Comp 790-058 Lecture 07: Autonomous Driving: Planning

October 3, 2017

Andrew Best
University of North Carolina, Chapel Hill

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

Administrative

- Homework due:
  - 11:59 PM October 4th (tomorrow)
- Project Proposals:
  - Next week
  - should make a WWW page of your project topic, 4 parts:
    - 1. What is the goal of your project? What is your motivation?
    - 2. What is the prior state of the art? Please include pointers to related work or WWW sites related to the prior work?
    - 3. What do you plan to accomplish over the semester?
    - 4. What is your timeline between Oct. 10 - Dec. 8? Remember the final project presentation would be after Dec. 8 deadline. I want you to come up with 2 week milestones (between Oct. 10 - Dec. 8) and put them on the WWW page? That way I want to make sure that you have thought in detail about the todo list for the project.
- 15-20 minute presentation slot on Oct. 10

Main Idea

- Motion Planning: term used in robotics for the process of breaking down a desired movement task into discrete **motions** that satisfy movement constraints and possibly optimize some aspect of the movement

Main Idea

- Motion Planning
  - Fuse prior information, sensing, mapping, etc. to generate:
    - Set of actions leading from some initial state to a goal
    - OR continuous action function from initial state to goal
  - Motion planning for navigation is:
    - Hierarchical
    - Recurrent

Structure

- Recap
  - Perception
  - Localization
  - Planning
- State, Kinematics, and Dynamics Models
- Planning
- AutonoVi-Sim

Autonomous Driving

- Autonomous vehicle: a motor vehicle that uses artificial intelligence, sensors and global positioning system coordinates to drive itself without the active intervention of a human operator
- Focus of enormous investment [$1b+ in 2015]

Tesla

Waymo

Nutonomy

Autonomous Driving: Levels of Autonomy

- 0: Standard Car
- 1: Assist in some part of driving
  - Cruise control
- 2: Perform some part of driving
  - Adaptive CC + lane keeping
- 3: Self-driving under ideal conditions
  - Human must remain fully aware
- 4: Self-driving under near-ideal conditions
  - Human need not remain constantly aware
- 5: Outperforms human in all circumstances

Autonomous Driving

- Urban driving is particularly challenging

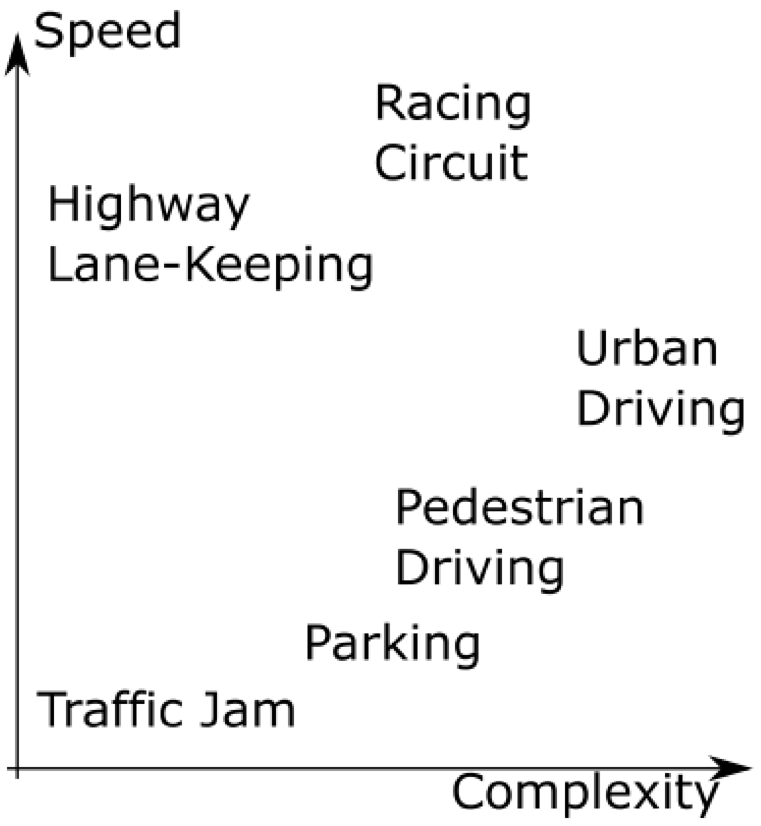


Figure 1. Complexity and operating velocity for various driving scenarios.

Autonomous Driving: Main Components

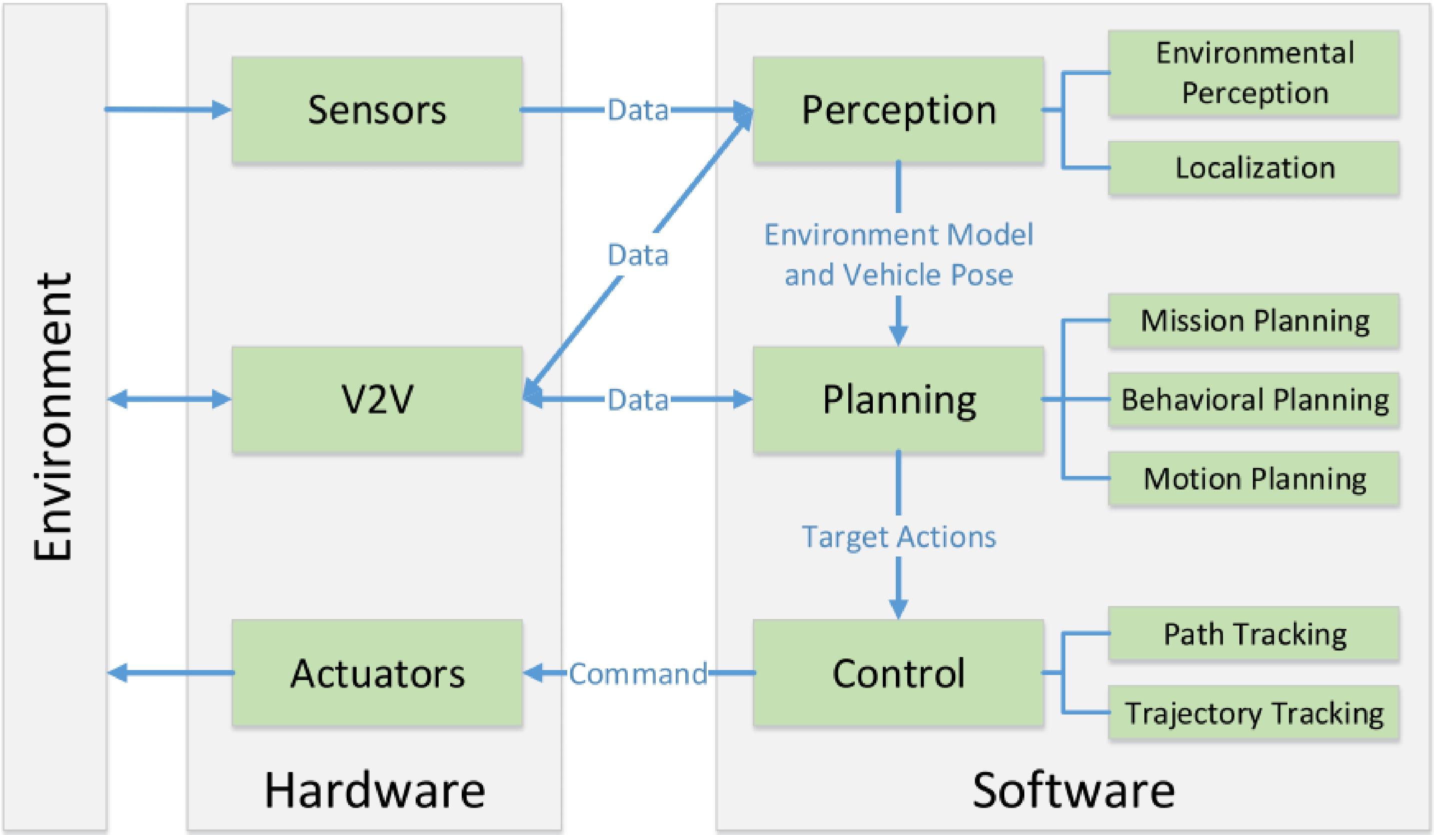


Figure 2. A typical autonomous vehicle system overview, highlighting core competencies.

Autonomous Driving: Main Components

- Perception
  - Collect information and extract relevant knowledge from the environment.

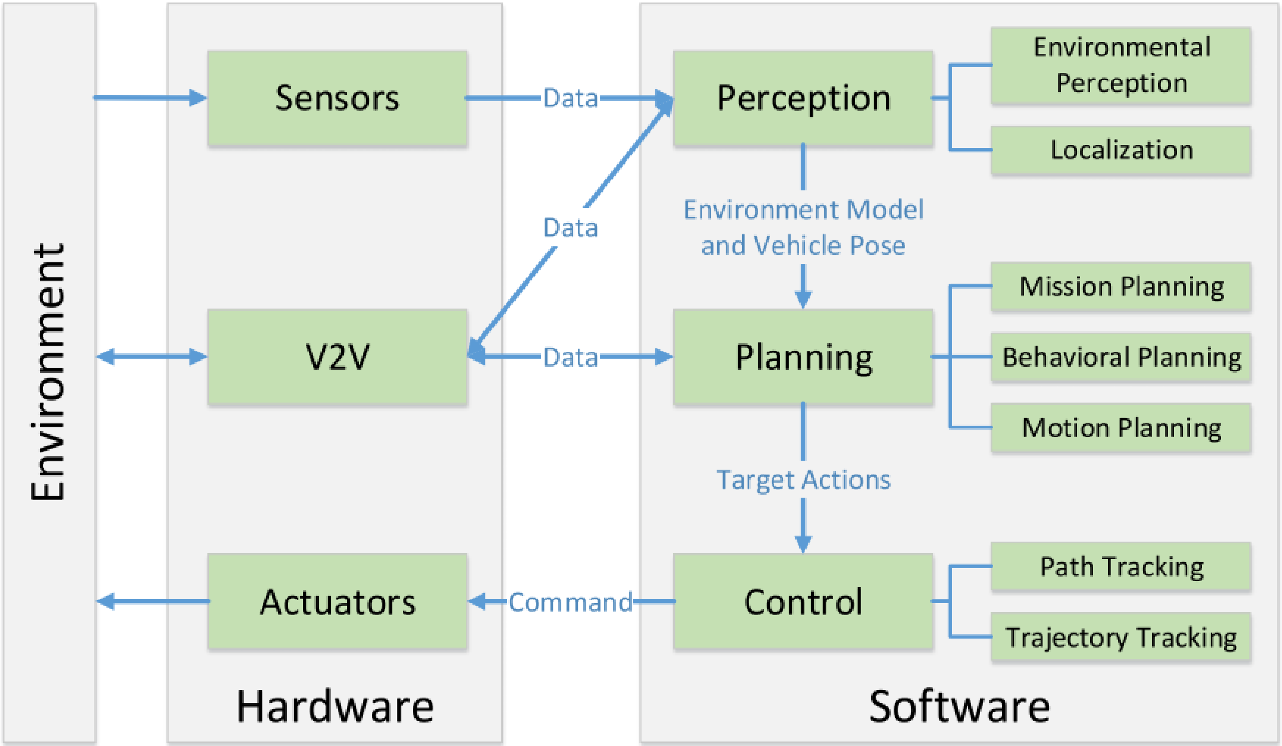


Figure 2. A typical autonomous vehicle system overview, highlighting core competencies.

Autonomous Driving: Main Components

- Planning
  - Making purposeful decisions in order to achieve the robot's higher order goals

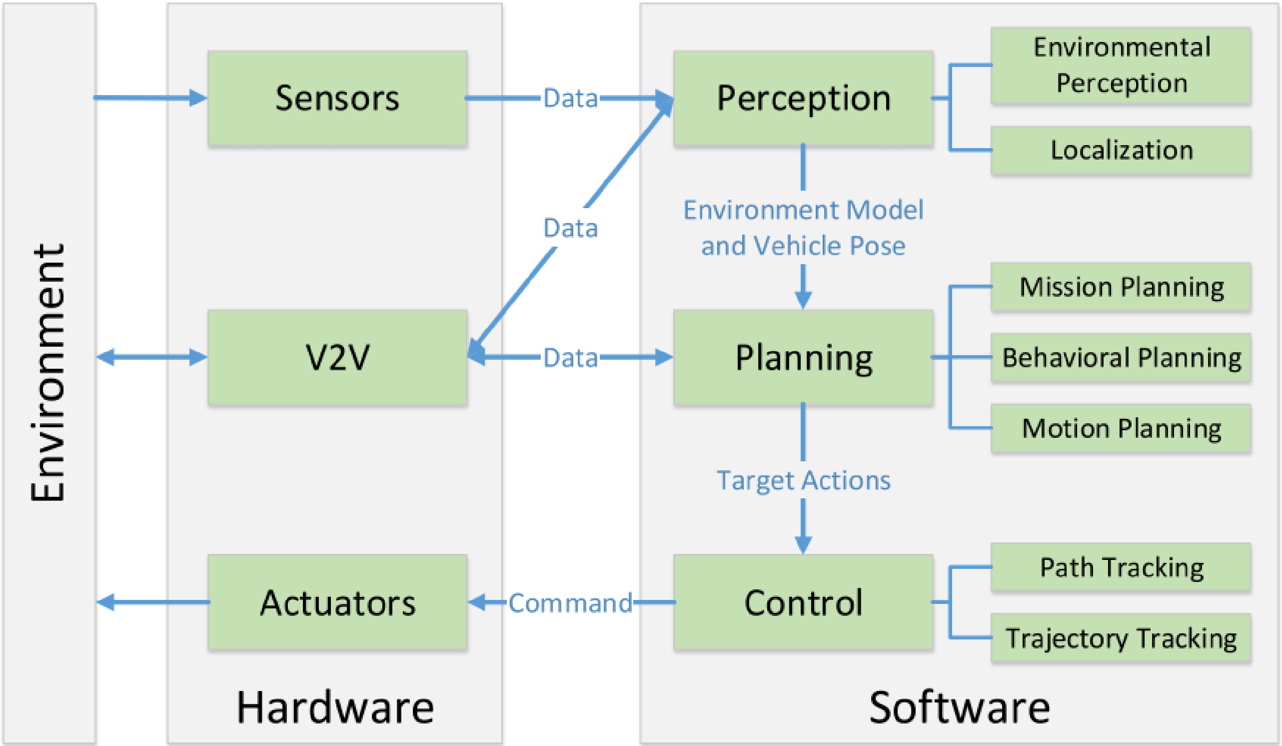


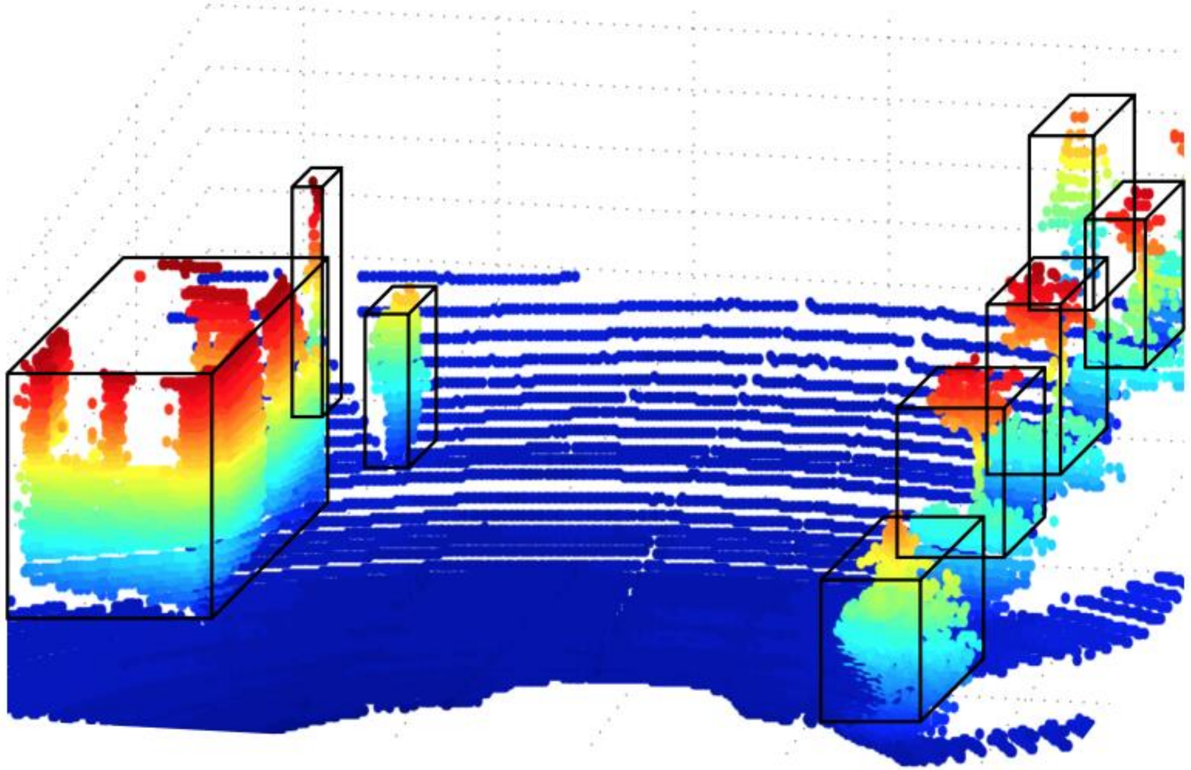
Figure 2. A typical autonomous vehicle system overview, highlighting core competencies.

Structure

- Recap
  - Perception
  - Localization
  - Planning
- State, Kinematics, and Dynamics Models
- Planning
- AutonoVi-Sim

Autonomous Driving: Perception using LIDAR

- Light Detection and Ranging
  - Illuminate target using pulsed laser lights, and measure reflected pulses using a sensor

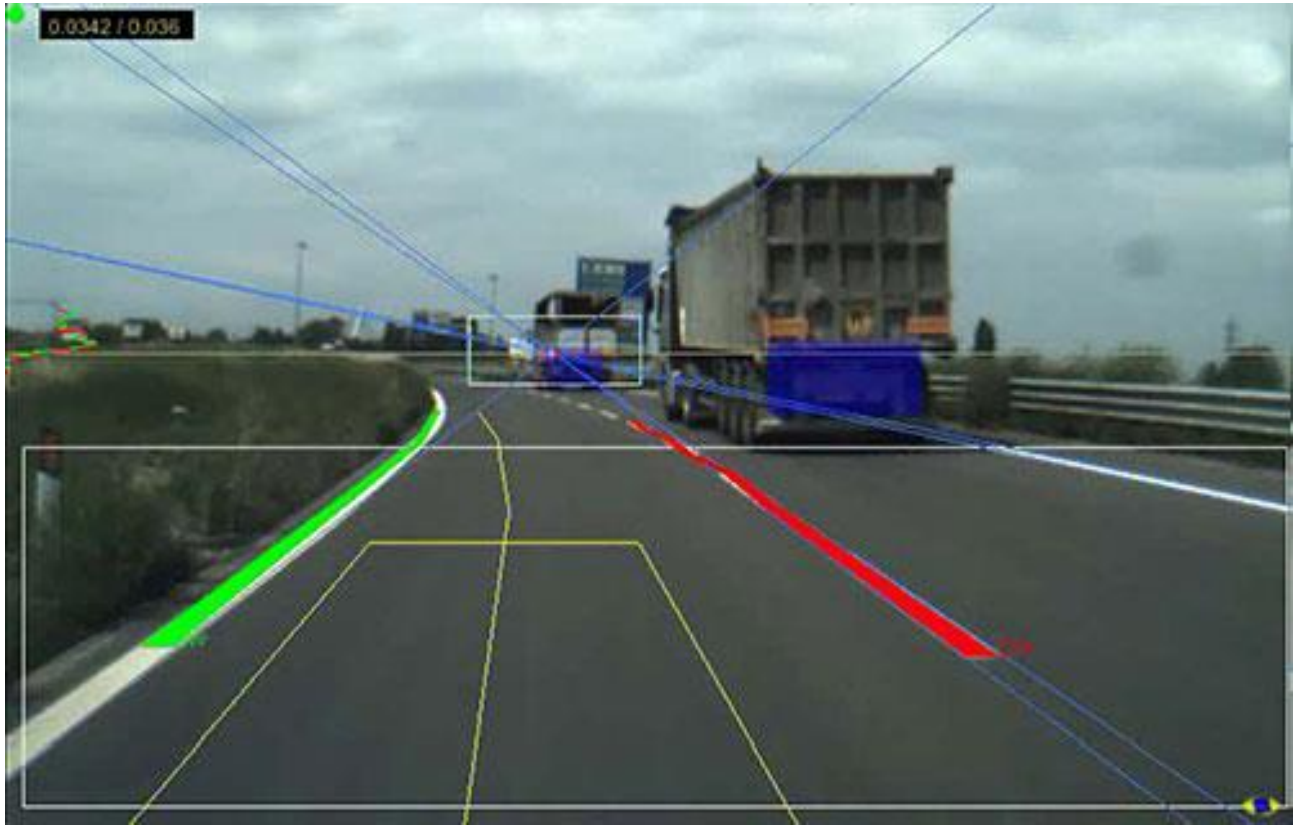


Autonomous Driving: Perception using LIDAR

- LIDAR in practice
  - Velodyne 64HD lidar
    - https://www.youtube.com/watch?v=nXlqv_k4P8Q

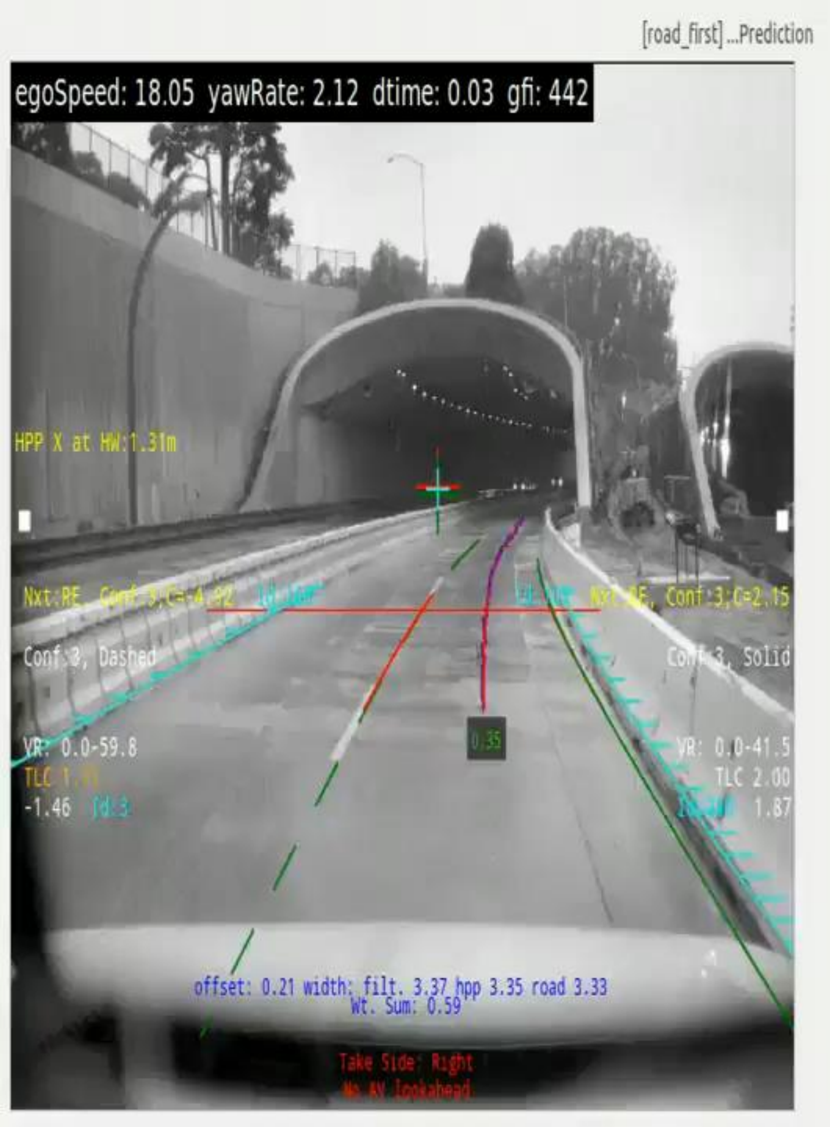
Autonomous Driving: Perception using Cameras

- Camera based vision
  - Road detection
    - Lane marking detection
    - Road surface detection
  - On-road object detection



Autonomous Driving: Perception using Cameras

- Sensing Challenges
  - Sensor Uncertainty
  - Sensor Configuration
  - Weather / Environment



Structure

- Recap
  - Perception
  - Localization
  - Planning
- State, Kinematics, and Dynamics Models
- Planning
- AutonoVi-Sim

Autonomous Driving: Vehicle Localization

- Determining the pose of the ego vehicle and measuring its own motion
- Fusing data
  - Satellite-based navigation system
  - Inertial navigation system
- Map aided localization
  - SLAM

Structure

- Recap
  - Perception
  - Localization
  - Planning
- Kinematics & Dynamics Models
- Planning
- AutonoVi-Sim

Autonomous Driving: Planning

- Compare to Pedestrian Techniques:
  - Route Planning: road selection (global)
  - Path Planning: preferred lanes (global)
  - Maneuver-search: high level maneuvers (local)
  - Trajectory planning: Lowest level of planning (local)

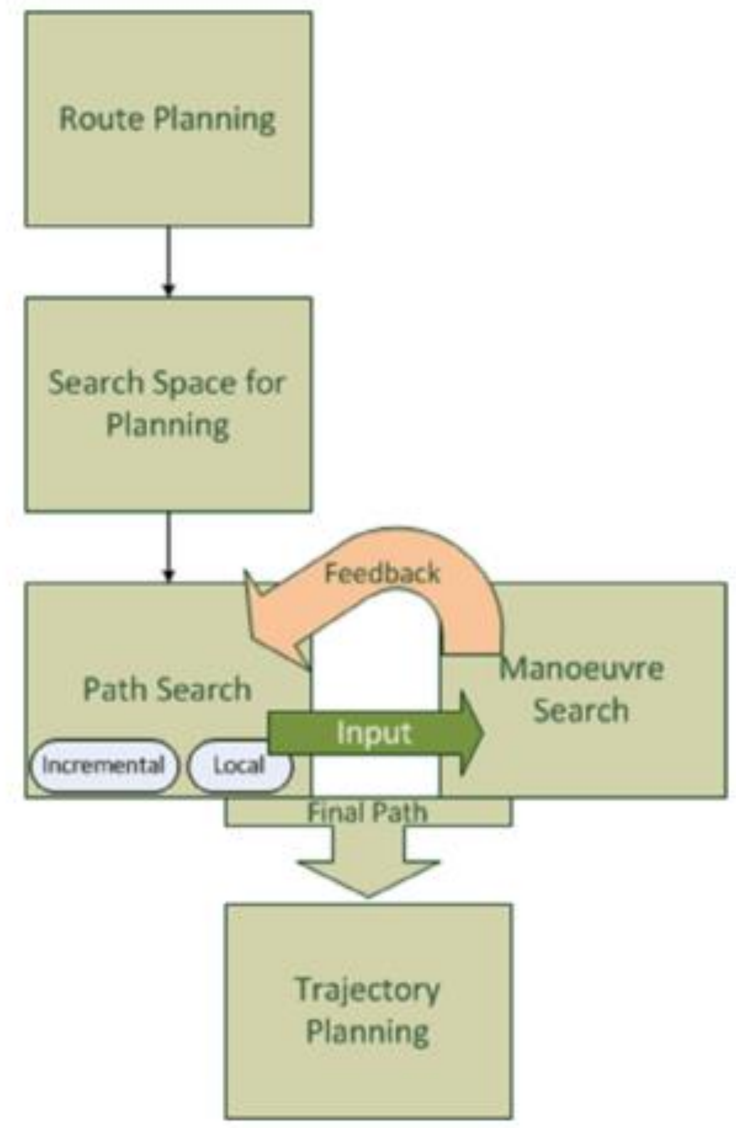


Fig. 2. A flow chart of planning modules.

