Autonomous Driving in Urban Centers

Roundabout Monitoring

Julian Scholle

May 18, 2017

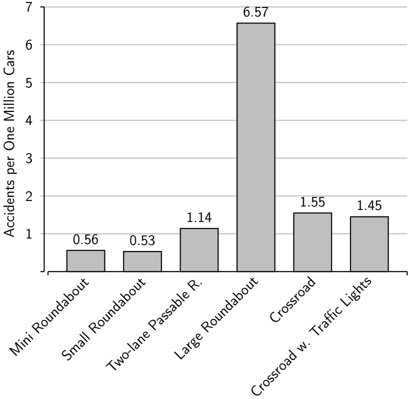
OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FAKULTÄT FÜR
INFORMATIK

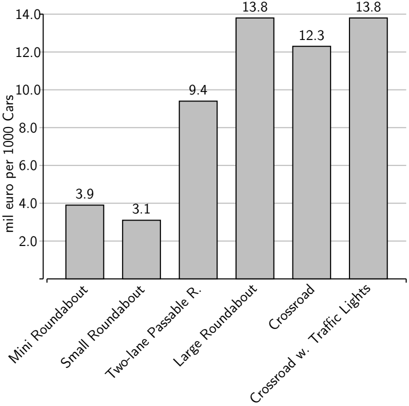
Why Roundabouts

Figure: Accident Rate in City Limits



Why Roundabouts

Figure: Accident Charge Rate in City Limits

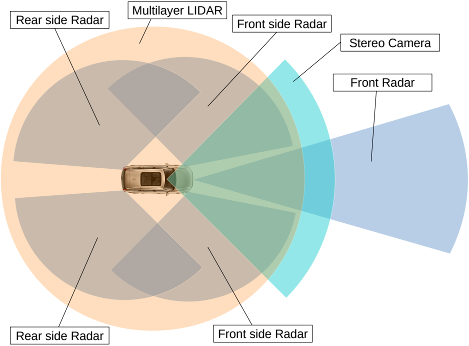


Test Platform

Figure: Snowfox

Test Platform

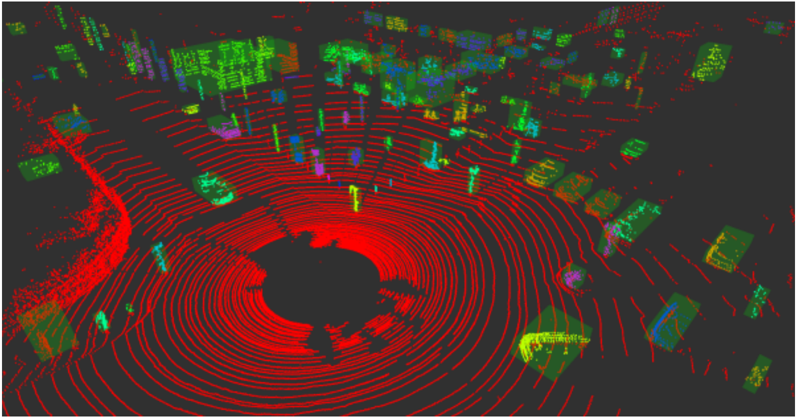
Figure: Snowfox Sensors



Main Problems

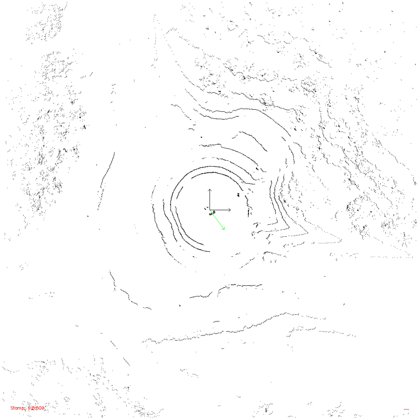
- Object Detection
  - Segmentation
  - Tracking
  - Classification
- Simulation
  - Scenario
  - Logic
- Evaluation
  - Simulation
  - Real Measurements
  - Performance

