SPOT: A Tool for Set-Based Prediction of Traffic Participants

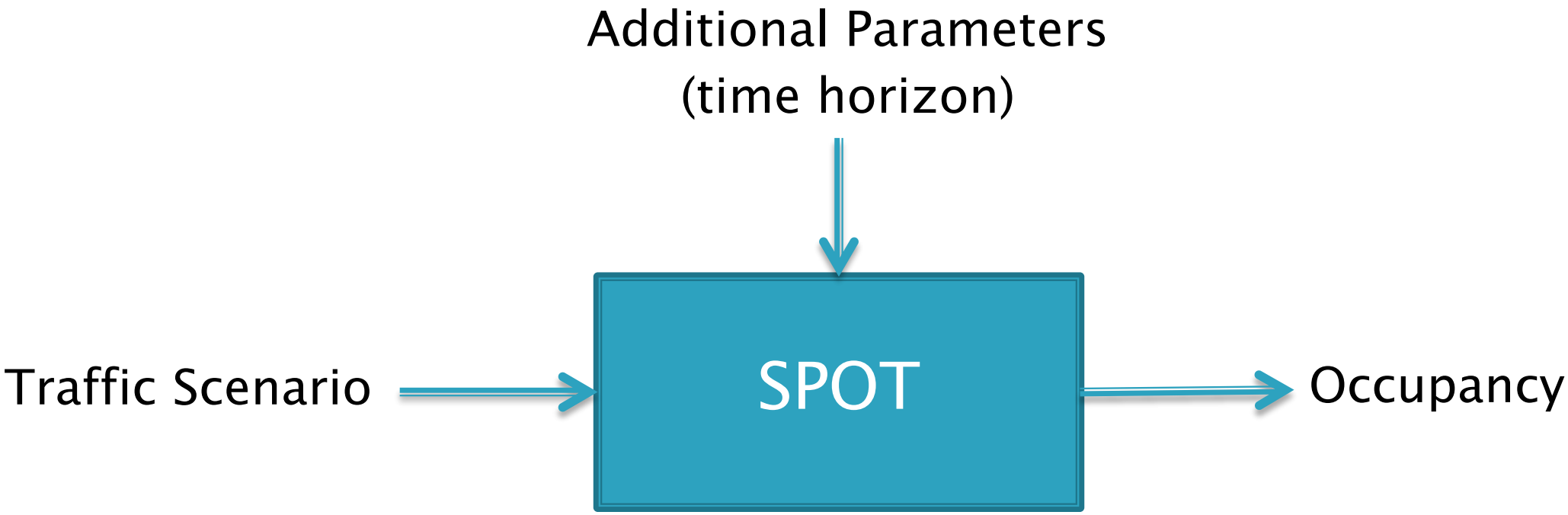
Sebastian Kaster
Technical University Munich
March 22, 2018

Motivation

- Trajectory safe → no intersection with occupancies of other traffic participants
- Calculation of a bunch of occupancies is time consuming

  → Use a fast programming language!

