Effective Map-matching on the Most Simplified Road Network

Kuien Liu
Yaguang Li
Fengcheng He
Jiajie Xu
Zhiming Ding

Institute of Software, Chinese Academy of Sciences, Beijing China
Map-matching on the Simplified Road Network: Why?

- **Motivation**
  - Some maps are simplified naturally.
  - Some terminals have limited memory.

- **Result**
  - Map size:
    - 273.5MB $\rightarrow$ 48.6MB.
  - Speed up
    - Indexing: 3s $\rightarrow$ 0.48s
    - Matching: 1.2s $\rightarrow$ 0.58s
  - Accuracy
    - 1hz GPS: 95.629%

<table>
<thead>
<tr>
<th>Type</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>id, name, type, length, speed limit, width, start, p_1, p_2, ..., p_n, end, etc.</td>
</tr>
<tr>
<td>Simplified</td>
<td>id, start, end</td>
</tr>
</tbody>
</table>
Proposed Method: *Passby* (1/2)

- **Example**
  - An object $o$ moving on road $e_1$, with sampled positions $P_{i-1}$, $P_i$, $P_{i+1}$, ...
  - Should $P_i$ be matched to road $e_1$ or road $e_2$?

- **Basic idea**
  - We can reduce the uncertainty of map-matching by considering both intersections which object $o$ has passed and will pass by next.

Unfortunately, it applies to high-sampling-rate trajectories only.
Proposed Method: *Passby* (2/2)

- **Challenges**
  - Low sampling rate trajectories
  - Other problems, e.g., Y-junction, Parallel pathways, tunnel

- **Improved approach**
  - **Search Space**
    1. Topological constrains
    2. Spatial constrains
    3. Temporal constrains
  - Ranking with weighted sum strategies

*Passby* shoots more than 60% (15s). Now, much better!
Some supplementary mechanisms & tools

- **Fast Angle Calculation**
  - Hash tables: $\arctan$, $\cosine$ and $\sin$e

- **Parallelized Matching Process**
  - Matching(1hz) with OpenMP: 0.58s $\Rightarrow$ 0.3s

- **Outlier Identification**
  - Topology connectivity and path reversibility

- **Visualization Tools**

(a) Matching result  (b) Error debugging  (c) Log analyzers (1hz)
On-going works

- **Passby under extending**
  - given a simplified map, how to improve the matching accuracy as much as possible.

- **CellularMap under construction**
  - given an acceptable tolerance, how to compress/simplify the underlying road network as much as possible.
Your Comments are Welcome!