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Structure

- Recap
- State, Kinematics, and Dynamics Models
  - State Space
  - Kinematic constraint models of the vehicle
  - Dynamic constraint models of the vehicle
- Planning
- AutonoVi-Sim

Autonomous Driving: State Space

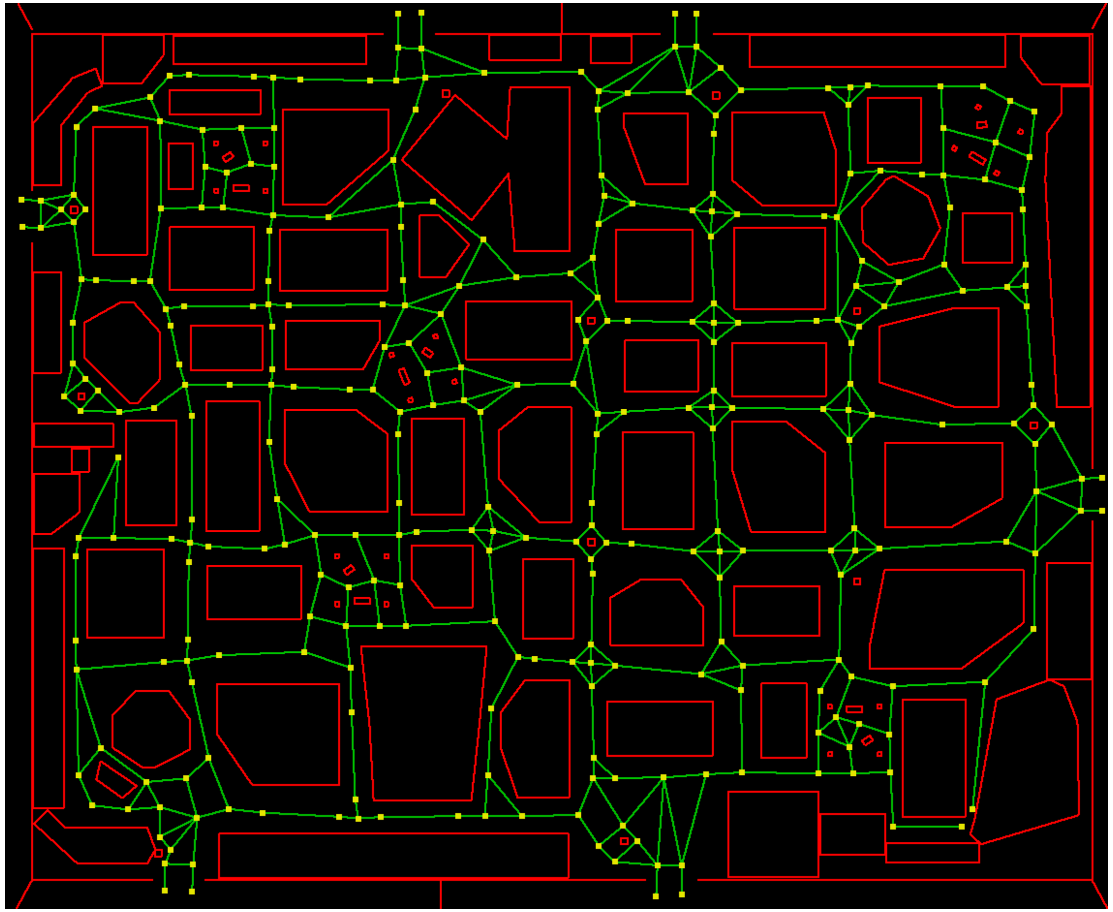
- “The set of attribute values describing the condition of an autonomous vehicle at an instance in time and at a particular place during its motion is termed the ‘state’ of the vehicle at that moment”
- Typically a vector with position, orientation, linear velocity, angular velocity
- State Space: set of all states the vehicle could occupy

Autonomous Driving: State Space

- “The set of attribute values describing the condition of an autonomous vehicle at an instance in time and at a particular place during its motion is termed the ‘state’ of the vehicle at that moment”
- Typically a vector with position, orientation, linear velocity, angular velocity
- State Space: set of all states the vehicle could occupy

Autonomous Driving: State Space

- Recall Pedestrian Planning:
  - Roadmap is essential a graph of potential agent states



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Autonomous Driving: State Space

- Examples:
  - 3D space with velocity
    - $(p_x, p_y, p_z, \theta_x, \theta_y, \theta_z, v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$
    - $(\vec{p}, \vec{\theta}, \vec{v}, \vec{\omega})$
  - 2D space with acceleration
    - $(p_x, p_y, \theta, v_x, v_y, \omega, a_x, a_y, \alpha)$
    - $(\vec{p}, \theta, \vec{v}, \omega, \vec{a}, \alpha)$

Autonomous Driving: State Space

- Examples:
  - 2D space with blinker booleans
    - $(\vec{p}, \theta, \vec{v}, \omega, bl_l, bl_r)$
  - State contains everything we need to describe the robot's current configuration!
  - Neglect some state variables when planning

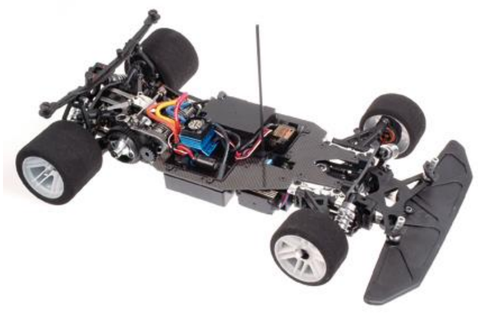
Structure

- Recap
- State, Kinematics, and Dynamics Models
  - State Space
  - Kinematic constraint models of the vehicle
  - Dynamic constraint models of the vehicle
- Planning
- AutonoVi-Sim

Autonomous Driving: Holonomicity

- “Holonomic” robots
  - Robots whose motion capability is independent of their orientation
  - Controllable DOF == total DOF
- Examples:
  - Quad-rotors
  - Omni-drive base
  - https://youtu.be/9ZCUxXajzXs

Autonomous Driving: Holonomicity



- Cars are “non-holonomic” robots
  - Typically 5 values describing physical
    - (2 Cartesian coordinates, orientation, linear speed, angular speed)
  - 2 “kinematic” constraints
    - Can only move forward or backward, tangent to body direction
    - Can only steer in bounded radius

Kinematic Constraints

- Kinematics of Motion
  - “the branch of mechanics that deals with pure motion, without reference to the masses or forces involved in it”
  - Equations describing conversion between control and motion
  - Control: inputs to the system
    - In vehicle: steering and throttle
    - Also referred to as “Action” in literature

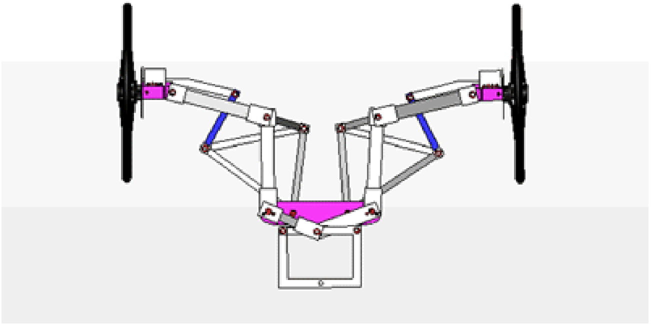
Autonomous Driving: Holonomicity

- kinematic and dynamic constraints can be considered "rules" governing the state evolution function
- For state $s_t \in S$, control input $u_t \in \mathrm{U}$, time $t \in T$:
  - $\mathrm{F}(\mathrm{s_t}, \mathrm{u}_t, \Delta t) \rightarrow \mathrm{s_{t+1}}$
- Ex:
  - A car cannot turn in place. No amount of steering will accomplish this
  - A Roomba can turn in place

Kinematic Constraints

- Kinematic models of a car
  - Single-track Bicycle (or simple car model)
    - 3-DOF configuration: (x,y,θ)
    - 2-DOF control: steering (φ), speed (v)
    - Full state: (x,y,θ,v, φ,L)
  - Equations of motion:
    - $\dot{p}_x = v * cos(\theta) \quad \dot{p}_y = v * sin(\theta)$

$$\dot{\theta} = \frac{v}{L} * \tan(\phi)$$



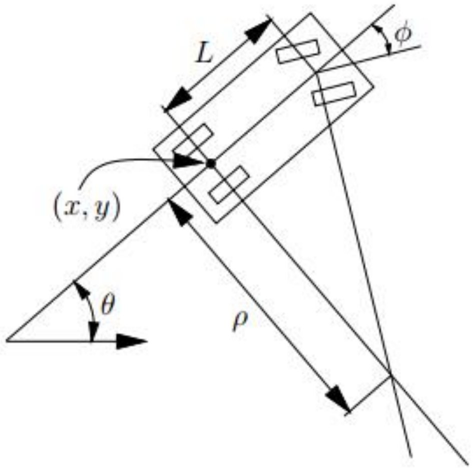


Figure 13.1: The simple car has three degrees of freedom, but the velocity space at any configuration is only two-dimensional.

Kinematic Constraints

- Kinematic models of a car
  - Single-track Bicycle example
  - https://www.youtube.com/watch?v=TyW1BPpHy18

  - Kinematic robot simulator provided as part of HW3

Kinematic Constraints

- Kinematic models of a car
  - Extended Car w. linear integrators
  - 6-DOF configuration (x,y,θ, φ,v,ω)
    - 2-DOF Control:

      steering rate (u_s), acceleration (u_v)
    - Full state ($x, y, \theta, v, \phi, \omega, u_s, u_v$, L)

Kinematic Constraints

- Extended Car w. linear integrators
  - Equations of motion
    - $\dot{p}_x = v * cos(\theta)$ $\dot{p}_y = v * sin(\theta)$

      $\dot{\theta} = \frac{\tan(\phi)}{L}$ $\dot{\phi} = \omega$ $\dot{\omega} = \mu_s$

      $\dot{v} = u_v$
  - Steering is continuous C^1
  - Velocity continuous
  - Control is more complex

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Kinematic Constraints

- Example: Stopping the car
  - Simple-car: $u_v = 0$
  - LI-car $u_v = -v$ iff $\max(U_v) \geq v$ *else* $\max(U_v)$
    - Car will not necessarily stop right away

- Error increases as we increase the number of integrators

Kinematic Constraints

- Kinematic models of a car
  - Extended Car w. linear integrators
  - https://www.youtube.com/watch?v=3Q31mA5Aj-c

Structure

- Recap
- State, Kinematics, and Dynamics Models
  - State Space
  - Kinematic constraint models of the vehicle
  - Dynamic constraint models of the vehicle
- Planning
- AutonoVi-Sim

Dynamic Constraints

- “the branch of mechanics concerned with the motion of bodies under the action of forces.”
- Tires subject to lateral and longitudinal force during steering / accelerating
  - If lateral force exceeds friction force
    - Fishtailing
  - If longitudinal force exceeds friction force
    - Peel out / skid

Dynamic Constraints

- No longer directly control acceleration and steering
  - Apply engine force
  - Apply steering force
- Diminishing returns on each force at limits of control

Dynamic Constraints

- Dynamic Bicycle model with linear tires
  - No load transfer between tires
  - Larger state space including tire stiffness
- F_x longitudinal force
- F_y lateral force
- m mass
- I_z yaw moment of intertia

$$
\begin{aligned}
F_{xf}\cos\delta - F_{yf}\sin\delta + F_{xr} &= m(\dot{v}_x - v_y\dot{\psi}) \\
F_{xf}\sin\delta + F_{yf}\cos\delta + F_{yr} &= m(\dot{v}_y + v_x\dot{\psi}) \\
(F_{xf}\sin\delta + F_{yf}\cos\delta)b - F_{yr}c &= I_z\ddot{\psi}
\end{aligned}
$$

$$F_y = C_\alpha \alpha$$

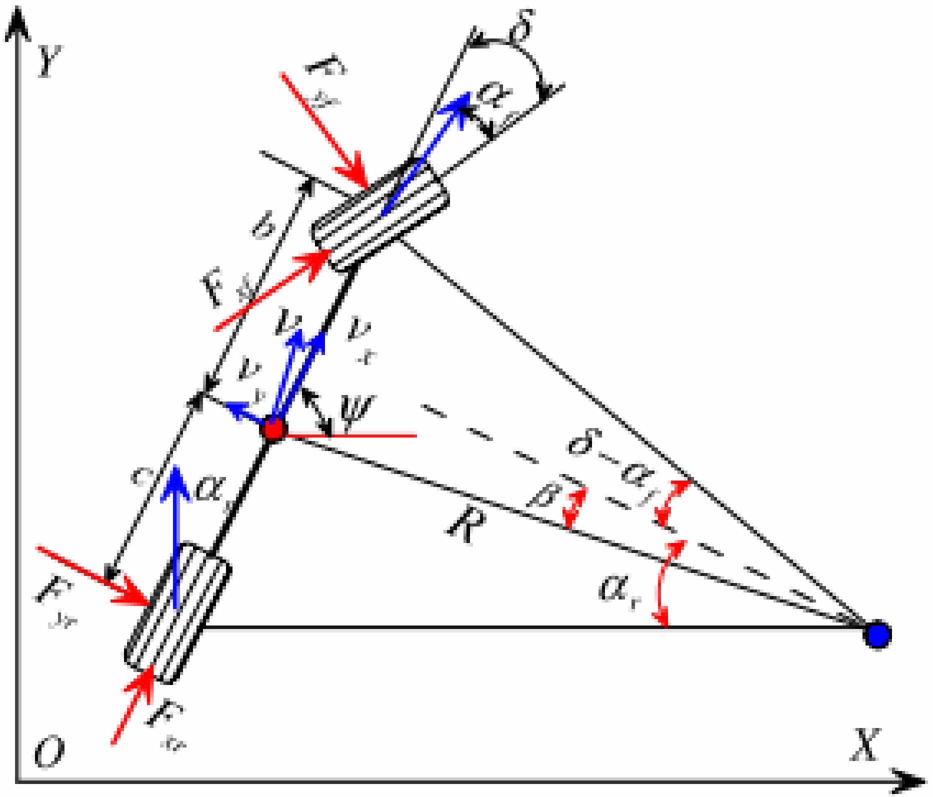
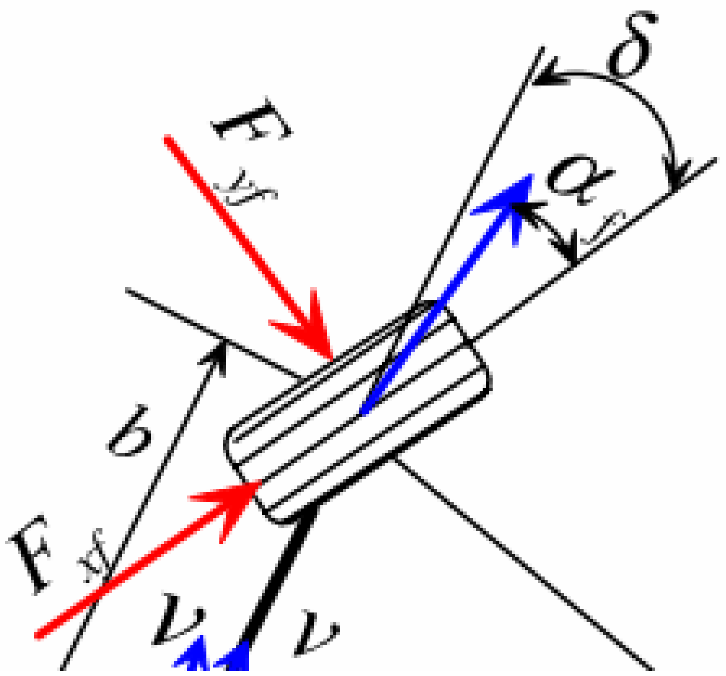


Fig. 1. Bicycle model of vehicle

Dynamic Constraints

- Dynamic Bicycle model with linear tires
  - F_y lateral force on tire
  - F_x longitudinal force on tire
  - α_f "slip angle" of tire
  - δ steering angle



Dynamic Constraints

- Dynamic constraints
  - Correcting for slip
  - https://www.youtube.com/watch?v=itggGQu_ECc

Dynamic Constraints

- Models increase in complexity as needed for performance tuning
  - Aerodynamic drag force $F_{wind} = (C_w A_w v_t^2 g)/16$
  - Maximum engine torque $\frac{F_{max}}{m} = 1 + \frac{3}{1 + e^{\left(\frac{v_t - 12}{4}\right)}}$
- Each layer of dynamics:
  - Increases accuracy of model
  - Increases computational complexity

Dynamic Constraints

- Dynamic constraints
  - Adjusting for drag & lateral forces
  - https://youtu.be/tesD4F-HOxs?t=1m24s

Dynamic Constraints

- Extended vehicle with load transfer

$$
\begin{aligned}
m\ddot{x} &= F_{x_{fl}} + F_{x_{fr}} + F_{x_{rl}} + F_{x_{rr}} - k_d \dot{x}^2 \\
m\ddot{y} &= -m\dot{x}\dot{\psi} + F_{y_{fl}} + F_{y_{fr}} + F_{y_{rl}} + F_{y_{rr}} \\
I\ddot{\psi} &= a(F_{y_{fl}} + F_{y_{fr}}) - b(F_{y_{rl}} + F_{y_{rr}}),
\end{aligned}
$$

$$
\begin{aligned}
F_{z_{fl}} &= \frac{bF_z - eF_x}{2(a+b)} - \frac{eF_y}{2c}, \quad & F_{z_{fr}} &= \frac{bF_z - eF_x}{2(a+b)} + \frac{eF_y}{2c}, \\
F_{z_{rl}} &= \frac{aF_z + eF_x}{2(a+b)} - \frac{eF_y}{2c}, \quad & F_{z_{rr}} &= \frac{aF_z + eF_x}{2(a+b)} + \frac{eF_y}{2c}.
\end{aligned}
$$

$$
\alpha_f = \frac{\dot{y} + a\dot{\psi}}{\dot{x}} - \delta, \quad \alpha_r = \frac{\dot{y} - b\dot{\psi}}{\dot{x}}
$$

Structure

- Recap
- State, Kinematics, and Dynamics Models
- Planning
  - Mission Planner
  - Behavior Planner
  - Maneuver Planner / Motion Planner
- AutonoVi-Sim

Autonomous Driving: Main Components

- Planning
  - Making purposeful decisions in order to achieve the robot's higher order goals

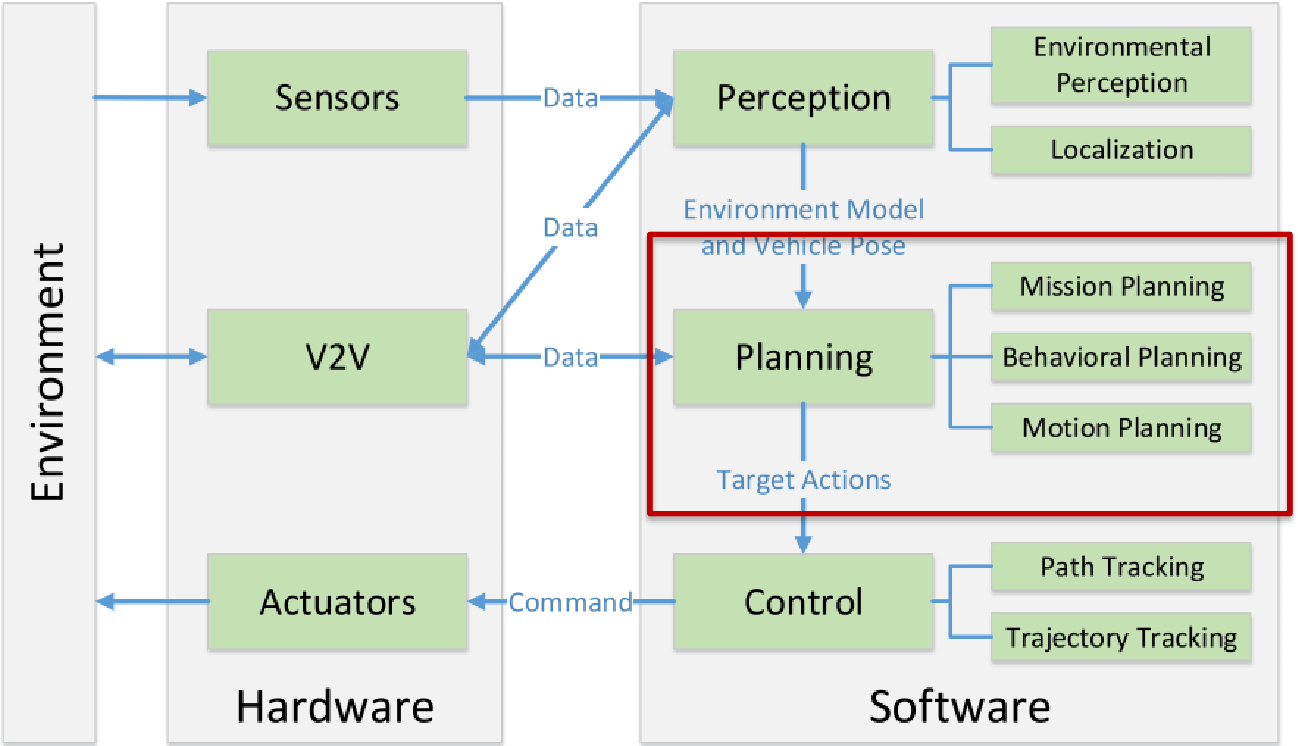


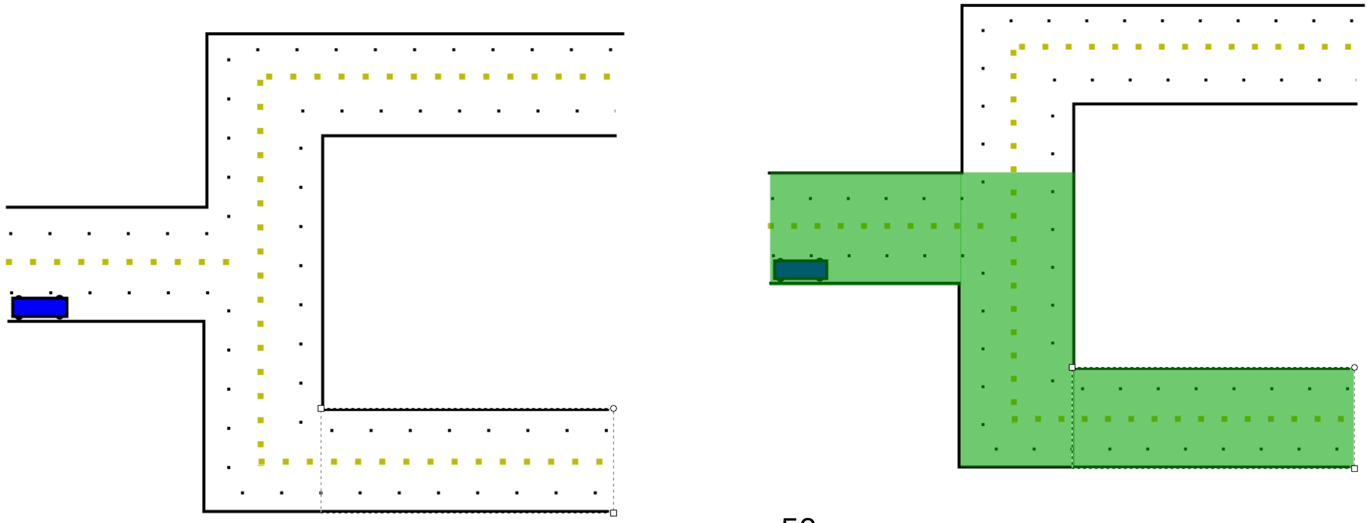
Figure 2. A typical autonomous vehicle system overview, highlighting core competencies.