Segmentation



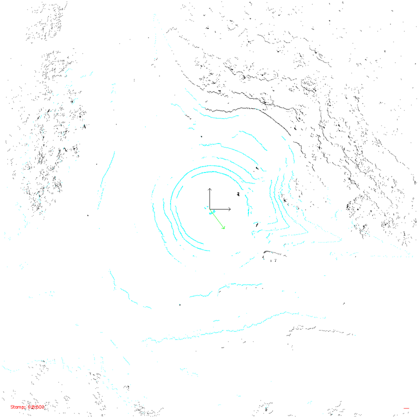
Segmentation - Ground Removal

Figure: Unsegmented Data



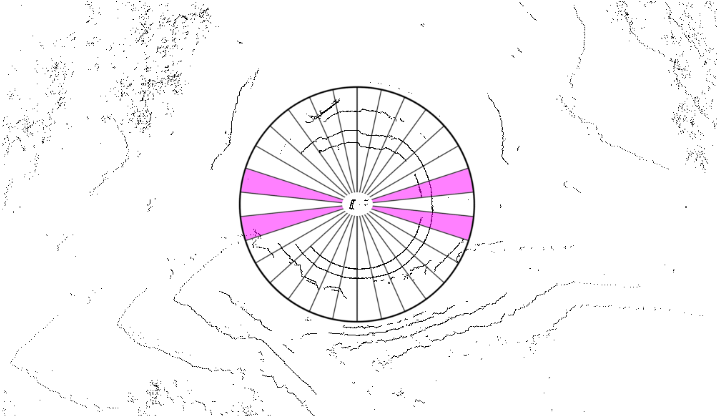
Segmentation - Ground Removal

Figure: Segmented Data



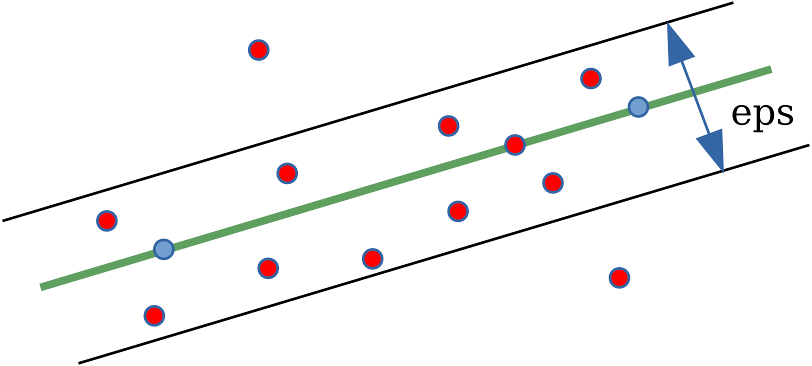
Segmentation - Ground Removal

Figure: LiDAR Segments



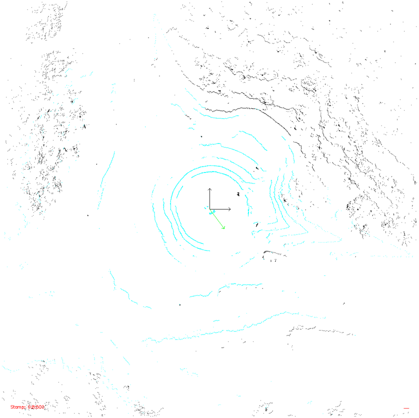
Segmentation - Ground Removal

Figure: RANSAC - Random Sample Consensus



Segmentation - Clustering

Figure: Segmented Data



Segmentation - Clustering

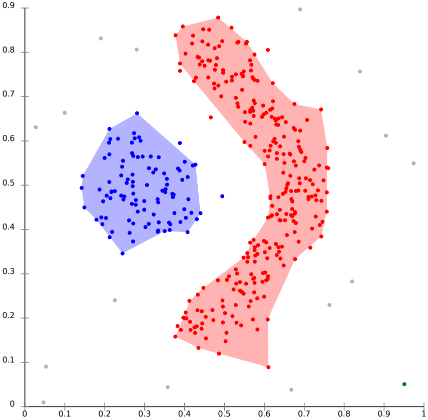
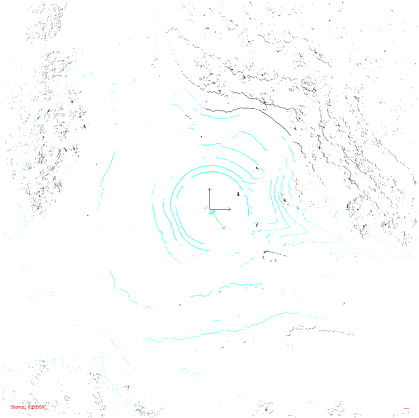


Figure: DBSCAN - Density-Based Spatial Clustering with Noise

Tracking

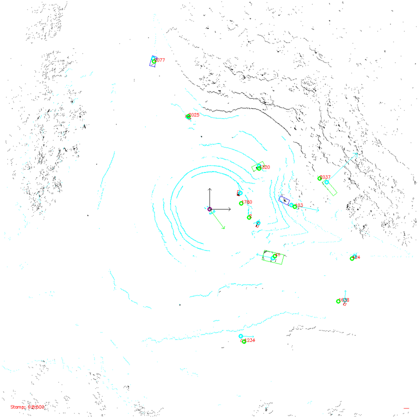
Tracking

Figure: Segmented Data



Tracking

Figure: Segmentation - Segmentation