Overview Model

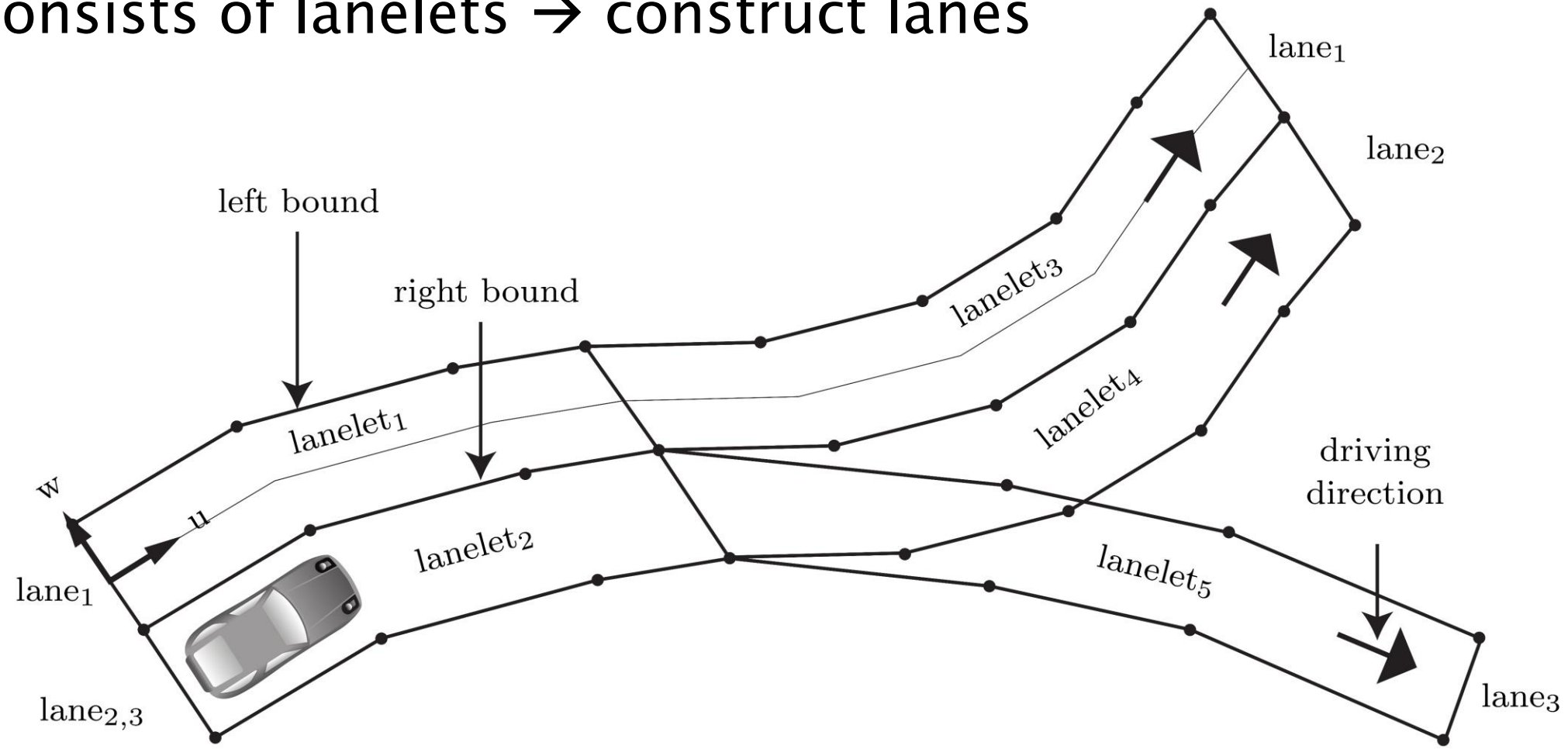


CommonRoad

- XML format for specifying road traffic scenarios
- Consists of represantations for
  - Road network
  - Obstacles (static, dynamic)
  - Ego vehicle
  - Goal region