Structure

- Recap
- State, Kinematics, and Dynamics Models
- Planning
  - Mission Planner
  - Behavior Planner
  - Maneuver Planner / Motion Planner
- AutonoVi-Sim

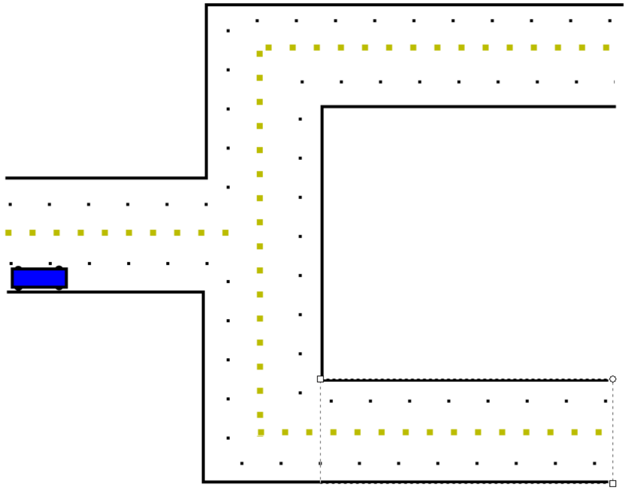
Mission Planner (Route Planning)

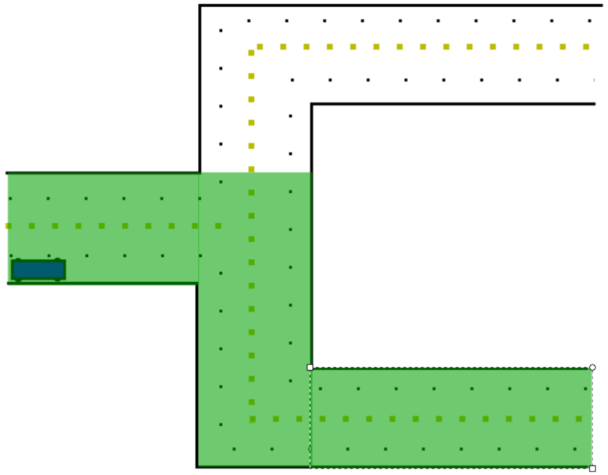
- Determine the appropriate macro-level route to take
- Typically road level i.e. which roads to take
- Katrakazas: “Route planning is concerned with finding the best global route from a given origin to a destination, supplemented occasionally with real-time traffic information”



Mission Planner (Route Planning)

- Pendleton: "considers high level objectives, such as assignment of pickup/dropoff tasks and which roads should be taken to achieve the task"
- Typical approaches:
  - RNG (Road-network Graph)
    - A*
    - Dijkstras
  - Scale poorly!



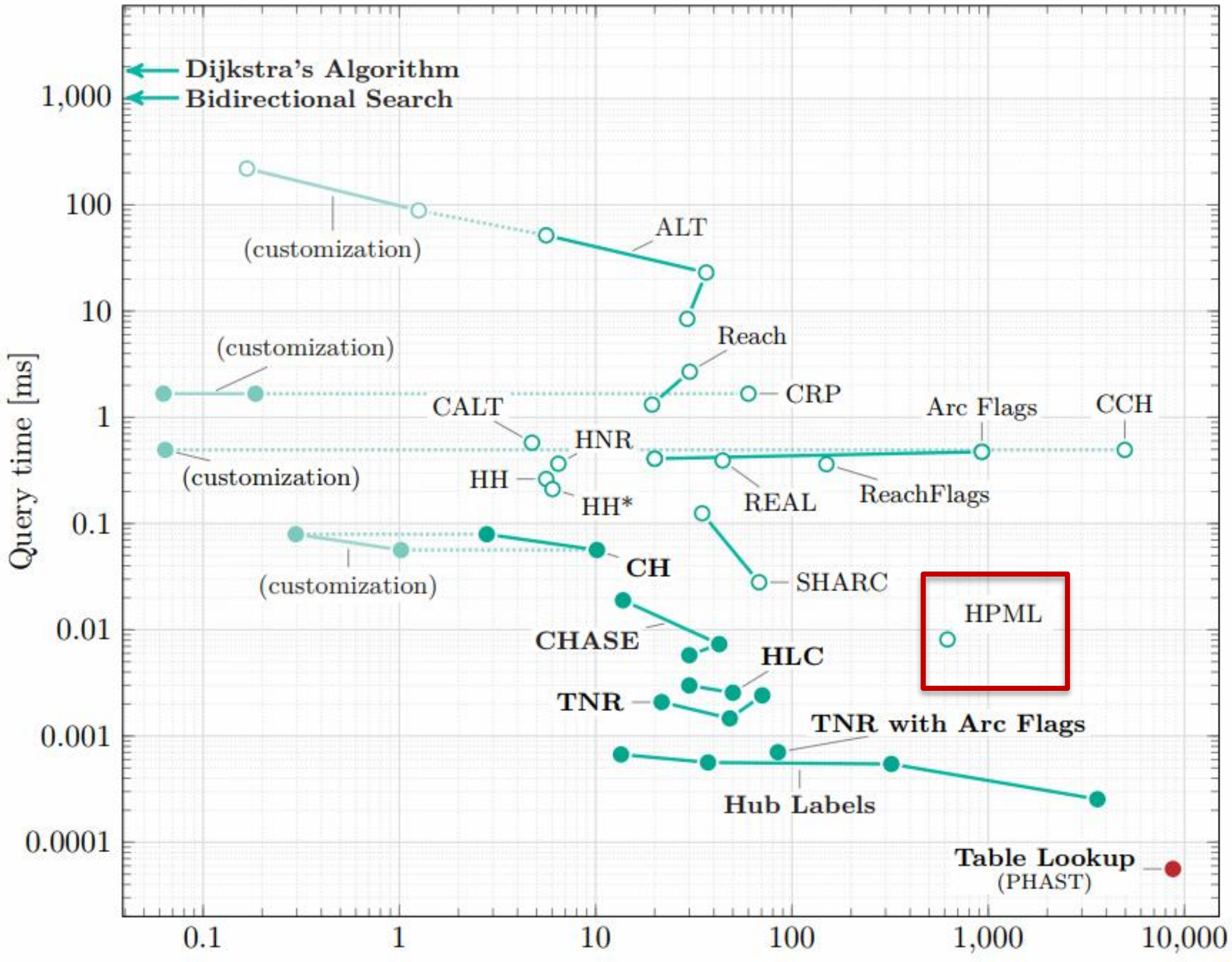


THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Mission Planner (Route Planning)

- Massive-scale algorithms needed for routing

- 18 million vertices, 42.5 million edges
  - Partial Western Europe dataset



Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., … Werneck, R. F. (2015). Route Planning in Transportation Networks. *Microsoft Research Technical Report*, 1–65.

Mission Planner

- High Performance Multi-Level (Delling et al.)
  - Hierarchical decomposition of input graph
  - Compute large set of partial graphs
  - Optimize subgraphs
    - Remove “unused” edges
    - Reorder graph to prioritize shortest paths

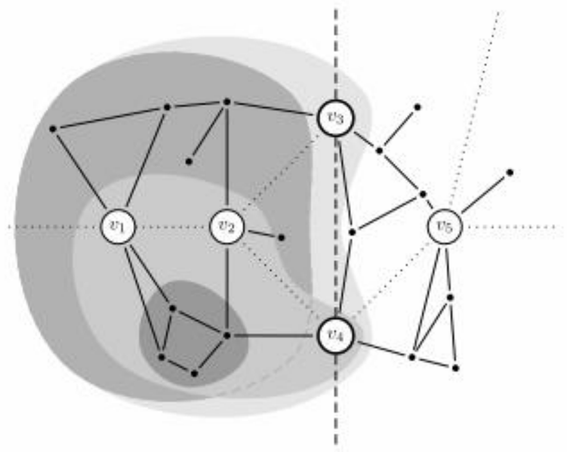


FIGURE 1. Hierarchy due to graph decomposition: components (darker shades) with belonging wrapped components (lighter shades) at levels 1 (smaller components) and 2 (larger components).

Mission Planner

- HPML (Delling et al.)
  - Optimize subgraphs
    - Remove “unused” edges
    - Reorder graph to prioritize shortest paths
  - Queries ~40μs on 18 million vertices

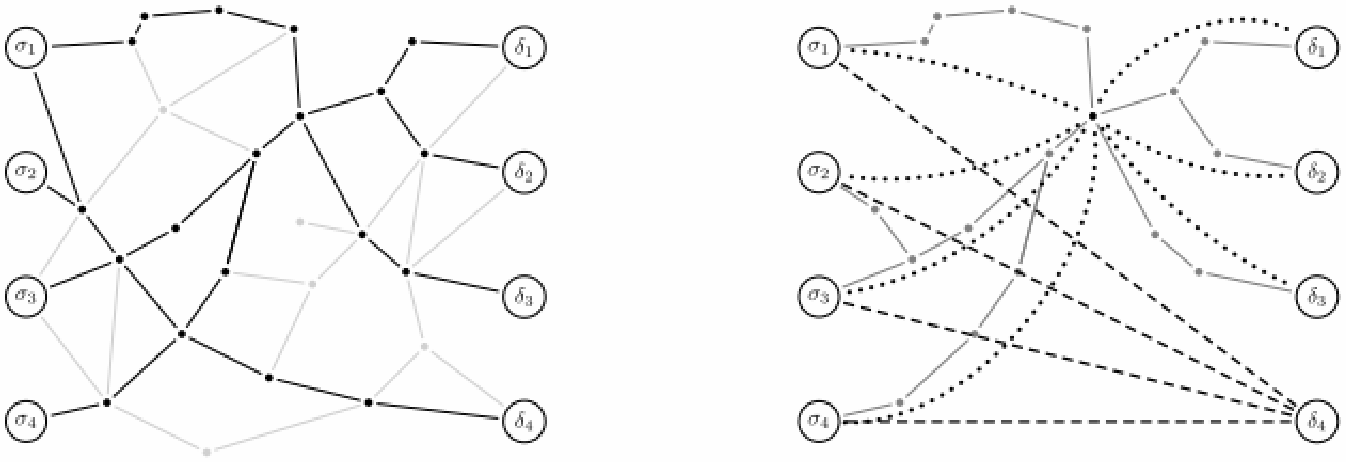
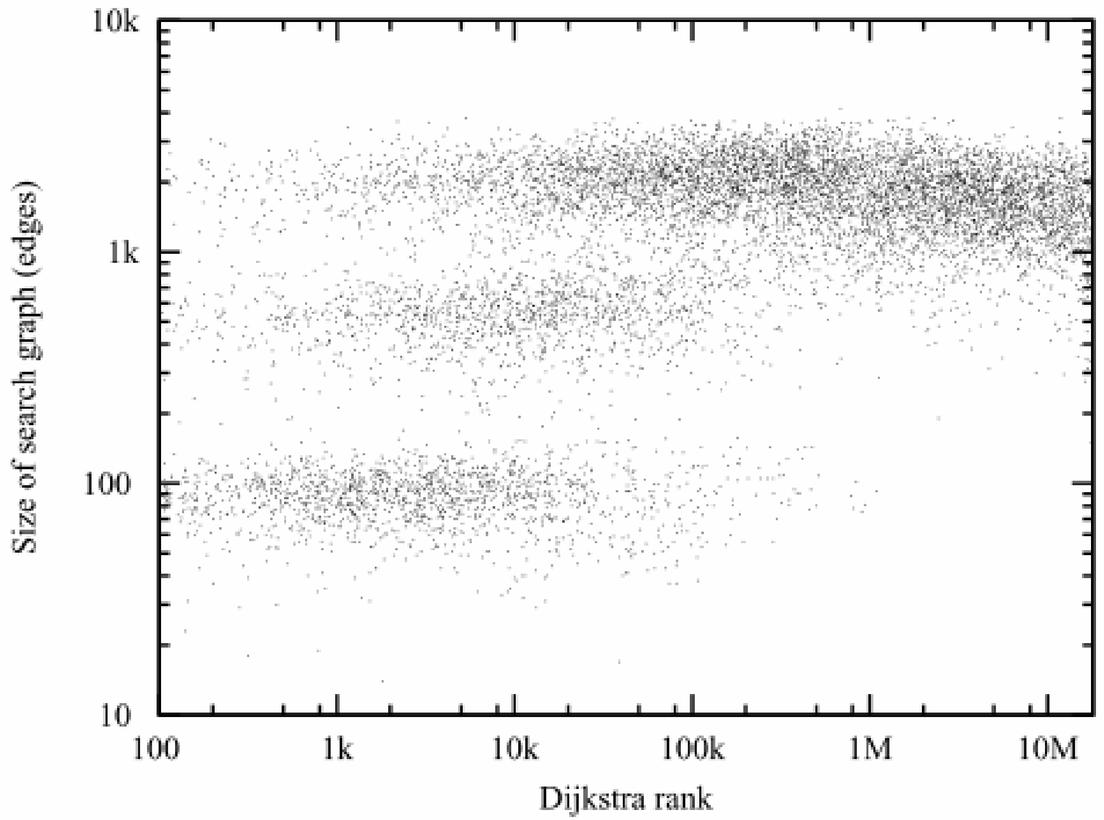


FIGURE 5. Constructing equivalent graphs. Left: sample graph; highlighted edges are contained in a shortest σ-δ path. Right: belonging search graph with edge compression applied; dotted and dashed edges are contained in the graph.

Delling, D., Holzer, M., Kirill, M., Schulz, F., & Wagner, D. (2008). High-Performance Multi-Level Routing, *2*, 1–19.

Mission Planner

- HPML (Delling et al.)



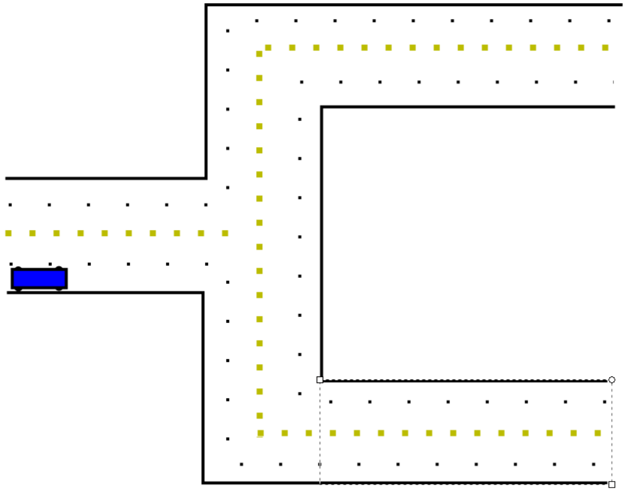
(a) Search space in terms of relaxed edges. Each dot depicts for one query the number of relaxed edges in relation to its Dijkstra rank.

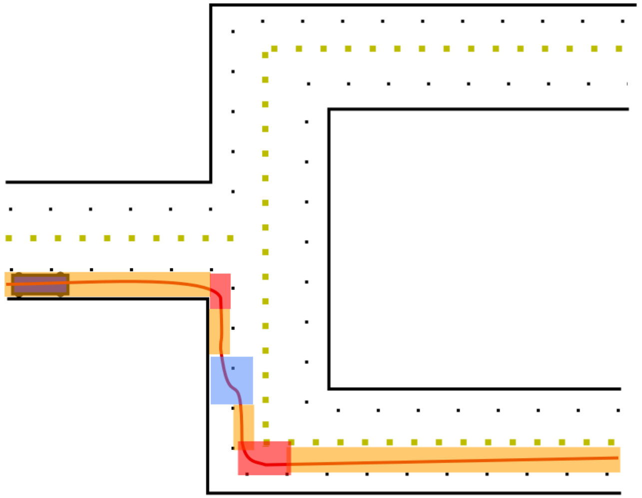
Structure

- Recap
- State, Kinematics, and Dynamics Models
- Planning
  - Mission Planner
  - Behavior Planner
  - Maneuver Planner / Motion Planner
- AutonoVi-Sim

Behavior Planner

- "makes ad hoc decisions to properly interact with other agents and follow rules restrictions, and thereby generates local objectives, e.g., change lanes, overtake, or proceed through an intersection"
  - Finite State Machines
  - Finite time maneuvers





Behavior Planner

- Finite State Machines
  - Set of "states" and transition functions between them
  - Separate from configuration state

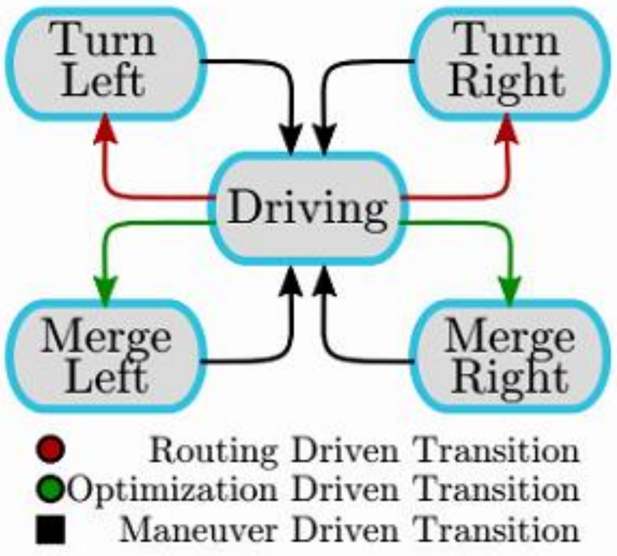
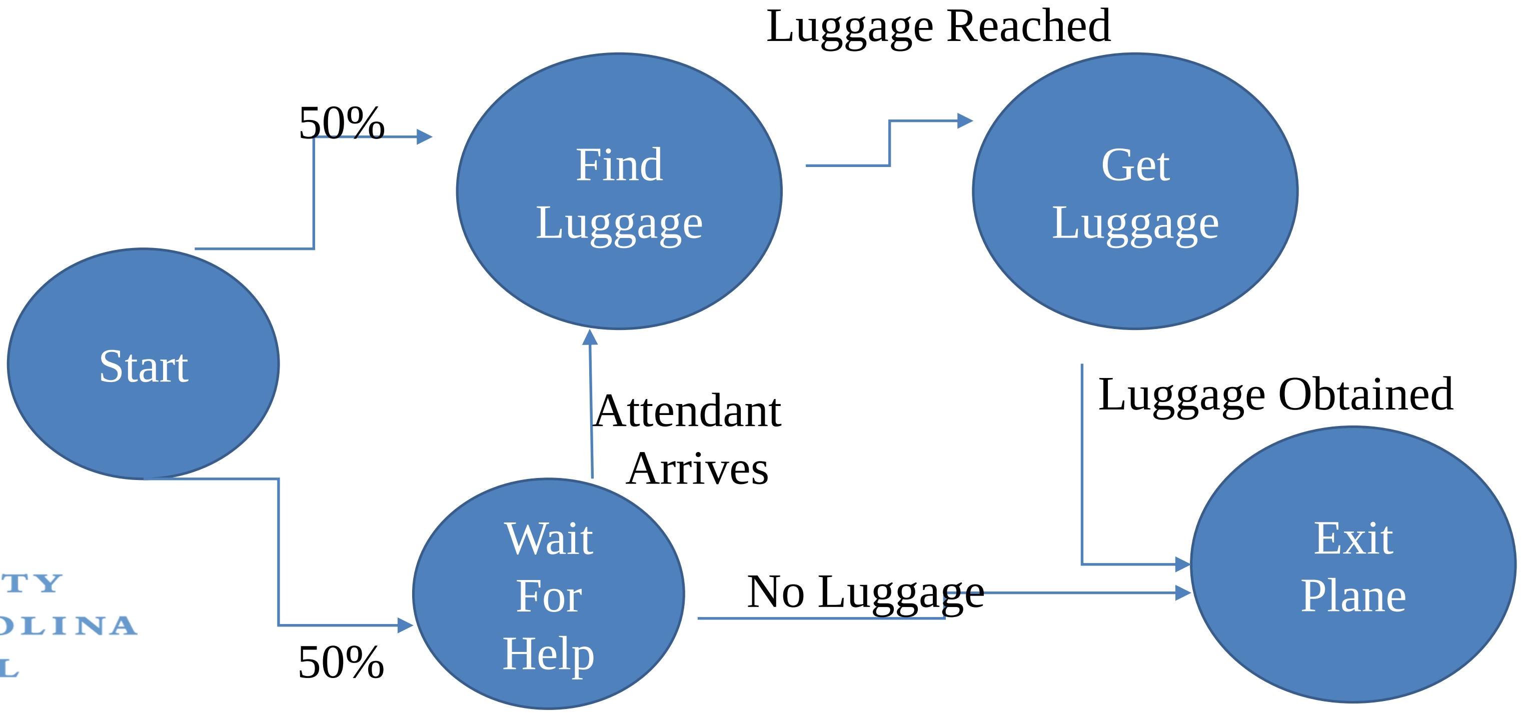


Fig. 2. **Finite State Machine:** We highlight different behavior states that are determined by the routing and optimization algorithms. When executing turns, the routing algorithm transitions the behavior state to a turning state. When the optimization-based maneuver algorithm plans a lane change, the behavior state is transitioned to merging.

