LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

LEHRSTUHL FÜR MOBILE UND VERTEILTE SYSTEME

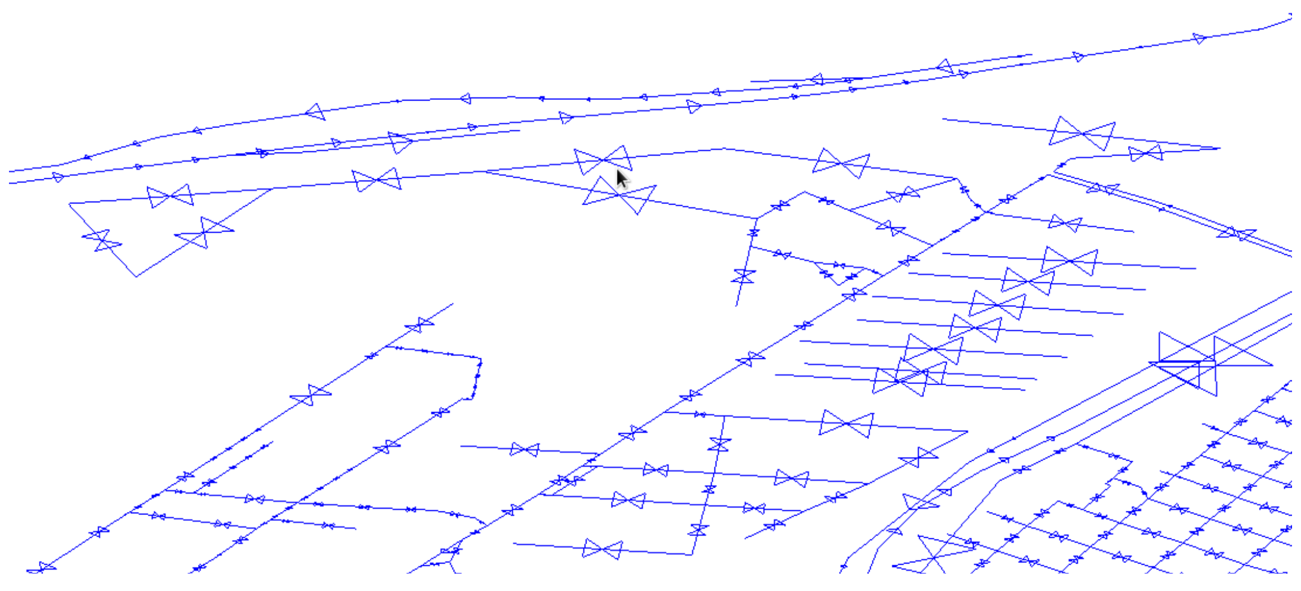
GISCUP 2015: Notes on Routing with Polygonal Constraints

Martin Werner, Mobile and Distributed Systems Group, Ludwig-Maximilians-Universität München
martin.werner@ifi.lmu.de

Introduction

The 4th SIGSPATIAL Cup is an GIS-focused algorithm competition. The current problem was about routing in street networks under polygonal constraints.