Road Network

- Consists of lanelets → construct lanes

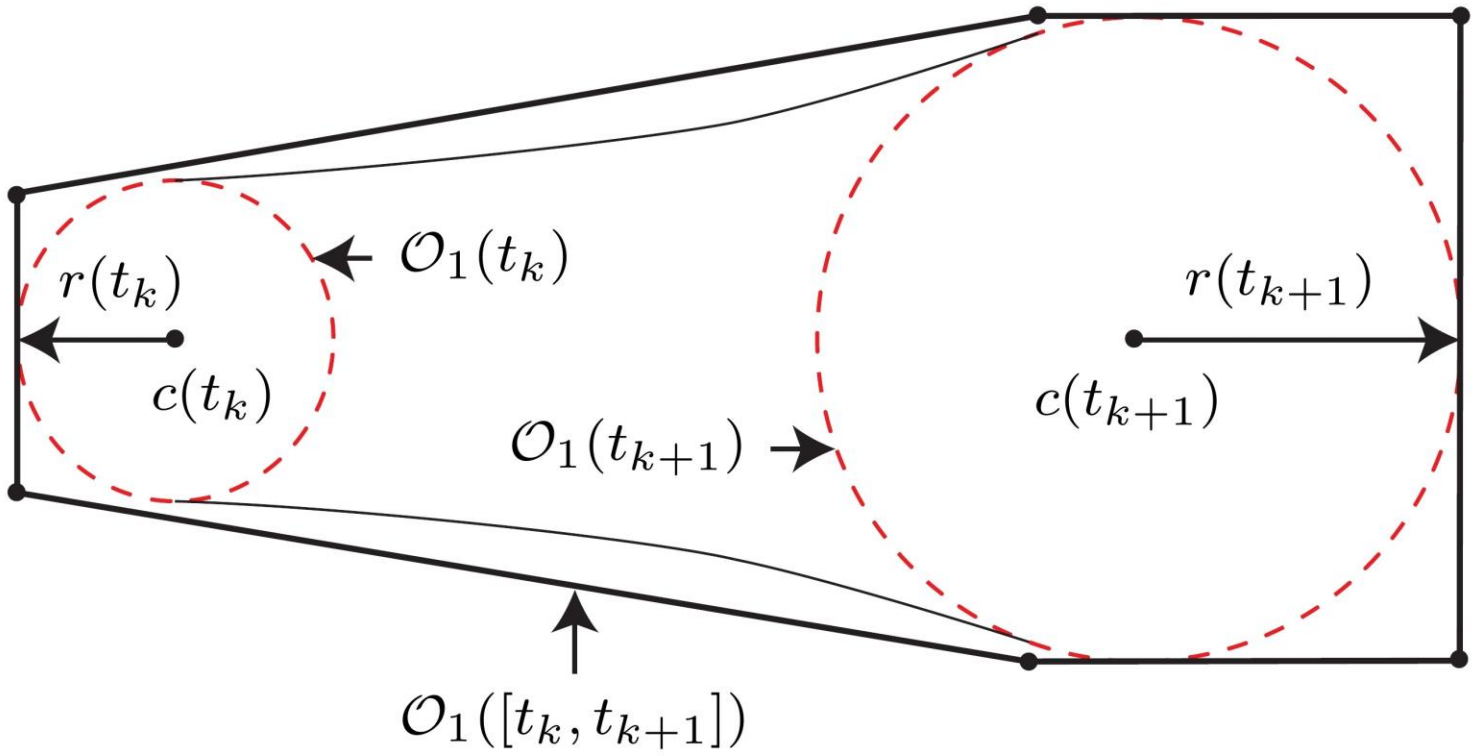


Constraints for Traffic Participants

Constraint	Description
$C_{a_{\max}}$	Maximum acceleration is limited.
$C_{v_{\max}}$	Longitudinal acceleration is stopped when above $v_{\max}$.
C_{engine}	Above a parameterized speed v_S, acceleration is limited.
C_{back}	Driving backwards in a lane is not allowed.
C_{lane}	Changing lanes only allowed if new lane has the same direction.
C_{safe}	Minimum distance to the ego vehicle ξ_{safe}

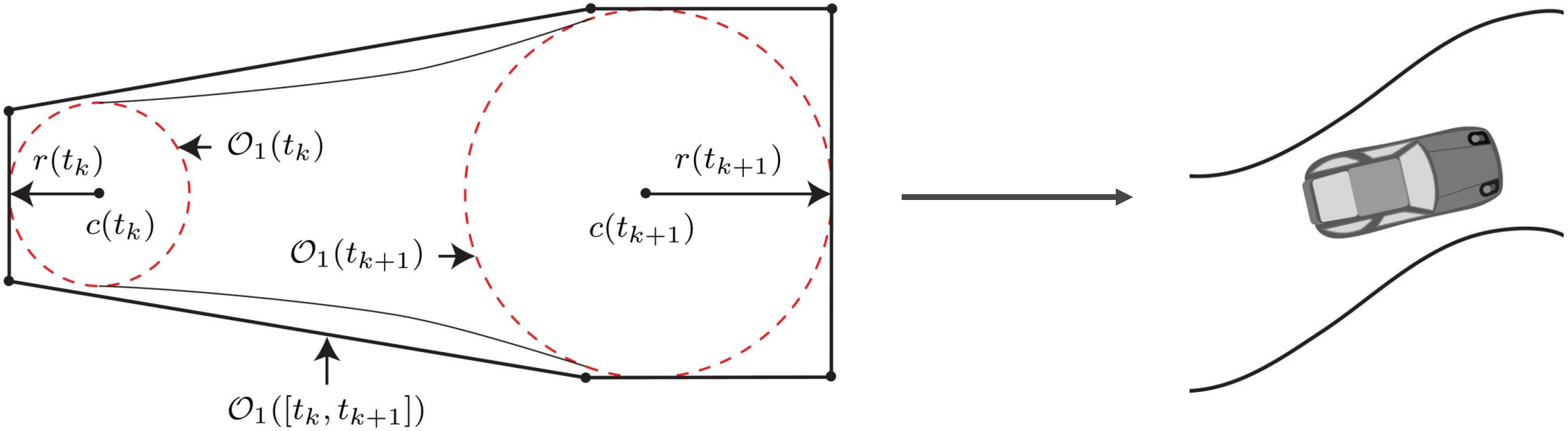
Acceleration-Based Occupancy (M1)

