Recursive state estimation for lane detection using a fusion of cooperative and map based data

IEEE Intelligent Transportation Systems Conference 2015

Bernd Schäufele, Daimler Center for Automotive IT Innovations

Oliver Sawade, Fraunhofer FOKUS

Patrick Lorenz, Fraunhofer FOKUS

Dr. Ilja Radusch, Daimler Center for Automotive IT Innovations

Agenda

- Motivation
- Approach
- Hypotheses generation
- Probabilty deduction
- Results

Motivation

- Many ADAS require lane-level accuracy positioning

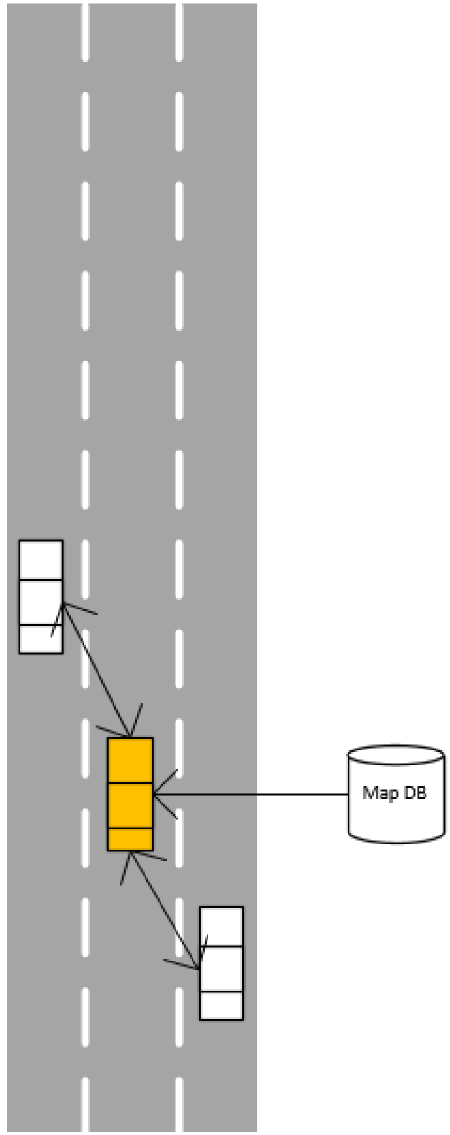
- Plain GNSS not precise enough
    - app. 10 meters

- In-vehicle sensors limited
    - Camera: Deterioration (e.g. snow)
    - Radar: 30 m

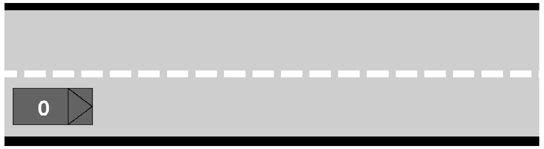
- Usage of GNSS characteristics
    - Limited precision of ephemeris and clock data
    - Signal propagation errors by atmospheric effects
    - Errors common for receivers in proximity

Approach

- Use cooperative positioning
- Fuse with map data
- Discrete Bayes Filter
- Recursive state estimation
- Generate lane hypotheses
- Deduct lane probability from orientation and perpendicular distance



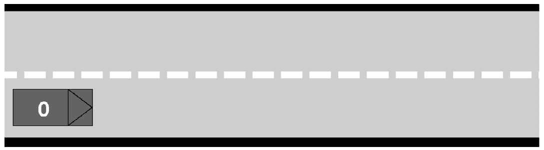
Hypotheses generation

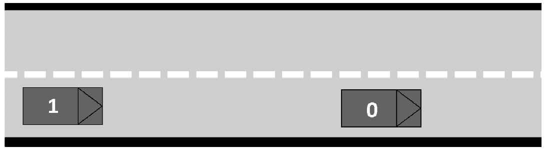


$$\mathbf{H} = \begin{Bmatrix} h_0 = ((v_0 \rightarrow l_0)\,,\; 0.5)\,, \\ h_1 = ((v_0 \rightarrow l_1)\,,\; 0.5) \end{Bmatrix}$$

- Fusion with map data in each step
- Hypotheses H consisting of hypothesis h_n
- Potential lane assignments, probability annotated