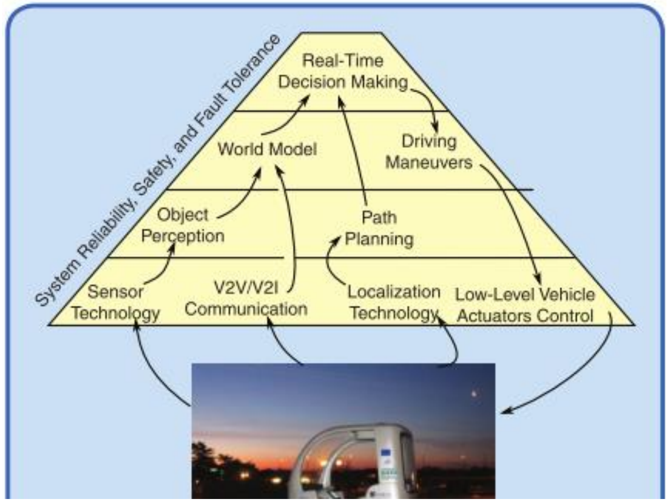
Behavior Planner

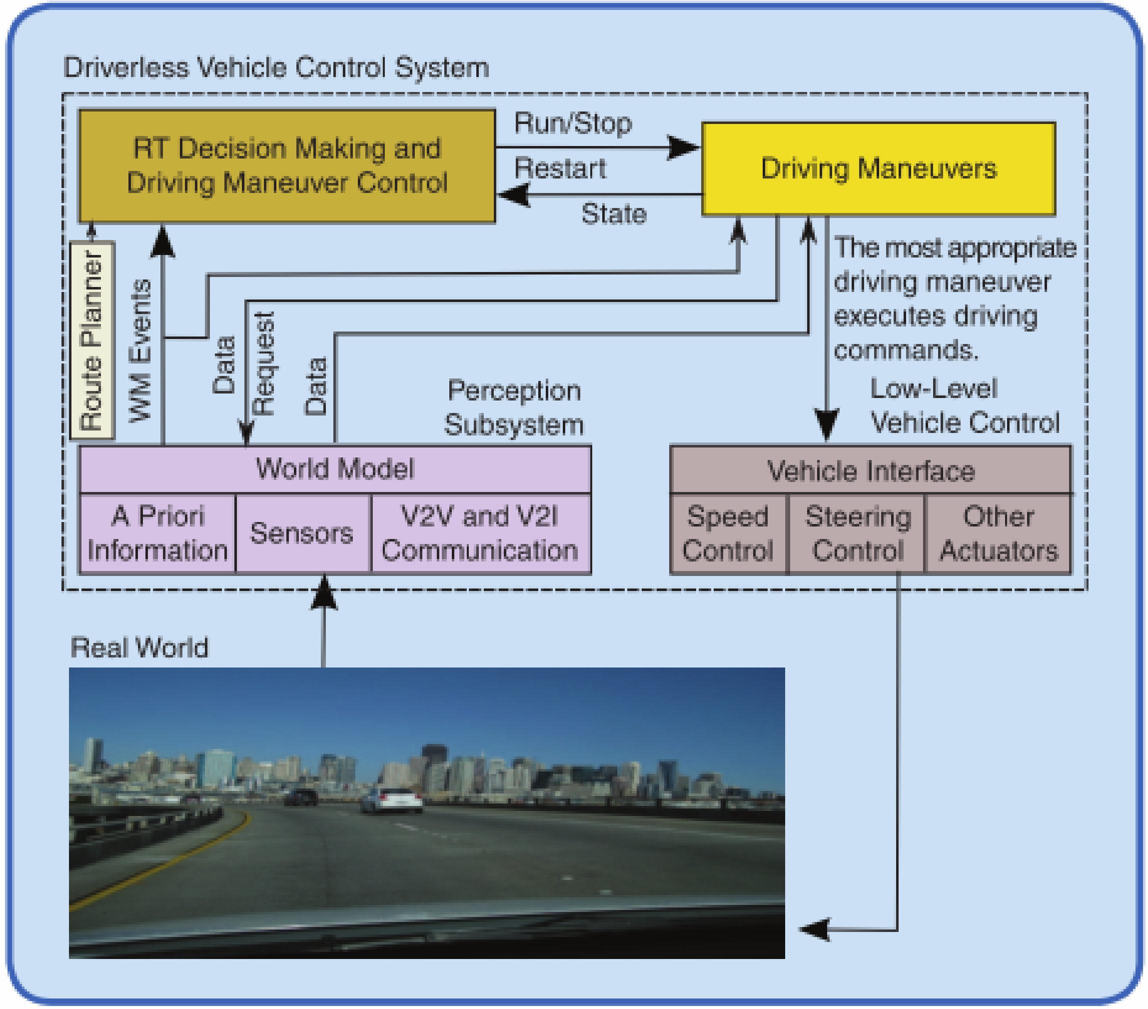
- Example from crowd sim
- AI Technique
  - Defines a set of States and Transition functions between them
  - Allows us to represent complex behaviors with simple components



Behavior Planner

- FSMs limited in some cases
  - What to do in unseen situations?
- Real-time decision making [Furda et al 2011]





Furda, A., & Vlacic, L. (2011). Enabling safe autonomous driving in real-world city traffic using Multiple Criteria decision making. *IEEE Intelligent Transportation Systems Magazine*, *3*(1), 4–17. http://doi.org/10.1109/MITS.2011.940472

Behavior Planner

- Limited discrete maneuver curve example
  - https://youtu.be/5ATo6hheV9U

Structure

- Recap
- State, Kinematics, and Dynamics Models
- Planning
  - ⑩ Mission Planner
  - ⑩ Behavior Planner
  - ⑩ Maneuver Planner / Motion Planner
- AutonoVi-Sim

Maneuver Planner / Motion Planner

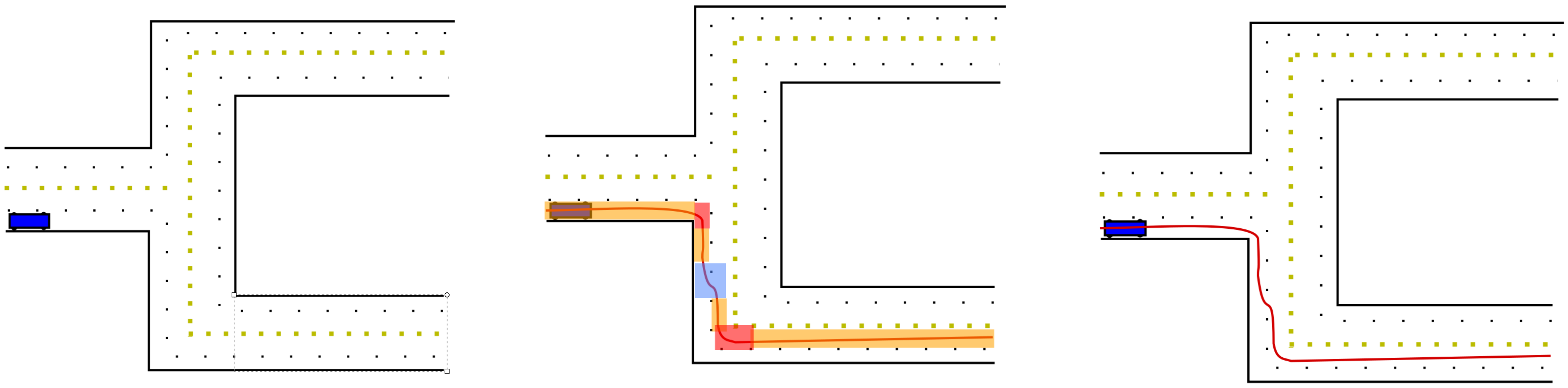
- Pendleton: generates appropriate paths and/or sets of actions to achieve local objectives, with the most typical objective being to reach a goal region while avoiding obstacle collision

Motion Planner

- Generally two stages:
  - Path planner - Computes the geometric representation of the path to be followed. I.e. the curve, spline, track, line, etc. we are following
  - Trajectory Planner / Path tracker - Computes the specific physical targets for following the path. I.e. velocity, acceleration, heading, steering, etc.

Motion Planner

- Pendleton: generates appropriate paths and/or sets of actions to achieve local objectives, with the most typical objective being to reach a goal region while avoiding obstacle collision



Motion Planner

- How do we evaluate them?
  - ⑩ Complexity (computation cost)
    - limits how frequently we can replan
    - NEVER get it perfectly right, so we focus on replanning as fast as possible

  - ⑩ Completeness (likelihood that a solution will be found if one exists)

    The piano-movers problem is PSPACE-HARD

    must guarantee safety

    i.e. must be sure we can deal with error and recover

Motion Planner

- Piano mover's problem
  - https://youtu.be/cXm3WW-geD8

Motion Planner

- Basic overview
  - Complete planning
  - Combinatorial Planning
  - Sample-Based planning

Motion Planner

- Basic overview
  - Complete planning - continuous plan in configuration space
    - Exponential in dimensions of c-space (curse of dimensionality)
    - "Complete"
  - Combinatorial Planning - discrete planning over an exact decomposition of the configuration space
  - Sample-Based planning:

Motion Planner

- Basic overview
  - Complete planning
  - Combinatorial Planning - discrete planning over an exact decomposition of the configuration space
    - Exponential in dimensions of c-space discretization (curse of dimensionality)
    - "resolution complete"
  - Sample-Based planning

Motion Planner

- Basic overview
  - Complete planning
  - Combinatorial Planning
  - Sample-Based planning - Sample in space to find controls / positions which are collision free and linked
    - Probabilistically complete
      - Some "probabilistically optimal"
    - NOT exponential in configuration space

Motion Planner: Combinatorial Planners

- General Approaches:
  - convex obstacle spaces
    - NP-Hard
  - visibility graph (shortest path)
  - voronoi diagram (highest clearance)
  - obstacle-cells using boundaries and borders

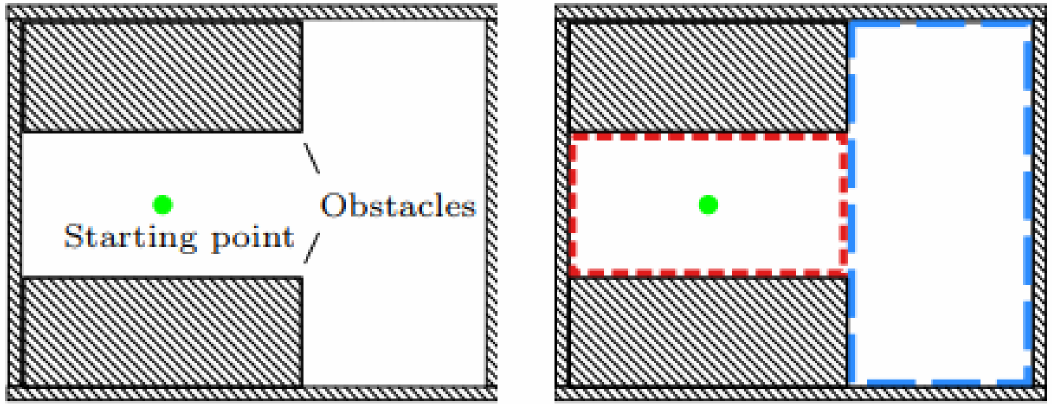


Fig. 1.

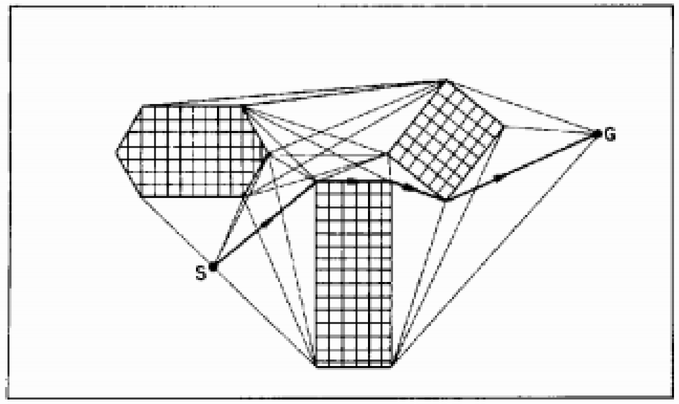
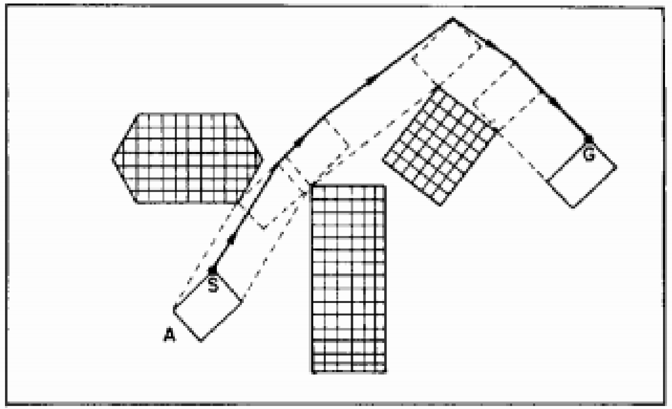


Fig. 3(a).

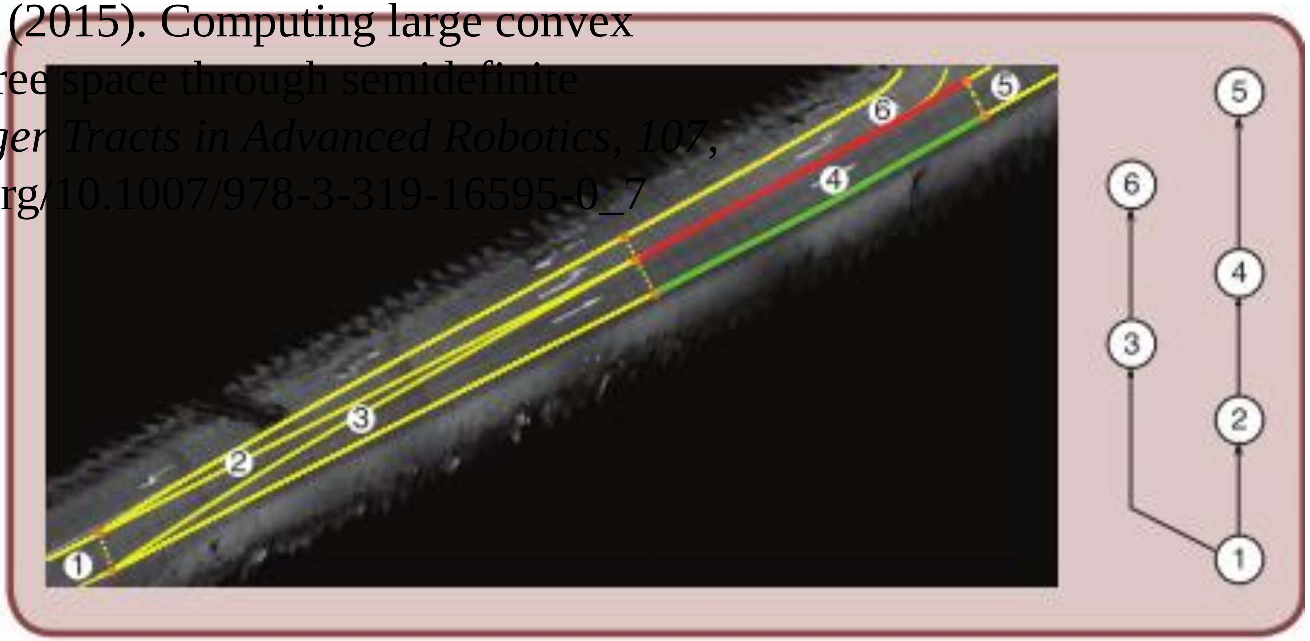


Deits, R., & Tedrake, R. (2015). Computing large convex regions of obstacle-free space through semidefinite programming. *Springer Tracts in Advanced Robotics*, *107*, 109–124. http://doi.org/10.1007/978-3-319-16595-0_7

Motion Planner: Combinatorial Planners

- Driving Corridors:
  - Decompose lanes into polygonal lanelets
  - Represent obstacles as polygonal bounding boxes or overlapping discs
  - Adjust lanelets to obstacle constraints

Deits, R., & Tedrake, R. (2015). Computing large convex regions of obstacle-free space through semidefinite programming. *Springer Tracts in Advanced Robotics*, *107*, 109–124. http://doi.org/10.1007/978-3-319-16595-0_7



Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., … Zeeb, E. (2014). Making bertha drive-an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, *6*(2), 8–20. http://doi.org/10.1109/MITS.2014.2306552

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Motion Planner: Combinatorial Planners

- Driving Corridors:
  - Decompose lanes into polygonal lanelets
  - Represent obstacles as polygonal bounding boxes or overlapping discs
  - Adjust lanelets to obstacle constraints

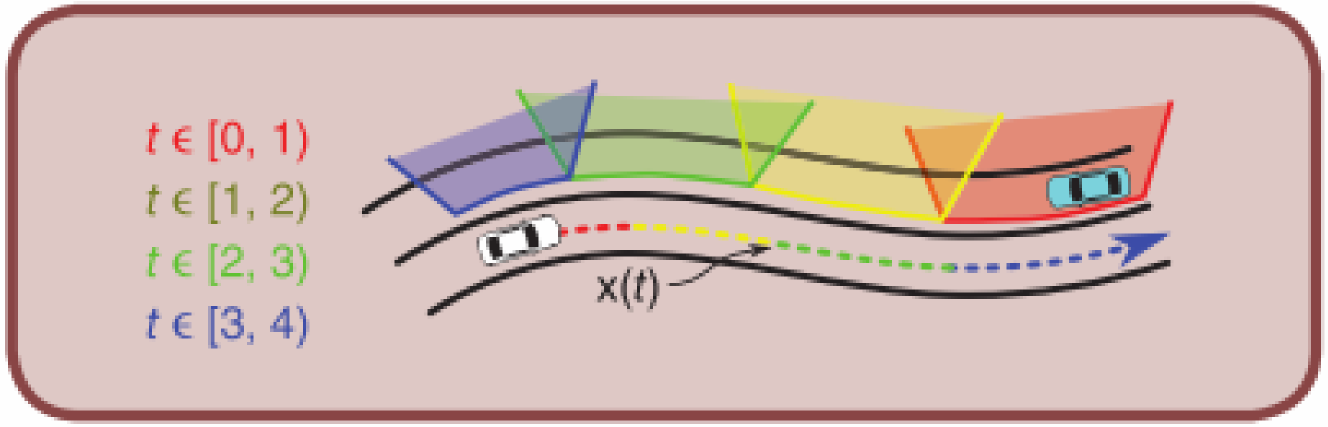


FIG 10 Constraints for an oncoming Object (cyan). The trajectory is only constrained by polygons of corresponding color.

Motion Planner: Combinatorial Planners

- Driving Corridors:
  - Decompose lanes into polygonal lanelets
  - Represent obstacles as polygonal bounding boxes or overlapping discs
  - Adjust lanelets to obstacle constraints

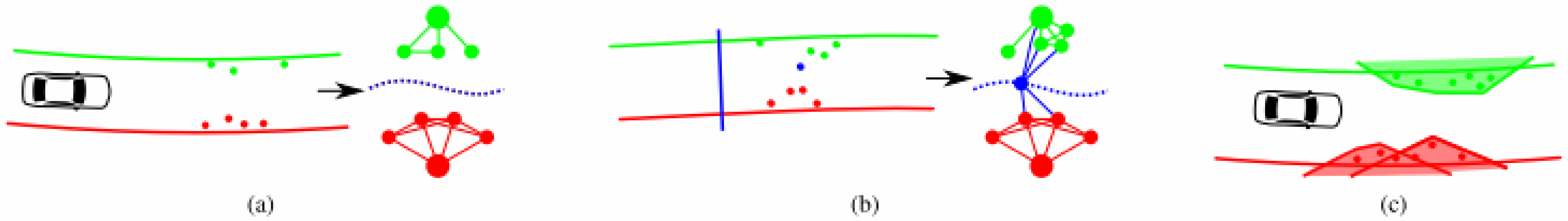


Figure 5: Building constraint polygons from sensor data.

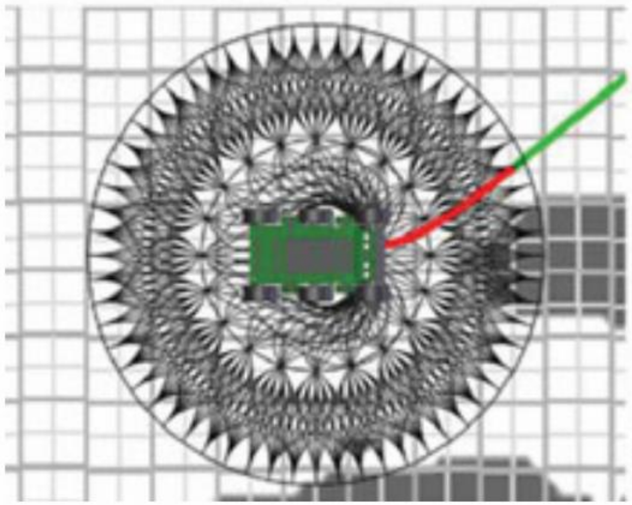
THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Motion Planner: Combinatorial Planners

- Driving Corridors:
  - https://youtu.be/GfXg9ux4xUw?t=2m5s

Motion Planner: Combinatorial Planners

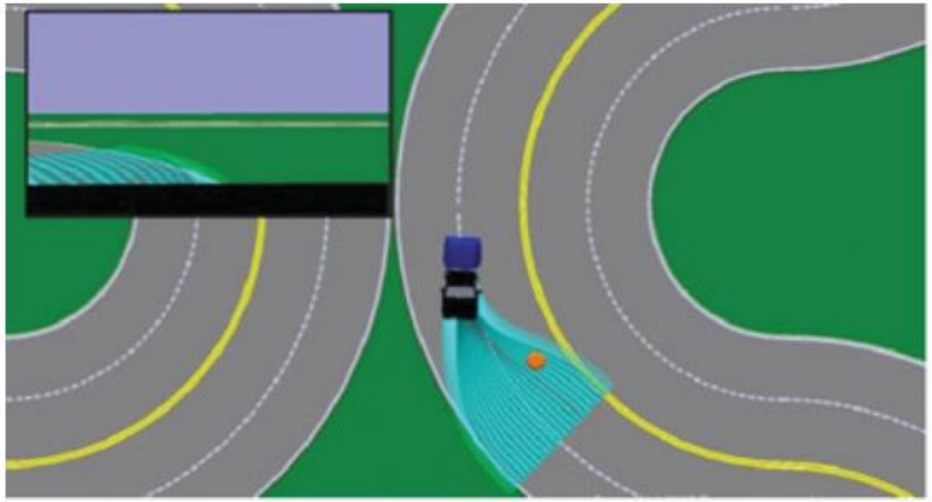
- Darpa Urban Challenge:
  - BOSS: kinodynamic reachable set
  - Trajectory planner generates candidate trajectories and goals
    - Done by precomputation of many curves
  - “best” trajectory chosen by optimization

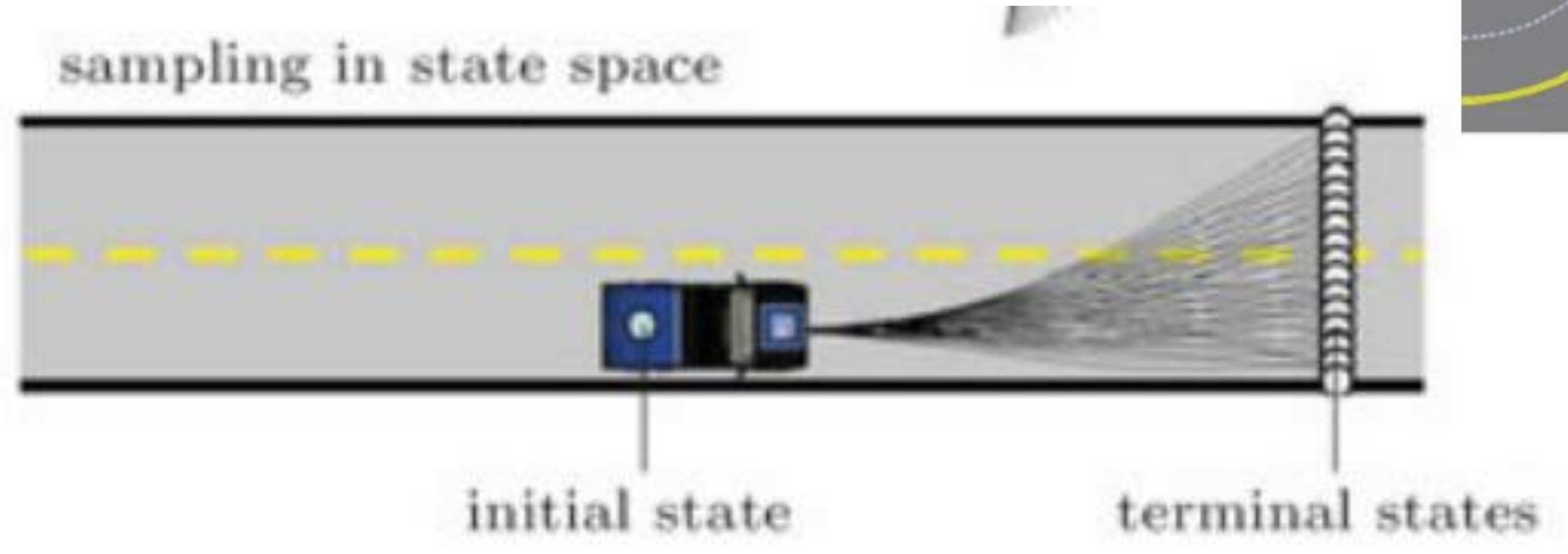


Urmson, C., Baker, C., Dolan, J., Rybski, P., Salesky, B., Whittaker, W., … Darms, M. (2009). Autonomous Driving in Traffic: Boss and the Urban Challenge. *AI Magazine*, *30*(2), 17–28. http://doi.org/10.1002/rob

Motion Planner: Combinatorial Planners

- Darpa Urban Challenge:
  - ⑩ BOSS: kinodynamic reachable set





Motion Planner: Combinatorial Planners

- Darpa Urban Challenge:
  - BOSS: kinodynamic reachable set
  - https://www.youtube.com/watch?v=lULl63ERek0&t=89s

  - Other combinatorial approaches:
  - https://www.youtube.com/watch?v=3FNPSld6Lrg

