Context-Aware Search Spaces for Queries with Spatial Relations

Rick Gelhausen
Context-aware search spaces are search spaces that we consider correct for a given context.

Setting the search radius for "hotels near cities" to 2km would not be context-aware.

Making the search radius for "hotels near cities" dependant on the city size would be context-aware.

Searching for "restrooms between countries" would be useless.
Motivation

- Web-mapping services are only using „near“ and „in“.
- Context-aware search spaces are represented by polygons.
- We introduce the new spatial relations „between“ and the compass directions.
Implementation

- I created a toolbox for every useful query, written in C++.
- For every query the input is a couple of points, polygons or polylines and the output is a polygon.
- I also created a small website that allows the usage of the toolbox.
The „between“ Relation

- We have clear boundary conditions.
- There are not many assumptions to make.
“countries between countries”
“countries between countries”
“cities between cities”
“cities between cities”
„hotels between cities“
„hotels between cities“
„gas stations between cities“
„gas stations between cities“
“hotels between streets”
„hotels between streets“
"hotels between streets"
„hotels between streets“
"hotels between streets"
„hotels between streets“
We do not have clear boundary conditions.
We have to make assumptions on many variables.
Examples for countries
„hotels north of cities“
„hotels north of streets“
“hotels north of streets”
„hotels west of streets“
## Overview

<table>
<thead>
<tr>
<th>“between”</th>
<th>country</th>
<th>city</th>
<th>street</th>
</tr>
</thead>
<tbody>
<tr>
<td>country</td>
<td>quadrilateral</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>city</td>
<td>special case</td>
<td>ellipse</td>
<td>x</td>
</tr>
<tr>
<td>hotel</td>
<td>special case</td>
<td>ellipse or tube structure</td>
<td>polygon structure</td>
</tr>
<tr>
<td>supermarkets</td>
<td>x</td>
<td>ellipse</td>
<td>polygon structure</td>
</tr>
<tr>
<td>gas stations</td>
<td>x</td>
<td>tube structure</td>
<td>polygon structure</td>
</tr>
<tr>
<td>restrooms</td>
<td>x</td>
<td>x</td>
<td>polygon structure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>compass directions</th>
<th>country</th>
<th>city</th>
<th>street</th>
</tr>
</thead>
<tbody>
<tr>
<td>country</td>
<td>quadrilateral</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>city</td>
<td>x</td>
<td>semi-ellipse</td>
<td>x</td>
</tr>
<tr>
<td>hotel</td>
<td>x</td>
<td>semi-ellipse</td>
<td>polygon structure</td>
</tr>
<tr>
<td>supermarkets</td>
<td>x</td>
<td>x</td>
<td>polygon structure</td>
</tr>
<tr>
<td>gas stations</td>
<td>x</td>
<td>x</td>
<td>polygon structure</td>
</tr>
<tr>
<td>restrooms</td>
<td>x</td>
<td>x</td>
<td>polygon structure</td>
</tr>
</tbody>
</table>
Conclusion

- The results for the „between“ relation are very satisfying.
- Most results for the compass directions are useful, but we have to conduct a survey to improve the results.
- The queries for countries in a compass direction are better solved using a non-polygon method.
Future Work

- conduct a survey to confirm the results
- improve the performance using e.g. databases
- create more spatial relations „near“, „inside“, „along“
- use the result polygons to improve web-mapping services
Thank you for your attention!
References