Chapter 1

Introduction

Visualization of data is a basic and important topic; it helps to analyze or extract information from the data, as well as to communicate it. If we restrict ourselves to spatial or geographical data, then we are talking about *cartography*, and, in particular, about the generation of maps.

The design of a map is a very complex task. A cartographer cannot just project the data onto a piece of paper, but he has to worry about its readability. The way to improve the quality of the map is by using so-called *white lies*. For example, in a road map of Europe, if the thickness of a road would be proportional to its width in real world, the user would not notice it on the map. Therefore, the cartographer needs to make it thicker on the map than it would be otherwise according to the map scale.

Furthermore, the design of a map is not only a complex task, but it also involves subjective decisions. For example, the cartographer has to decide what information is not relevant and can be omitted from the map, how to improve its readability in cluttered areas, where to put labels with relevant features, what legend to use, and so on. "How to Lie with Maps", by Monmonier [103], is a classical, nice-to-read book on how these decisions affect the map.

1.1 Automated cartography

The appearance of computers in the 20th century has affected many fields, and cartography has not been kept aside of this revolution, leading to the so-called automated cartography research field. Initially the topics consisted of automating certain tasks originally done by cartographers; later, the area mixed with the research on geographic information systems.

Automated generalization is, probably, the most recurrent topic within automated cartography. According to Heywood et al. [87], *generalization* is "the process by which information is selectively removed from a map in order to simplify pattern without distortion of overall content". Another relevant topic is automated *labelling* of a map, that is, placing labels with the entities of a map,

Figure 1.1: Detail of the underground map of London, a classical example of a schematic map.

such as names of cities or rivers. These two processes are done so often during the design of a map that automating them saves hours of work.

In recent years, the concept of interactive maps and maps on demand, that is, maps that are tailored to the user's wishes or necessities, are getting an increased interest, mostly due to the widespread use of Internet. These maps include route maps based on queries as a special case. Given that there is a lot of demand for such maps, their construction has to be fully automated.

One of the types of maps that allow for automated construction is the schematized map, which inspired most of the research presented in this thesis. In a *schematic map*, a set of nodes and their connections are displayed in a highly simplified form, since the precise shape of the connections and position of the nodes is not so important; see Figure 1.1. To preserve the recognizability for map readers, the approximate layout must be maintained, however.

Cartograms are another interesting type of map that gives rise to challenging computational problems; see Figure 1.2. A *cartogram* is a map in which the size of each entity is proportional to some value associated with the entity [35, Chapter 14][47, Chapter 10]. *Area cartograms* are the most common example, in which the area of each region is proportional to some function of the region, like for example, its population. In *linear cartograms*, we want to display a network in such a way that the length of a connection is related to some characteristic of the connection. In ordinary maps, this length is correlated (through a planar projection of the sphere) to the length of the connection in the real world. However, we may be interested in showing, for instance, the travel time for each connection, or the amount of traffic on each connection. Part of this thesis is concerned with linear cartograms.

We have analyzed some of the steps, or considerations, that cartographers face when designing schematic maps and linear cartograms. We have abstracted them, and converted them into mathematically formulated computational prob-

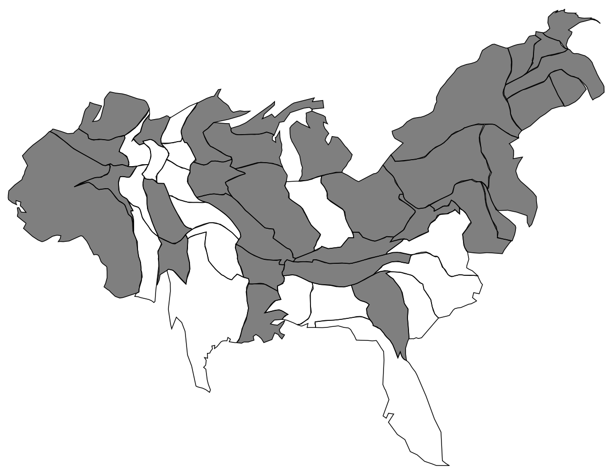

Figure 1.2: Cartogram of the United States based on the electoral votes in the 1992 presidential elections (from Edelsbrunner and Waupotitsch [61]).

lems. The abstraction process is fundamental to be able to deal with the problems in the context of computational geometry. Furthermore, this allows that the considered problems find applications not only in cartography, but also in robotics, visualization of data, and graph drawing, to name a few other research areas.

We want to stress that our research is, by no means, trying to completely solve the problem of automatically generating maps, but is aimed at providing tools that will successfully perform specific tasks that cartographers may find useful when designing a cartographic network. This is the main purpose of the research contained in this thesis.

In the following sections, we describe the context in which the results of this thesis are embedded: computational geometry. Then, we discuss the related work that has been done. At the end of this chapter, we give a thesis overview, explaining the computational problems that are analyzed in subsequent chapters, and discussing their motivation within cartography.

1.2 Computational geometry

We have considered the problems from a computational geometry perspective. *Computational geometry* is the branch of algorithmics that deals with problems with a strong geometric flavour. Generally, the problems are considered in constant-dimensional spaces. In fact, most of the research has been done in two or three dimensions because of the numerous applications.

In computational geometry, effort is put into designing efficient algorithms, where efficiency is measured both in the asymptotic running time and in the required memory space.

The field started in the seventies, and the first books on the topic were by Edelsbrunner [58], and by Preparata and Shamos [110]. The book by de Berg et al. [45] contains (more than) the appropriate background you need to understand this thesis. A more programming oriented book of computational geometry is by O'Rourke [108], and a more discrete and combinatorial slant can be found in the book by Matoušek [100].

Some of the problems that we considered are instances of *geometric optimization* problems, that is, optimization problems with an essential geometric component. In this context, the concepts of approximation algorithms and approximation schemes play an important role. A good introduction to these concepts with a geometric flavour is by Bern and Eppstein [20]. The surveys about geometric optimization by Agarwal and Sharir [6], and by Arora [10] are also relevant.

Besides cartography and geographical information systems, the field of computational geometry also shares interests with other research areas: data structures, motion planning, virtual environments, computational biology, graph drawing, discrete and combinatorial geometry, computer graphics, computer vision, shape matching and recognition, computational topology, and many more.

1.3 Related work

In this section, we give an overview of some of the research that has been done in automated cartography. We start with general references, and then we concentrate on more relevant work on schematic maps and linear cartograms. In each chapter, we give relevant references for the specific problem that we consider.

A good introduction to cartography is the book by Dent [47], and a more informal one is the one by Monmonier [103]. Recent research results on automated cartography are presented at the International Cartographic Conference, Auto-Carto (Proceedings of the International Symposium on Computer-Assisted Cartography), the International Symposium on Spatial Data Handling, the Dagstuhl seminars on Computational Cartography (September 1999, May 2001, September 2003), and some other conferences.

Schematic maps have become a quite standard way to convey information, and therefore the work on this area has increased over the past years. For example, the PhD thesis by Avelar [11] and an ArcGIS Schematic package, produced by ESRI [1], have appeared recently. However, the automated construction of schematic maps has already been studied in earlier papers. Elroi [65, 66, 67] describes an approach where the paths are first simplified, then they are placed on a grid to assure restricted orientations, and then crowded areas are locally enlarged to avoid regions with too high density. No technicalities of the algorithm or running-time analysis are given in these papers. Avelar and Müller [13] describe an iterative procedure that attempts to rotate all links of an input network into one of the four main orientations. No upper bound on the running time can be given. Also, the output may contain links in other orientations than the desired ones. Barbowsky, Latecki and Richter [15] have used an iterative discrete curve evolution, based on a local measure, to simplify the curves while keeping the local spatial ordering. All these methods rely on iterative approaches where the connections should converge to the major orientations, while displacing the junctions. For this type of techniques, it is difficult or even impossible to guarantee the convergence of the procedure.

The paper by Neyer [107] describes a line-simplification algorithm where the final paths must have links in one of c given orientations only, and stay close enough to the original path. The algorithm minimizes the number k of links in the output in $O(n^2 k \log n)$ time, and when it is applied to disjoint paths, the output may have intersections, which change the topology of the map. In a paper by Raghavan et al. [112], a wiring is made by connecting pairs of points by non-intersecting, 2-link orthogonal paths. This can be seen as a schematic map where only two different schematic paths are possible for each pair of points. The problem can be solved in $O(n \log n)$ time, as shown by Imai and Asano [90], but the model is too restrictive and the relative positions in the resulting network may be different than in the original network, making the recognizability of the features harder. For depicting the schematic paths, the work by Duncan et al. [55] is also worth mentioning, where paths are redrawn in the same homotopic class and with maximal separation.

We are not aware of any research done for the construction of linear cartograms. However, there is much research done on the problem of drawing graphs with specified edge lengths. For cartographic purposes we have the natural restriction that the drawing should be planar, and then the problem has been considered by Di Battista and Vismara [52], Eades and Wormald [56], and Whitesides [120].

If we drop the planarity condition, then the problem has been studied in the fields of computational geometry [41, 69, 115, 122], rigidity theory [40, 83, 91], sensor networks [36, 114], and structural analysis of molecules [19, 42, 84]. It appears frequently when only distance information is known about a given structure, such as the atoms in a protein [19, 42, 84] or the nodes in an ad-hoc wireless network [36, 111, 114].

Other research related to schematic maps is map generalization for road networks and line simplification. However, the objectives in such problems are quite different. In general one does not consider achieving a given number of links per path and/or having restricted orientations, but instead tries to keep the main features of each path while reducing the number of coordinates to describe it. One of the most popular algorithms for simplification of paths is by Douglas and Peucker [54], but much more research has been done; see the notes by Weibel [119] for a general reference.

Also related is the research on VLSI layout design [76, 97], where the number of edges in the output is generally not considered critical, and research on graph drawing [51, 118], where the positions of the endpoints are usually not fixed. A recent, related topic that is relevant to schematization is the rendering of particular routes under queries [7, 12]. In this case, the paths are also simplified, but there is more flexibility to distort the input map because only the objects in the surroundings of the route are displayed. Another interesting research topic within cartographic networks is using graphs for displaying the train interconnection data in the railway network. This problem has been treated by Brandes et al. [23, 24].

1.4 Thesis overview

Consider a drawing of a transportation network, such as road or railway map. To design a schematic version of it, or to produce a linear cartogram, the cartographer performs actions that can be classified into three types: displace the nodes (or junctions) of the network, modify the shape of the connections, or both together. Let us analyze these actions independently.

1.4.1 Displacing nodes

Consider the actions that displace nodes, which from now on we will imagine to be points. Independent of the reason to displace a point, its new position has to be close to the original one. A natural way to abstract this is by restricting the new position of a point to be inside a fixed region around the original position, and then the problem becomes: given a collection $S_1, \ldots, S_n$ of regions in $\mathbb{R}^2$, find a "good"placement $p_1, \ldots, p_n$ with $p_i \in S_i$. There are two issues we have to handle: what is a "good" placement, and which regions are the appropriate ones?

The cartographer displaces the points with respect to their original position to improve the quality of the map under construction. So, let us assume that we have a quality function $Q : S_1 \times \cdots \times S_n \to \mathbb{R}$ that measures the quality of a placement $(p_1, \ldots, p_n) \in S_1 \times \cdots \times S_n$. We can then say that our objective would be to provide a best placement, that is, points $p_1^*, \ldots, p_n^*$, with $p_i \in S_i$ such that

$$Q(p_1^*, \ldots, p_n^*) = \max_{(p_1,\ldots,p_n)\in S_1\times\cdots\times S_n} Q(p_1, \ldots, p_n).$$

A quality function that evaluates all the aspects of the placement would be very complex, and becomes infeasible from the computational point of view. Instead, we have considered two particular quality functions that play a fundamental role in the design of schematic maps for networks.

The first quality function captures the fact that, in a schematic map, the connections between nodes are usually displayed by a small number of straight-line segments, and usually horizontal, vertical, or also diagonal (45 or 135 degrees) segments are preferable; see Figure 1.1. This leads to the problem of, given a graph on the points $p_1, \ldots, p_n$, find a placement that maximizes the number of straight-line edges that have horizontal, vertical, and perhaps diagonal orientation. Because a picture is worth a thousand words, take a look at Figure 1.3. We refer to this problem as the aligning-points problem. We show that finding a best placement under this criterion is NP-hard, and provide several approximation schemes whose performance depend on various parameters. These results, previously published in [34], are described in Chapter 2.

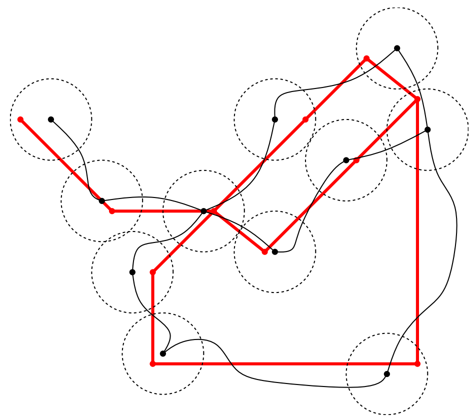

Figure 1.3: The aligning-points problem. In thin black, the original network is shown. The disk around each point represents the region where it can be moved. We want to maximize the number of edges with horizontal, vertical, or diagonal orientation. In this case, a possible optimal solution is the thick network shown.

Regarding the second quality function that we will consider, a rule of thumb tells that the readability of the map improves as the separation between its features increases. This leads to the problem of maximizing the distance between the placed points, and, in particular, the problem of maximizing the distance between any pair of points as much as possible. Take a look at Figure 1.4 to see an example. We refer to this problem as the spreading-points problem. The problem was shown to be NP-hard by Baur and Fekete [16], and Fiala, Kratochvíl, and Proskurowski [71]. When the regions are disks, we provide efficient approximation algorithms with constant-factor approximation. These results were previously published in [28], and we report them in Chapter 3.

Let us remark that in these problems, we did not deal with the actual choice of the regions, or which regions are preferable, but we assume that the regions are already given. Under this assumption, both problems belong to the area of geometric optimization.

1.4.2 Deforming connections

Let us consider one of the connections in the original map, and the corresponding connection in the schematic map. If a road passes to the North of an important

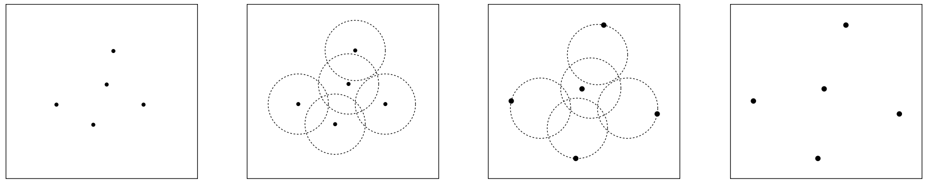

Figure 1.4: The spreading-points problem. On the left, we have a collection of points. We consider a disk around each of them (center left), and then we allow each point to be displaced within its disk. The objective is to maximize the distance of the closest pair, and for the depicted example, we would get the situation shown on the right.

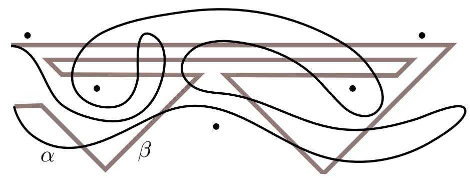

Figure 1.5: Are the paths α and β homotopic, that is, can α be continuously deformed into β without touching any of the points?

city, we do not want the schematized version to pass to the South of that city, but to keep the relative positions. If we think of the construction of the schematic path as a continuous deformation of the original path, we can formulate this requirement as follows: the original path is transformed into the schematic one in a continuous way, fixing its endpoints and without crossing any "important point", where "important point" refers to a city or any other point feature whose relative position with respect to the path we want to maintain.

Homotopy of paths is a well-known concept in topology that captures this idea. Therefore, to understand the problem and extract its features, we consider the basic decision problem: given two paths and a set of "important points" P, can we deform one path into the other one without passing over any point in P? Or, using the topological terminology, are the two paths homotopic in the plane minus P? See Figure 1.5 for an example. If both the paths and the point set P have complexity n, then we can decide it in $O(n \log n)$ time in case the paths are simple, and in $O(n^{3/2} \log n)$ time in case the paths self-intersect. Lower bounds for both cases (simple and self-intersecting paths) are also presented, and, in particular, they show that the algorithm for the simple case is asymptotically optimal. These results were previously published in [32], and we present them in Chapter 4.

Next we return to the problem of constructing a schematic map. We assume that the input is a planar embedding of a graph consisting of polygonal paths between specified points called endpoints. We are interested in producing another planar embedding where all endpoints have the same positions, and every path is displayed as a two-link or three-link path where links are restricted to certain orientations; see Figure 1.6. Furthermore, as discussed above, the output map should be equivalent to the input map in the sense that a continuous deformation exists such that no path passes over an endpoint during the transformation. For maps whose paths do not intersect, this equivalence implies that the cyclic order of paths around endpoints is maintained.

Chapter 5 contains our formalization of the problem, a computationally optimal algorithm to deal with it, and discusses the quality of the schematic maps given by our implementation. These results were previously published in [30], improving our previous works [29, 33]. Figure 1.6 has been produced by our implementation, and more examples are shown in the figures of Chapter 5, where we also discuss extensions of the algorithm.

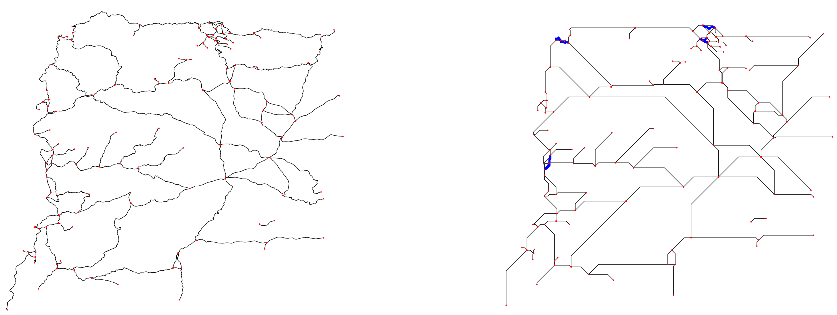

Figure 1.6: Northwest of the Iberic Peninsula. Left: the original map. Right: the schematized version made by the implementation described in Chapter 5.

1.4.3 Doing everything at the same time

To construct a schematic map by displacing points and deforming paths, both simultaneously, is too complex. Let us be more specific: so far, no algorithm provides provable results in this context. Some authors [11, 12, 13, 15, 65, 66, 67] give iterative processes that do everything at once, and that in general may give nice, pleasant results. However, there is no provable guarantee that the output will be optimal, or even good. For example, it may well happen that the algorithm cannot modify anything in the original map. More specifically, no combinatorial algorithms have been proposed that can handle connections and nodes together.

If there are no such theoretical, provable results, where is then the difficulty? The basic issue is that many simpler problems are already computationally intractable. For example, consider the problem of generating a linear cartogram.

The construction of such a map can be modeled by defining the length of each edge appropriately and trying to realize the graph with these edge lengths. So we can abstract the generation of a linear cartogram to the following natural problem: given a graph G, can we construct a planar straight-line embedding of G where the edges have a prescribed length? Observe that in real-life applications, we would also like to keep some resemblance with the original network, and so we may restrict where the vertices of the graph can be embedded. However, as we show in Chapter 6, the problem is already NP-hard without this restriction; that is, there is little hope that efficient algorithms will exist for this problem. This is true even for a very restricted class of graphs, such as 3-connected, bounded degree, and bounded face degree graphs. This problem is discussed in Chapter 6, whose contents were previously published in [31].

Bibliography

[1] Webpage: `http://www.esri.com/software/arcgis/arcgisxtensions/schematics/index.html`.

[2] `http://www.cs.uu.nl/archive/sw/schematic-map/`.

[3] M. Abellanas, F. Hurtado, and P.A. Ramos. Structural tolerance and Delaunay triangulation. *Information Processing Letters*, 71:221–227, 1999.

[4] P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29:912–953, 1999.

[5] P. K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22(4):794–806, 1993.

[6] P. K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.*, 30:412–458, 1998.

[7] M. Agrawala and C. Stolte. Rendering effective route maps: Improving usability through generalization. In *Proc. SIGGRAPH 2001*, pages 241–250, 2001. `http://graphics.stanford.edu/papers/routemaps/`.

[8] R. Aharoni and P. Haxell. Hall's theorem for hypergraphs. *Journal of Graph Theory*, 35:83–88, 2000.

[9] M. A. Armstrong. *Basic Topology.* McGraw-Hill, London, UK, 1979.

[10] S. Arora. Approximation schemes for NP-hard geometric optimization problems: A survey, 2002. `http://www.cs.princeton.edu/~arora/pubs/arorageo.ps`.

[11] S. Avelar. *Schematic Maps on Demand: Design, Modeling and Visualization.* PhD thesis, Swiss Federal Institute of Technology, Zurich, 2002.

[12] S. Avelar and R. Huber. Modeling a public transport network for generation of schematic maps and location queries. In *Proc. 20th Int. Cartographic Conference*, pages 1472–1480, 2001.

[13] S. Avelar and M. Müller. Generating topologically correct schematic maps. In *Proc. 9th Int. Symp. on Spatial Data Handling*, pages 4a.28–4a.35, 2000.

[14] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41:153–180, 1994.

[15] T. Barbowsky, L.J. Latecki, and K. Richter. Schematizing maps: Simplification of geographic shape by discrete curve evolution. In *Spatial Cognition II, LNAI 1849*, pages 41–48, 2000.

[16] C. Baur and S.P. Fekete. Approximation of geometric dispersion problems. *Algorithmica*, 30:451–470, 2001. A preliminary version appeared in *APPROX'98*.

[17] M. Ben-Or. Lower bounds for algebraic computation trees. In *Proc. 15th Annu. ACM Sympos. Theory Comput.*, pages 80–86, 1983.

[18] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28(9):643–647, September 1979.

[19] B. Berger, J. Kleinberg, and T. Leighton. Reconstructing a three-dimensional model with arbitrary errors. In *Proc. 28th Annu. ACM Sympos. Theory Comput.*, pages 449–458, May 1996.

[20] M. W. Bern and D. Eppstein. Approximation algorithms for geometric problems. In D. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 8, pages 296–345. PWS Publishing, 1996.

[21] S. Bespamyatnikh. Computing homotopic shortest paths in the plane. In *SODA03*, pages 609–617, 2003. To appear in J. of Algorithms.

[22] F. Brandenberg, D. Eppstein, M.T. Goodrich, S.G. Kobourov, G. Liotta, and P. Mutzel. Selected open problems in graph drawing. In *Graph Drawing (Proc. GD'03)*, LNCS, 2003. To appear.

[23] U. Brandes, G. Shubina, R. Tamassia, and D. Wagner. Fast layout methods for timetable graphs. In J. Marks, editor, *Graph Drawing, Proceedings of 8th International Symposium, GD 2000*, volume 1984 of *Lecture Notes in Computer Science*, pages 127–138. Springer-Verlag, 2001.

[24] U. Brandes and D. Wagner. Using graph layout to visualize train interconnection data. *Journal of Graph Algorithms and Applications*, 4(3):135–155, 2000. `http://www.cs.brown.edu/publications/jgaa/volume04.html`.

[25] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. A strong and easily computable separation bound for arithmetic expressions involving radicals. *Algorithmica*, 27(1):87–99, 2000.

[26] C. Burnikel, S. Funke, K. Mehlhorn, S. Schirra, and S. Schmitt. A separation bound for real algebraic expressions. In F. Meyer auf der Heide, editor, *Algorithms — ESA 2001, 9th Annual European Symposium, Aarhus, Denmark, August 28–31, 2001, Proceedings*, volume 2161 of *Lecture Notes in Computer Science*, pages 254–265. Springer-Verlag, 2001.

[27] B. Buttenfield. Treatment of the cartographic line. *Cartographica*, 22:1–26, 1985.

[28] S. Cabello. Approximation algorithms for spreading points. Technical report UU-CS-2003-040, Available at `http://www.cs.uu.nl/research/techreps/UU-CS-2003-040.html`, 2003.

[29] S. Cabello, M. de Berg, S. van Dijk, M. van Kreveld, and T. Strijk. Schematization of road networks. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 33–39, 2001.

[30] S. Cabello, M. de Berg, and M. van Kreveld. Schematization of networks. Technical report, University Utrecht. Submitted to journal, 2002.

[31] S. Cabello, E.D. Demaine, and G. Rote. Planar embeddings of graphs with specified edge lengths. In *Graph Drawing 2003*, LNCS, 2004. To appear.

[32] S. Cabello, Y. Leo, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. *Discrete & Computational Geometry*. To appear. A preliminary version appeared in *SoCG'02*.

[33] S. Cabello and M. van Kreveld. Schematic networks: an algorithm and its implementation. In D.E. Richardson and P. van Oosterom, editors, *Advances in Spatial Data Handling*, pages 475–486. Springer, 2002.

[34] S. Cabello and M. van Kreveld. Approximation algorithms for aligning points. *Algorithmica*, 37:211–232, 2003. A preliminary version appeared in *SoCG'03*.

[35] J. Campbell. *Map Use and Analysis*. McGraw-Hill, Boston, 4th edition, 2001.

[36] S. Čapkun, M. Hamdi, and J. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Hawaii International Conference on System Sciences*, pages 3481–3490, January 2001.

[37] B. Chandra and M. M. Halldórsson. Approximation algorithms for dispersion problems. *J. Algorithms*, 38:438–465, 2001.

[38] B. Chazelle. An algorithm for segment-dragging and its implementation. *Algorithmica*, 3:205–221, 1988.

[39] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.

[40] R. Connelly. On generic global rigidity. In P. Gritzman and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 147–155. AMS Press, 1991.

[41] C. Coullard and A. Lubiw. Distance visibility graphs. *Internat. J. Comput. Geom. Appl.*, 2(4):349–362, 1992.

[42] G. M. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformation.* John Wiley & Sons, 1988.

[43] F.H. Croom. *Basic Concepts of Algebraic Topology.* Springer Verlag, Berlin, 1978.

[44] M. de Berg and O. Schwarzkopf. Cuttings and applications. *Internat. J. Comput. Geom. Appl.*, 5:343–355, 1995.

[45] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, Berlin, Germany, 2nd edition, 2000.

[46] M. de Berg, M. van Kreveld, and S. Schirra. Topologically correct subdivision simplification using the bandwidth criterion. *Cartography and GIS*, 25:243–257, 1998.

[47] B.D. Dent. *Cartography: Thematic Map Design.* McGraw-Hill, 5th edition, 1999.

[48] O. Devillers, G. Liotta, F. P. Preparata, and R. Tamassia. Checking the convexity of polytopes and the planarity of subdivisions. *Comput. Geom. Theory Appl.*, 11:187–208, 1998.

[49] T. K. Dey and S. Guha. Transforming curves on surfaces. *Journal of Computer and System Sciences*, 58:297–325, 1999.

[50] T. K. Dey and H. Schipper. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete Comput. Geom.*, 14:93–110, 1995.

[51] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing.* Prentice Hall, Upper Saddle River, NJ, 1999.

[52] G. Di Battista and L. Vismara. Angles of planar triangular graphs. *SIAM Journal on Discrete Mathematics*, 9(3):349–359, 1996. A preliminary version appeared in *STOC'93.*

[53] R. Diestel. *Graph Theory.* Springer-Verlag, New York, 2nd edition, 2000.

[54] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer*, 10(2):112–122, December 1973.

[55] C.A. Duncan, A. Efrat, S.G. Kobourov, and C. Wenk. Drawing with fat edges. In *Graph Drawing 2001*, volume 2265 of *LNCS*, pages 162–177, 2002.

[56] P. Eades and N. Wormald. Fixed edge length graph drawing is NP-hard. *Discrete Appl. Math.*, 28:111–134, 1990.

[57] H. Edelsbrunner. A note on dynamic range searching. *Bull. EATCS*, 15:34–40, 1981.

[58] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Heidelberg, West Germany, 1987.

[59] H. Edelsbrunner, Leonidas J. Guibas, J. Hershberger, R. Seidel, Micha Sharir, J. Snoeyink, and Emo Welzl. Implicitly representing arrangements of lines or segments. *Discrete Comput. Geom.*, 4:433–466, 1989.

[60] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9(1):66–104, 1990.

[61] H. Edelsbrunner and E. Waupotitsch. A combinatorial approach to cartograms. *Comput. Geom. Theory Appl.*, 7:343–360, 1997.

[62] A. Efrat, A. Itai, and M. J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001.

[63] A. Efrat, M. J. Katz, F. Nielsen, and M. Sharir. Dynamic data structures for fat objects and their applications. *Comput. Geom. Theory Appl.*, 15:215–227, 2000. A preliminary version appeared in *WADS'97, LNCS 1272*.

[64] A. Efrat, S.G. Kobourov, and A. Lubiw. Computing homotopic shortest paths efficiently. In *10th Annual European Symposium, ESA'02*, volume 2461 of *LNCS*, pages 411–423, 2002.

[65] D. Elroi. Designing a network line-map schematization software enhancement package. In *Proc. 8th Ann. ESRI User Conference*, 1988. `http://www.elroi.com/fr2_publications.html`.

[66] D. Elroi. GIS and schematic maps: A new symbiotic relationship. In *Proc. GIS/LIS'88*, 1988. `http://www.elroi.com/fr2_publications.html`.

[67] D. Elroi. Schematic views of networks: Why not have it all. In *Proc. of the 1991 GIS for Transportation Symposium*, pages 59–76, 1991. `http://www.elroi.com/fr2_publications.html`.

[68] J. Erickson. New lower bounds for Hopcroft's problem. *Discrete Comput. Geom.*, 16:389–418, 1996.

[69] H. Everett, C. T. Hoàng, K. Kilakos, and M. Noy. Distance segment visibility graphs. Manuscript, 1999. `http://www.loria.fr/~everett/publications/distance.html`.

[70] S.P. Fekete and H. Meijer. Maximum dispersion and geometric maximum weight cliques. To appear in *Algorithmica* 38(3), 2004.

[71] J. Fiala, J. Kratochvíl, and A. Proskurowski. Geometric systems of disjoint representatives. In *Graph Drawing, 10th GD'02, Irvine, California*, number 2528 in Lecture Notes in Computer Science, pages 110–117. Springer Verlag, 2002.

[72] J. Fiala, J. Kratochvíl, and A. Proskurowski. Systems of sets and their representatives. Technical Report 2002-573, KAM-DIMATIA, 2002. Available at `http://dimatia.mff.cuni.cz/`.

[73] H.N. Gabow. A matroid apporach to finding edge connectivity and packing arborescences. *J. Comput. Systems Sci.*, 50:259–273, 1995.

[74] S. Gao, M. Jerrum, M. Kaufmann, K. Mehlhorn, W. Rülling, and C. Storb. On continuous homotopic one layer routing. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 392–402, 1988.

[75] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, NY, 1979.

[76] S.H. Gerez. *Algorithms for VLSI Design Automation.* John Wiley & Sons, Chichester, 1999.

[77] A. V. Goldberg, S. A. Plotkin, and P. M. Vaidya. Sublinear-time parallel algorithms for matching and related problems. *Journal of Algorithms*, 14:180–213, 1993. A preliminary version appeared in *FOCS'88.*

[78] M. T. Goodrich and R. Tamassia. Dynamic ray shooting and shortest paths in planar subdivisions via balanced geodesic triangulations. *J. Algorithms*, 23:51–73, 1997.

[79] J. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity.* American Mathematical Society, 1993.

[80] R. Grossi and E. Lodi. Simple planar graph partition into three forests. *Discrete Applied Mathematics*, 84:121–132, 1998.

[81] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48:798–859, 2001.

[82] A. Hatcher. *Algebraic Topology.* Cambridge University Press, 2001.

[83] B. Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput.*, 21(1):65–84, February 1992.

[84] B. Hendrickson. The molecule problem: Exploiting structure in global optimization. *SIAM J. on Optimization*, 5:835–857, 1995.

[85] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl.*, 4:63–98, 1994.

[86] J. Hershberger and Subhash Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18:403–431, 1995.

[87] I. Heywood, S. Cornelius, and S. Carver. *An Introduction to Geographical Information Systems*. Addison Wesley Longman, New York, 1998.

[88] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32:130–136, 1985.

[89] J. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973.

[90] H. Imai and T. Asano. Efficient algorithms for geometric graph search problems. *SIAM J. Comput.*, 15(2):478–494, 1986.

[91] B. Jackson and T. Jordán. Connected rigidity matroids and unique realizations of graphs. Manuscript, March 2003.

[92] K. Kedem, R. Livne, J. Pach, and Micha Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.

[93] D. E. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM J. on Discrete Mathematics*, 5(3):422–427, August 1992.

[94] C. Li and C. Yap. A new constructive root bound for algebraic expressions. In *Proc. 12th Annu. ACM–SIAM Sympos. Discrete Algorithms*, pages 496–505, 2001.

[95] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[96] F. M. Maley. *Single-Layer Wire Routing and Compaction*. MIT Press, Cambridge, MA, 1990.

[97] F.M. Maley. Testing homotopic routability under polygonal wiring rules. *Algorithmica*, 15:1–16, 1996.

[98] J. Matoušek. More on cutting arrangements and spanning trees with low crossing number. Technical Report B-90-2, Fachbereich Mathematik, Freie Universität Berlin, Berlin, 1990.

[99] J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.*, 8:315–334, 1992.

[100] J. Matousek. *Lectures on Discrete Geometry*. Springer Verlag, Berlin, 2002.

[101] N. Megiddo. Combinatorial optimization with rational objective functions. *Math. Oper. Res.*, 4:414–424, 1979.

[102] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983.

[103] M. Monmonier. *How to Lie with Maps*. The University of Chicago Press, Chicago, 1991.

[104] C. W. Mortensen. Fully-dynamic two dimensional orthogonal range and line segment intersection reporting in logarithmic time. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 618–627. Society for Industrial and Applied Mathematics, 2003.

[105] C. W. Mortensen. Personal communication, 2003.

[106] J. R. Munkres. *Topology: A first course*. Prentice Hall, Englewood Cliffs, NJ, 1975.

[107] G. Neyer. Line simplication with restricted orientations. In *Algorithms and Data Structures, WADS'99*, volume 1663 of *LNCS*, pages 13–24, 1999.

[108] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 2nd edition, 1998.

[109] L. Palazzi and J. Snoeyink. Counting and reporting red/blue segment intersections. *CVGIP: Graph. Models Image Process.*, 56(4):304–311, 1994.

[110] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 3rd edition, October 1990.

[111] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proceedings of 6th Annual International Conference on Mobile Computing and Networking*, pages 32–43, Boston, MA, August 2000.

[112] R. Raghavan, J. Cohoon, and S. Sahni. Single bend wiring. *J. Algorithms*, 7:232–257, 1986.

[113] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.

[114] C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad-hoc wireless sensor networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 2037–2040, Salt Lake City, UT, May 2001.

[115] J. B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proc. 17th Allerton Conf. Commun. Control Comput.*, pages 480–489, 1979.

[116] A. Schrijver. A course in combinatorial optimization. Lecture Notes. Available at `http://homepages.cwi.nl/~lex/files/dict.ps`, 2003.

[117] B. Simons. A fast algorithm for single processor scheduling. In *Proc. 19th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 246–252, 1978.

[118] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.

[119] R. Weibel. Generalization of spatial data: Principles and selected algorithms. In *Lecture notes from CISM Advanced School on Algorithmic Foundations of Geographic Information Systems*, pages 99–152. Springer-Verlag, 1996.

[120] S. Whitesides. Algorithmic issues in the geometry of planar linkage movement. *Australian Computer Journal*, 24(2):42–50, May 1992.

[121] G. Woeginger. Personal communication, 2003.

[122] Y. Yemini. Some theoretical aspects of position-location problems. In *Proc. 20th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 1–8, 1979.