Motion Planner: Combinatorial Planners

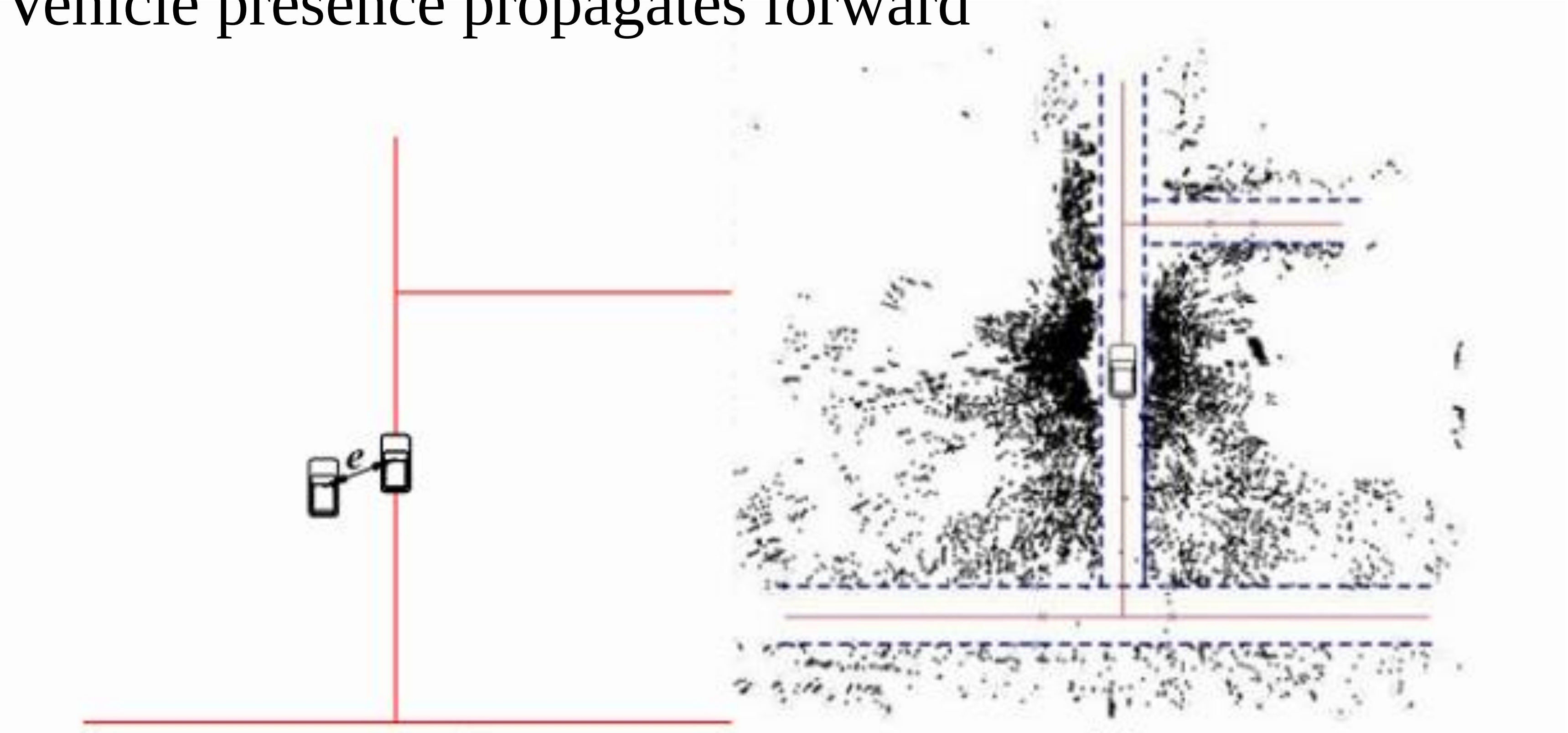
- Grid Decomposition approaches:
  - Generate cellular-grid representation of local space
  - Cells encode probability of occupancy
  - Moving obstacles propagate occupancy probability

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

Motion Planner: Combinatorial Planners

- Grid Decomposition approaches:
  - Vehicle presence propagates forward

Broggi, A., Medici, P., Zani, P., Coati, A., & Panciroli, M. (2012). Autonomous vehicles control in the VisLab Intercontinental Autonomous Challenge. *Annual Reviews in Control*, *36*(1), 161–171. http://doi.org/10.1016/j.arcontrol.2012.03.012



(a) (b)

Motion Planner: Combinatorial Planners

- Grid Decomposition approaches:
  - https://youtu.be/CRQfhhICSj0
  - https://youtu.be/MzpBzrtEGrA

Motion Planner: Combinatorial Planners

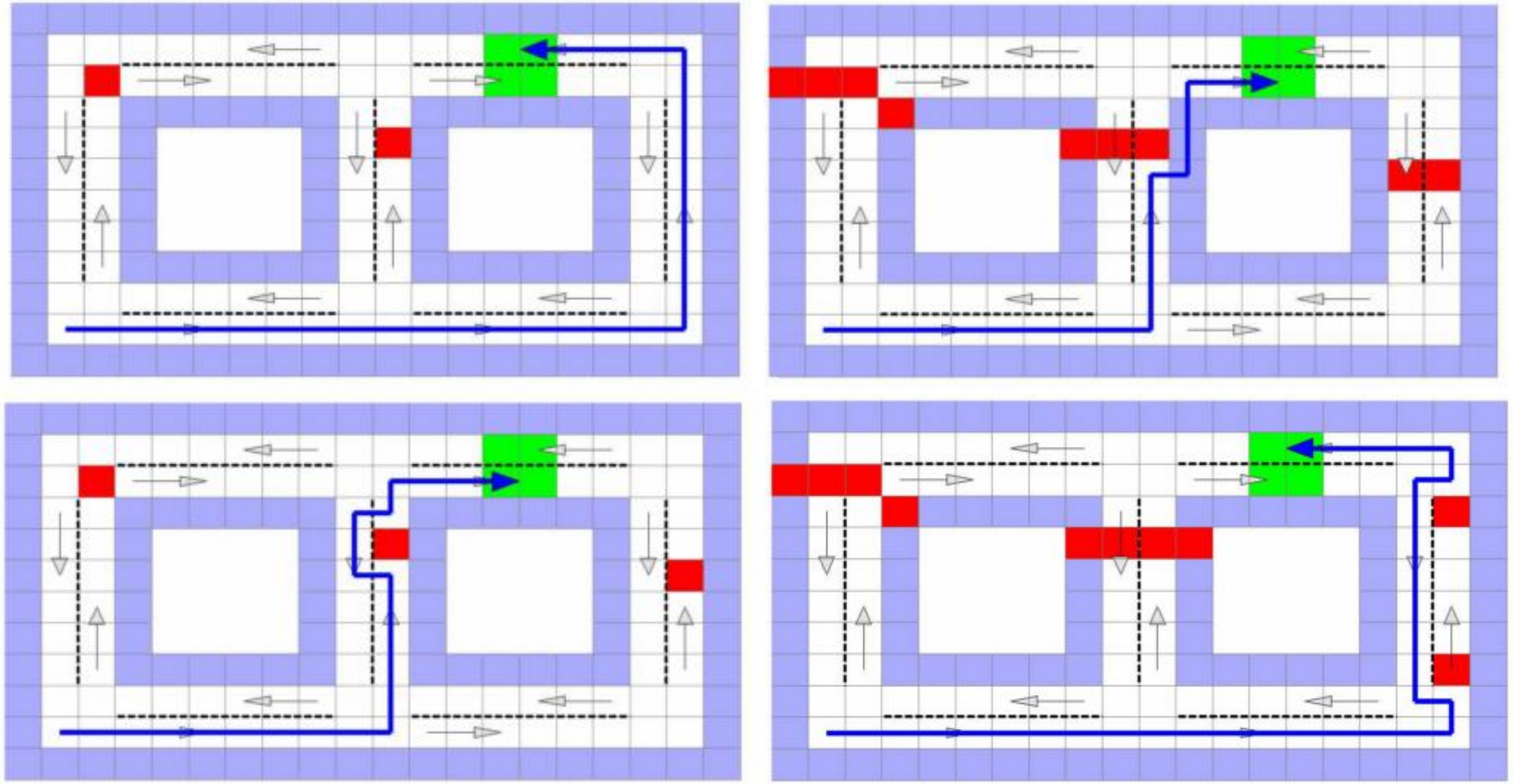
- Correct by construction planners:
  - Concept: Encode discrete rules and available actions
    - Rules assigned priority in Finite Linear Temporal Logic
    - Rules define "cost" penalty for violation
  - Generate plan over discrete action space guaranteeing least-violation of rules
    - Essentially least-violating state-space search

Motion Planner: Combinatorial Planners

- Correct by construction planners:
  - Example rules:
    - Do not collide with traffic
    - Never head in wrong direction
    - Do not drive on sidewalk
    - Go to the goal

Motion Planner: Combinatorial Planners

- Correct by construction planners:



- Green: Goal
- Red: Obstacle
- Lavendar: Sidewalk

Tumova, J., Hall, G. C., Karaman, S., Frazzoli, E., & Rus, D. (2013). Least-violating control strategy synthesis with safety rules. *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*, 1–10. http://doi.org/10.1145/2461328.2461330

Maneuver Planner: Sample-based Planners

- Pendleton: popular for their guarantees of probabilistic completeness, that is to say that given sufficient time to check an infinite number of samples, the probability that a solution will be found if it exists converges to one.
- General approaches:
  - PRM: Probabilistic Roadmaps
  - RRT: Rapidly-Exploring Random Tree
  - FMT: Fast-Marching Trees

Maneuver Planner: Sample-based Planners

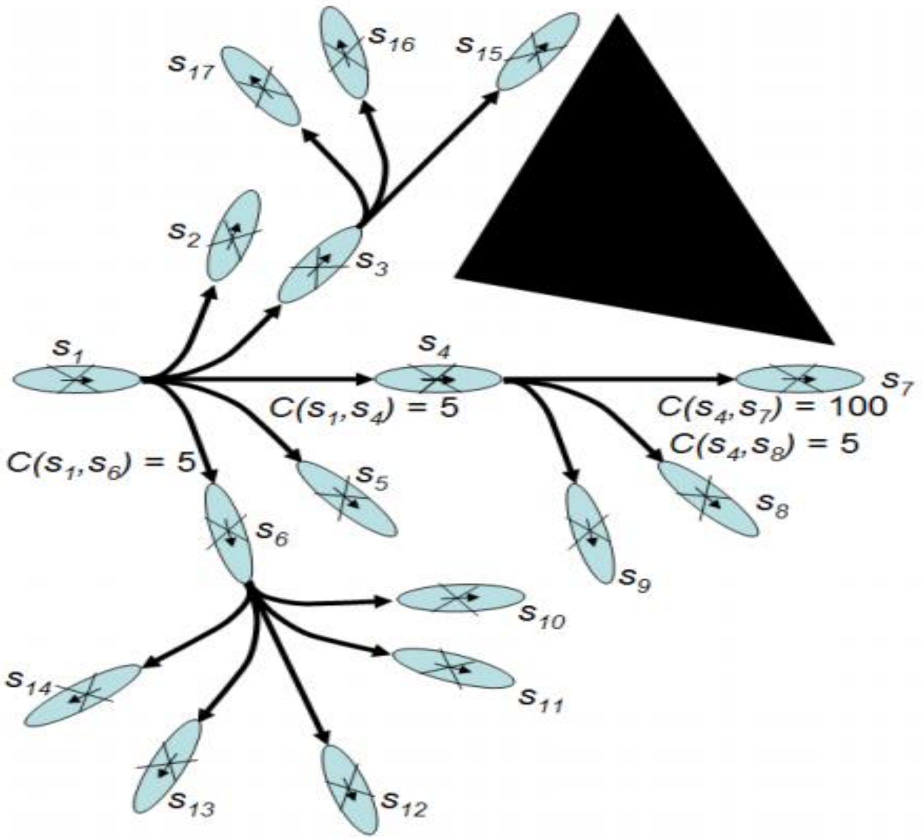
- Sample-based Planning specifically for cars:
  - Dynamics computation
  - Inevitable collision states
  - "Space-time planning approaches"
- Pendleton: "Incorporating differential constraints into state-sampling planners is still a challenging matter, and requires a steering function to draw an optimal path between two given states which obeys control constraints (if such a path exists), as well as efficient querying methods to tell whether a sampled state is reachable from a potential parent state"

Maneuver Planner: Sample-based Planners

- RRT:
  - Given at-least one initial configuration in free-space and a goal configuration
    - Sample a point p in configuration space, determine if it is collision free
    - If so, find nearest node n to the point, move some δ towards the point
    - If n to $n + \delta$ is CLEAR, connect to the tree

Maneuver Planner: Sample-based Planners

- RRT



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Maneuver Planner: Sample-based Planners

- RRT:
  - https://www.youtube.com/watch?v=rPgZyq15Z-Q
  - https://www.youtube.com/watch?v=mEAr2FBUJEI
  - https://www.youtube.com/watch?v=p3p0EWT5lpw

Maneuver Planner: Sample-based Planners

- PRM: Incorporating dynamics: Sampling directly from admissible controls
- [Hsu et al]
  - Extends existing PRM framework
  - State × time space formulation
  - state typically encodes both the configuration and the velocity of the robot

Hsu, D., Kindel, R., Latombe, J.-C., & Rock, S. (2002). Randomized Kinodynamic Motion Planning with Moving Obstacles. *The International Journal of Robotics Research, 21*(3), 233–255. http://doi.org/10.1177/027836402320556421

Maneuver Planner: Sample-based Planners

- Incorporating dynamics: Sampling directly from admissible controls
- [Hsu et al]
  - Represents kinodynamic constraints by a control system
  - set of differential equations describing all possible local motions of a robot
- Define set of piecewise constant control functions for finite time horizons

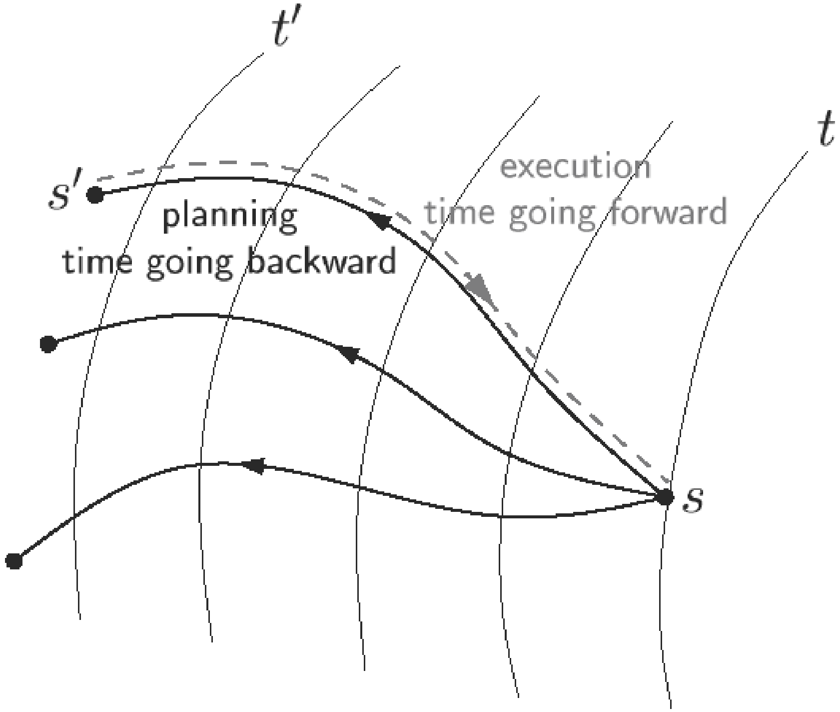
Maneuver Planner: Sample-based Planners

Algorithm 1 Control-driven randomized expansion.

Insert m_b into T; $i \leftarrow 1$.
repeat
    Pick a milestone m from T with probability $\pi_T(m)$.
    Pick a control function u from $\mathcal{U}_\ell$ uniformly at random.
    $m' \leftarrow$ PROPAGATE(m, u) .
    if $m' \neq nil$ **then**
        Add m' to T; $i \leftarrow i + 1$.
        Create an edge e from m to m'; store u with e.
        if $m' \in$ ENDGAME **then** exit with SUCCESS.
        if $i = N$ **then** exit with FAILURE.

Maneuver Planner: Sample-based Planners

- Check if m is in a ball of small radius centered at the goal.
  - Limitation: relative volume of the ball -> 0 as the dimensionality increases.
- Check whether a canonical control function generates a collision-free trajectory from m to (s_g, t_g)
- Build a secondary tree T' of milestones from the goal with motion constraints equation backwards in time.
- Endgame region is the union of the neighborhood of milestones in T'



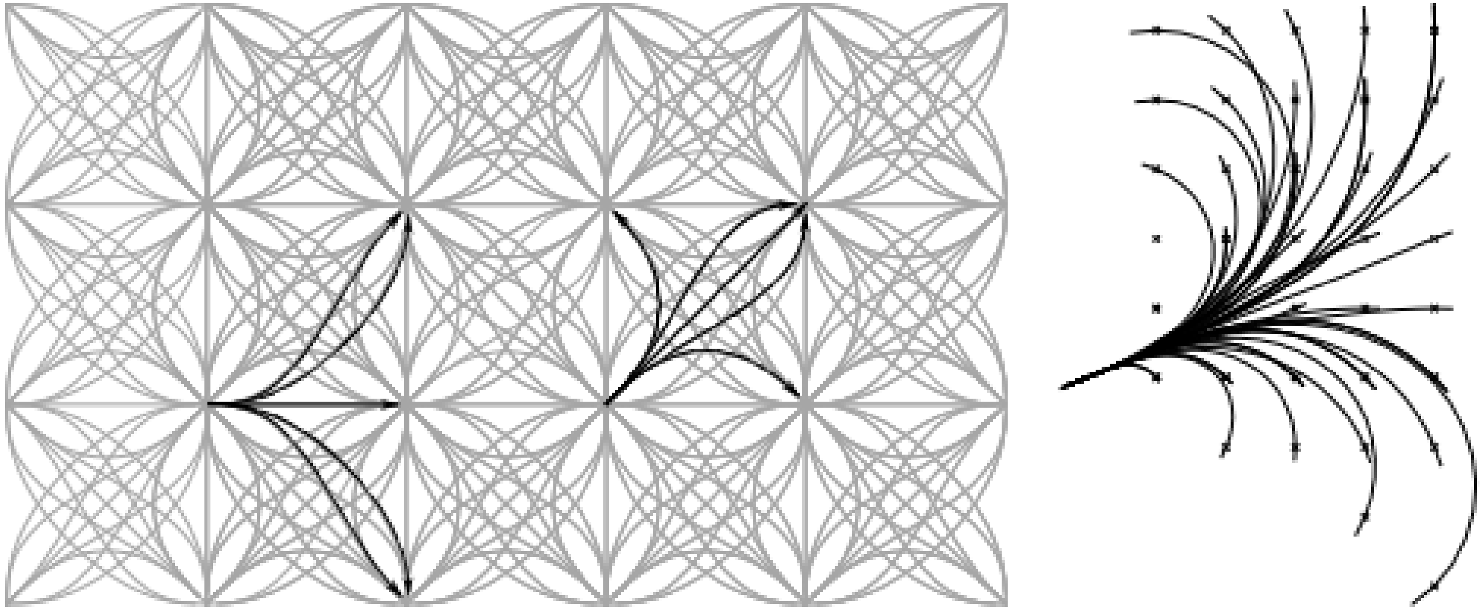
Maneuver Planner: Sample-based Planners

- State-lattice planners
  - Generate set of potential future states through solving boundary-value problem
  - Generate connected “lattice” of potential future states expanding in time and space

Maneuver Planner: Sample-based Planners

- State-lattice planners
  - Ex: Configurations in space

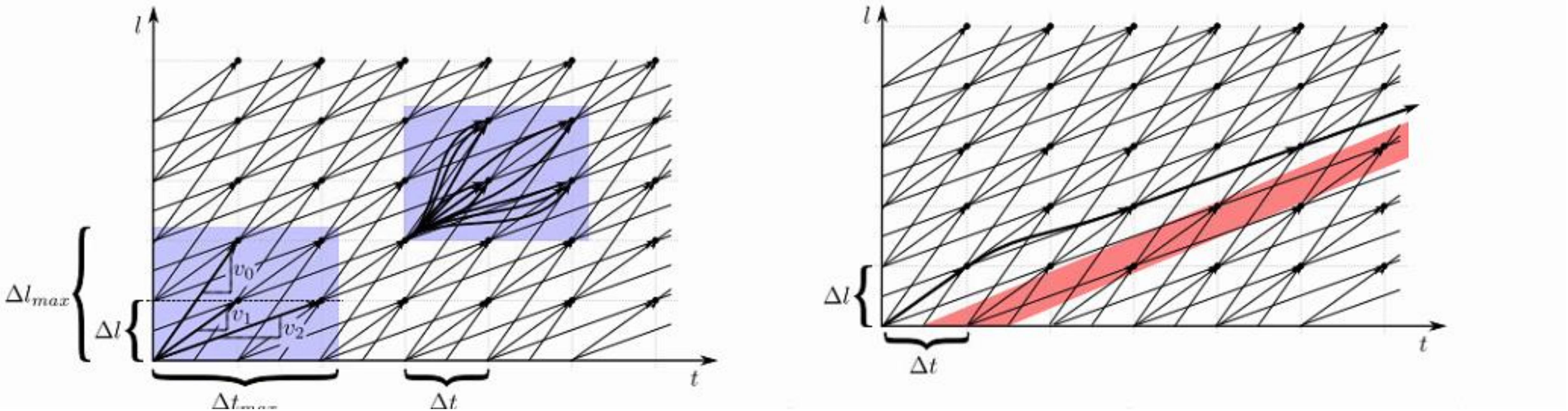
Ziegler, J., & Stiller, C. (2009). Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, 1879–1884. http://doi.org/10.1109/IROS.2009.5354448



(a) (b)

Maneuver Planner: Sample-based Planners

- State-lattice planners
  - 1D Example in "l," obstacle in red



Maneuver Planner: Sample-based Planners

- State-lattice planners
  - Transform road representation to longitudinal and lateral segments
  - Generate potential paths in parametrized space
  - Best path chosen by cost metric
    - Time, comfort, length

Fig. 5: State transitions on the transformed grid. The successors of one vertex are shown in black.

Maneuver Planner: Sample-based Planners

- State-lattice planners
  - https://www.youtube.com/watch?v=I5hL8vSo6DI
  - Notice the discrete maneuver points

Maneuver Planner: Sample-based Planners

Martinez-Gomez, L., & Fraichard, T. (2009). Collision avoidance in dynamic environments: An ICS-based solution and its comparative evaluation. *Proceedings - IEEE International Conference on Robotics and Automation*, 100–105. http://doi.org/10.1109/ROBOT.2009.5152536

- ICS-Avoidance
  - Theoretically define "inevitable collision states"
    - Set of collision-avoiding controls is null
  - Iterative check each candidate control s.t. subsequent controls are not ICS
  - Effective but very costly

$$\mathrm{ICS}(\mathcal{B}) = \{s \in \mathcal{S} \mid \forall \tilde{u} \in \tilde{\mathcal{U}}, \exists t, \mathcal{A}(\tilde{u}(s,t)) \cap \mathcal{B}(t) \neq \emptyset\}$$

Maneuver Planner: Sample-based Planners

- ICS-Avoidance
  - Area inside red region represents inevitable collisions
  - Different movements of B dramatically change $ICO(B, \phi)$

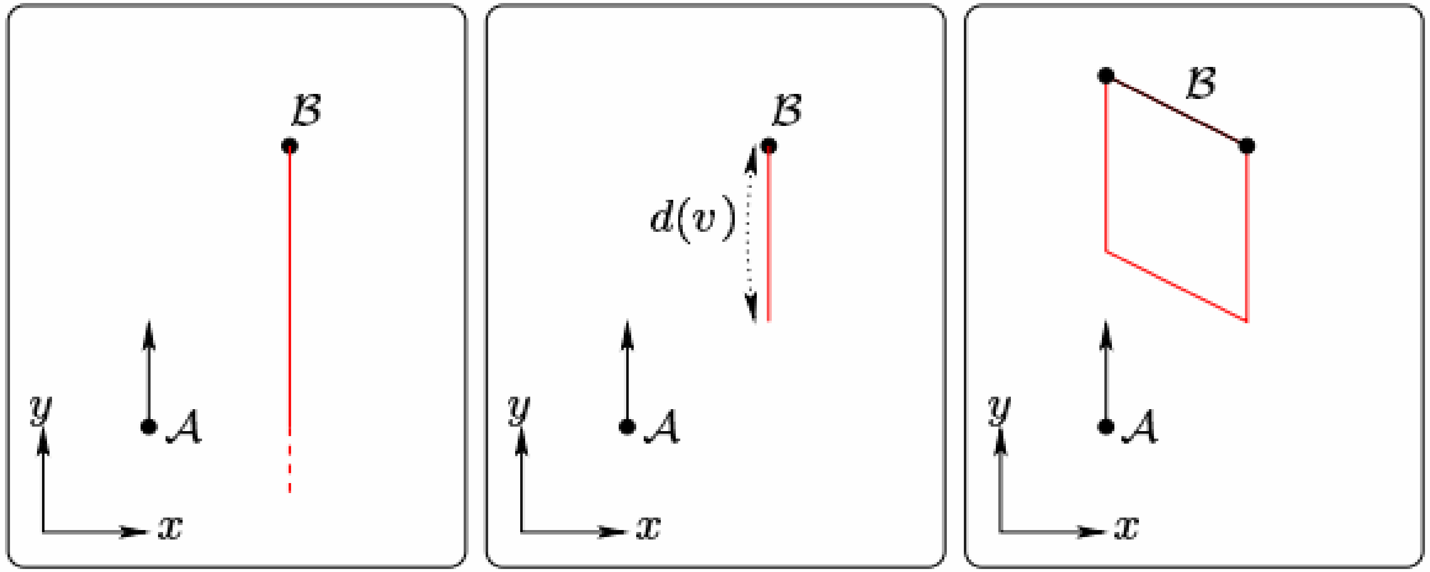


Fig. 10. $ICO(\mathcal{B}, \phi)$ for ϕ such that $\xi = 0$ ($\mathcal{A}$ is moving straight). $a = 0$ (left), a is changing (middle and right).