Tracking - Tracking of simplified Clusters

- clusters are defined through there mean center point
- tracking is done through a simple distance criteria

Tracking - Tracking of simplified Clusters

- now, we can track the Objects from timestep to timestep
- but.. we need more information
  - direction and speed of the object movement
  - prediction of the movement

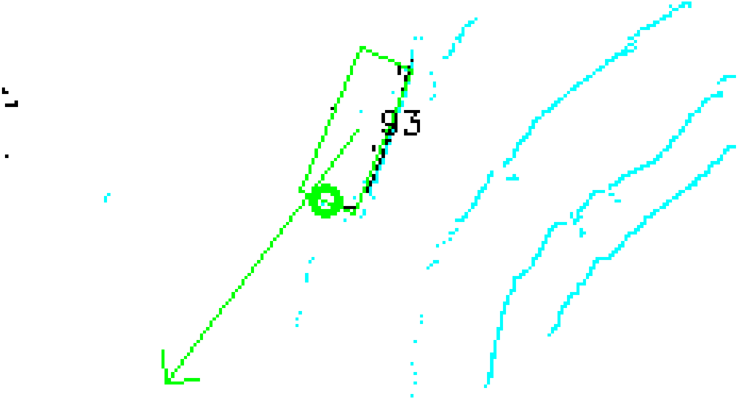
Tracking - Tracking of simplified Clusters

- calculation oth the movement through:

$$\begin{aligned}\Delta x &= P_x(t) - P_x(t_{-2m}) + \Delta C_x\\ \Delta y &= P_y(t) - P_y(t_{-2m}) + \Delta C_y\\ \theta &= \mathrm{atan2}(\Delta y, \Delta x)\end{aligned}$$

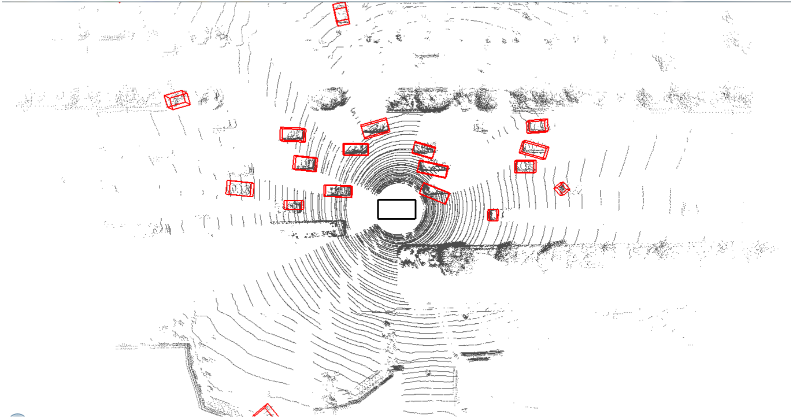
Tracking - Tracking of simplified Clusters