Hypotheses generation



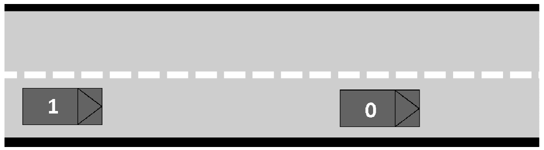


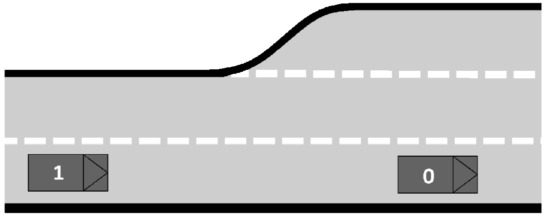
$$\mathbf{H} = \begin{Bmatrix} h_0 = ((v_0 \rightarrow l_0)\,,\ 0.5)\,, \\ h_1 = ((v_0 \rightarrow l_1)\,,\ 0.5) \end{Bmatrix}$$

$$\mathbf{H} = \begin{Bmatrix} h_0 = ((v_0 \rightarrow l_0,\ v_1 \rightarrow l_0)\,,\ 0.25)\,, \\ h_1 = ((v_0 \rightarrow l_0,\ v_1 \rightarrow l_1)\,,\ 0.25)\,, \\ h_2 = ((v_0 \rightarrow l_1,\ v_1 \rightarrow l_0)\,,\ 0.25)\,, \\ h_3 = ((v_0 \rightarrow l_1,\ v_1 \rightarrow l_1)\,,\ 0.25) \end{Bmatrix}$$

- New vehicle in proximity
- Distribute probability equally

Hypotheses generation



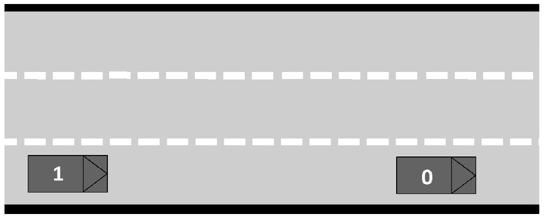


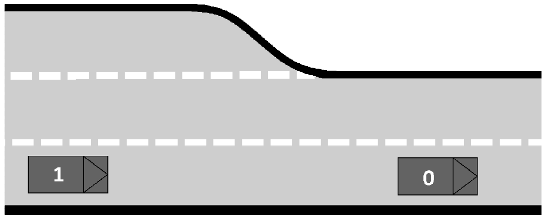
$$\mathbf{H} = \begin{Bmatrix} h_0 = ((v_0 \to l_0,\ v_1 \to l_0),\ 0.7), \\ h_1 = ((v_0 \to l_0,\ v_1 \to l_1),\ 0.1), \\ h_2 = ((v_0 \to l_1,\ v_1 \to l_0),\ 0.1), \\ h_3 = ((v_0 \to l_1,\ v_1 \to l_1),\ 0.1) \end{Bmatrix}$$

$$\mathbf{H} = \begin{Bmatrix} h_0 = ((v_0 \to l_0,\ v_1 \to l_0),\ 0.4667), \\ h_1 = ((v_0 \to l_0,\ v_1 \to l_1),\ 0.0667), \\ h_2 = ((v_0 \to l_1,\ v_1 \to l_0),\ 0.0667), \\ h_3 = ((v_0 \to l_1,\ v_1 \to l_1),\ 0.0667), \\ h_4 = ((v_0 \to l_2,\ v_1 \to l_0),\ 0.1667), \\ h_5 = ((v_0 \to l_2,\ v_1 \to l_1),\ 0.1667) \end{Bmatrix}$$

- New lane available for ego vehicle
- Expand state space
- Distribute probability of new assignments equally
- Normalize previous assignments

Hypotheses generation





$$\mathbf{H} = \left\{\begin{array}{l} h_0 = ((v_0 \to l_0,\ v_1 \to l_0),\ 0.3111), \\ h_1 = ((v_0 \to l_0,\ v_1 \to l_1),\ 0.0444), \\ h_2 = ((v_0 \to l_0,\ v_1 \to l_2),\ 0.1111), \\ h_3 = ((v_0 \to l_1,\ v_1 \to l_0),\ 0.0444), \\ h_4 = ((v_0 \to l_1,\ v_1 \to l_1),\ 0.0444), \\ h_5 = ((v_0 \to l_1,\ v_1 \to l_2),\ 0.1111), \\ h_6 = ((v_0 \to l_2,\ v_1 \to l_0),\ 0.1111), \\ h_7 = ((v_0 \to l_2,\ v_1 \to l_1),\ 0.1111), \\ h_8 = ((v_0 \to l_2,\ v_1 \to l_2),\ 0.1111) \end{array}\right\}$$