Maneuver Planner: Sample-based Planners

- ICS-Avoidance
  - Area inside red region represents inevitable collisions
  - Different movements of B dramatically change $ICO(B, \phi)$

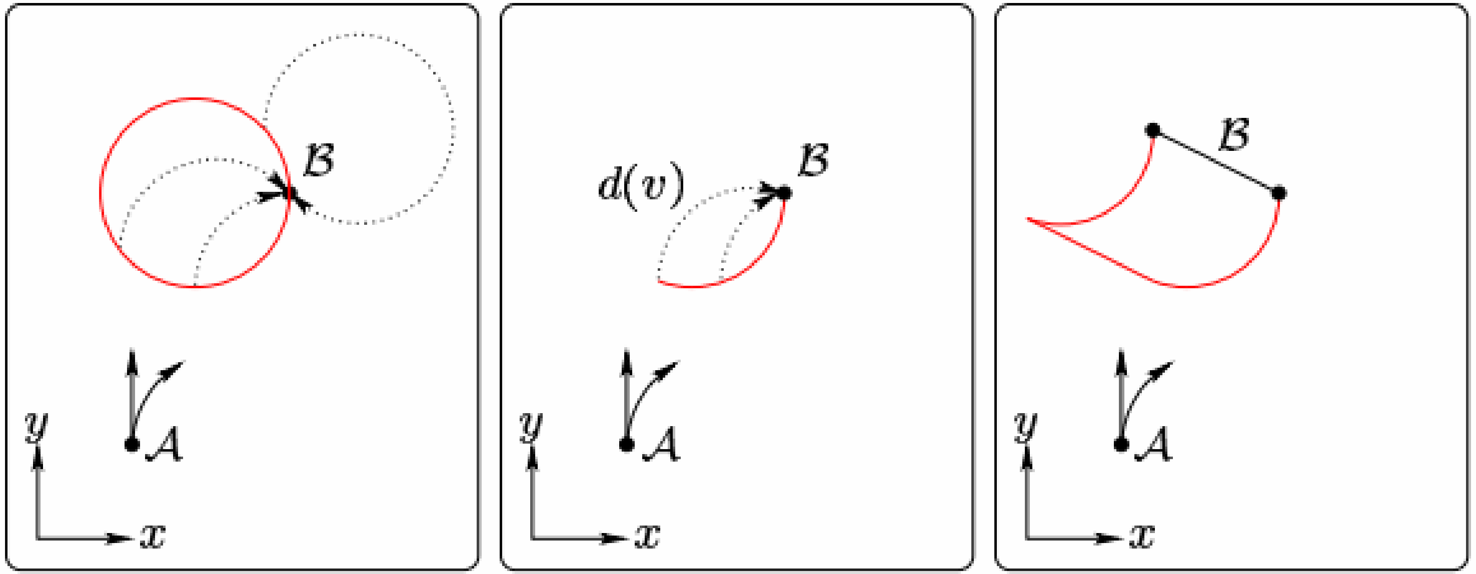


Fig. 11. $ICO(\mathcal{B}, \phi)$ for ϕ such that $\xi \neq 0$ ($\mathcal{A}$ is turning with a constant steering angle). $a = 0$ (left), a is changing (middle and right).

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Maneuver Planner: Obstacle Representation

- Depending on our planning approach, we have options on how we want to represent obstacles
- Obstacle-avoidance approaches
  - Space-time conics
  - RVOS
  - Critical-space planning

Maneuver Planner: Obstacle Representation

- Space-time conics
  - Choice in obstacle representation over time

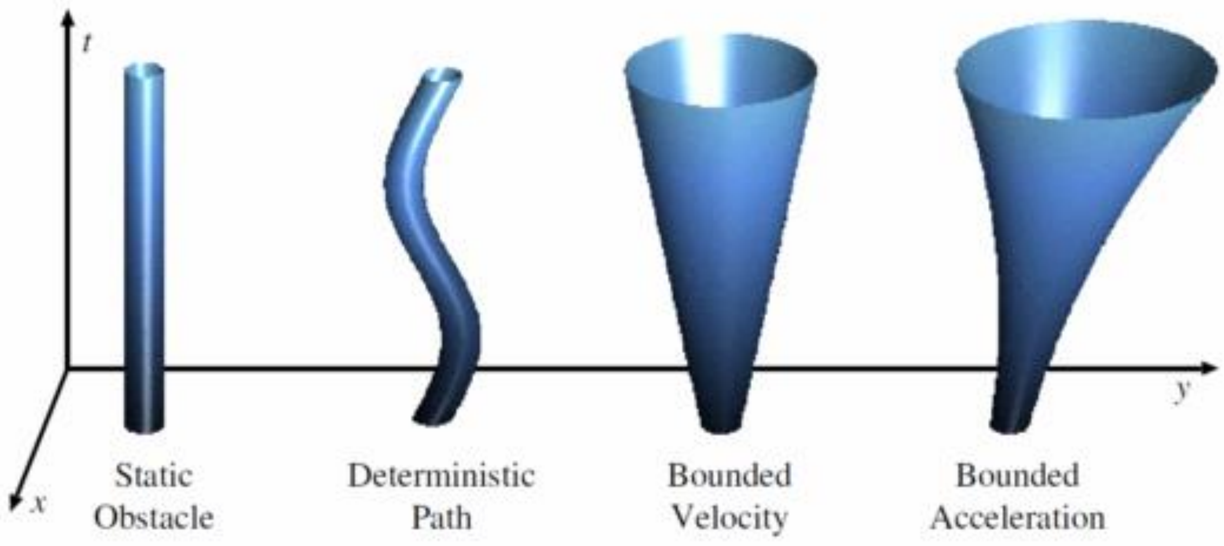
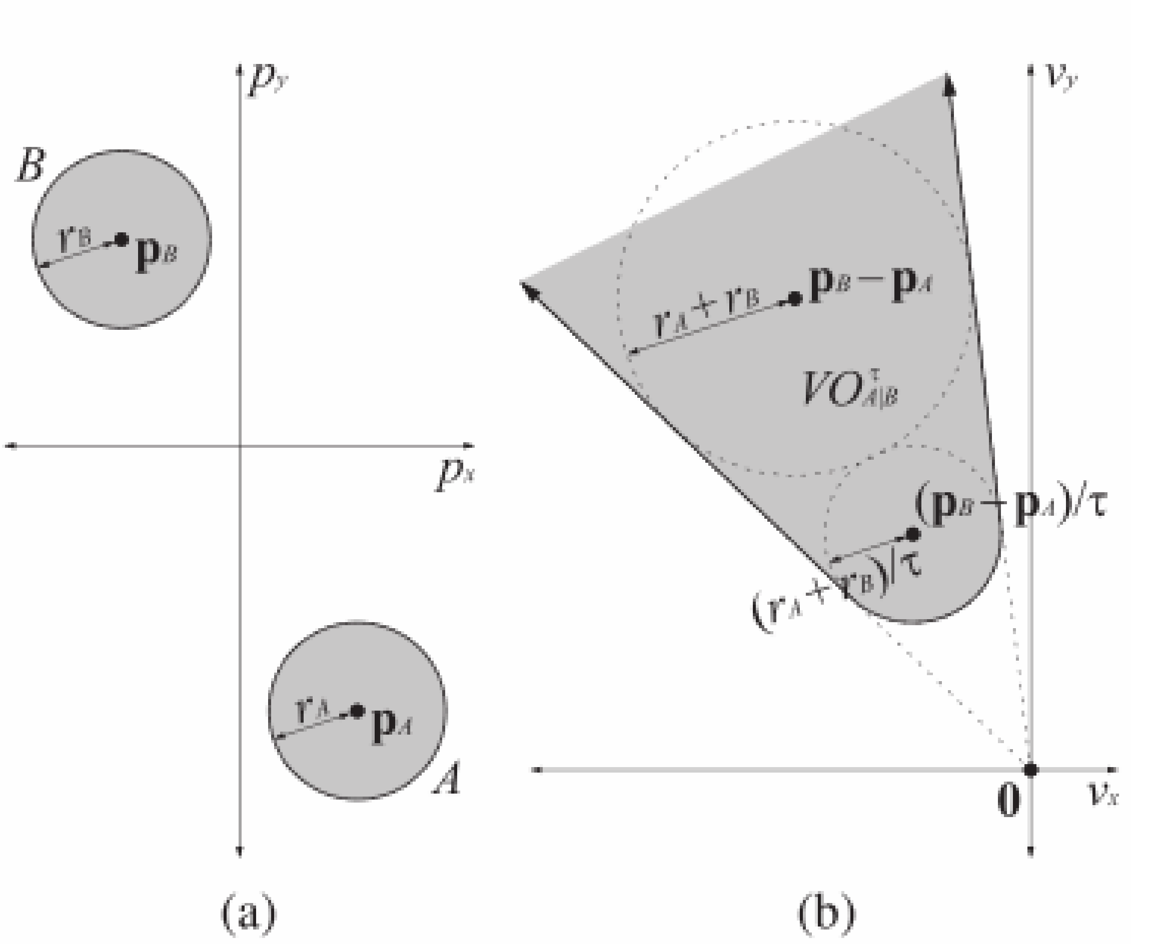


Figure 4. Obstacles as space-time volumes in $\Re^2 \times$ Time space [235]. Time is shown in vertical axis. When accounting for uncertainty, obstacle size grows with respect to time.

Maneuver Planner: Obstacle Representation

- RVOs: Reciprocal-velocity Obstacles
  - Prohibit velocity choices leading to collision within a time horizon assuming reciprocity
  - Originally proposed for discs

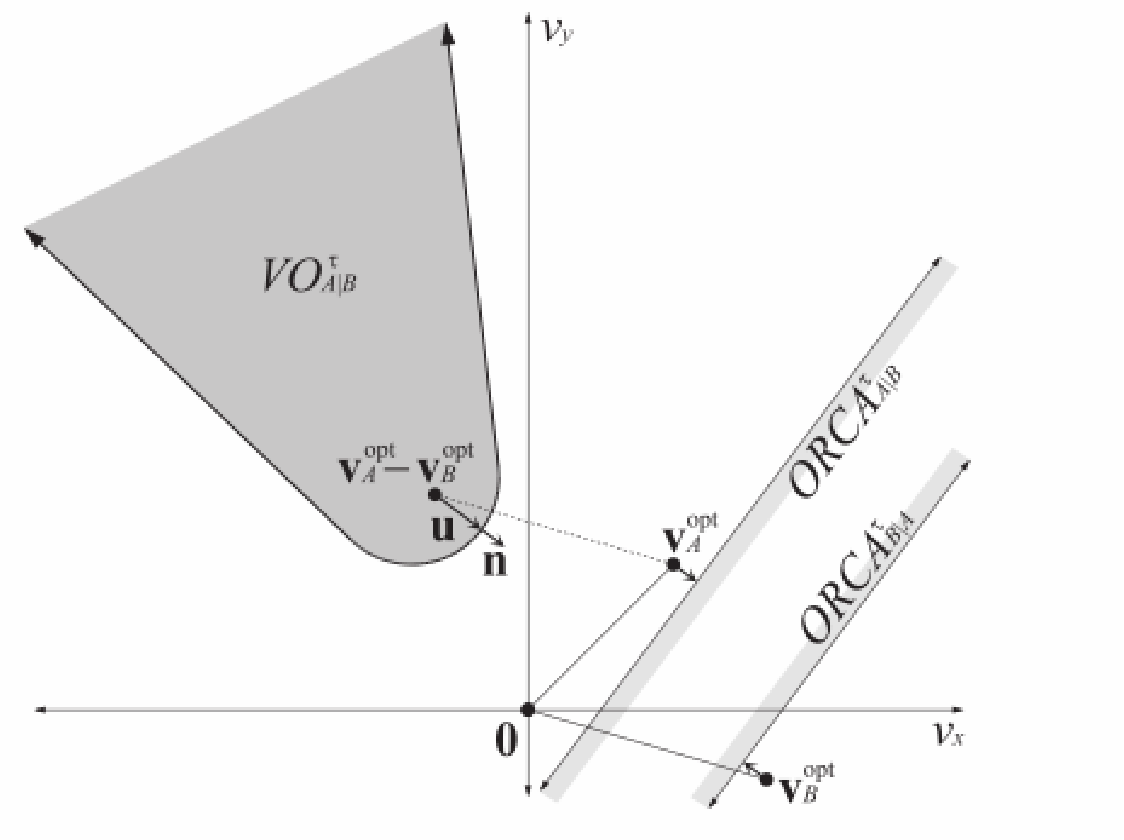
http://gamma.cs.unc.edu/ORCA/



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Maneuver Planner: Obstacle Representation

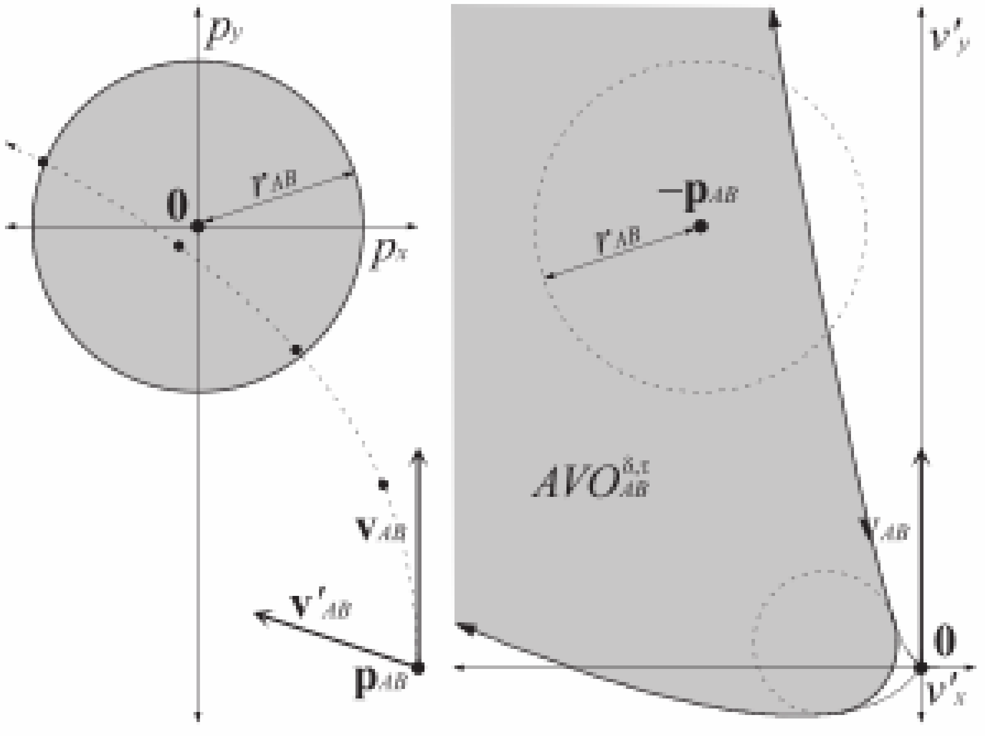
- RVOs: Reciprocal-velocity Obstacles
  - Constructs mutually exclusive velocity set choices for multiple robots
  - https://youtu.be/1Fn3Mz6f5xA?t=1m24s



THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Maneuver Planner: Obstacle Representation

- AVOs: Acceleration-Velocity Obstacles
  - Extends RVO concept to acceleration bounded shapes
  - https://youtu.be/BeNIPfWRLrY?t=23s



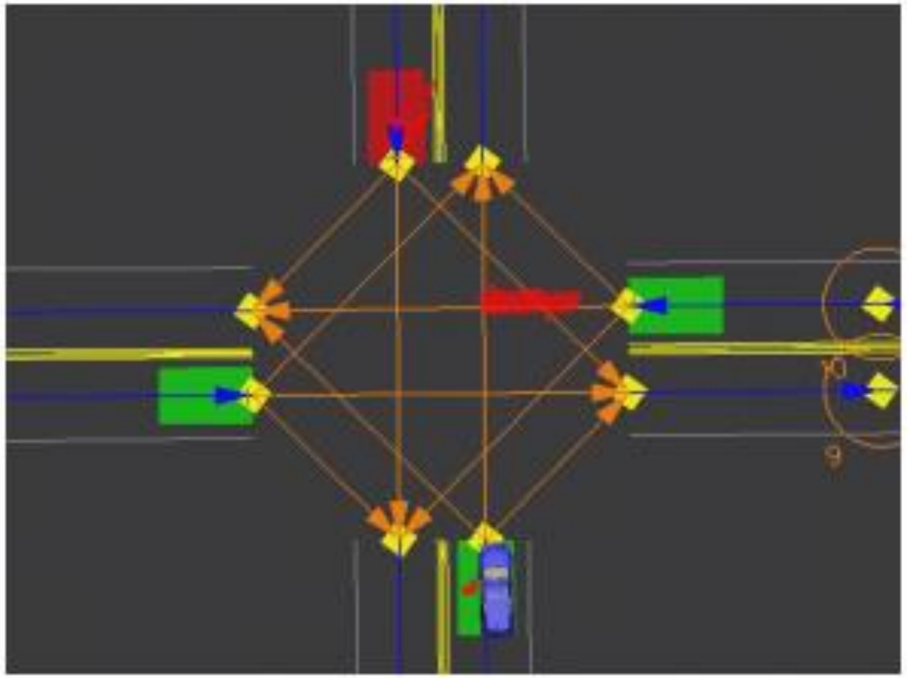
(b) (c)

Maneuver Planner: Obstacle Representation

- Control-Obstacles:
  - Plan avoidance directly in control space for arbitrary dynamics robots
  - https://youtu.be/X5nsubTAaWg?t=19s

Maneuver Planner: Obstacle Representation

- Critical-zone planning:
  - Determine “Critical zones” which trigger automatic stopping
  - Allows specific behavior encoding at intersections and stop signs



Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., … Thrun, S. (2009). Junior: The stanford entry in the urban challenge. *Springer Tracts in Advanced Robotics*, *56*(October 2005), 91–123. http://doi.org/10.1007/978-3-642-03991-1_3

Motion Planner

- Generally two stages:
  - Path planner - Computes the geometric representation of the path to be followed. I.e. the curve, spline, track, line, etc. we are following
  - Trajectory Planner / Path tracker - Computes the specific physical targets for following the path. I.e. velocity, acceleration, heading, steering, etc.

Maneuver Planner: Trajectory planning

- Given a determined path, we must compute local inputs to track the path
- Control theory, feedback applied over error in system
- Several approaches
  - Pure-pursuit tracker
  - Stanley Method

Structure

- Recap
- Kinematics & Dynamics Models
- Planning
- AutonoVi-Sim

Structure

- Recap
- Kinematics & Dynamics Models
- Planning
- AutonoVi-Sim

AutonoVi-Sim:

Modular Autonomous Vehicle Simulation Platform Supporting Diverse Vehicle Models, Sensor Configuration, and Traffic Conditions

Andrew Best, Sahil Narang, Lucas Pasqualin, Daniel Barber, Dinesh Manocha

University of North Carolina at Chapel Hill

UCF Institute for Simulation and Training

http://gamma.cs.unc.edu/AutonoVi/

Motivation

- 1.2 billion vehicles on the roads today
  - 84 million new vehicles in 2015
    - China: 24 m U.S.: 2.7m
    - India: 3.7 m S.E Asia: 3.8m
- Many markets expected to grow exponentially through 2030

New Delhi

Bangkok

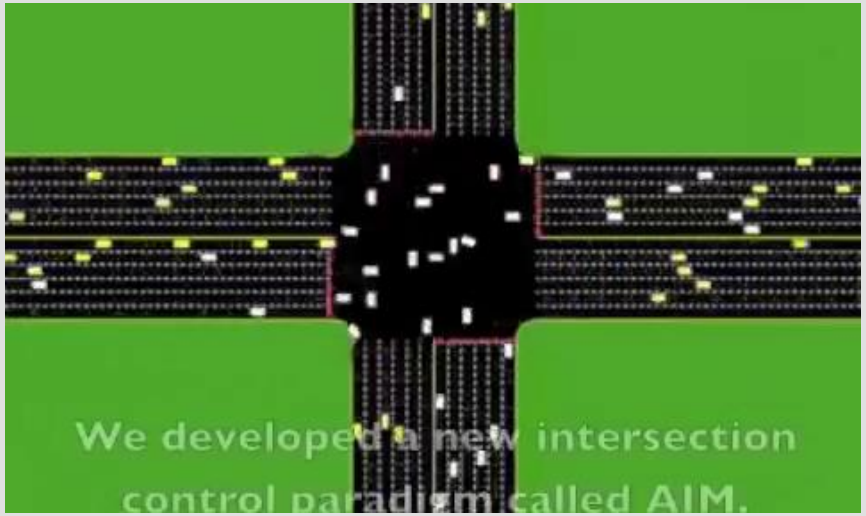
Motivation

- Majority of new vehicles in developing markets (30+ million)
- Limited infrastructure, loose traffic conventions
- Average vehicle life: 10+ years (17 years in U.S)



Motivation

- Long before autonomy will reach this:



Au et al. 2012

Kabbaj, TED 2016

Motivation

- It will deal with situations like these:

Motivation

- It will deal with situations like these:

Motivation

- It will deal with situations like these:

Challenges

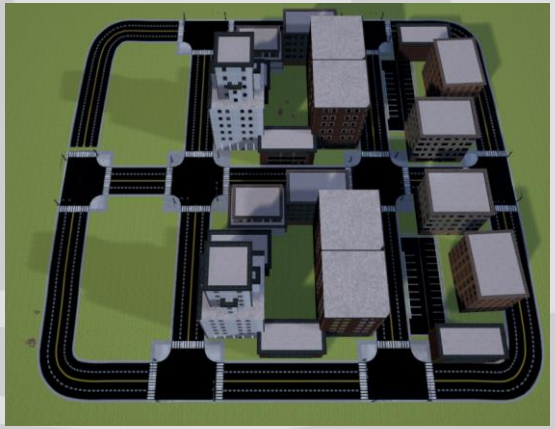
- Safety guarantees are critical
- Drivers, pedestrians, cyclists difficult to predict
- Road and environment conditions are dynamic
- Laws and norms differ by culture
- Huge number of scenarios


Challenges

- Development and testing of autonomous driving algorithms
  - On-road experiments may be hazardous
  - Closed-course experiments may limit transfer
  - High costs in terms of time and money

- Solution: develop and test robust algorithms in simulation
  - Test novel driving strategies & sensor configurations
  - Reduces costs
  - Allows testing dangerous scenarios
  - Vary traffic and weather conditions

Parking lot mock-up



Simulated city

Contributions

- AutonoVi-Sim : high fidelity simulation platform for testing autonomous driving algorithms
  - Varying vehicle types, traffic condition
  - Rapid Scenario Construction
  - Simulates cyclists and pedestrians
  - Modular Sensor configuration, fusion
  - Facilitates testing novel driving strategies

Contributions

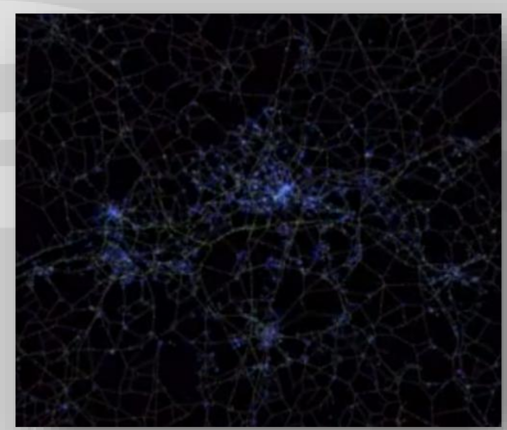
- AutonoVi: novel algorithm for autonomous vehicle navigation
  - Collision-free, dynamically feasible maneuvers
  - Navigate amongst pedestrians, cyclists, other vehicles
  - Perform dynamic lane-changes for avoidance and overtaking
  - Generalizes to different vehicles through data-driven dynamics approach
  - Adhere to traffic laws and norms

Overview

- Motivation
- **Related Work**
- Contributions:
  - Simulation Platform: Autonovi-Sim
  - Navigation Algorithm: Autonovi
- Results

Related work:

- Traffic Simulation
  - MATSim [Horni 2016], SUMO [krajzewicz 2002]
- Autonomous Vehicle Simulation
  - OpenAI Universe, Udacity
  - Waymo Carcraft, Righthook.io
- Simulation integral to development of many controllers & recent approaches [Katrakazas2015].