Figure: Direction of movement



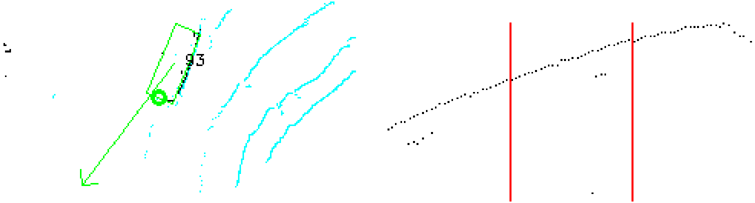
Tracking - How to Calculate Bounding Boxes

Figure: Minimum Bounding Box



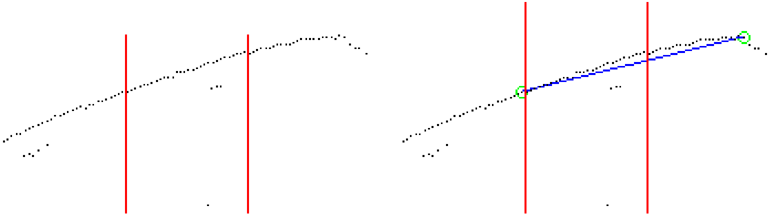
Tracking - How to Calculate Bounding Boxes

Figure: Object Rotation and Division



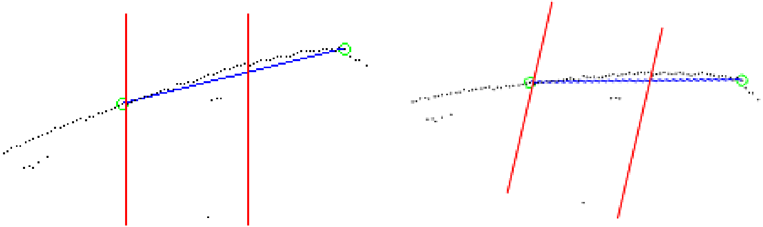
Tracking - How to Calculate Bounding Boxes