- Considers $C_{a_{max}}$



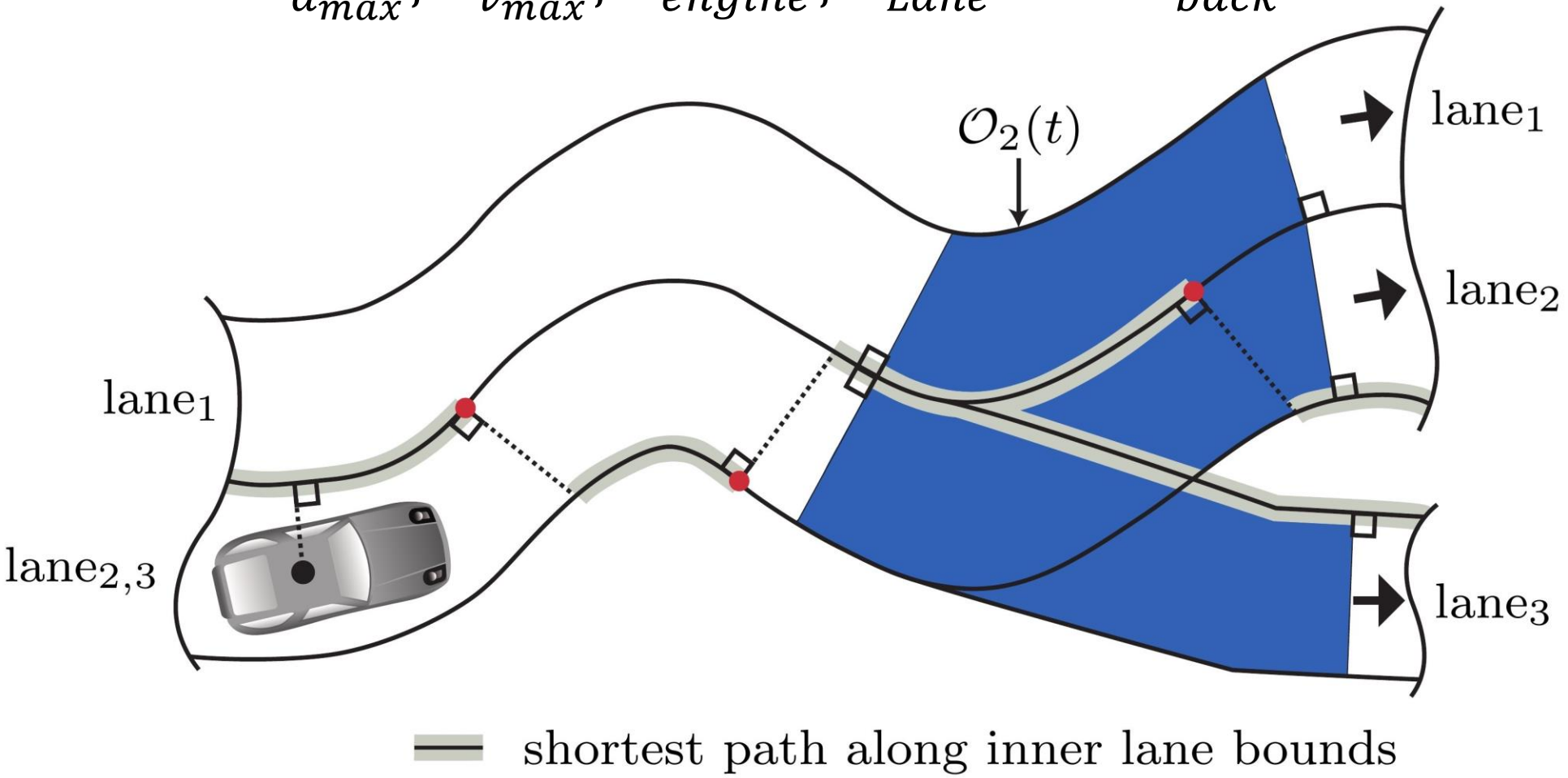
Acceleration-Based Occupancy (M1)

