Learning to Localize Using a LiDAR Intensity Map

Andrei Bărsan\(^{*1,2}\), Shenlong Wang\(^{1,3,2}\), Andrei Pokrovsky\(^3\), Raquel Urtasun\(^1,2\)

\(^1\)Uber ATG, \(^2\)University of Toronto

1 Motivation

- Robust and accurate localization is one of the cornerstones of an autonomous driving stack.
- Goal: Perform real-time online localization w.r.t. an HD LiDAR intensity map with centimeter-level accuracy.
- Challenges:
  - Dynamic Objects
  - Lack of Geometric Cues
  - Different LiDAR Types
- Past approaches:
  - Suffer in geometrically degenerate environments, e.g., bridges
  - Cannot generalize to different LiDARs without calibration.

2 Probabilistic Localization

- We learn to match between online sensory observations and a map.
- We incorporate this learned component into a histogram filter with GPS information:

\[
\text{Bel}_t(x) = \eta \cdot \frac{P_{\text{LiDAR}}(T_t; x; \omega)}{P_{\text{GPS}}(G_t|x)} \cdot \text{Bel}_{t-1}(x|\chi_{t-1})
\]

This gives us a probability distribution over the vehicle pose in world coords. The discretization is centered around the dead reckoning pose.

**Online and Map Embedding Networks**

\[
P_{\text{LiDAR}} \propto \exp \left( -\frac{1}{\sigma_{\text{GPS}}^2} \left( \sum g_x(x_t; \theta; \mathcal{M}; w_{\mathcal{M}}) + y_t \right) \right)
\]

2D rigid transform

\[
P_{\text{GPS}} \propto \exp \left( -\frac{1}{\sigma_{\text{GPS}}^2} \left( \sum g_x(x_t; \theta; \mathcal{M}; w_{\mathcal{M}}) + y_t \right) \right)
\]

**Motion Model**

\[
\text{Bel}_{t-1}(x|\chi_{t-1}) = \sum_{x_{t-1}} P(x|\chi_t, x_{t-1}) \text{Bel}_{t-1}(x_{t-1})
\]

Motion model uses a Gaussian to model dynamics uncertainty.

- At each time step we exhaustively search the space \(x = (x, y, \theta)\) around the dead reckoning pose for the best match.
- Obtain current pose from Bel:

\[
x_t^* = \arg\max_\chi \text{Bel}_t(x)\]

**Matching in (x, y) is equivalent to a 2D correlation, which we perform in the Fourier domain for performance reasons.**

**Matching in spatial domain:** 26.7ms

**Matching in Fourier domain:** 1.4ms

Real-time system performance: **15Hz on a GPU**

- The learned component of our system is the LiDAR matching, i.e., \(P_{\text{LiDAR}}\) in the above diagram.
- The embedding nets use the LinkNet architecture.
- Embeddings are learned by backpropagating through the cross-correlation matching. We do not include the temporal filtering or GPS components at train time.
- We use a cross-entropy loss whereby the score volume corresponding to the ground truth is a one-hot encoding of the true offset between the online and the map data in a sample.

Examples of high-definition maps with centimeter-level resolution.

3 Results

- Tested on 280km of highway.
- 99th percentile error < 20cm (lane marker = 15cm wide).

<table>
<thead>
<tr>
<th>Method</th>
<th>Motion</th>
<th>Prob</th>
<th>Median Error (cm)</th>
<th>Failure Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lat</td>
<td>Lon</td>
</tr>
<tr>
<td>Dynamics</td>
<td>Yes</td>
<td>No</td>
<td>1.84</td>
<td>91.02</td>
</tr>
<tr>
<td>Raw LiDAR</td>
<td>Yes</td>
<td>No</td>
<td>1.52</td>
<td>5.04</td>
</tr>
<tr>
<td>ICP</td>
<td>Yes</td>
<td>No</td>
<td>3.81</td>
<td>4.53</td>
</tr>
<tr>
<td>Ours (LinkNet)</td>
<td>No</td>
<td>No</td>
<td>3.87</td>
<td>4.99</td>
</tr>
<tr>
<td>Ours (LinkNet)</td>
<td>Yes</td>
<td>Yes</td>
<td>3.00</td>
<td>4.33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Motion</th>
<th>Prob</th>
<th>Median Error (cm)</th>
<th>Failure Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lat</td>
<td>Lon</td>
</tr>
<tr>
<td>Dynamics Only</td>
<td>Yes</td>
<td>No</td>
<td>1.14</td>
<td>65.66</td>
</tr>
<tr>
<td>ICP</td>
<td>Yes</td>
<td>No</td>
<td>2.57</td>
<td>15.29</td>
</tr>
<tr>
<td>Ours (Transfer)</td>
<td>Yes</td>
<td>No</td>
<td>6.95</td>
<td>6.38</td>
</tr>
</tbody>
</table>