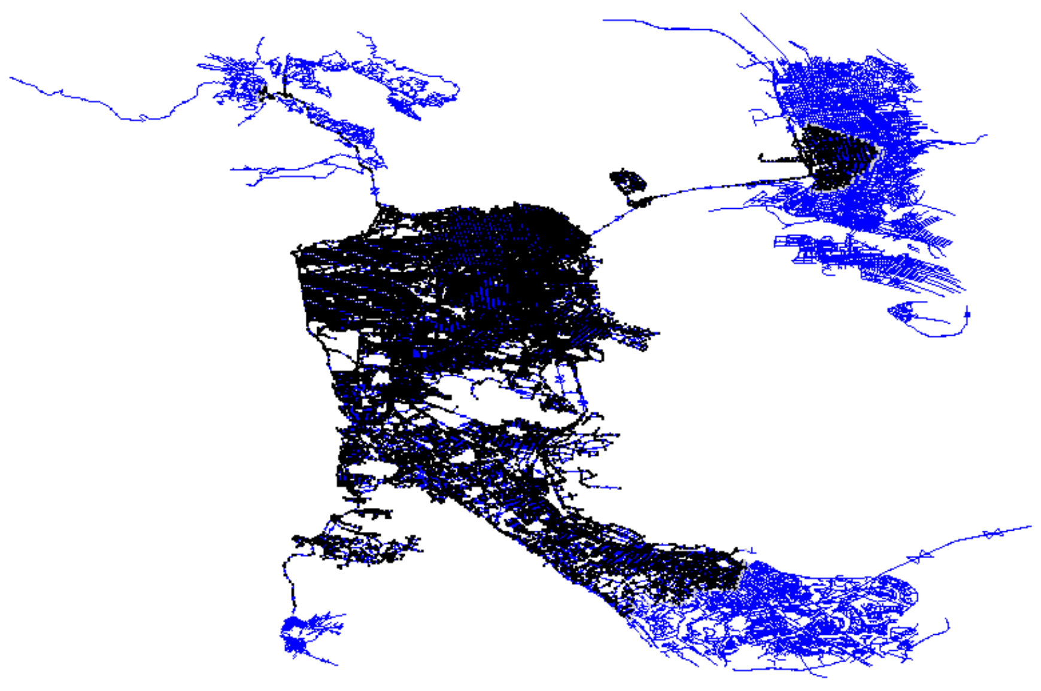
Dataset



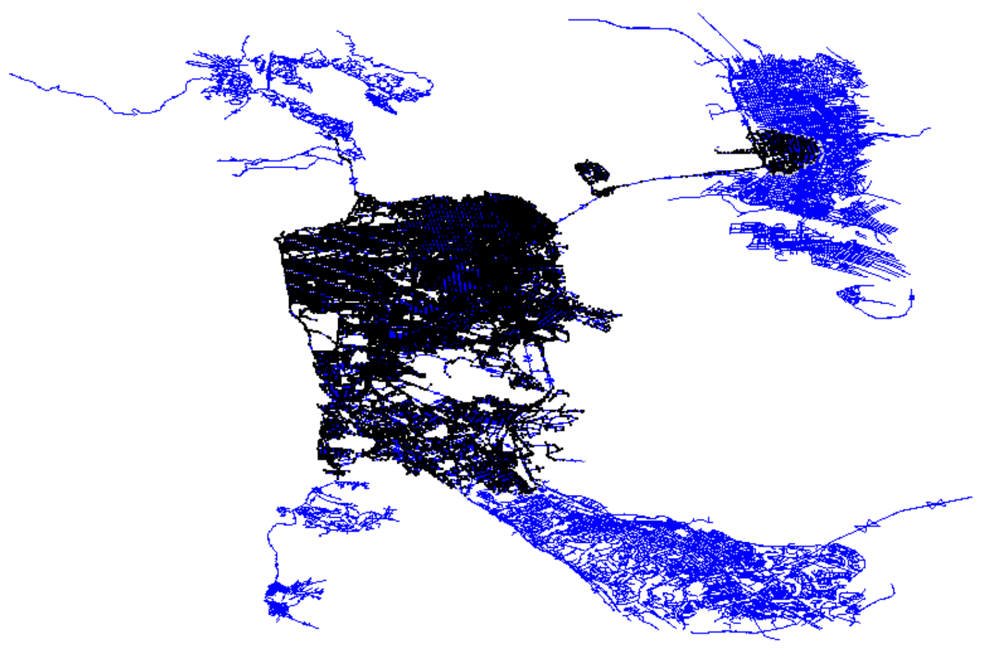
The dataset has been extracted from OpenStreetMap and given as a set of shapefiles describing roads, nodes, and polygonal constraints. The dataset is given in a WebMercator projection.

