	Vehicle shall be capable of controlling stop position with <50cm error.	GPS acquisition coupled to percep not finalized
Proven	DD-2630.4	Vehicle shall be capable of handling a case of near simultaneous arrival with other vehicles.	
Proven	DA-2580.3	Vehicle shall be capable of respecting precedence order at intersections and not proceed out of turn.	

Proven	DD-2590.3	Vehicle shall be capable of identifying if entry points at an intersection are occupied.	

Testing

- 45+ days of pre-race acceptance testing
  - 6+ hours/ day of active robot test time
- Ongoing, weekly system tests
  - Regressive and developmental system tests
  - Published report with requirements gap analysis
- Daily component & development testing
- > 3200km of testing at 4 different sites across US

Results

- 11 vehicles in race
- 6 completed
- Great day for robotics

TartanRacing @ Carnegie Mellon

Thanks!

GM
CATERPILLAR
DARPA
Continental
Carnegie Mellon
ibeo
Google
intel
NetApp
APPLANIX
vector
Viewpoint
hp
MACOM