Figure: Maximize Y-Values



Tracking - How to Calculate Bounding Boxes

Figure: Calculating Correction



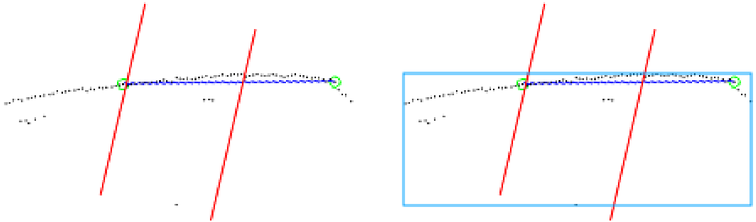
$$\Delta x = R_x - L_x$$

$$\Delta y = R_y - L_y$$

$$\theta_{correction} = \operatorname{atan2}(\Delta y, \Delta x)$$

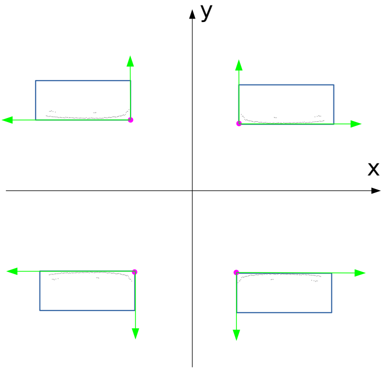
Tracking - Size calculation

Figure: Size Calculation



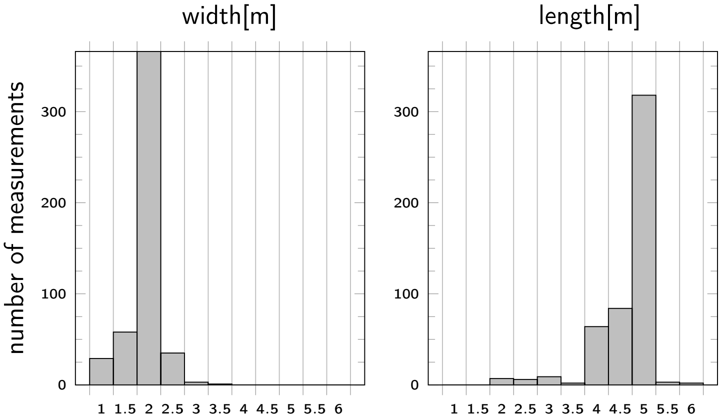
Tracking - Size calculation

Figure: Bounding Box Calculation Cases



Tracking - Size calculation

Figure: Object Size Histogram



Tracking - Confidence

Increase confidence value by one, if the object could be tracked and

- The width of the object is less than the length of the obstacle plus 1.5m
- The length of the obstacle is less than 10m
- The width of the obstacle less less than 4m

and halved if not

Classification

Using simple size based approach

pedestrian: length $<$ 1.5 m and width $<$ 1.5 m

cyclist: length $<$ 2 m and width $<$ 1.5 m

car: length $<$ 10 m and width $<$ 4 m

undefined: length $>=$ 10 m and width $>=$ 4 m

State Estimation

using constant turn rate and velocity model and extendet Kalman filter

$$\vec{x}(t) = \begin{bmatrix} x & y & \theta & v & \omega \end{bmatrix}^T$$

x - x-Axis

y - y-Axis

θ - Object Yaw Angle

v - Object Velocity

ω - Yaw Rate

State Estimation - Extendet Kalman Filter

$$f = \vec{x}(t+\Delta t) = \begin{bmatrix} \frac{v}{\omega}(-\sin(\theta) + \sin(\Delta t\omega + \theta)) + x(t) \\ \frac{v}{\omega}(\cos(\theta) - cos(\Delta t\omega + \theta)) + y(t) \\ \omega\Delta t + \theta \\ v \\ \omega \end{bmatrix}$$

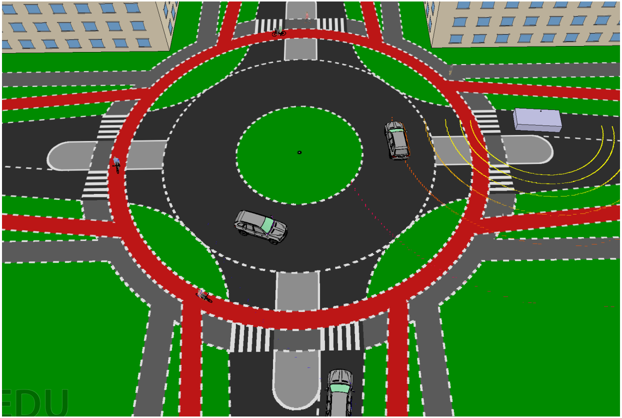
- a lot more equations..
- hacks for numerical stability (yawrate near zero)
- singularity in rotation (because $[-180^\circ \leq \theta \leq 180^\circ]$)

State Estimation - Extendet Kalman Filter

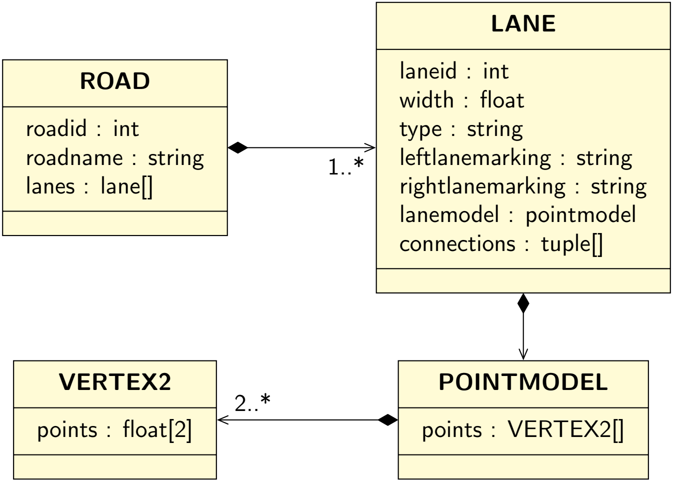
- filtering the data
- prediction of the future position if an object could not be detected in a timestep
- reassigning in case of redetection, using the predicted position