Problem Statement

Calculate as quickly as possible (including index build and update times) shortest path under polygonal constraints using time and distance metrics.



Dijkstra is a natural choice - it does not involve distance computations and is fairly efficient given the small dataset. The image shows a shortest path calculation between two islands - black vertices are vertices that have been used in the given search. Essentially, Dijkstra creates a circular search space around the shortest path.

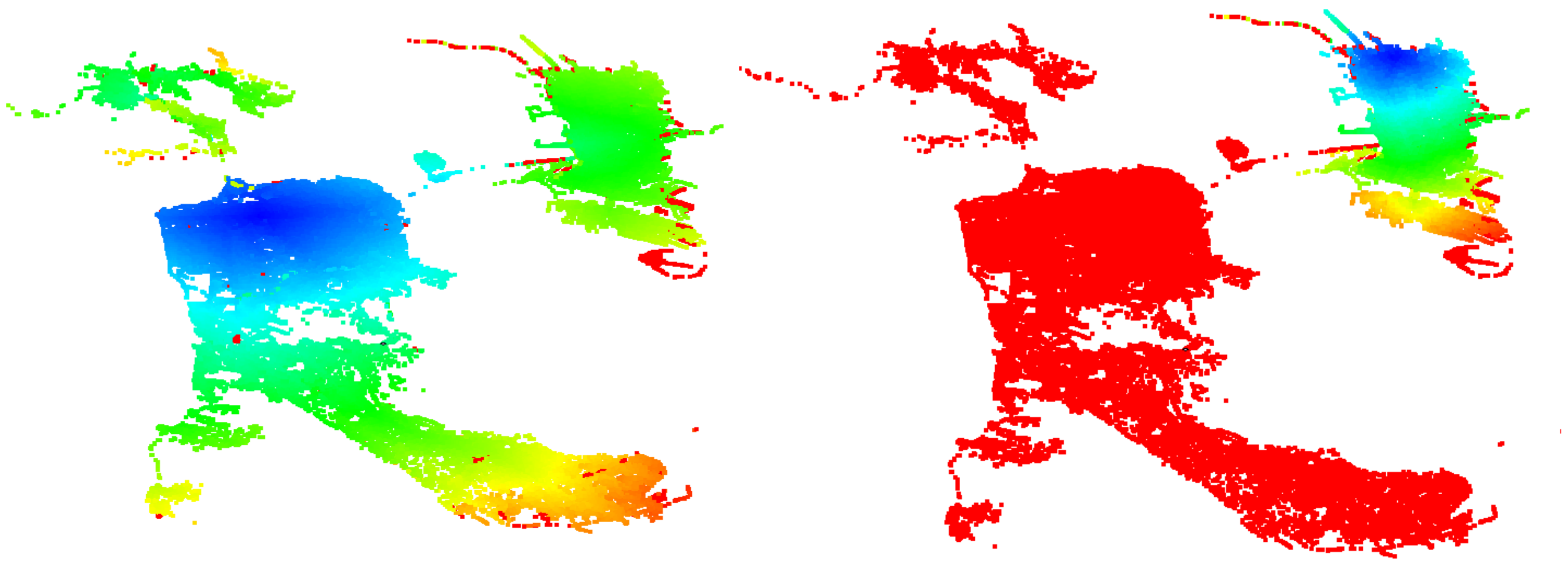


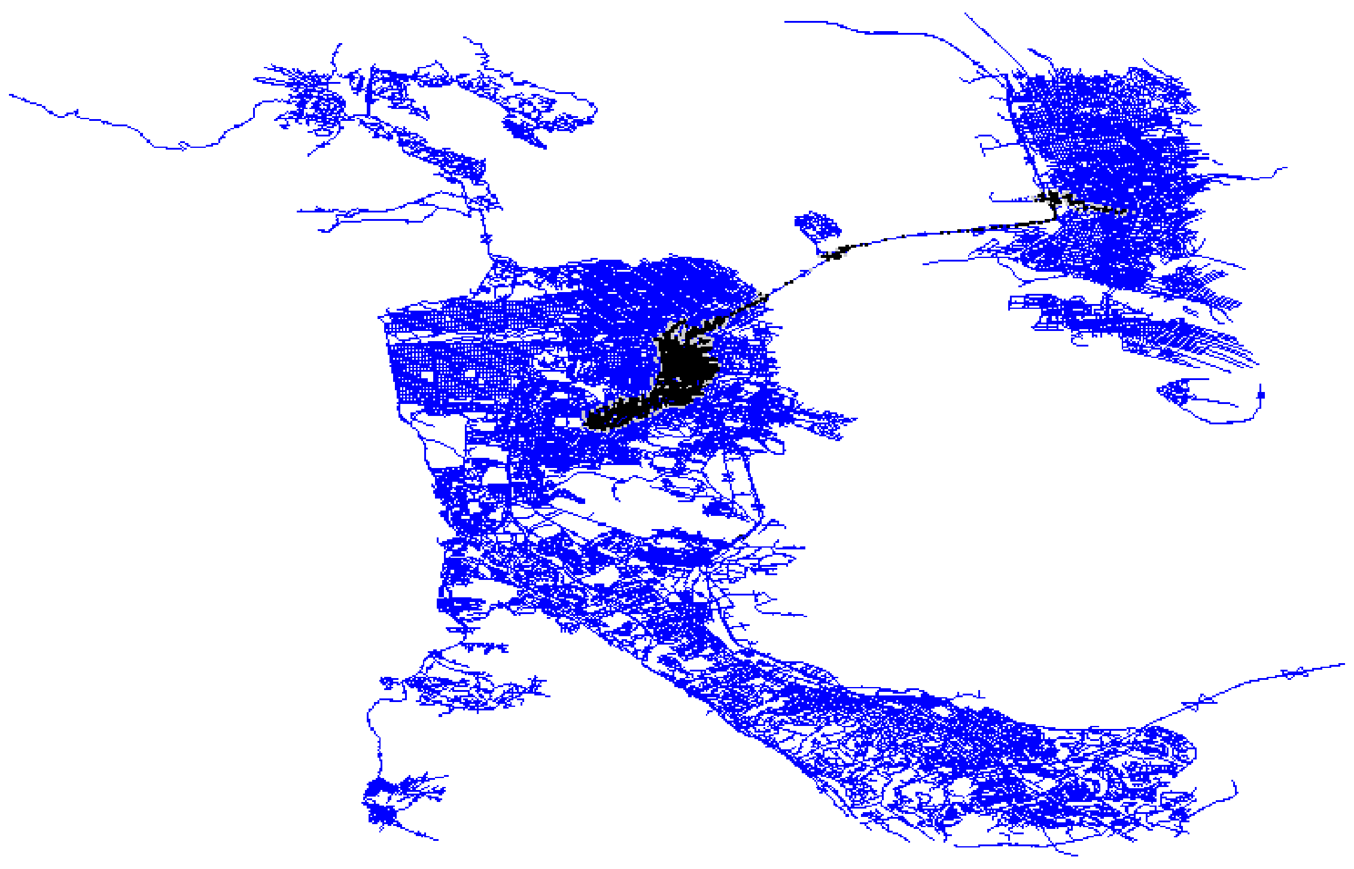
A* is a speedup technique for Dijkstra in which the spatial distance (line of sight) between the current vertex processed during a search is used in order to reduce the search space. A* is optimal and optimally efficient in theory (e.g., there is no algorithm without precalculation calculating shortest paths visiting fewer vertices) - however A* can be slow in practice due to numerous distance calculations.