- Add object dimensions, rotate, translate



Lane-Following Occupancy (M2)

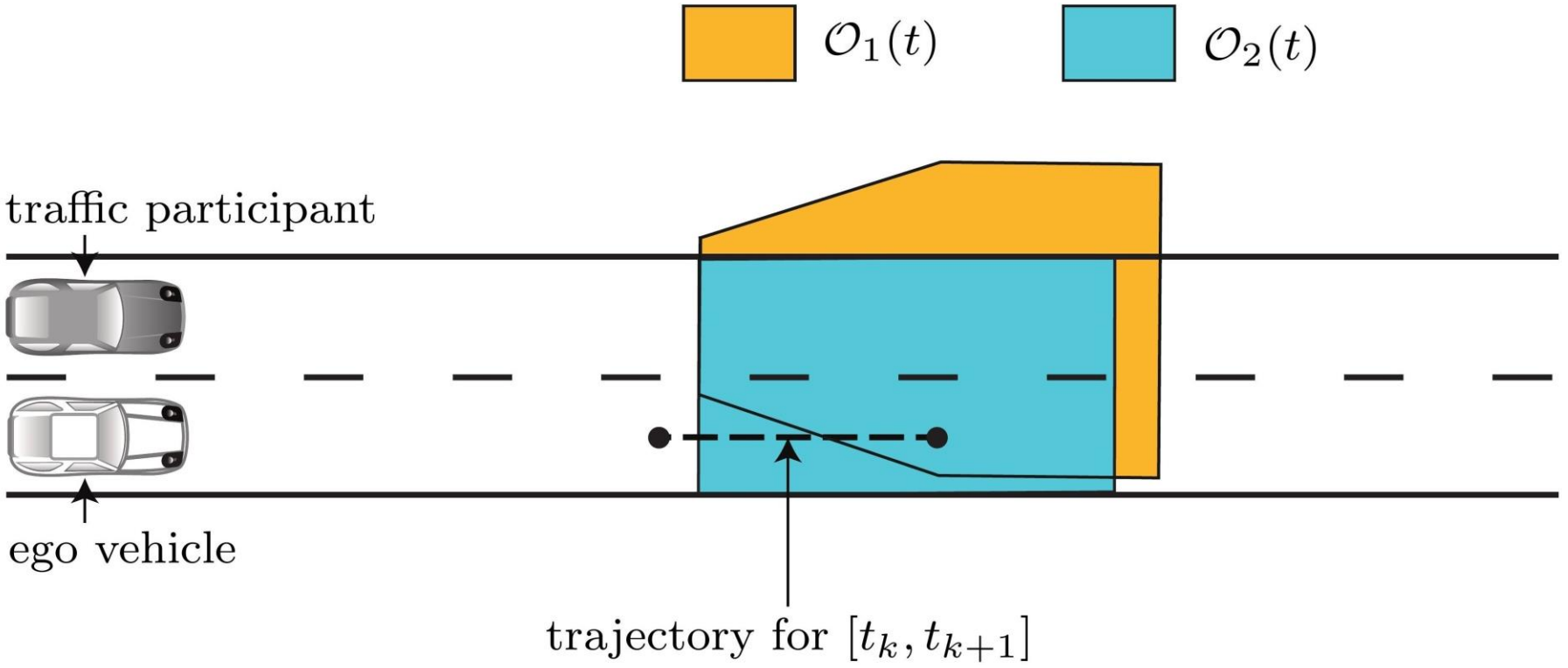
- Considers $C_{a_{max}}$, $C_{v_{max}}$, C_{engine}, C_{Lane} and C_{back}