Simulation

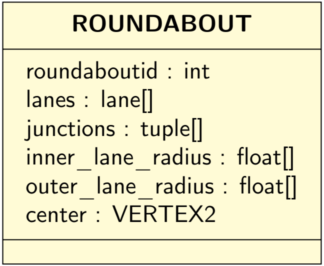
Figure: Simulation - Scenario



Mapping


ROAD
roadid : int
roadname : string
lanes : lane[]
1..*
LANE
laneid : int
width : float
type : string
leftlanemarking : string
rightlanemarking : string
lanemodel : pointmodel
connections : tuple[]
VERTEX2
points : float[2]
2..*
POINTMODEL
points : VERTEX2[]

Mapping

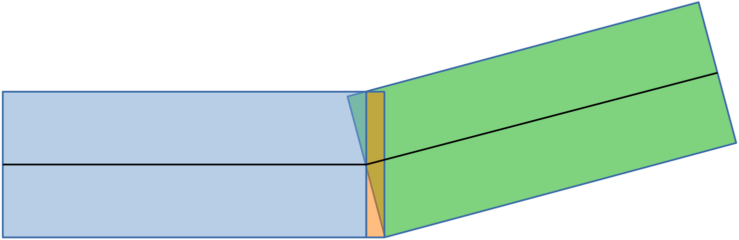


ROUNDABOUT

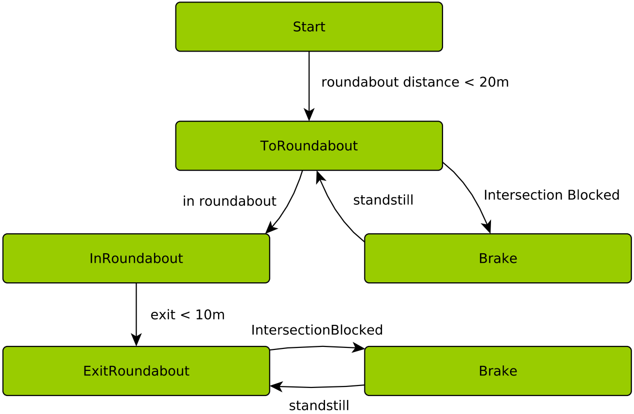
roundaboutid : int
lanes : lane[]
junctions : tuple[]
inner_lane_radius : float[]
outer_lane_radius : float[]
center : VERTEX2

