Personalized Routing for Car Navigation Systems

Jukka M. Krisp, Andreas Keler, Nicole Karrais
University of Augsburg, Institute of Geography, Applied Geoinformatics, Alter Postweg 118, 86159 Augsburg, Germany
Mobile & Location Based Services – e.g. Routing

• Current routing options are somewhat “primitive”.
  – limited (basically shortest / fastest route) // + some limited options; + near real time traffic information (RTTI)

• Research - Personalized (in-car) routing, in this case “Fahranfängerrouting – drivers beginners routing” avoiding complicated crossings (Krisp et.al. MobileTartu2014)
Personalized Routing e.g. for inexperienced drivers (Fahranfänger-Routing)

• The idea here is to provide “inexperienced drivers” with an in-car routing system that computes an “easy to drive” route

• What is that an “easy to drive” route?
  – 21 driving situations have been described
  – 80 questionnaires have been distributed among potential driving beginners at selected driving schools; rating each situation / 39 returned
Top situation regarding difficulty

Situations are the as top three regarding difficulty #1 and #2

- **(1) big unclear „complicated“ crossings**
- **(2) Left turns on busy roads, without a traffic light**
- In general left turns are regarded more difficult than right turns

For example Munich, Lenbachplatz
How to compute an „easy to drive“ route?

• Data (attributes of road data, traffic lights, etc.)
  – Which attributes in existing road datasets can be used to compute a route that acknowledges the inexperienced drivers preferences? (e.g. are traffic lights geocoded in the road datasets?)

• Algorithms
  – Can we implement additional routing algorithms to support the computation of an “easy to drive” route? (e.g. implement an algorithm that will compute a route from A to B with only right turns in a network)
Data (attributes of road data, traffic lights, etc.) – e.g. (OSM) data as a basis for routing

- Study area - road network of Munich
- Selected road type values (Road-Extracts)
Method - Extraction of OSM elements and generation of obstacle polygons

1. Road-Extracts → node-extraction
2. Generation of polygons → Extraction by threshold
3. Nodes with all attributes → merging points
4. Point elements

Density values → KDE

Obstacle polygons → Point elements

UNIA Universität Augsburg University
Method - generation of obstacle polygons

- Overview of generated obstacle polygons based on the number of nodes (of the considered road elements)
## Obstacle Polygons

<table>
<thead>
<tr>
<th>Map view: obstacle polygon</th>
<th>Street view</th>
<th>Description of the crossing</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Map view" /></td>
<td><img src="image2" alt="Street view" /></td>
<td>- complicated left turns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Amount of counted road element nodes: <strong>97</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Search radius: 60 m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cell size for KDE: 5 m</td>
</tr>
<tr>
<td><img src="image3" alt="Map view" /></td>
<td><img src="image4" alt="Street view" /></td>
<td>Amount of counted road element nodes: <strong>67</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Search radius: 60 m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cell size for KDE: 5 m</td>
</tr>
<tr>
<td><img src="image5" alt="Map view" /></td>
<td><img src="image6" alt="Street view" /></td>
<td>Amount of counted road element nodes: <strong>33</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Search radius: 60 m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cell size for KDE: 5 m</td>
</tr>
</tbody>
</table>

*UNIA Universität Augsburg University*
Results - Alternative routing through the study area

- “standard” route between Theresienwiese and Ludwig-Maximilians-Universität (LMU)
Results - Alternative routing through the study area

- route avoiding complicated crossings as obstacles between Theresienwiese and Ludwig-Maximilians-Universität (LMU)
### Results – Comparison of the two routes

<table>
<thead>
<tr>
<th></th>
<th>Standard Route</th>
<th>Route avoiding complicated crossings as obstacles</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Length</strong></td>
<td>5.14 [km]</td>
<td>6.19 [km]</td>
</tr>
<tr>
<td><strong>Time to drive</strong></td>
<td>11 [minutes]</td>
<td>14 [minutes]</td>
</tr>
<tr>
<td><strong>Number of turns</strong></td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td><strong>Number of right turns</strong></td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td><strong>Number of left turns</strong></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td><strong>Number of crossings</strong></td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td><strong>Number of avoided</strong></td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>“complicated crossings”</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
...useful?

• Is the route avoiding complicated crossings in every case easier to drive than the standard route?
  – a **proof of concept** a comprehensive user test needs to be conducted
  – As “easy”, “difficult” are **fuzzy** concepts, why not build a fuzzy logic model?
  – elaborate on **personalized model** a fuzzy continuum ranging from an "easy to drive" route to a "difficult to drive route"
To come...

• the first step in the development of a more comprehensive system, that suggests users an “easy to drive” route


• personalized parameters

• including dynamic objects (bikes)?

• *We want “complicated routes” for driver training...?*

• + application areas “driver test (Führerscheinprüfung)”
Thanks.

Contact & Further information

Prof. Dr. Jukka M. Krisp
Universität Augsburg
Institut für Geographie
Alter Postweg 118
86159 Augsburg, Germany