MATSim

SUMO

Related work:

- Collision-free navigation
  - Occupancy grids[Kolski 2006], driving corridors [Hardy 2013]
  - Velocity Obstacles [Berg 2011], Control obstacles [Bareiss 2015], polygonal decomposition [Ziegler 2014], random exploration [Katrakazas 2015]
  - Lateral control approaches [Fritz 2004, Sadigh 2016]

- Generating traffic behaviors
  - Human driver model [Treiber 2006], data-driven [Hidas 2005], correct by construction [Tumova 2013], Bayesian prediction [Galceran 2015]

Related work:

- Modelling Kinematics and Dynamics
  - kinematic models [Reeds 1990, LaValle 2006, Margolis 1991]
  - Dynamics models [Borrelli 2005]

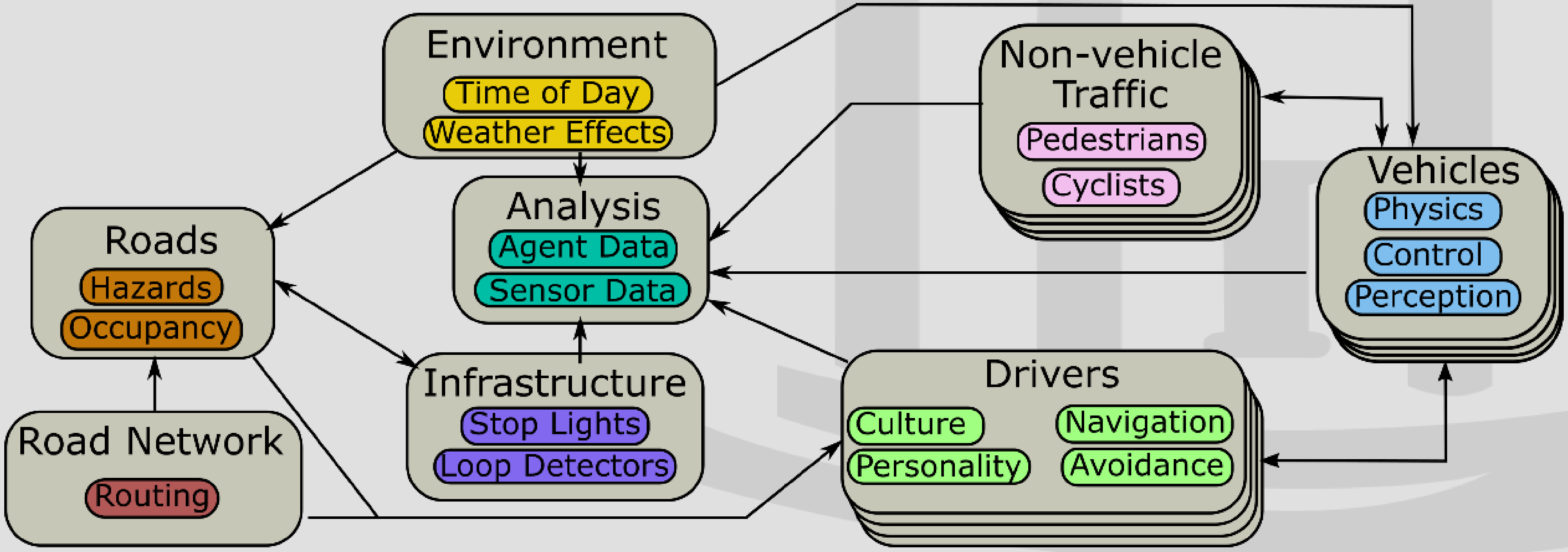
- Simulation for Vision Training
  - Grand Theft Auto 5 [Richter 2016, Johnson-Roberson 2017]

Overview

- Motivation
- Related Work
- **Contributions:**
  - **Simulation Platform: Autonovi-Sim**
  - Navigation Algorithm: Autonovi
- Results

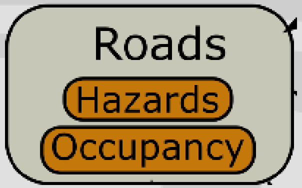
Autonovi-Sim

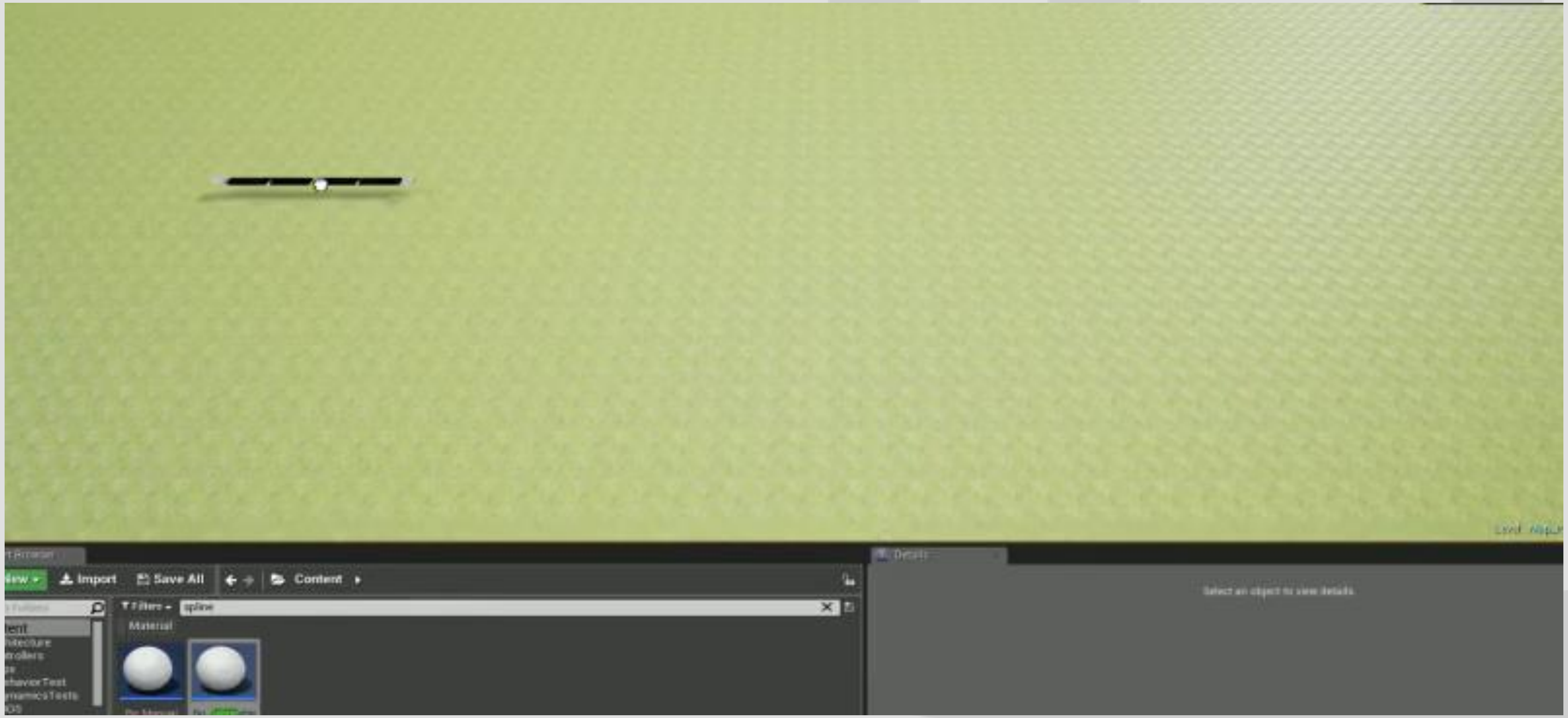
- Modular simulation framework for generating dynamic traffic conditions, weather, driver profiles, and road networks
- Facilitates novel driving strategy development



Autonovi-Sim: Roads & Road Network

- Roads constructed by click and drag
- Road network constructed automatically

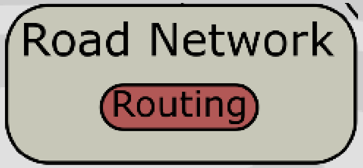


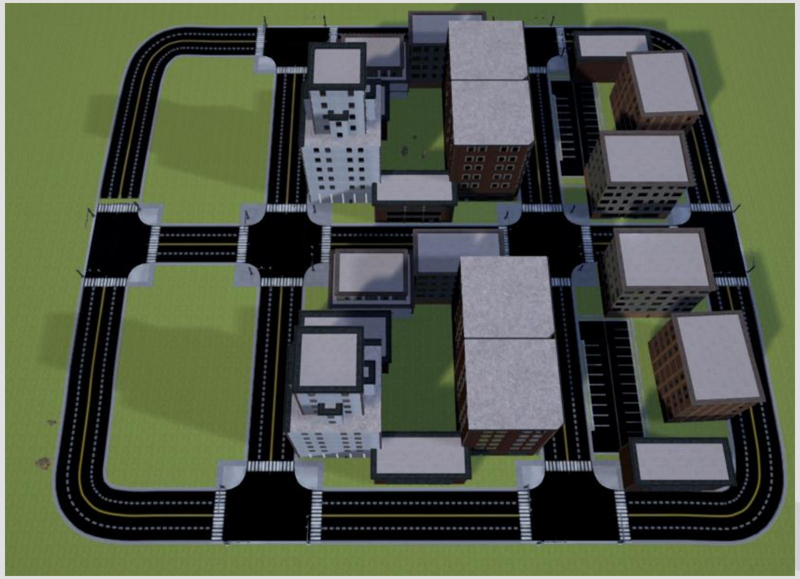


Road layouts

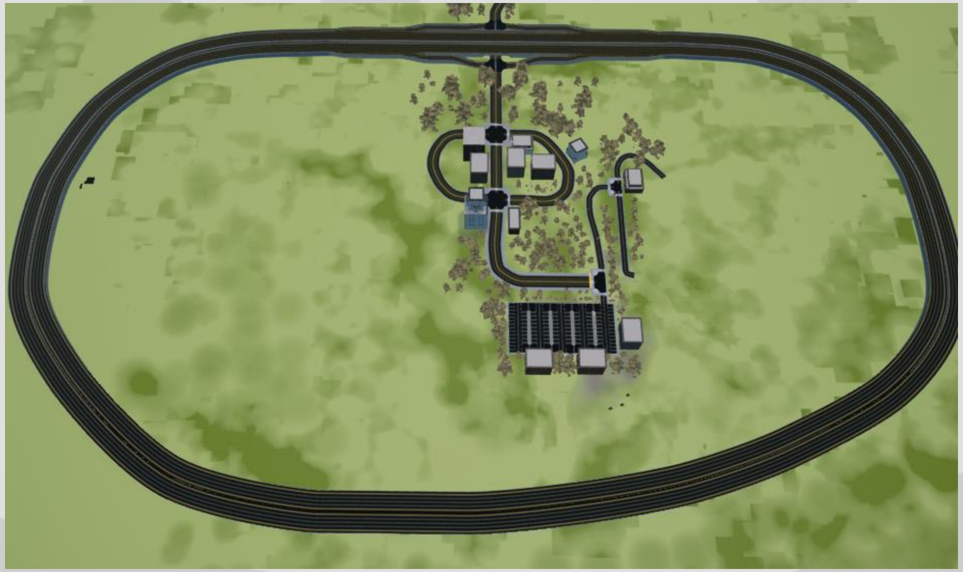
Autonovi-Sim: Roads & Road Network

- Construct large road networks with minimal effort
- Provides routing and traffic information to vehicles
- Allows dynamic lane closures, sign obstructions





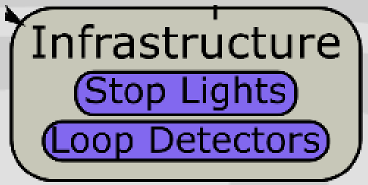
Urban Environment for pedestrian & cyclist testing

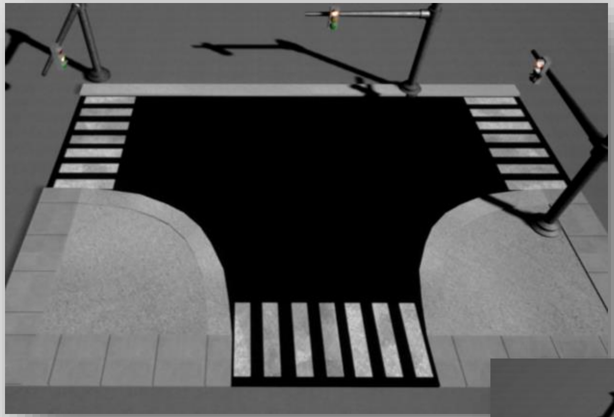


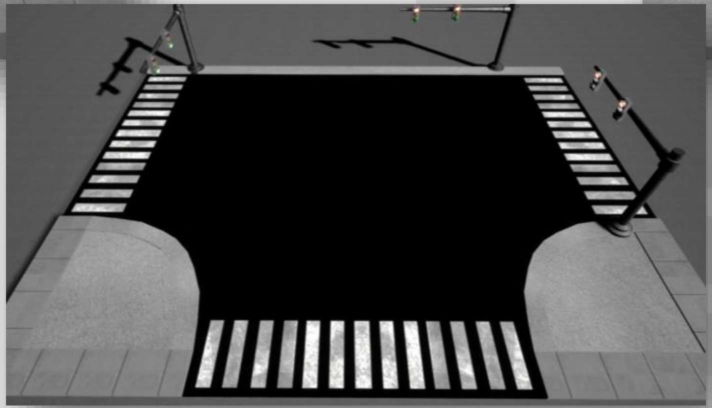
4 kilometer highway on and off loop

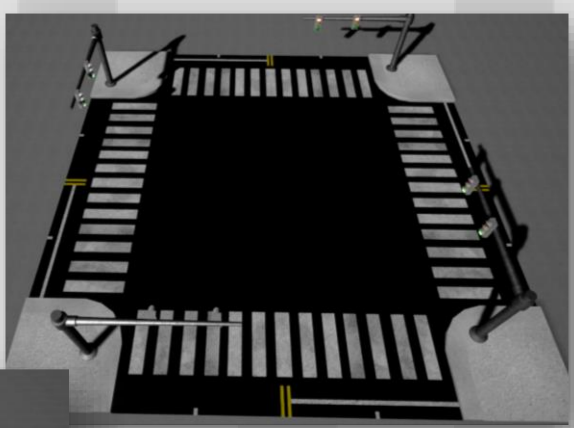
Autonovi-Sim: Infrastructure

- Infrastructure placed as roads or overlays
- Provide cycle information to vehicles, can be queried and centrally controlled



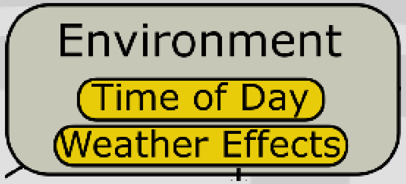

3 way, one lane


3 way, two lane


4 way, two lane

Autonovi-Sim: Environment

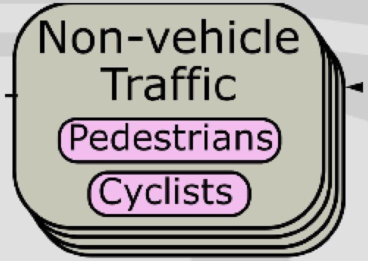
- Goal: Testing driving strategies & sensor configuration in adverse conditions
- Simulate changing environmental conditions
  - Rain, fog, time of day
  - Modelling associated physical changes



Fog reduces visibility

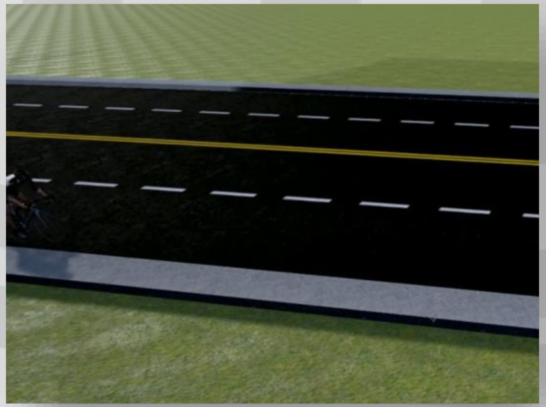
Heavy rain reduces traction

Autonovi-Sim: Non-vehicle Traffic



- Cyclists
  - operate on road network
  - Travel as vehicles, custom destinations and routing
- Pedestrians
  - Operate on roads or sidewalks
  - Programmable to follow or ignore traffic rules
  - Integrate prediction and personality parameters

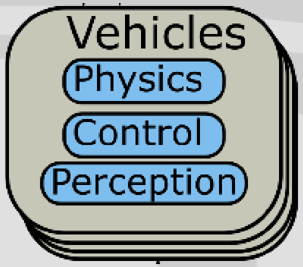
Pedestrian Motion

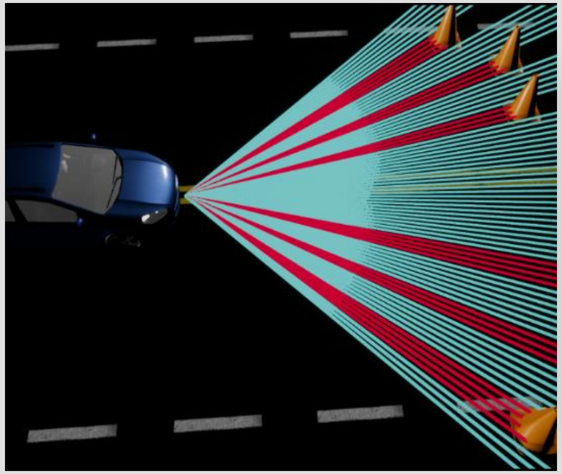


Cyclist Motion

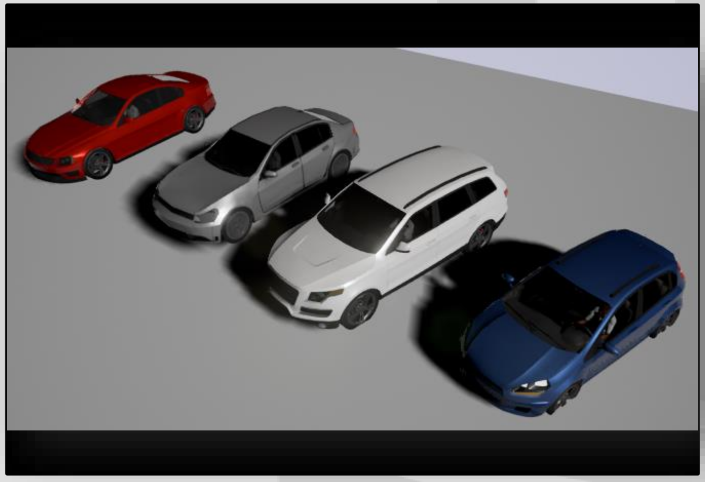
Autonovi-Sim: Vehicles

- Various vehicle profiles:
  - Size, shape, color
  - Speed / engine profile
  - Turning / braking
- Manage sensor information

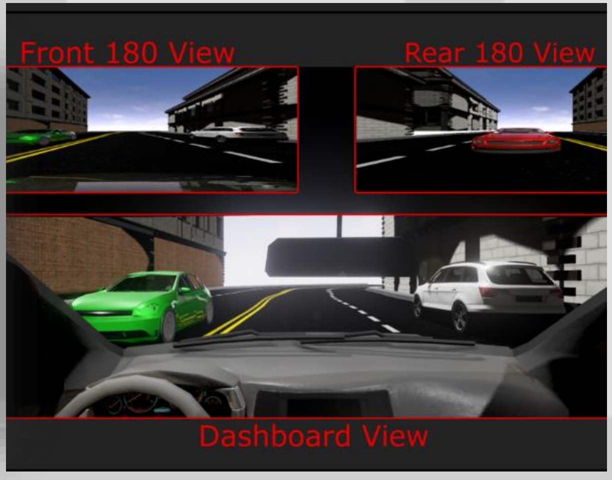




Laser Range-finder



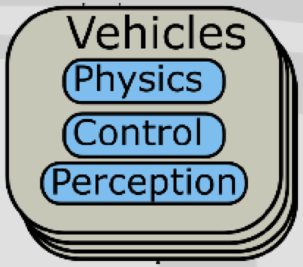
Multiple Vehicle Configurations



Multi-camera detector

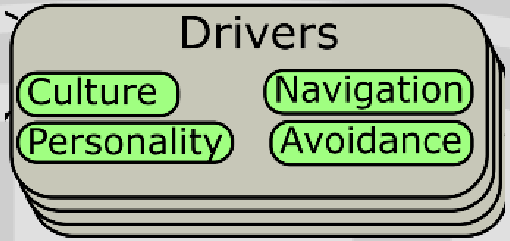
Autonovi-Sim: Vehicles

- Sensors placed interactively on vehicle
  - Configurable perception and detection algorithms



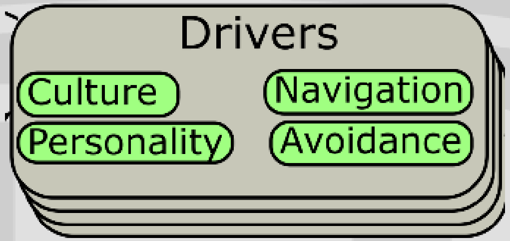
Autonovi-Sim: Drivers

- Control driving decisions
  - Fuse sensor information
  - Determine new controls (steering, throttle)
- Configurable parameters representing personality
  - Following distance, attention time, speeding, etc.
- Configure proportions of driver types
  - i.e. 50% aggressive, 50% cautious



Autonovi-Sim: Drivers

- 3 Drivers in AutonoVi-Sim
  - Manual
  - Basic Follower
  - AutonoVi



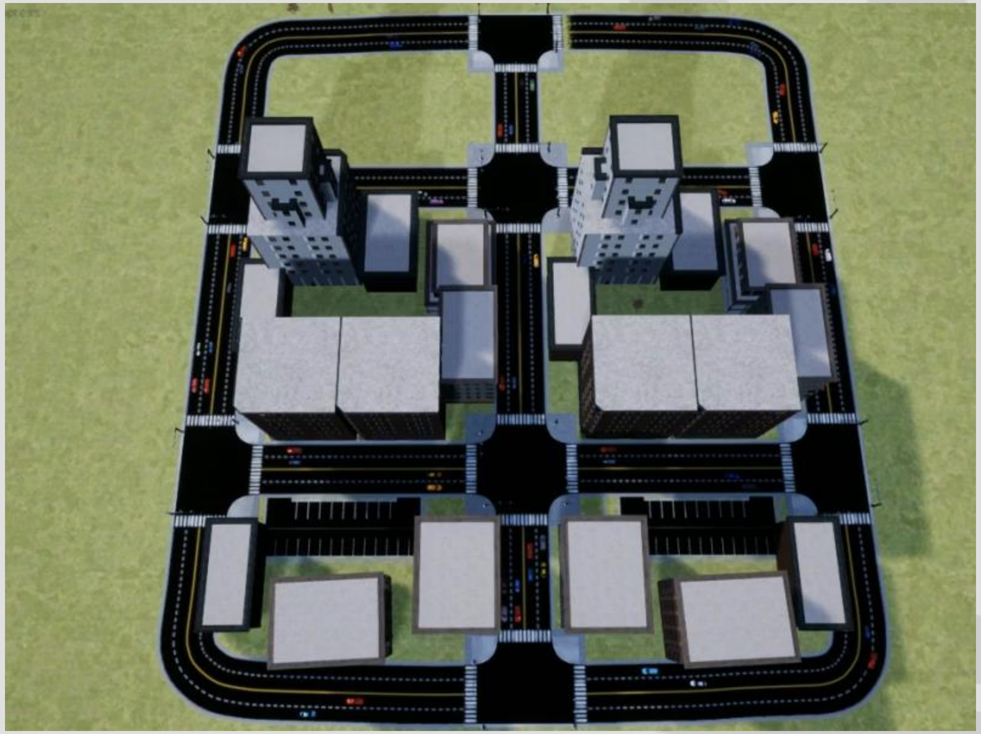
Manual Drive

Basic Follower

AutonoVi

Autonovi-Sim: Results

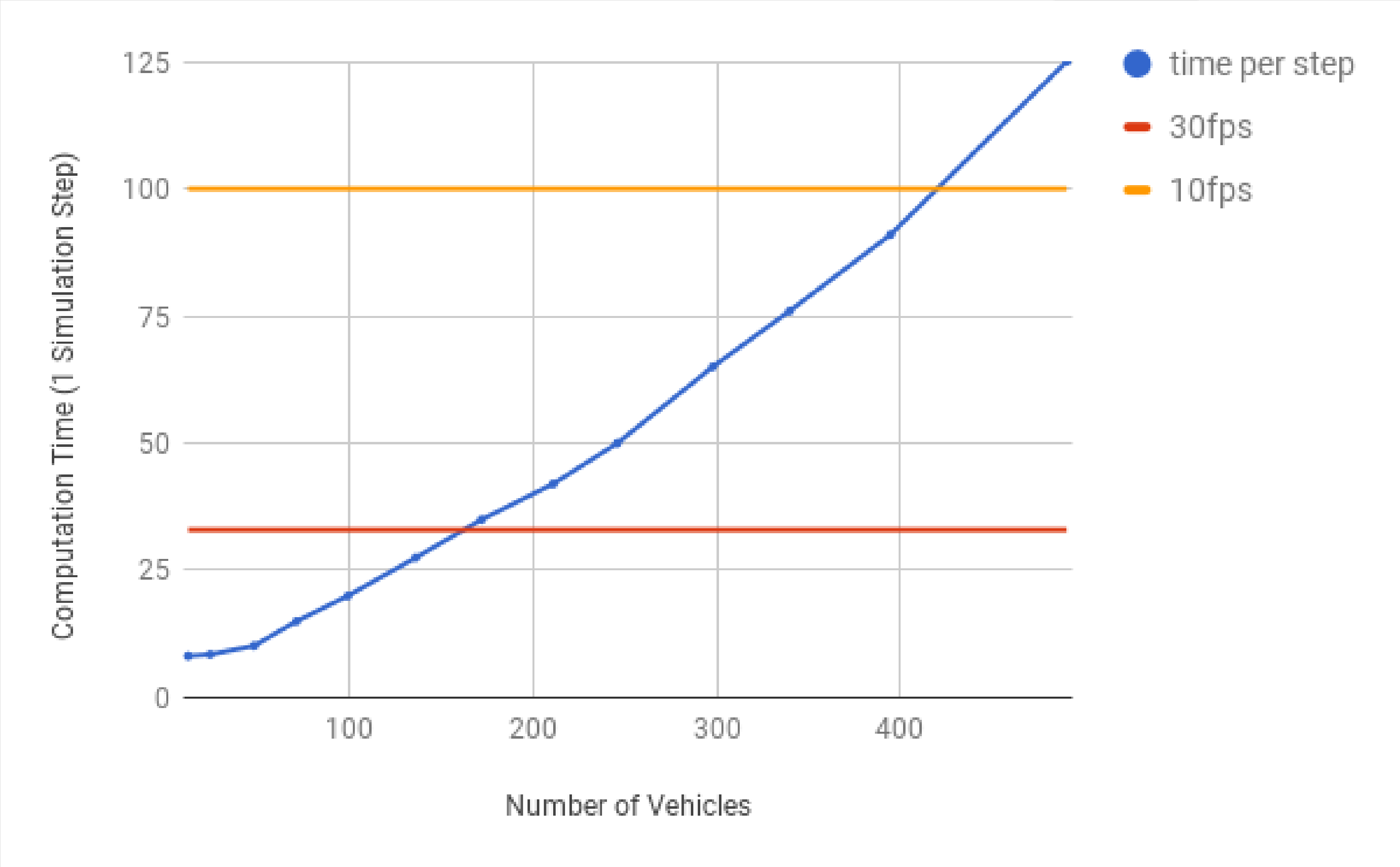
- Simulating large, dense road networks
- Generating data for analysis, vision classification, autonomous driving algorithms