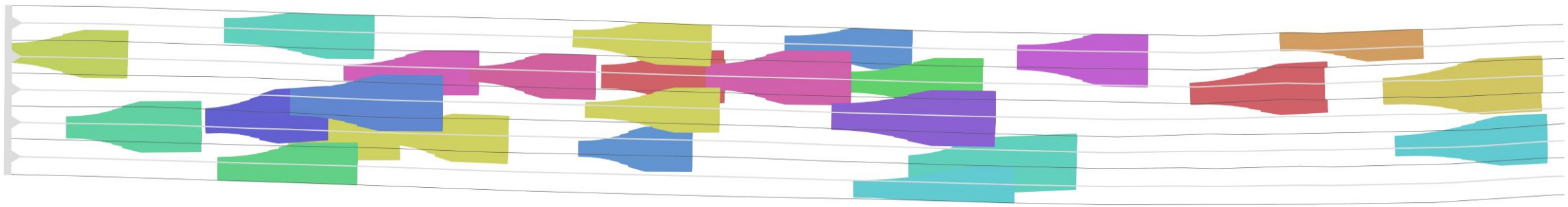
Overall Occupancy

- Intersection of M1 and M2 for every time interval

$$O_1(t) \cap O_2(t)$$

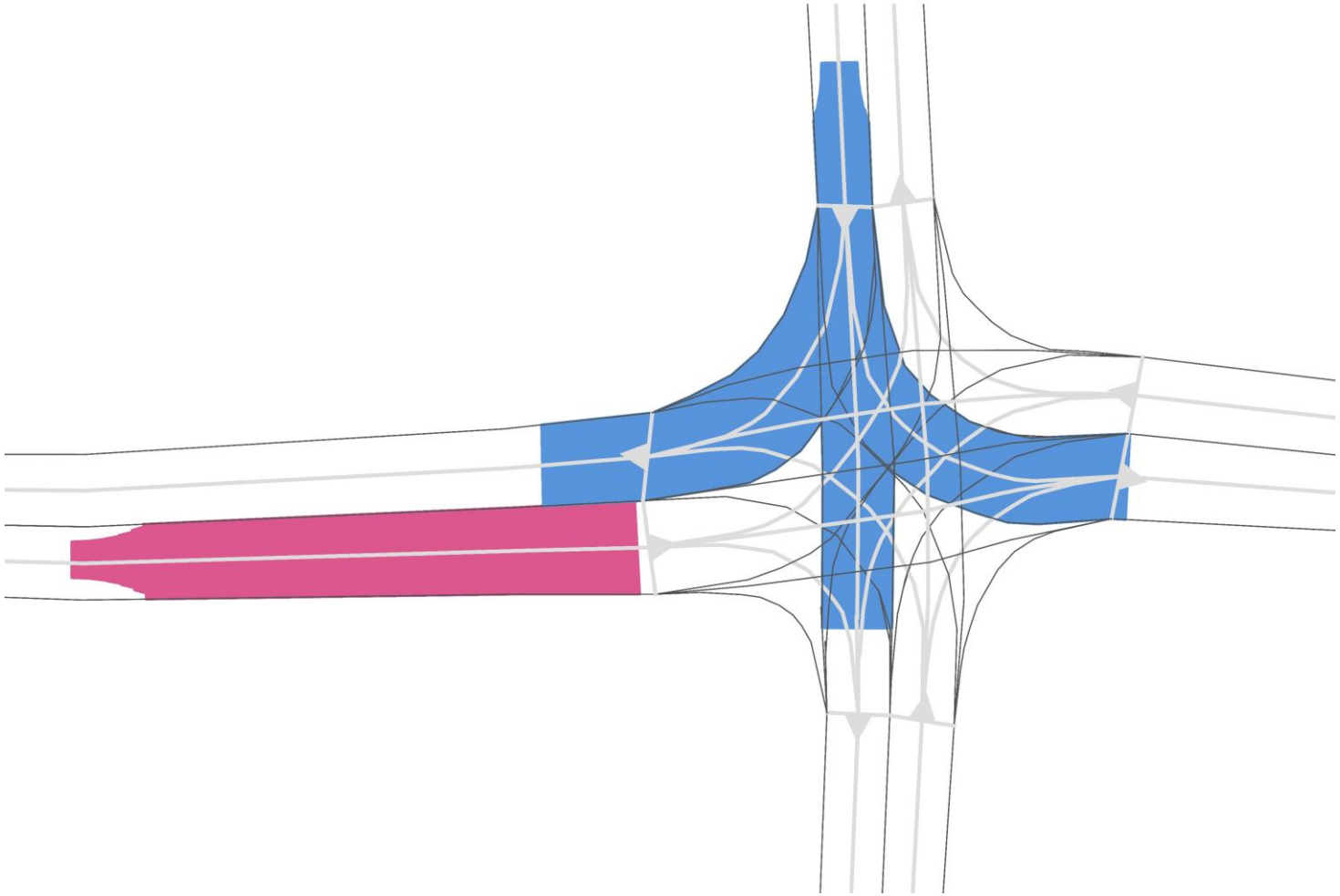


Overall Occupancy – Example 1



Scenario: NGSIM_US101_5

Overall Occupancy – Example 2



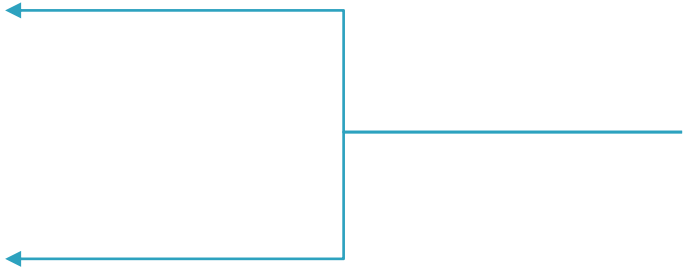
Scenario: GER_Ffb_1

Overview Implementation

1. Get lanelets
2. Construct lanes
3. Calculate shortest path
4. Get obstacles
5. Compute occupancy
