Ego-Lane Estimation by Modeling Lanes and Sensor Failures

HIGHWAY-LIKE SCENARIOS

A.L. Ballardini, D. Cattaneo, D.G. Sorrenti, Università degli Studi di Milano - Bicocca, Italy
R. Izquierdo, I. Parra, M.A. Sotelo, University of Alcalá, Alcalá de Henares, Spain

Tuesday, October 17th, 2017
Proposed Approach Overview

- **Given** a rough localization estimate (road segment)
- **Leverage** a line detector & tracker (we do not investigate this topic)
- **Exploit** a Hidden Markov Model with a transient failure model
- **Infer**: vehicle’s ego-lane in a probabilistic fashion
In an ideal scenario, we could directly infer the vehicle ego-lane from the road lines.

Possible issues:
- Cluttering elements (other cars)
- Faded Road Markings
- Illumination Issues
Limit the uncertainties

Given partial line detections, we can exploit their distances (lateral offset *wrt* vehicle) to limit the ego-lane uncertainty.

Only leftmost line detected due to illumination issues

Lanes compatible *w.r.t.* line distance
01
Exploit *consecutive* yet partial observations *over time*

02
Exploit a HMM approach with \(2n\)-states corresponding to the \(n\)-lanes

03
“\(n\)” is the number of lanes retrieved from OpenStreetMap

Track the Vehicle ego-lane
Line Detector & Tracker Interface

output = {  line_1, <property_{1...k}>  
            line_2, <property_{1...k}>  
            ...  }

1. Find Contours in image (BEV)
2. Fit Geometric Primitives on Contours (clothoids, polylines...)
3. Track Geometric Primitives
   1. lateral offset from vehicle [m]
   2. is line continuous? (y/n)
Temporal Line Reliability

Reliability Index

line detection ratio over the last k-frames

We add the Reliability Index and the isValid properties to the Line properties vector.
These results can be used to infer the ego-lane using the simple geometric considerations shown before.
The HMM Model

State Definition

HMM ( n, \( \sigma_1, \sigma_2, P_1, P_2, BV, w \) )

- \( n \) number of lanes, retrieved from OpenStreetMap
- \( \sigma_1, \sigma_2 \) parameters used for the lane transition model generation
- \( P_1, P_2 \) How likely to stay in SensorOK/SensorBad state (prediction phase)
- \( BV \) Bonus Value for continuous lines (gives richer information)
- \( w \) Inertia used in the Sensor Matrix to propagate the SensorBad state

The HMM allow us to represent and track over time the probability of being on each lane and having either a properly operating or a faulty Sensor

\( X_t = \langle \text{Lane}_1 \text{SensorOK} \ldots \text{Lane}_n \text{SensorOK} , \text{Lane}_1 \text{SensorBad} \ldots \text{Lane}_n \text{SensorBad} \rangle \)

\( |X_t| = |\text{Sensor}| * |\text{ego\_lane}| \)
The HMM Tracking Model

State Prediction

\[ X_t = \langle \text{Lane}_1...\text{n SensorOK} ; \text{Lane}_1...\text{n SensorBad} \rangle \]

\[ X_{t+1} = X_t \cdot \text{StateTransitionMatrix} \]

The State Transition Matrix is built using a Basic Transition Matrix, which in turn is based on the following consideration:

the probability of moving to another lane is normally distributed, the average is on the current lane, \( \sigma \) is a model parameter
The HMM Tracking Model

State Prediction

\[ X_{t+1} = X_t \cdot \text{StateTransitionMatrix} \]

\[ STM_{2n \times 2n} = \begin{pmatrix} A_{n \times n} & B_{n \times n} \\ C_{n \times n} & D_{n \times n} \end{pmatrix} \]

- A → (Lane Transition from SensorOK to SensorOK) * \( P_1 \)
- B → (Lane Transition from SensorOK to SensorBad) * \( (1-P_1) \)
- C → (Lane Transition from SensorBad to SensorOK) * \( (1-P_2) \)
- D → (Lane Transition from SensorBad to SensorBad) * \( P_2 \)
The HMM Tracking Model