Mapping

Figure: Mapping - Lanesegment

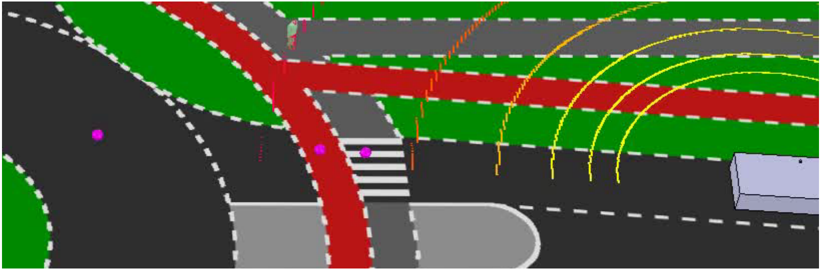


State Machine



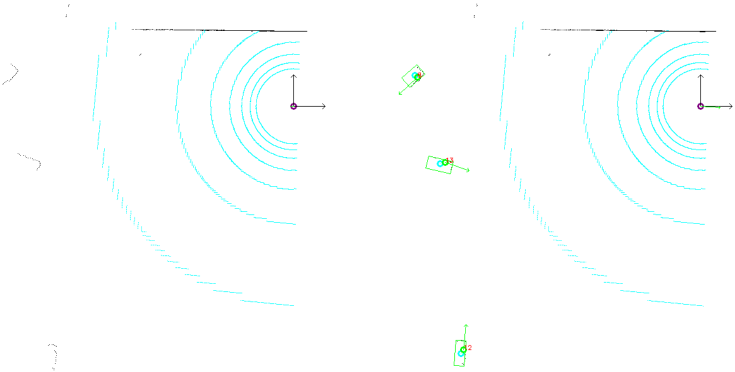
Simulation

Figure: Simulation - Intersection Position



Evaluation

Figure: Detection Distance Performance

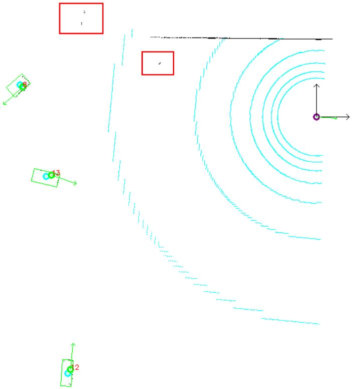


Timestep 1

Timestep 2

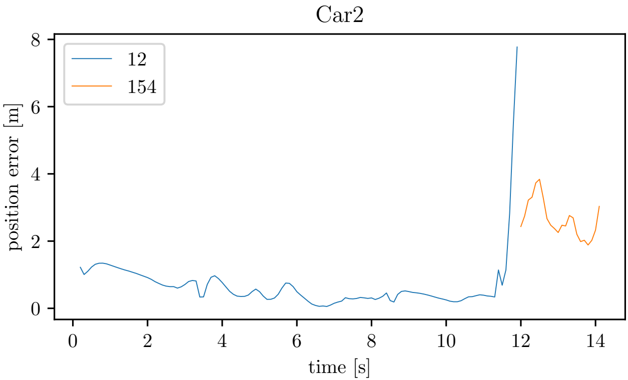
Evaluation

Figure: Detection Distance Performance Pedestrians/Cyclists



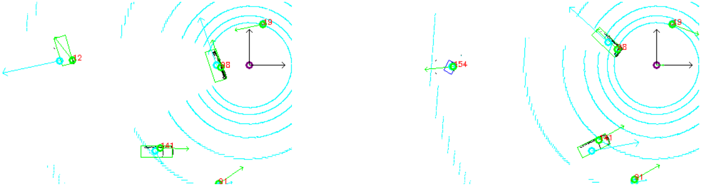
Evaluation

Figure: Car Position Error



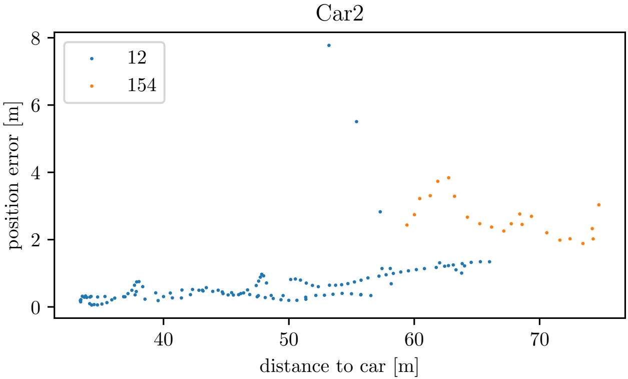
Evaluation

Figure: Tracking Error



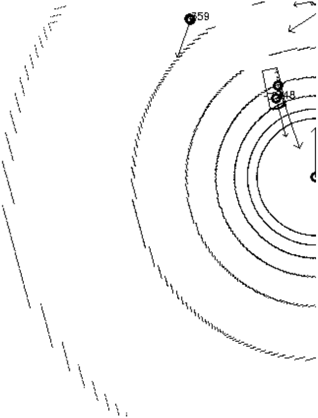
Evaluation

Figure: Car Position Error / Distance



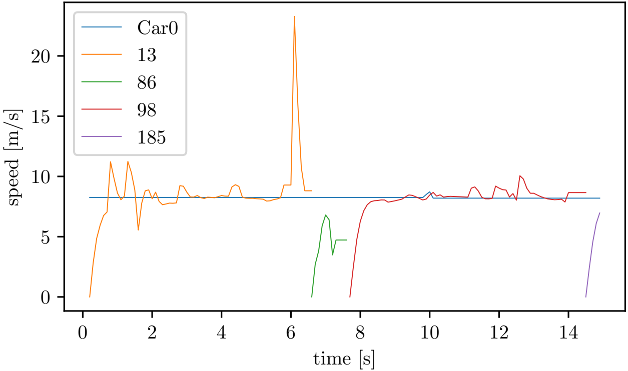
Evaluation