   - Acceleration-Based Occupancy (M1)
   - Lane-Following Occupancy (M2)
   - Intersection of M1 and M2
6. Repeat from step 4 with $t + \Delta t$

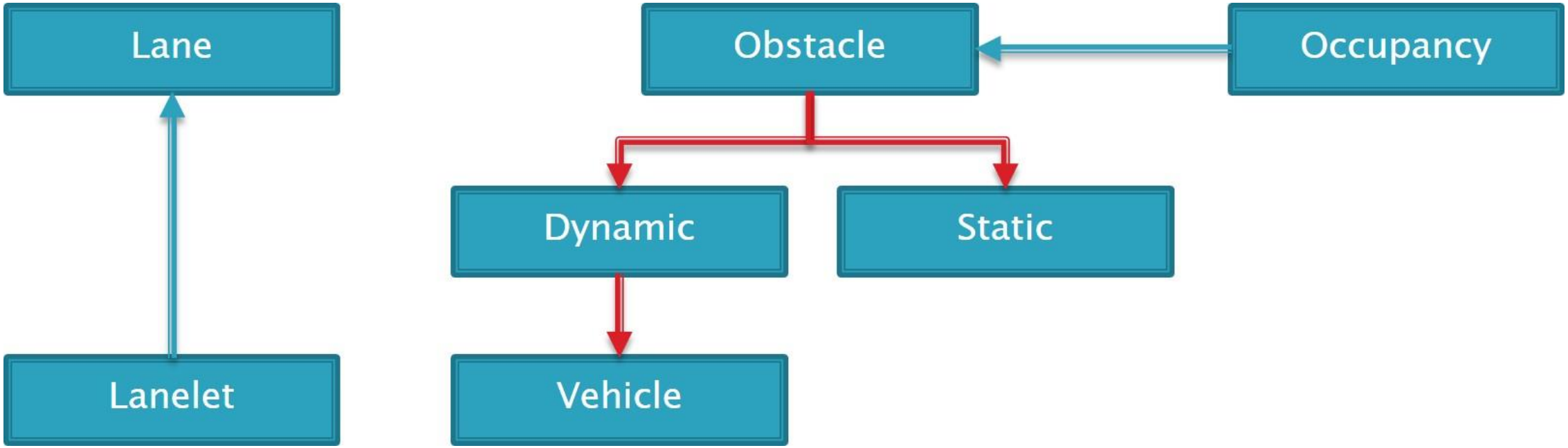
Overview Implementation

1. Get lanelets
2. Construct lanes
3. Calculate shortest path
4. Get obstacles
5. Compute occupancy
   - Acceleration-Based Occupancy (M1)
   - Lane-Following Occupancy (M2)
   - Intersection of M1 and M2
6. Repeat from step 4 with $t + \Delta t$



Parallelize!