State Prediction

\[ X_{t+1} = X_t \cdot \text{StateTransitionMatrix} \]

- A → (Lane Transition from SensorOK to SensorOK) \(* P_1\)
- B → (Lane Transition from SensorOK to SensorBad) \(* (1-P_1)\)
- C → (Lane Transition from SensorBad to SensorOk) \(* (1-P_2)\)
- D → (Lane Transition from SensorBad to SensorBad) \(* P_2\)

Idea:
the sensor mainly gives long runs of correct outputs, so \( P_1 \) is “large” (and \( 1-P_1 \) is “small”) when the sensor makes mistakes for a short period of time, so \( 1-P_2 \) is “large” (and \( P_2 \) is “small”)
Updating the Belief 1

Counting Scheme

To update the belief exploiting the output of the Line Detector & Tracker we use an ad-hoc sensor model which uses the Line Properties 
(LateralOffset; ReliabilityIndex; isValid; Continuous)

\[ P = \{ \frac{3}{9}, \frac{3}{9}, \frac{2}{9}, \frac{1}{9} \} \]
Updating the Belief 2

Exploiting the Sensor Reliability

To update the belief exploiting the output of the Line Detector & Tracker we use an ad-hoc sensor model which uses the Line Properties \((\text{Lateral Offset} ; \text{Reliability Index} ; \text{isContinuous})\)

\[
SensorScoreOK = \frac{\sum_{i=1}^{m} \text{isValid}_i \times RI_i}{10 \times (n + 1)} = \frac{9 + 8 + 5}{50} = 0.44
\]

\[
SensorScoreBad = 1 - 0.44 = 0.66
\]

The normalizer equals the maximum number than RI can take times the maximum number of lines
To correctly deal with either a *properly operating* or *faulty* sensor, the HMM model includes different strategies for the two cases.

**Sensor Reliability**

\[
S_1 = \text{SensorScoreOK} \cdot \text{tentative}
\]

\[
S_2 = \text{SensorScoreBad} \cdot [(\text{tentative} \cdot w) + \frac{X_t}{X_{t+1}} \cdot (1 - w)]
\]

\[
Z = (S_1 | S_2)
\]
Experimental Configuration

We verified the improvements of our model using two datasets recorded in real driving conditions.

Differently from KITTI datasets here we have hundreds of lane transitions.

A4 Highway, Milan-Bergamo, Italy
4-Lanes Configuration

A2 Highway, Alcalá de Henares, Spain
3-Lanes Configuration

Detector Only
Our Model

Wrong Lane Estimate
Correct Lane Estimate
Results 1

Detected Ego Lane Dispersion

<table>
<thead>
<tr>
<th>Correct Lane</th>
<th>Detectory Only</th>
<th>Our Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset 1</td>
<td>3744</td>
<td>2762</td>
</tr>
<tr>
<td>Offset 2</td>
<td>779</td>
<td>212</td>
</tr>
<tr>
<td>Offset 3</td>
<td>153</td>
<td>0</td>
</tr>
</tbody>
</table>

| GT Lane 1  | 2230  | 320  | 21   | 3    | 2574 | 0.866 |
| GT Lane 2  | 904   | 2005 | 275  | 16   | 3200 | 0.627 |
| GT Lane 3  | 373   | 1666 | 927  | 5    | 2971 | 0.312 |
| GT Lane 4  | 150   | 369  | 574  | 114  | 1207 | 0.094 |
| Total      | 3657  | 4360 | 1797 | 138  |      |       |

| Precision | 0.61  | 0.46  | 0.516 | 0.826 |
| F1 Score  | 0.7158| 0.53  | 0.389 | 0.17  |

Support-Column: how many GT lanes. Total-Row: how many ego-vehicle detections over the n-lane.
Results 2

Ground Truth

Line Detector & Tracker Only

Proposed Model
Thank you

Dataset will be available for download at http://www.ira.disco.unimib.it/ego-lane-estimation-by-modeling-lanes-and-sensor-failures

or just scan QR code

Updated Version of the Paper also on our website soon
Results 2