Figure: Simulation Sensor Resolution



Evaluation

Figure: Speed



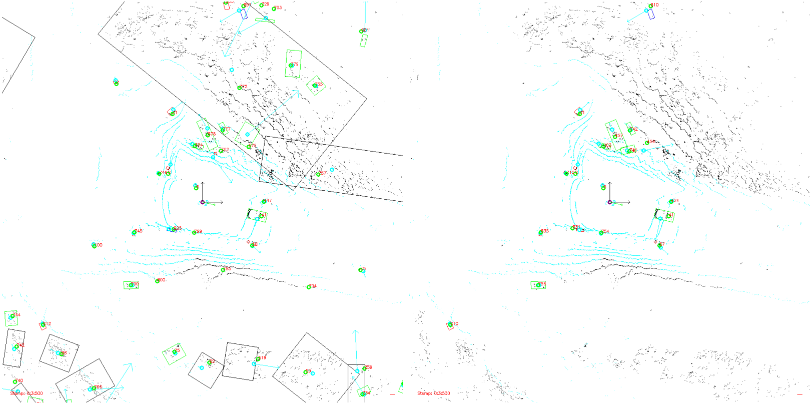
Evaluation

Figure: Classification

Name	Car [%]	Bike [%]	Pedestrian [%]	UnCl [%]
Pedestrian	0	0	100	0
Car0	100	0	0	0
Car1	100	0	0	0
Car2	98.2	0.73	0.98	0
Bike0	0	0.5	99.5	0
Bike1	0	0	100	0
Bike2	1.9	1.2	96.9	0

Evaluation

Figure: Confidence Filtering

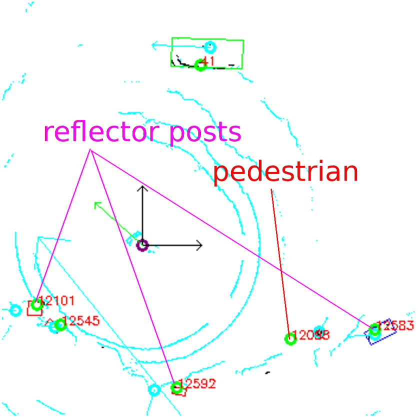


without filtering

with filtering

Evaluation

Figure: Reflector Posts

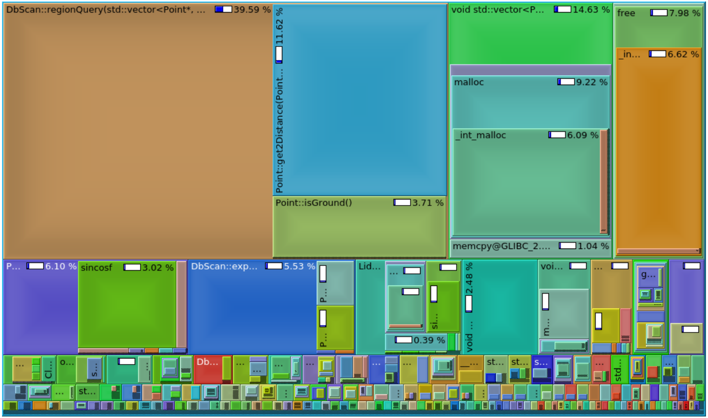


reflector posts

pedestrian

Evaluation

Figure: Performance



Autonomous Driving in Urban Centers

Roundabout Monitoring

Julian Scholle

May 18, 2017

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FAKULTÄT FÜR
INFORMATIK