50 vehicles navigating (3x)

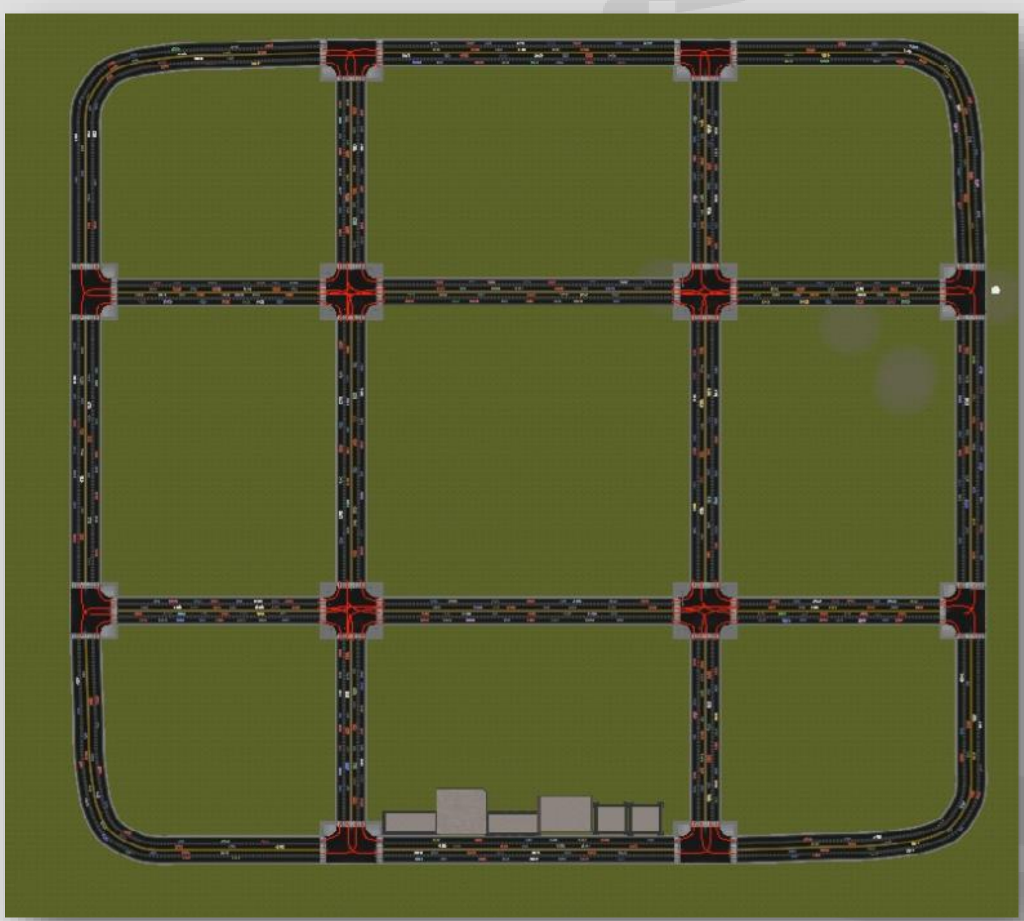
Autonovi-Sim: Results

- Interactive Simulation of hundreds of vehicles



Autonovi-Sim: Results

- 600+ vehicles on 3.5 km

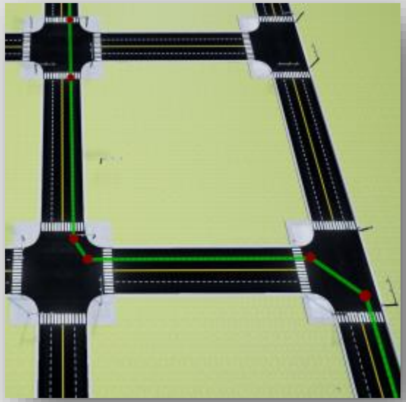


Overview

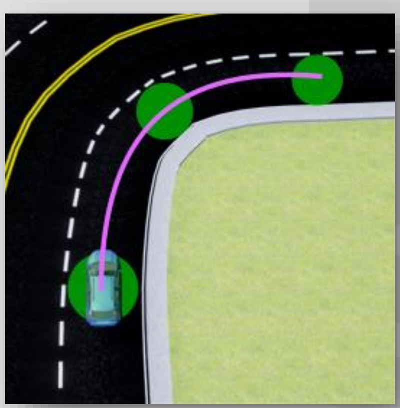
- Motivation
- Related Work
- **Contributions:**
  - Simulation Platform: Autonovi-Sim
  - **Navigation Algorithm: Autonovi**
- Results

Autonovi

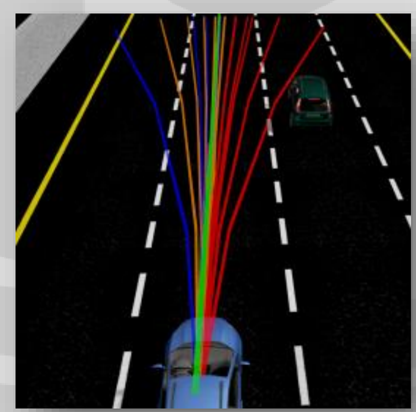
- Computes collision free, dynamically feasible maneuvers amongst pedestrians, cyclists, and vehicles
- 4 stage algorithm
  - Routing / GPS
  - Guiding Path Computation
  - Collision-avoidance / Dynamics Constraints
  - Optimization-based Maneuvering



GPS Routing



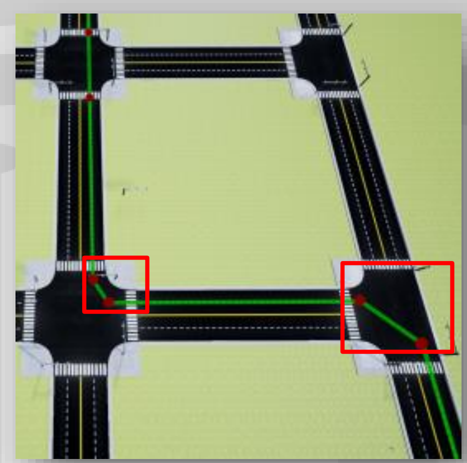
Guiding Path



Optimization-based Maneuvering

Autonovi: Routing / GPS

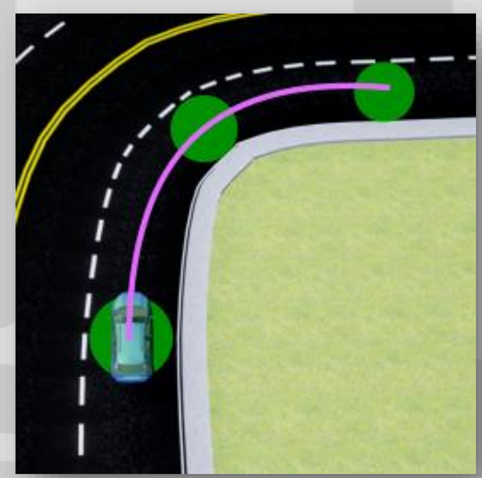
- Generates maneuvers between vehicle position and destination
- Nodes represent road transitions
- Allows vehicle to change lanes between maneuvers



GPS Routing

Autonovi: Guiding Path

- Computes “ideal” path vehicle should follow
- Respects traffic rules
- Path computed and represented as arc
- Generates target controls



Guiding Path

Autonovi: Collision Avoidance / Dynamics

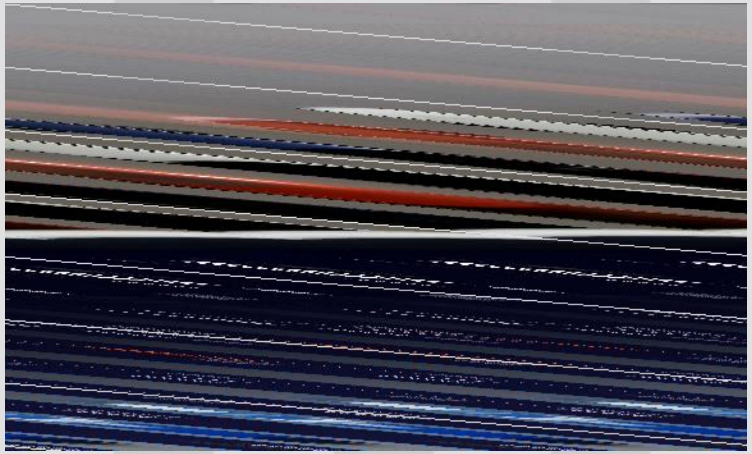
- Control Obstacles [Bareiss 2015]
  - “Union of all controls that could lead to collisions with the neighbor within the time horizon, τ”
  - Plan directly in control space (throttle, steering)
  - Construct “obstacles” for nearby entities
- Key principles / Assumptions
  - Reciprocity in avoidance (all agents take equal share)
  - Bounding discs around each entity
  - Controls / decisions of other entities are observable
  - New controls chosen as minimal deviation from target s. t. the following is not violated:

$$\forall (j \neq i, 0 \leq t < \tau) :: (\mathcal{O}_i \oplus \{\mathbf{q}_i(\mathbf{g}_i(t, \mathbf{x}_i, \mathbf{u}_i + \Delta \mathbf{u}_i))\}) \cap (\mathcal{O}_j \oplus \{\mathbf{q}_j(\mathbf{g}_j(t, \mathbf{x}_j, \mathbf{u}_j + \Delta \mathbf{u}_j))\}) = \emptyset$$

[Bareiss 2015]

Autonovi: Collision Avoidance / Dynamics

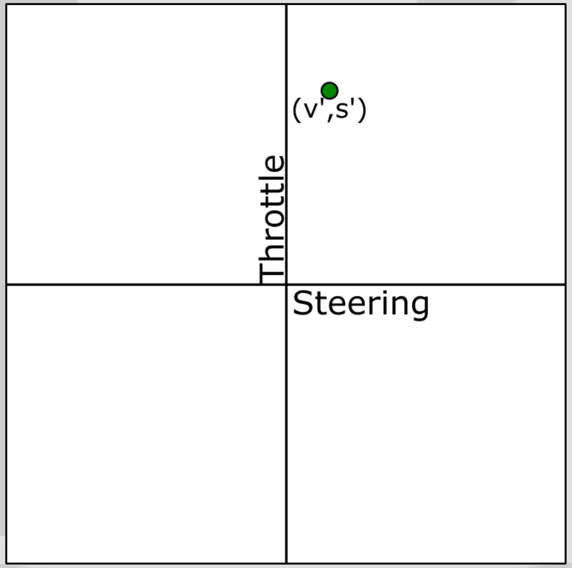
- Goal: Augment control obstacles with dynamics constraints
- Generate dynamics profile for vehicles through profiling
  - repeated simulation for each vehicle testing control inputs
- Represent underlying dynamics without specific model
- Gather data to generate approximation functions for non-linear vehicle dynamics
  - S(μ) : target controls are safe given current vehicle state
  - A(μ) : Expected acceleration given effort and current state
  - Φ(μ) : Expected steering change given effort and current state



Dynamics Profile Generation

Autonovi: Collision Avoidance / Dynamics

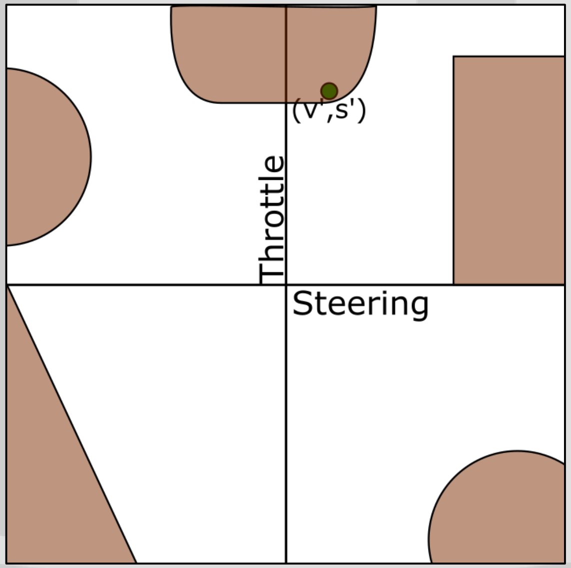
- Augmented Control Obstacles
  - Reciprocity is not assumed from others
  - Use tightly fitting bounding polygons
  - Do not assume controls of others are observable
  - New controls chosen from optimization stage



Autonovi: Collision Avoidance / Dynamics

- Augmented Control Obstacles
  - Reciprocity is not assumed from others
  - Use tightly fitting bounding polygons
  - Do not assume controls of others are observable
  - New controls chosen from optimization stage

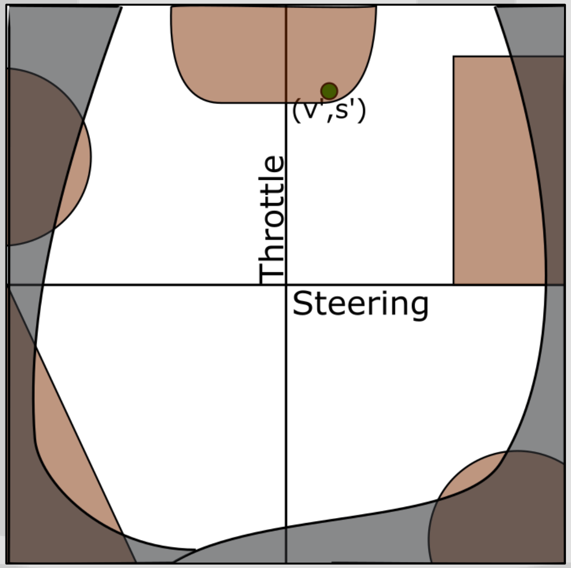
- Obstacles constructed from avoidance



Autonovi: Collision Avoidance / Dynamics

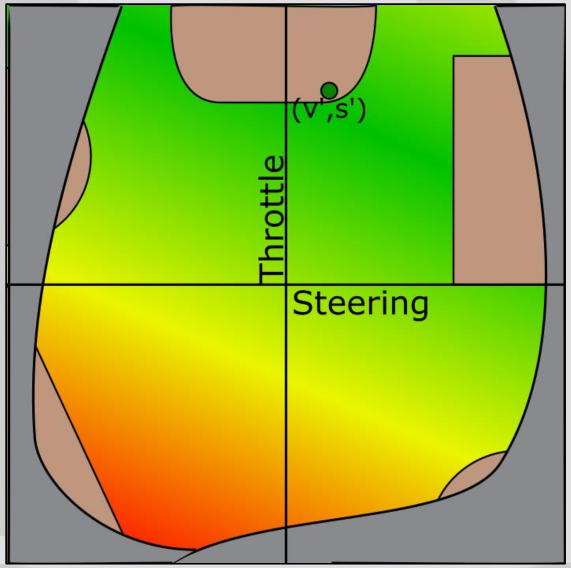
- Augmented Control Obstacles
  - Reciprocity is not assumed from others
  - Use tightly fitting bounding polygons
  - Do not assume controls of others are observable
  - New controls chosen from optimization stage

- Obstacles constructed from avoidance
- Obstacles constructed from dynamics



Autonovi: Collision Avoidance / Dynamics

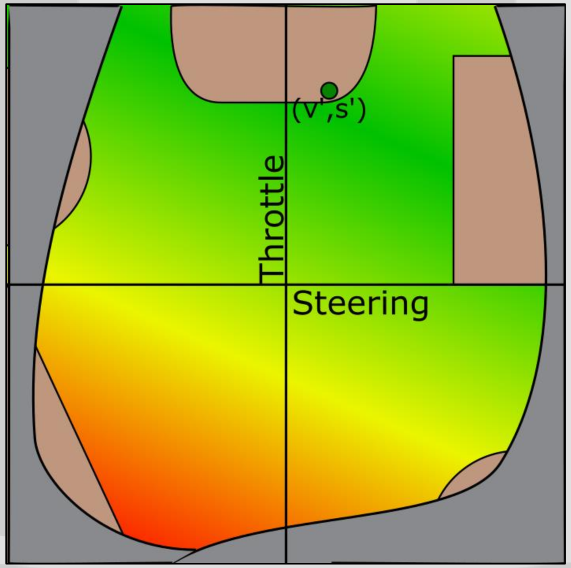
- Augmented Control Obstacles
  - Reciprocity is not assumed from others
  - Use tightly fitting bounding polygons
  - Do not assume controls of others are observable
  - New controls chosen from optimization stage



- Obstacles constructed from avoidance
- Obstacles constructed from dynamics
- New velocity chosen by cost-optimization

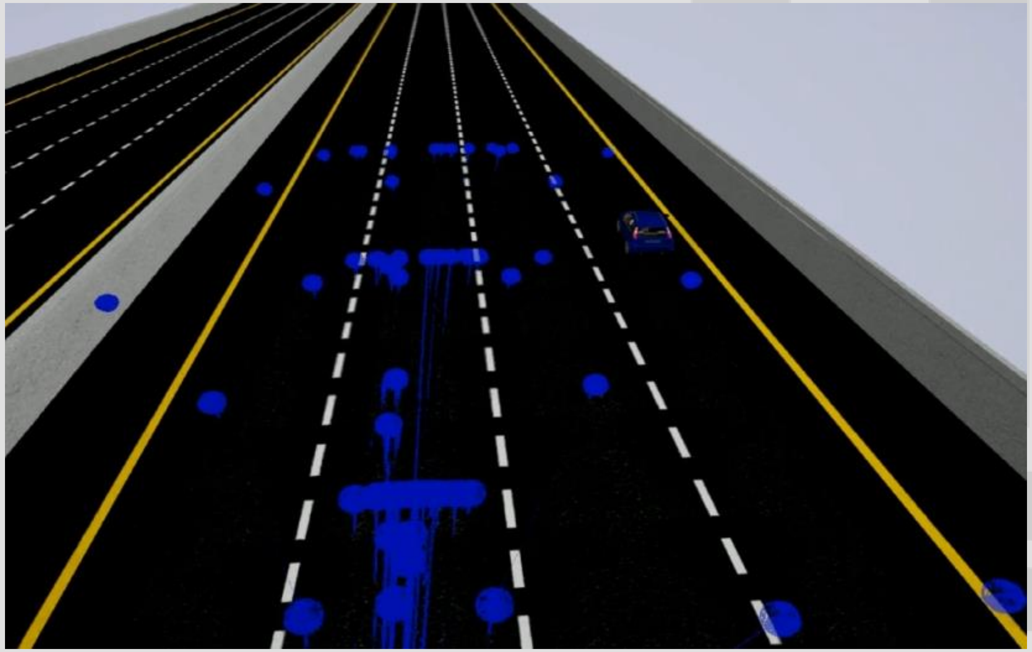
Autonovi: Collision Avoidance / Dynamics

- Advantages of augmented control obstacles:
  - Free-space is guaranteed feasible and safe
  - Conservative linear constraints from surface of obstacles
- Disadvantages:
  - Closed-form of surface may not exist
  - Space may be non-convex
  - Computationally expensive



Autonovi: Collision Avoidance / Dynamics

- Sampling approach
  - Construct candidate controls via sampling near target controls
- Evaluate collision-avoidance and dynamics constraints
  - Forward integrate safe controls to generate candidate trajectories
- Choose “optimal” control set in optimization stage



Autonovi: Optimization-Based Maneuvering

- Choose “optimal” controls through multi-objective cost function
- Path (velocity, drift, progress)
- Comfort (acceleration, yaw)
- Maneuver (lane change, node distance)
- Proximity (cyclists, vehicle, pedestrians)

$$C = \sum_{i=0}^{I} c_{path}(i) + c_{cmft}(i) + c_{mnvr}(i) + c_{prox}(i)$$

Autonovi: Optimization-Based Maneuvering

- Choose "optimal" controls through multi-objective cost function
- Path (velocity, drift, progress)
- Comfort (acceleration, yaw)
- **Maneuver (lane change, node distance)**
  - Static cost for lane changes
  - Cost inverse to distance if vehicle occupies incorrect lane as maneuver approaches
- Proximity (cyclists, vehicle, pedestrians)

$$C = \sum_{i=0}^{I} c_{path}(i) + c_{cmft}(i) + c_{mnvr}(i) + c_{prox}(i)$$

Autonovi: Optimization-Based Maneuvering

- Choose "optimal" controls through multi-objective cost function
- Path (velocity, drift, progress)
- Comfort (acceleration, yaw)
- Maneuver (lane change, node distance)
- **Proximity (cyclists, vehicle, pedestrians)**
  - Configurable cost per entity type
  - Generates safe passing buffers

$$C = \sum_{i=0}^{I} c_{path}(i) + c_{cmft}(i) + c_{mnvr}(i) + c_{prox}(i)$$

Overview

- Motivation
- Related Work
- Contributions:
  - Simulation Platform: Autonovi-Sim
  - Navigation Algorithm: Autonovi
- **Results**

Results: Sudden Hazards @ 20 mph

- Vehicle responds quickly to sudden hazards
  - Braking and swerving to avoid collisions

Results: Sudden Hazards @ 60 mph

- Vehicle responds quickly to sudden hazards
  - Respects unique dynamics of each car

Results: Jaywalking Pedestrian

- Vehicle accounts for pedestrians and comes to a stop

Results: Jaywalking Pedestrian

- Vehicle accounts for pedestrians and comes to a stop
  - Respects unique dynamics of each car

Results: Passing Cyclists

- Vehicle changes lanes to safely pass cyclist

Results: Passing Cyclists

- Vehicle changes lanes to safely pass cyclist
  - Lane change only when possible

Results: Next Steps

- Generating data for deep-learning
  - Growing consensus that synthetic data is necessary for AV training

Results: Next Steps

- Using real-world training data, behaviors can be optimized to improve realism
- Ex: Drivers behave more like human drivers
- Ex: Infrastructure tuned to specific real patterns

- Vehicle sensors can be similarly calibrated

Maneuver Planner: Project ideas

- Improving tracking using a deep learnt pedestrian detection framework
- Biometric Walk: Learning and classifying pedestrian trajectories/behavior to a specific person to improve person identification
- Autonomous intelligent navigation of robots in a crowd (Pepper)
- Anomaly Detection using machine learning on a synthetic dataset
- Designing models for robots to be more socially-tolerant. Improve the personal space from SocioSense to more than just a fixed circle - a probabilistic comfort zone.

Maneuver Planner: Project ideas

- Sampling-based planner / Parameter optimization
- Trajectory Analysis / simulation data logging and analysis
- Perception models for detection (pedestrian detection from simulation)
- Modelling sensors (virtual lidar etc)
- Driver behavior learning and classification
- Implementing alternate planners (elastic band / rrt / state lattice / etc)
- Cyclist and Pedestrian planning expansion in AutonoVi-Sim
- Modelling better fidelity weather and its impact on sensor information

Maneuver Planner: Related reading

- Katarakazas: Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions
- Pendleton et al.: Perception, Planning, Control, and Coordination for Autonomous Vehicles
- Lefèvre et al. : A survey on motion prediction and risk assessment for intelligent vehicles
- Saifuzzaman et al: Incorporating human-factors in car-following models: a review of recent developments and research needs
- Bast et al.: Route planning in transportation networks

Questions

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL