An Algorithmic Approach to Geographic Information Science

Zusammenfassung der schriftlichen Habilitationsleistung von

Dr.-Ing. Jan-Henrik Haunert

Lehrstuhl für Informatik I
Fakultät für Mathematik und Informatik
Universität Würzburg

Fachmentorat
Prof. Dr. Alexander Wolff, Universität Würzburg
Prof. Dr.-Ing. Monika Sester, Universität Hannover
Prof. Dr. Marc Erich Latoschik, Universität Würzburg

Würzburg, den 14. Mai 2013

Ich versichere, dass ich die schriftliche Habilitationsleistung selbstständig verfasst und die Herkunft des verwendeten oder zitierten Materials ordnungsgemäß kenntlich gemacht habe.

Ich habe keinen anderen Antrag auf Zulassung zur Habilitation eingereicht, mir wurde kein akademischer Grad entzogen, und gegen mich ist kein Verfahren anhängig, das die Entziehung eines akademischen Grades zur Folge haben könnte.

Dr.-Ing. Jan-Henrik Haunert
Würzburg, den 6.5.2013

Contents

Submitted Publications

[1] T. C. van Dijk and J.-H. Haunert.
A probabilistic model for road selection in mobile maps. In *Proc. 12th International Symposium on Web and Wireless Geographical Information Systems (W2GIS'13)*, volume 7820, series *LNCS*, pages 214–222. Springer-Verlag, 2013.

[2] M. Fink, J.-H. Haunert, A. Schulz, J. Spoerhase, and A. Wolff.
Algorithms for labeling focus regions. *IEEE Transactions on Visualization and Computer Graphics (Proc. Information Visualization 2012)*, 18(12):2583–2592, 2012.

[3] A. Gemsa, J.-H. Haunert, and M. Nöllenburg.
Boundary-labeling algorithms for panorama images. In *Proc. 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM-GIS'11)*, pages 289–298, 2011.

[4] J.-H. Haunert.
A symmetry detector for map generalization and urban-space analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 74:66–77, 2012.

[5] J.-H. Haunert and C. Brenner.
Vehicle localization by matching triangulated point patterns. In *Proc. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM-GIS'09)*, pages 344–351, 2009.

[6] J.-H. Haunert and B. Budig.
An algorithm for map matching given incomplete road data. In *Proc. 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM-GIS'12)*, pages 510–513, 2012.

[7] J.-H. Haunert, A. Dilo, and P. van Oosterom.
Constrained set-up of the tGAP structure for progressive vector data transfer. *Computers & Geosciences*, 35(11):2191–2203, 2009.

[8] J.-H. Haunert and L. Sering.
Drawing road networks with focus regions. *IEEE Transactions on Visualization and Computer Graphics (Proc. Information Visualization 2011)*, 17(12):2555–2562, 2011.

[9] J.-H. Haunert and A. Wolff.
Optimal and topologically safe simplification of building footprints. In *Proc. 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM-GIS'10)*, pages 192–201, 2010.

1 Introduction

Geographic information (GI) science has become a well-established discipline with a growing number of peer-reviewed journals, about a dozen international conferences, and Bachelor's as well as Master's courses in many countries. While GI science is often considered as the discipline of working with geographic information systems (GIS), it is more accurately described as *the science behind GIS*. As such, it aims at methods for the acquisition, administration, analysis, and visualization of spatial information.

Basic tasks that users solve with GIS are the computation of a map overlay, a polygon buffer, or a shortest route in a road network. Most of the algorithms for these basic tasks have been developed by computer scientists with a focus on computational geometry or graph theory—and *not* by GI scientists. These tasks are simply of such general relevance (also in disciplines other than GI science, for example, in robotics, logistics, and computer-aided design) that sophisticated solutions have been found by evolution over time. Since appropriate mathematical models (e.g., graphs) for real-world phenomena (e.g., road networks) have been established, engineering a shortest-path algorithm for a transport application requires little knowledge of GI science.

Given that, which role do GI scientists actually play? To answer this question, it is important to understand that most of the open problems in GI science are very different from the above-mentioned basic problems, each of which can be unambiguously defined with a single sentence—*polygon buffering*, for example, can be defined as follows: Given a polygon P and a number $d \geq 0$, compute the union of all points whose distance to P is at most d. In contrast, GI scientists usually work on problems that involve multiple, sometimes vaguely defined constraints. Often, mathematical models are missing or there are multiple models that seem to be appropriate for a task. Consequently, GI scientists do not start with a given problem statement that is rigorously defined in mathematical terms. Instead, they first need to build a good model that describes all important aspects of their task. Typically, this requires knowledge in surveying, cartography, or geography. Thus, GI science can be seen as the link between several application-driven research disciplines and (a more theoretically oriented) computer science.

This paper summarizes nine publications [1–9] that present methods for a rather broad range of problems in GI science, including map generalization, information visualization, and map matching. Despite the diversity of the problems, all these methods share, on an abstract level, the same approach, which is shown in Fig. 1. We term this approach our *research cycle*.

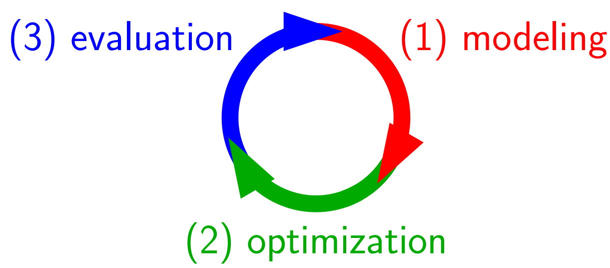

Figure 1: Our research cycle that refines a model that formalizes a task.

Our research cycle combines three stages, namely, (1) modeling, (2) optimization, and (3) evaluation, which can be repeatedly applied to incrementally refine a model that describes a given task and to adjust the algorithm for solving it.

We first exemplify the research cycle by discussing its application to map generalization (Sect. 2). Next, we discuss new methods for information visualization, in particular, focus-and-context visualization (Sect. 3) and label placement (Sect. 4). Then, we summarize results on the problem of matching data (e.g., a GPS trajectory) recorded by a mobile user with a reference map (Sect. 5). Finally, we draw general conclusions and discuss possible directions of future research (Sect. 6).

2 Map Generalization

Map generalization is the problem of generating a map of a smaller scale from a given map. To ensure that the generated map is legible, some of the features in the map need to be simplified, aggregated, displaced, or eliminated through selection. Though many algorithms have been developed to solve special problems in map generalization, it is difficult to decide which algorithm to choose in a given situation and how to sequence algorithms for different subtasks. To model the overall problem of map generalization, researchers have formalized constraints that the output map has to satisfy [WD98, HW07]. This corresponds to the *modeling stage* in our research cycle. Usually, constraints for map generalization can be classified into constraints that aim at preserving characteristics of the input map and constraints that aim at increasing the readability of the map. Some of the constraints in map generalization are conflicting and can only be satisfied to a certain degree. We term such constraints *soft* and thereby distinguish them from hard constraints, which have to be satisfied in any case.

Once we have established a set of constraints, we can ask for an optimization algorithm that yields a solution fulfilling all hard constraints while satisfying the soft constraints as much as possible. Since even basic map generalization problems have been shown to be NP-hard [EM01], most often general-purpose heuristics, for example, simulated annealing [WJT03], have been applied. Generally, developing and applying an optimization algorithm is subsumed under the *optimization stage* in our research cycle.

In the *evaluation stage* in our research cycle we assess the quality of our results. For example, we can compare the maps generated by an optimization method with maps generated manually by cartographers, or study how users perform on certain tasks when given our automatically generalized maps. Ideally, this evaluation allows us to conclude whether our model is appropriate or whether it has to be refined, for example, because important constraints are missing. In the latter case, we simply reiterate the research cycle in Fig. 1. This process allows us to incrementally improve our algorithm. Moreover, with each cycle, we increase the accuracy of our model and, thereby, obtain valuable knowledge of the criteria that make good maps. This knowledge can be used for measuring the quality of maps, which is a problem of general interest—not only in the context of automated map generalization.

If we apply heuristic algorithms in the optimization stage, we have to be careful, however, about what to infer from our evaluation. Deficits in the quality of our generated map can be due to two reasons, that is, (1) either the model that we established in the modeling

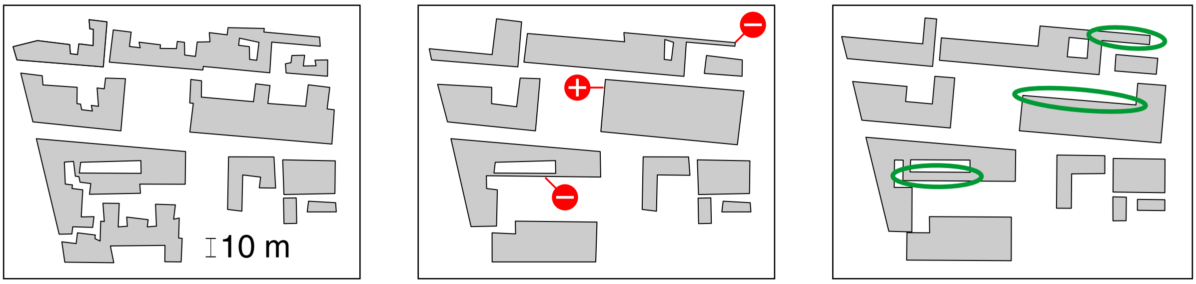

Figure 2: A set of buildings (left) that are used as input for our simplification algorithm [9], using an error tolerance of 10 m. Minimizing the number of vertices (middle), we sometimes obtain undesirable results, for example, the sizes of the polygons increase (indicated with “+”) or decrease, which results in narrow polygon parts (indicated with “−”). With a more advanced cost function we can avoid such deficits (right).

stage does not adequately reflect cartographic quality or (2) the heuristic algorithm applied in the optimization stage tends to yield solutions far from optimum. Often, it is difficult to distinguish the influences of model parameters (for example, the weights of different soft constraints) from the influences of tuning parameters of a heuristic algorithm (for example, the cooling rate in simulated annealing). This limits the conclusiveness of our evaluation. A possibility to overcome this drawback is to apply exact algorithms, which guarantee a globally optimal solution. Such algorithms may be too slow to solve large real-world instances in adequate time. Still, they can be useful to compute optimal solutions for small instances. Since the quality of these solutions is independent of the algorithm used, they can be used to verify a model and, later, can serve as quality benchmarks for heuristic algorithms.

We have successfully implemented the described three-stage approach for the simplification of polygons that represent building footprints [9]. Building simplification is in fact a classical map generalization problem, for which several algorithms have been proposed earlier [May98, KL06]. Our algorithm is the first, however, that yields an optimum solution with respect to a well-defined problem statement. Initially, our objective was to simplify a given polygon such that, among all simplifications satisfying a set of hard constraints, the output polygon has the minimum number of vertices. We showed that finding an optimum solution is NP-hard if we require a basic set of hard constraints and that the output polygon is simple. Still, we were able to solve the problem exactly by integer linear programming. Figure 2 visualizes some of our results. Though the results of minimizing the number of vertices were promising, we identified additional soft constraints that are needed to keep the output polygons sufficiently similar to the input polygons, in particular, to preserve the sizes of the polygons and right angles, which are typical for buildings. We defined a cost function that penalizes violations of these constraints and adjusted our method to find a solution of minimum total cost. When testing our method for a set of buildings of Boston, the minimum-cost solution contained only a few (2%) more vertices than the solution with the minimum number of vertices. The overall cost was reduced, however, by 39%. After a visual inspection, we concluded that the results of our new method are much better.

A shortcoming of our algorithm for building simplification that we sometimes observed

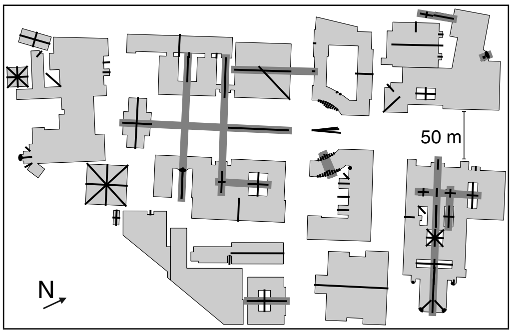

Figure 3: A set of building footprints with automatically detected mirror axes. Black lines represent basic symmetry relations between pairs of polylines that are contained in the boundaries of the building footprints. Every fat gray line represents a group of basic symmetry relations with approximately collinear mirror axes.

(even with our improved cost function) was that characteristic symmetries of the input polygons were not present in the output [4]. In order to find out whether symmetries should be considered in map generalization, we conducted a user study, in which we asked 30 participants to manually solve three tasks, namely, to simplify a set of building footprints, to identify characteristic symmetries in a set of building footprints, and to identify groups of buildings in a map. To conclude, the participants preserved most of the symmetry relations in the simplification task. We therefore developed an algorithm that detects symmetries in building footprints and compared its results with the symmetries the users had detected in the second task. Figure 3 visualizes mirror symmetries that were detected with this algorithm. Currently, we are about to integrate the preservation of symmetries as a soft constraint into our model for building simplification. This means that, after our attempt with the objective of minimizing the number of edges and our approach with the improved cost function, we will complete our research cycle for the third time.

A major research direction in map generalization is the development of methods that enable continuous zooming. We have discussed this problem in the context of land cover maps [7], where areas of different land cover classes are merged when the scale decreases. Our method is based on a model that defines constraints for a small-scale map. Given a large-scale map, we first search for a corresponding small-scale map that satisfies the constraints best. Then, to support zoom interactions, we apply an algorithm that yields intermediate representations between both maps. An interesting idea for future research is to model constraints not only for a small-scale map but for a whole *sequence of maps* that is traversed when a user zooms from one scale to another scale. Probably, this approach can reduce abrupt changes in the map, which frequently occur with current software for map viewing.

3 Focus-and-Context Visualization

Information visualization studies the visual display of abstract data to help people extract useful information. Though cartography can be subsumed under this definition, it focuses on

(a) input map (b) result of fish-eye projection (c) our result

Figure 4: The road network of Boston (a) with a focus region (black circle) and two maps in which the focus region is enlarged by a factor of three. The map in (b) was generated with the fish-eye projection by Yamamoto et al. [YOT09] and the map in (c) with our method [8] that minimizes distortions.

the visualization of spatial data with maps. In contrast, researchers in information visualization often consider non-spatial data and other types of visualization than maps, for example, diagrams or drawings of graphs. Interesting problems in the intersection of information visualization and cartography are the schematization and distortion of geographic networks. In metro maps, geographical correctness is often sacrificed for the sake of a clearer, diagram-like network layout that allows users to quickly decide where to change trains [NW11]. Similarly, when visualizing a route a driver has to follow, long sections of the route that do not require crucial driving maneuvers (e.g., highway sections) can be drastically scaled down to save space for more critical sections (e.g., inner-city sections with many turns) [AS01]. Generally, the approach of attracting the user's attention to certain parts of a data set while de-emphasizing other parts is termed *focus-and-context visualization*.

A classical approach to focus-and-context visualization in cartography is to use a fish-eye projection [HSL02, YOT09]. This allows certain areas of a map to be displayed at a larger scale than other parts of the map, which is particularly helpful for users of small mobile devices. A tourist walking through an unknown city, for example, often needs a large-scale representation of his vicinity, but also more global context information to build a mental map of the city. Technically, fish-eye projections are often defined with a mapping function $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ that maps each point p in a map to a point $f(p)$ in a distorted output map. Usually, f displaces p away from a focus center c.

A common problem with fish-eye projections is that they introduce large distortions, see Fig. 4(a) and (b). This motivated us to develop a new method [8] that keeps the distortions small but still allows the scale in a focus region to be enlarged, see Fig. 4(c). More precisely, we designed a method for focus-and-context visualization of a geometric graph that represents a geographic network. Other than existing fish-eye techniques, our method is not based on a predefined mapping function. Instead, we applied our research cycle that strictly separates the modeling stage from the algorithmic solution. Since our work was motivated by the high distortion of fish-eye projections, we started off with formalizing a model of distortion. This model can be summarized as follows:

- We measure distortion *locally*, that is, for each node u of the given graph, we consider the star $S(u)$ of all edges incident to u.
- We compare $S(u)$ with its version $S'(u)$ in the output map. To this end, we scale and translate $S(u)$ such that it fits $S'(u)$ best. Now, the distortion is the square sum of the differences between the coordinates of the vertices of $S(u)$ and the corresponding vertices of $S'(u)$.
- To measure distortion *globally*, we sum the local measure over all nodes.

Our objective is to minimize the global distortion measure subject to a set of hard constraints. As a basic set of constraints, we require that a user-selected focus region is scaled up with a user-set factor and that the output map fits into the bounding box of the input map. Additionally, we forbid edge crossings in the output map.

In order to solve the problem, we have chosen an approach based on convex quadratic programming. This approach allows us to solve our problem (with the basic set of constraints) with an algorithm that is both efficient and exact. The constraint that forbids edge crossings had to be implemented such that some feasible solutions became excluded.

To evaluate our method, we have compared its results with those generated with the fish-eye technique proposed by Yamamoto et al. [YOT09]. On average, our method reduces the distortion (as defined in our model) by 75%. Also, our results visually appear to be far less distorted than those of the fish-eye technique. For example, in Fig. 4(b), we observe large distortions in the region north east of the focus region, which do not occur in Fig. 4(c).

An obvious idea to extend our method is to combine it with map generalization, that is, to adjust the level of detail of the map (locally) to the scale. To this end, we have developed a method that ranks roads according to their relevance for a user, where relevance is defined based on a probabilistic model of user movement [1]. Based on the ranks that we obtain, we decide which roads to display. Generally, our aim is to maximize the total rank of all displayed roads, but again we have to satisfy a set of hard constraints that, for example, restrict the number of selected edges. Algorithmically, we compute the ranks of the roads with the famous PageRank algorithm [PBMW99] that Google uses to rank web pages.

4 Map Labeling

Labeling geographic maps is a central problem in cartography. Labeling maps manually is a tedious task that, in the 1980's, was estimated to consume 50% of a map's production time [Mor80]. Until today, labeling static topographic maps has been automated to a high degree. New solutions are needed, however, to label maps that support continuous zooming or focus-and-context maps, in which a certain region is of particular importance.

Motivated by the fact that the focus and the context region of a focus-and-context map serve different purposes, we have developed a special labeling technique for such maps [2]. We regard map space as a resource that is more expensive in the focus region, thus text annotations and iconic labels for points of interest (POIs) or sites in the focus region should preferably be moved to the context region. The difficulty herein is that correspondences

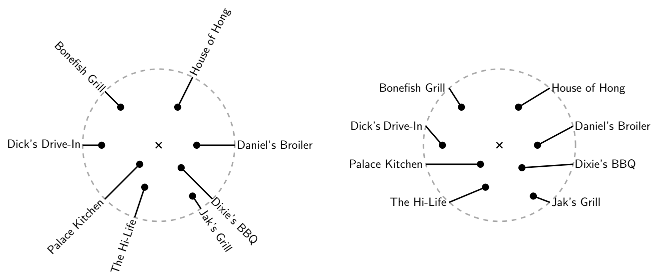

Figure 5: A labeling in the radial-leader model (left) and a labeling in the free-leader model (right).

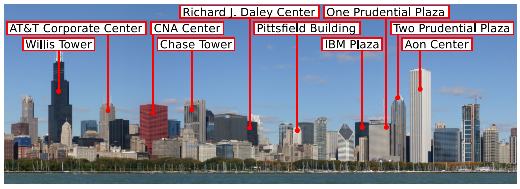

Figure 6: A labeling that uses our labeling model for panorama images.

between labels and sites have to remain clear. One possibility to achieve this is to display such correspondences as linear connections, which in the literature on map labeling are commonly termed *leaders*. Placing labels at the boundary of a map and connecting them via leaders with the map objects is commonly termed *boundary labeling* [BKSW07]. We do not place the labels at the boundary of the map but at the boundary B of a circular focus region. We have implemented this general idea with two labeling models, namely the *radial-leader model* and the *free-leader model*, see Fig. 5. In the radial-leader model, every leader is a straight-line segment that starts at some point on B and is oriented towards the focus center. In the free-leader model, a leader can be any straight-line segment that connects a point on B with a site. Though the free-leader model offers more flexibility, it can result in more visual clutter. For both labeling models, we have developed optimization algorithms that select a set of sites to become labeled and determine the label positions. Our algorithms allow us to optimize various objectives and to handle a set of hard constraints. In particular, we aim at labeling many sites while avoiding crossing leaders or overlapping labels. Moreover, we aim at short leaders and we consider the case that different sites have different priorities, which we model with weights—in this case, we maximize the total weight of all labeled sites rather than their bare number. An interesting extension of our model is that we also consider leaders that are Bézier curves.

In a second paper on boundary labeling [3], we have presented algorithms for labeling panorama images, which, since the start of the Google Street View project, have become a popular source of geographic information. Our algorithms use the fact that most of the interesting objects in panorama images lie below a horizontal line (e.g., the horizon) or a line that is monotone in the horizontal direction (e.g., the skyline of a city). Usually, the space above this line (the sky) is empty and can accommodate labels. We decided to use horizontal labels, to place them in the sky, and to connect them with their corresponding sites via vertical leaders, see Fig. 6. We forbid leaders and labels to intersect and labels to overlap each other and, as in our work on labeling focus regions, consider the case that we cannot label all sites. Again, we designed our algorithms to make a selection, either by maximizing the number of labeled sites or their total weight.

The question of which labeling model to use for a given application is still open. We are currently extending our algorithms for labeling panorama images to also consider tilted labels. We plan to evaluate the appropriateness of our models in a user study.

5 Map Matching

Map matching is the problem of finding a path in a given road data set that corresponds to a given trajectory. The trajectory is a sequence of positions with time stamps, which usually has been recorded with GPS. Several map matching algorithms have been developed that work well, even if the GPS measurements are noisy and the sampling rate of the GPS measurements is low [QON07, NK09]. A problem that has not been solved satisfactorily yet, however, is map matching with incomplete road data. This problem is particularly relevant as user-generated road data, for example, data from the Open Street Map project, is becoming increasingly popular. Though user-generated data is often more detailed than official or commercial data, its quality varies with the number of contributing users in a particular region, the time they spend on the project, and their skills. While some regions are mapped with large amounts of detail, other regions are sparsely mapped.

The existing map matching algorithms usually search for a path in a graph representing the road network. Among all paths in the graph, some algorithms find a path of minimum distance to the trajectory (according to some distance measure). If, for example, an important bridge is missing in the road data, such algorithms will generate long detours. In order to overcome this drawback, we have developed a map matching algorithm that introduces new edges if they are needed to generate an output path sufficiently similar to the trajectory [6]. Figure 7 shows a result of our method. Underlying our algorithm is a hidden Markov model (HMM). Generally, given a sequence of observations of a system, this allows the likelihood of a sequence of system states to be inferred. In our case, each observation is a GPS point and each system state is a possible position of the user, which for each time step (that is, each GPS measurement) we restrict to a discrete set of candidate positions. In contrast to the method of Newson and Krumm [NK09], which is also based on a HMM, we include, in each candidate set, a point that does not lie on an edge of the road network. This allows us to cope with roads missing in the data. To complete the HMM, we have modeled a set of conditional probabilities, namely, the probability density of observing a GPS point p if the user is at a candidate position a and the probability that a user moves to a candidate position b if he was in a candidate position a before. After having completed the HMM, we find a state sequence of maximum likelihood with the Viterbi algorithm [RJ86]. In order to evaluate our map matching algorithm, we tested it with OSM data and GPS trajectories that we recorded during four hikes. Since we knew the actual routes we had taken, a visual inspection of our data allowed us to determine that in 23 cases we had used paths not contained in the OSM data. To summarize, our map matching algorithm found the same 23 sections of our trajectory. A general problem with HMM-based map matching is, however, that the conditional probabilities depend on the user and the transportation mode applied. We therefore aim to extend our method to train the probabilities with reference data.

Since GPS sometimes fails in street canyons of dense urban areas, we have also considered to match data of sensors other than GPS with a map. In particular, we have developed an algorithm that localizes vehicles equipped with laser scanners or cameras [5]. These sensors allow salient features in the vicinity of a vehicle to be detected, for example, vertical pole-like objects such as traffic signs. Our algorithm, which is similar to the Viterbi algorithm, matches such features with point features in a reference map. An interesting direction for

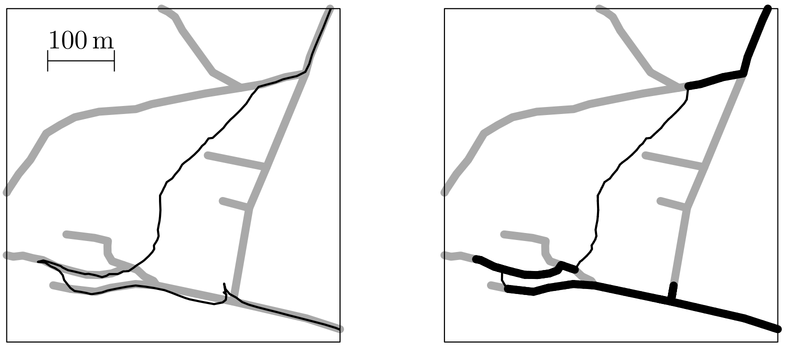

Figure 7: A trajectory (left, black) and a road data set (gray) with the output path of our map matching algorithm (right, black). The output path contains sections that are part of the road network (fat), but also sections that were not matched (thin).

future research is to integrate measurements of different sensors, for example, cameras and GPS, into a common probabilistic model.

6 Conclusion and Future Research

Modeling the criteria that make a good solution is the key to success in solving a problem. This may seem trivial. Many algorithms in GI science, however, are *not* motivated with a good model. Often, they have been designed to simulate procedures that humans might choose to reach a rather vaguely defined goal. In practice, such algorithms often yield satisfactory results. If they fail, however, diagnosing the reason of the failure is difficult. In contrast, we have argued—and exemplified on multiple problems—that a clear separation of the modeling stage, a solution by optimization, and the evaluation allows us to incrementally refine our models and to improve our methods. For each of these three stages of our approach, we now discuss our conclusions.

Modeling. Most of the models that we have used are based on constraints. While some of the constraints are hard (that is, they must be satisfied in any case) other constraints are soft (that is, we quantitatively measure their degree of violation). Generally, this requires appropriate measures and weights that model the importance of constraints. Most of our labeling algorithms [2, 3] as well as our algorithms for the selection of roads [1] allow us to select a set of features of maximum total weight (subject to a set of hard constraints). A user can set these weights arbitrarily, thus our algorithms are rather generally applicable. On the other hand, it is interesting to consider concrete weight settings that, for example, allow for a probabilistic interpretation. In our method for road selection [1], this was achieved with a probabilistic model of user movement. Similarly, our map matching algorithm [6] is based on a probabilistic model of observations and transitions in the road network. We conclude that we should avoid restrictive assumptions about the input of our algorithms, but also need better models to specify their parameters.

Our research cycle in Fig. 1 may suggest that finding a perfect model requires an infinite number of cycles of modeling, optimization, and evaluation. We should not aim, however, at exhaustive models, which subsume every criterion that possibly might be of relevance. Instead, it is important to identify the *core* of a problem and to grasp the criteria that matter.

Optimization. Most of the algorithms that we have discussed are both efficient and exact, have been implemented, and shown to be applicable on real-world instances of sizes that are relevant in practice. In multiple cases, this was achieved by dynamic programming [2, 3, 5], but also other approaches, for example, a greedy strategy [1] or convex quadratic programming [8], allowed us to solve some of our problems efficiently and exactly.

Algorithms engineering is a growing field of research, which aims at algorithms of a good performance in practice rather than at algorithms of a low asymptotic worst-case running time. For example, though integer programming is NP-hard (and thus the existing solvers for integer programs have an exponential asymptotic worst-case running time) it often allows moderately-sized problem instances to be solved within reasonable time. In fact, we successfully implemented this approach to simplify building footprints without introducing unwanted intersections [9].

To conclude, to tackle the diversity of problems in GI science successfully, we need to be aware of the large diversity of optimization algorithms. For GI scientists, techniques from algorithms engineering are particularly relevant.

Evaluation. We have evaluated all our methods through informal discussions with experts, colleagues, or potential users. Often, such discussions are valuable sources of ideas for possible improvements and future research directions. At some point it becomes important, however, to evaluate in a more formal, quantitative way. For each of our automatic methods, we have followed at least one of the following approaches of assessing a result:

(A1) We have assessed the differences between the result and a reference solution.
(A2) We have assessed the quality of the result with a quantitative measure of quality.
(A3) We have conducted user tests.

For example, we used approach A1 to assess the results of our map matching algorithm [6] and approach A2 to assess the focus-and-context maps that we generated with our optimization algorithm [8] or the fish-eye technique of Yamamoto et al. [YOT09]. In order to assess the symmetries found by our symmetry detector [4], we conducted user tests (approach A3) in which we asked the participants to visually detect symmetries; we then considered these symmetries as reference solutions and compared them with the symmetries found by our algorithm (approach A1).

Generally, it is important to conduct user tests for problems in which the criteria that make good solutions are not well understood—this particularly holds for problems of information visualization, for example, map labeling. Currently, we are preparing a user study to assess our algorithms for labeling panorama images. In this study, we will ask the participants to rate the quality of automatically generated labelings and to solve a certain task with a given labeling, for example, to quickly find the site of a given name.

References

[AS01] M. Agrawala and C. Stolte. Rendering effective route maps: Improving usability through generalization. In *Proc. 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 241–249. ACM, 2001.

[BKSW07] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry: Theory and Applications*, 36(3):215–236, 2007.

[EM01] R. Estkowski and J. S. B. Mitchell. Simplifying a polygonal subdivision while keeping it simple. In *Proc. 17th Annual Symposium on Computational Geometry*, SoCG '01, pages 40–49. ACM, 2001.

[HSL02] L. Harrie, L. T. Sarjakoski, and L. Lehto. A mapping function for variable-scale maps in small-display cartography. *Journal of Geospatial Engineering*, 4(2):111–123, 2002.

[HW07] L. Harrie and R. Weibel. Modelling the overall process of map generalization. In W. Mackaness, A. Ruas, and T. L. Sarjakoski, editors, *Generalisation of geographic information: Cartographic modelling and applications*, chapter 4, pages 67–88. Elsevier, 2007.

[KL06] M. Kada and F. Luo. Generalisation of building ground plans using half-spaces. In *Proc. ISPRS Commission IV Symposium on Geospatial Databases for Sustainable Development*, volume 36 (part 4) of *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2006.

[May98] H. Mayer. Model-generalization of building outlines based on scale-spaces and scale-space events. In *Proc. ISPRS Commission III Symposium on Object Recognition and Scene Classification from Multispatial and Multisensor Pixels*, volume 37 (part 3) of *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 530–536, 1998.

[Mor80] J. L. Morrison. Computer technology and cartographic change. In D.R.F. Taylor, editor, *The Computer in Contemporary Cartography*. Johns Hopkins University Press, 1980.

[NK09] P. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. In *Proc. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 336–343. ACM, 2009.

[NW11] M. Nöllenburg and A. Wolff. Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):626–641, 2011.

[PBMW99] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, 1999.

[QON07] M. A. Quddus, W. Y. Ochieng, and R. B. Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312–328, 2007.

[RJ86] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE Acoustics, Speech, and Signal Processing Magazine*, pages 4–15, January 1986.

[WD98] R. Weibel and G. Dutton. Constraint-based automated map generalization. In *Proc. 8th International Symposium on Spatial Data Handling*, pages 214–224, 1998.

[WJT03] J. M. Ware, C. B. Jones, and N. Thomas. Automated map generalization with multiple operators: a simulated annealing approach. *International Journal of Geographical Information Science*, 17(8):743–769, 2003.

[YOT09] D. Yamamoto, S. Ozeki, and N. Takahashi. Focus+glue+context: an improved fisheye approach for web map services. In *Proc. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 101–110. ACM, 2009.