$$\mathbf{H} = \left\{\begin{array}{l} h_0 = ((v_0 \to l_0,\ v_1 \to l_0),\ 0.4083), \\ h_1 = ((v_0 \to l_0,\ v_1 \to l_1),\ 0.1000), \\ h_2 = ((v_0 \to l_0,\ v_1 \to l_2),\ 0.1667), \\ h_3 = ((v_0 \to l_1,\ v_1 \to l_0),\ 0.0583), \\ h_4 = ((v_0 \to l_1,\ v_1 \to l_1),\ 0.1000), \\ h_5 = ((v_0 \to l_1,\ v_1 \to l_2),\ 0.1667) \end{array}\right\}$$

- Lane disappears for ego vehicle
- Shrink state space
- Distribute probability of reduced states to hypothesis with same remaining assignments

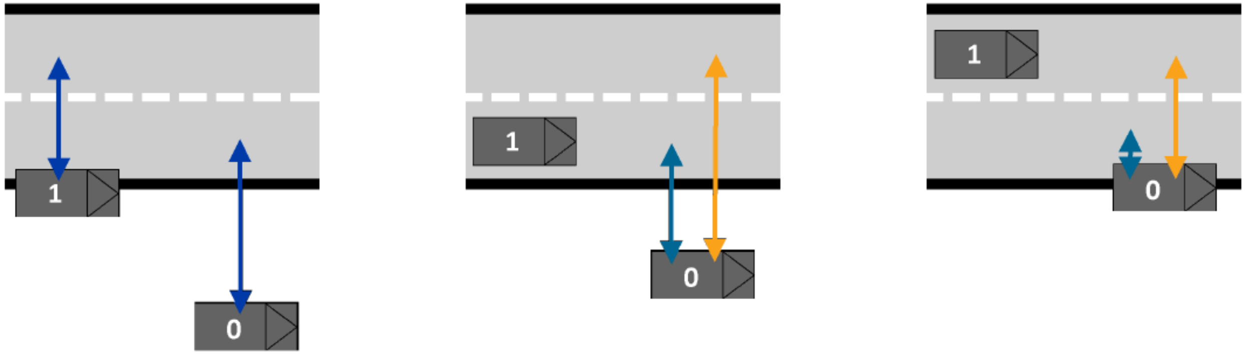
Probability deduction

- Discrete Bayes Filter

- Prediction
  - Based on heading of vehicle and street
  - Angle of vehicle and lane similar:
    - Low probability for lane change
  - Angle of vehicle yields towards another lane:
    - High probability for lane change

Probability deduction

- Discrete Bayes Filter



$v_1 \rightarrow l_1$	$v_1 \rightarrow l_0$	$v_1 \rightarrow l_1$
$v_0 \rightarrow l_0$	$v_0 \rightarrow l_0 \; l_1$	$v_0 \rightarrow l_0 \; l_1$

- Update
  - Take reference vehicle (v_1) and translate it to lane according to a hypothesis
  - Move all other vehicles by the same translation
  - Translate all other vehicles according to the hypothesis
  - Sum of translations of other vehicles influences probability of hypothesis
  - Probabilities summed up per vehicle

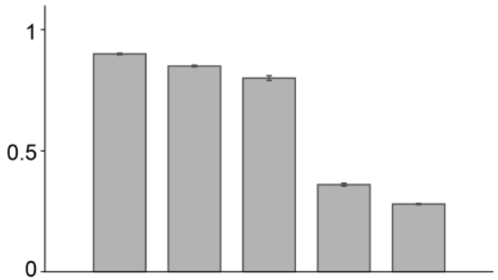
Evaluation

- Simulated GNSS errors
- Gaussian bias and Gaussian receiver error

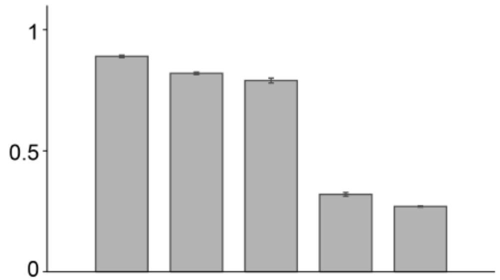
Configuration	$\sigma_{\mathbf{bias}}$	$\max_{\mathbf{bias}}$	$\sigma_{\mathbf{rec}}$	$\max_{\mathbf{rec}}$
Bias Gauss 1	8	15	0.5	2
Bias Gauss 2	8	15	1	4
Bias Gauss 3	8	15	2	6
Bias Gauss 4	8	15	4	10
Bias Gauss 5	8	15	6	12

Results

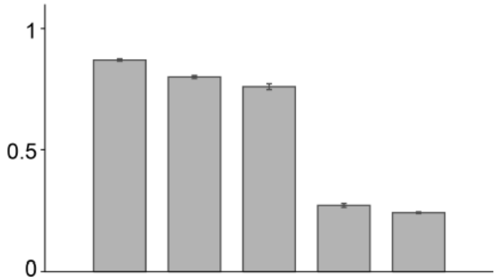
- Test with different scenarios, different Gaussian errors and different thresholds



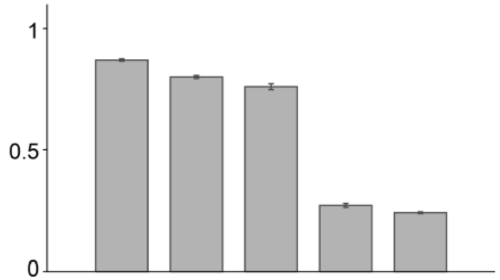
(a) $threshold = 80\%$



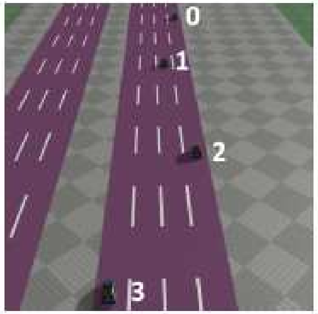
(b) $threshold = 85\%$



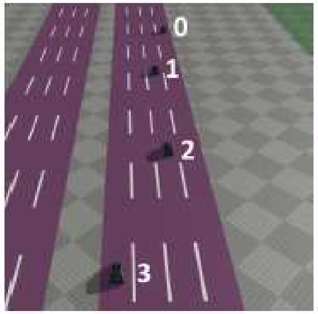
(c) $threshold = 90\%$



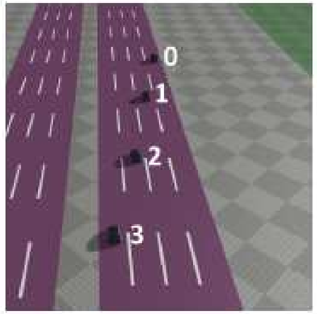
(d) $threshold = 95\%$



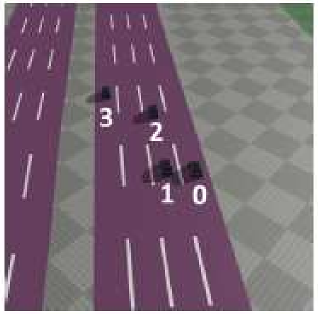
(a) $t_0 \leq t \leq t_1$



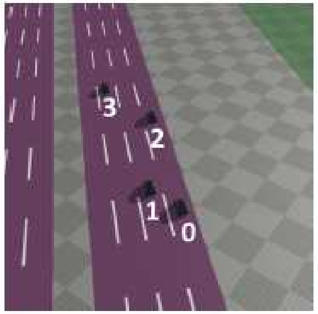
(b) $t_1 < t \leq t_2$



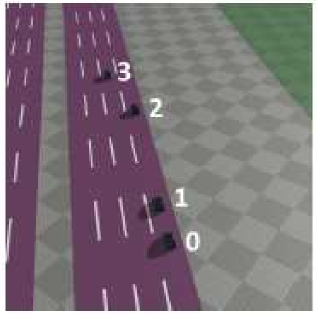
(c) $t_2 < t \leq t_3$



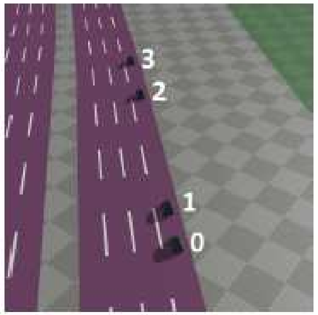
(d) $t_3 < t \leq t4$



(e) $t_4 < t \leq t_5$



(f) $t_5 < t \leq t_6$



(g) $t_6 < t \leq d$

Conclusion

- Lane detection based on Cooperative Positioning fused with map data
- Discrete Bayes Filter
- Hypotheses generation based on map information
- Prediction based on vehicle orientation
- Update based on relative vehicle positions
- Evaluated with different scenarios and configurations
    - Over 80% reliability with GNSS receiver error of up to 6 meters