Automatic Generation of Route Sketches

Andreas Gemsa[1], Martin Nöllenburg[1,2], Thomas Pajor[1], and Ignaz Rutter[1]

[1] Institute of Theoretical Informatics, Karlsruhe Institute of Technology (KIT), Germany
[2] Department of Computer Science, University of California, Irvine, USA

1   Introduction

Generating route sketches is a graph redrawing problem, where we are given an initial drawing of a graph G and want to find a new, *schematized* drawing of G that reduces the drawing complexity while preserving the structural characteristics of the input. The motivation of our work is the visualization of routes in road networks as sketches for driving directions. An important property of a route sketch is that it focuses on road changes and important landmarks rather than exact geography and distances. Typically the start and destination lie in populated areas that are locally reached via a sequence of relatively short road segments. On the other hand, the majority of the route typically consists of long highway segments with no or only few road changes. This property makes it difficult to display driving directions for the whole route in a single traditional map since some areas require much smaller scales than others. The strength of route sketches for this purpose is that they are not drawn to scale but rather use space proportionally to the route complexity.

In summary, we wish to find drawings of routes that have a low visual complexity, yet capture all important parts of a route, such as turning points or road changes. Further, the schematized drawing should in some sense reflect the layout of the original geographic route. We model the problem as a mixed-integer linear program (MIP) that computes schematizations of paths or unions of alternative paths. The solution of the MIP satisfies all mandatory constraints and optimizes the visual quality criteria.

Related Work   Agrawala and Stolte [1] presented a system called *LineDrive* that draws route sketches with heuristic methods based on simulated annealing. Nöllenburg and Wolff [4] used a MIP approach to compute schematic metro maps of public transport graphs. Brandes and Pampel [2] studied a path schematization problem in the presence of orthogonal order constraints for preserving the mental map. They showed that deciding the existence of a rectilinear schematization is NP-hard. Delling et al. [3] gave an efficient algorithm for schematizing monotone paths under orthogonal order constraints.

2   Model

Our aim is to produce a drawing of the input graph that reduces the drawing complexity but at the same time maintains the user's mental map of the route. As a preprocessing step we simplify the input using the standard Douglas-Peucker line-simplification algorithm to reduce the number of edges in the graph while maintaining its overall shape as

well as all important vertices (e.g., road changes and turning points). For further reducing the drawing complexity, we use a limited set $\mathscr{C}_d = \{z \cdot 90°/d \mid z \in \mathbb{Z}\}$ of admissible edge slopes for $d \geq 1$.

To maintain the mental map, we preserve the *orthogonal order* of the input, i.e., the left-to-right and top-to-bottom order of all vertices. Furthermore, each edge is preferably drawn with the slope in $\mathscr{C}_d$ that is closest to its input slope; an appropriate minimum edge length is used so that each edge is well visible.

We also experimented with two supplementary approaches that help create sketches that are often visually more pleasing. In the original formulation edge lengths in the output do not carry information about the true distances. While it is an important feature of a route sketch that distances are distorted, we may require that the input length order of the edges is preserved in the output. Second, preserving the orthogonal order for vertices that are far apart is of limited importance. So any pair of vertices whose distance in one of the coordinates is at least one third of the total extent of the path in that coordinate does not need to preserve its order in the respective other coordinate.

All these constraints can be modeled as the linear constraints and the linear objective function of a MIP.

3   Evaluation

We have implemented our approach and have tested it on 1000 random routes in the German road network. Two main observations can be made in this study: 1) More than 50% of the instances did not have a valid rectilinear schematization, whereas only 0.7% of the instances were infeasible in the octilinear case $d = 2$ and for any $d \geq 3$ there were at most 0.1% infeasible instances. 2) The average running time of the MIP optimizer increases with increasing d from 107.36 ms for $d = 1$ and 649.49 ms for $d = 3$ to 1347.86 ms for $d = 5$ on a single AMD Opteron 2218 CPU with the MIP optimizer Gurobi 3.0.1.

Additionally, we present examples that display sketches of alternative routes in a single picture.

References

1. Maneesh Agrawala and Chris Stolte. Rendering effective route maps: Improving usability through generalization. In *Proc. 28th Ann. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH'01)*, pages 241–249. ACM, 2001.
2. Ulrik Brandes and Barbara Pampel. On the hardness of orthogonal-order preserving graph drawing. In *Proc. 16th Internat. Symp. Graph Drawing (GD'08)*, volume 5417 of *Lecture Notes Comput. Sci.*, pages 266–277. Springer-Verlag, 2009.
3. Daniel Delling, Andreas Gemsa, Martin Nöllenburg, and Thomas Pajor. Path schematization for route sketches. In *Proc. 12th Scand. Symp. & Workshops on Algorithm Theory (SWAT'10)*, volume 6139 of *Lecture Notes Comput. Sci.*, pages 285–296. Springer-Verlag, 2010.
4. Martin Nöllenburg and Alexander Wolff. Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE Trans. Visualization and Computer Graphics*, 2010. Preprint available online.