ALT is a variant of A* in which the distance calculations are replaced with table lookups using triangle inequality. Therefore, the distance to and from every vertex is precomputed in a preprocessing step. ALT is compatible with increasing edge weights.

$$d_{\text{from } L}(w) - d_{\text{from } L}(v) \leq d(v,w)$$

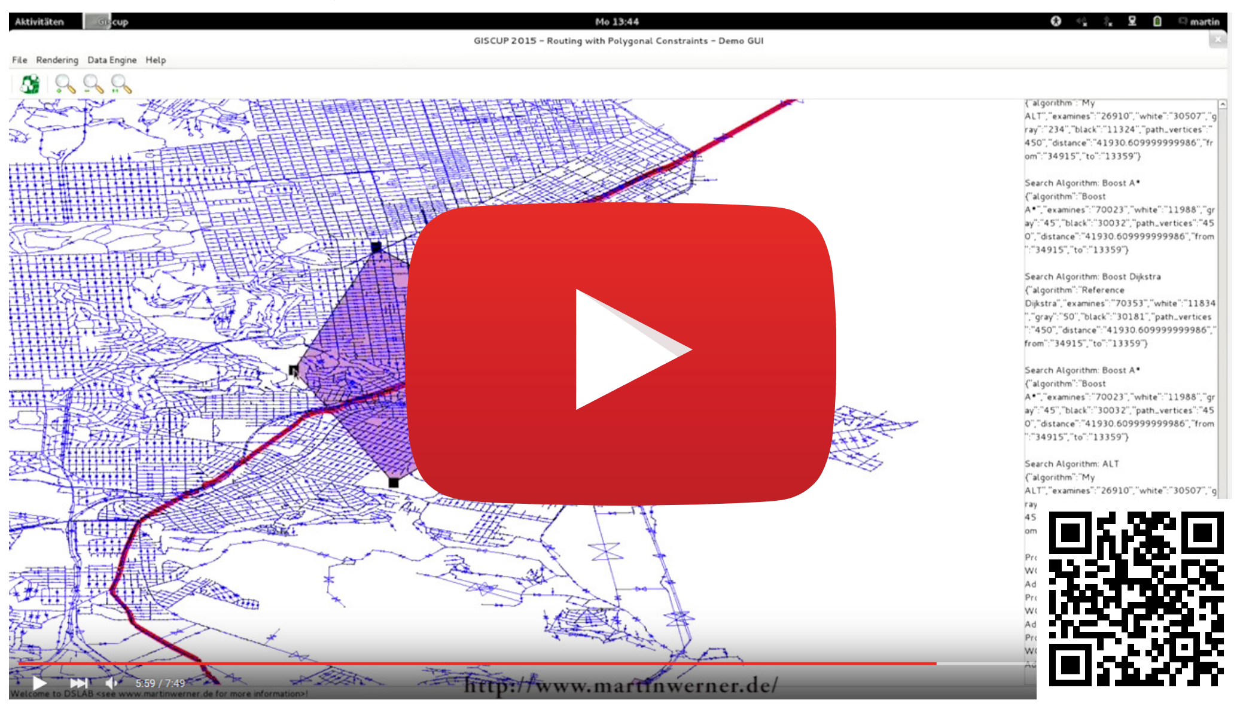
$$d_{\text{to } L}(v) - d_{\text{to } L}(w) \leq d(v,w)$$





The resulting search space is rather small and - as there are no distance compuations - the system is efficient. Note that other methods (contraction hierarchies, geometric containers, etc.) would have been an alternative for ALT. Still, I expected that their complex update strategies would not pay off on the small network given in the challenge.

Further information



You find the *source code* on my webpage, a *video demonstration* on YouTube, and the *DSLAB Data Science Lab* environment on GitHub. Feel free to contact me!

Calculating 1000 Shortest Paths

```
Built a graph. Vertices: 42065, Edges: 96850
Done init in 2.87s
Preparing 25 landmarks

Basic Dijkstra
Elapsed 4.37s
Approx. 228.833 operations per second
Basic A*
Elapsed 8.61s
Approx. 116.144 operations per second
ALT
Elapsed 1.84s
Approx. 543.478 operations per second
```

pervasive
data
science