Class Diagram

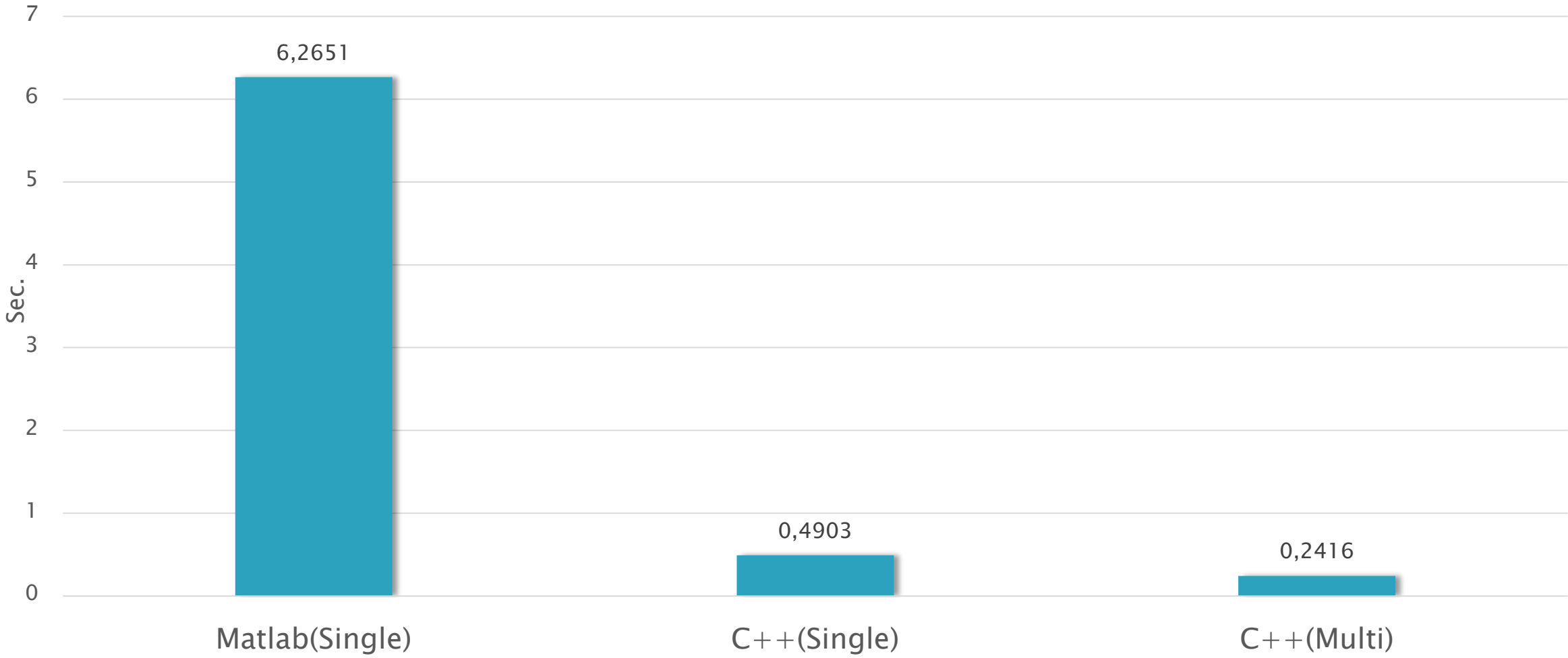


Some Implementation Details

- pugi-xml
- std::vector used for lanelets, lanes and obstacles
- Boost-library:
  - Intersection of polygons
  - Convex-Hull
- OpenMP
  - Parallelization

Benchmark

Scenario: NGSIM_US101_3, 15 cars, 6 Lanes, 2 sec. time horizon, 20 time intervals



Future & Current Work

- Add other traffic participants: cyclists, pedestrians
- Add arbitrary shape of cars
- Shrink Over-Approximations:
  - Reduce velocity in curves
- Optimize Performance
- Integration into ROS

Safe Distance Space $S_3(t)$

- Vienna Convention: One should not endanger another

$$O_1(t) \cap O_2(t) / S_3(t)$$