59 GEOGRAPHIC INFORMATION SYSTEMS

Marc van Kreveld

INTRODUCTION

Geographic information systems (GIS) facilitate the input, storage, manipulation, analysis, and visualization of geographic data. Geographic data generally has a location, size, shape, and various attributes, and may have a temporal component as well. Geographical analysis is important for a GIS. It includes combining different spatial themes, relating the dependency of phenomena to distance, interpolating, studying trends and patterns, and more. Without analysis, a GIS could be called a spatial database.

Not all aspects of GIS are relevant to computational geometry. Human-computer interaction, and legal aspects of GIS, are also considered part of GIS research. This chapter focuses primarily on those aspects that are susceptible to algorithms research. Even here, the approach taken within GIS research is different from the approach a computational geometer would take, with much less initial abstraction of the problem, and less emphasis on theoretical efficiency. The GIS research field is multi-disciplinary: it includes researchers from geography, geodesy, cartography, and computer science. The research areas geodesy, surveying, photogrammetry, and remote sensing primarily deal with the data input, storage, and correction aspects of GIS. Cartography mainly concentrates on the visualization aspects.

Section 59.1 deals with spatial data structures important to GIS. Section 59.2 discusses the most common spatial analysis methods. Section 59.3 discusses the visualization of spatial data, in particular automated cartography. Section 59.4 deals with Digital Elevation Models (DEMs) and their algorithms. Section 59.5 discusses algorithmic results on trajectory data. Section 59.6 overviews miscellaneous other occurrences of geometric algorithms in GIS. Section 59.7 discusses the most important contributions that can be made from the computational geometry perspective to research problems in GIS.

59.1 SPATIAL DATA STRUCTURES

GIS store different types of data separately, such as land cover, elevation, and municipality boundaries. Therefore, each such data set is stored in a separate data structure that is tailored to the data, both in terms of representation and searching efficiency.

Geometric data structures for intersection, point location, and windowing are a mainstream topic in computational geometry, and are treated at length in Chapters 38, 40, and 42. This section concentrates on concepts and results that are specific and important to GIS. We overview raster and vector representation of data, problems that appear in data input and correction in a GIS, and a well-known spatial indexing structure.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

GLOSSARY

Thematic map layer: Separately stored and manipulatable component of a map that contains the data of only one specific theme or geographic variable.

(Geographic) feature: Any geographically meaningful object.

Raster structure: Representation of geometric data based on a subdivision of the underlying space into a regular grid of square pixels.

Vector structure: Representation of geometric data based on the representation of points with coordinates, and line segments between those points.

Digitizing: Process of transforming cartographic data such as paper maps into digital form by tracing boundaries with a mouse-like device.

Conflation: Process of rectifying a digital data set by comparison with another digital data set that covers the same region (cf. rubber sheeting).

Topological vector structure: Vector structure in which incidence and adjacency of points, line segments, and faces is explicitly represented.

Quadtree: Tree where every internal node has four children, and which corresponds to a recursive subdivision of a square into four subsquares. The standard quadtree is for raster data. Leaves correspond to pixels or larger squares that appear in the recursive subdivision and are uniform in the thematic value.

R-tree: Tree based on a recursive partitioning of a set of objects into subsets, where every internal node stores a number of pairs (BB, S) equal to the number subtrees, where BB is the axis-parallel bounding box of all objects that appear in the subtree S. All leaves have the same depth and store the objects. R-trees have high (but constant) degree and are well-suited for secondary memory.

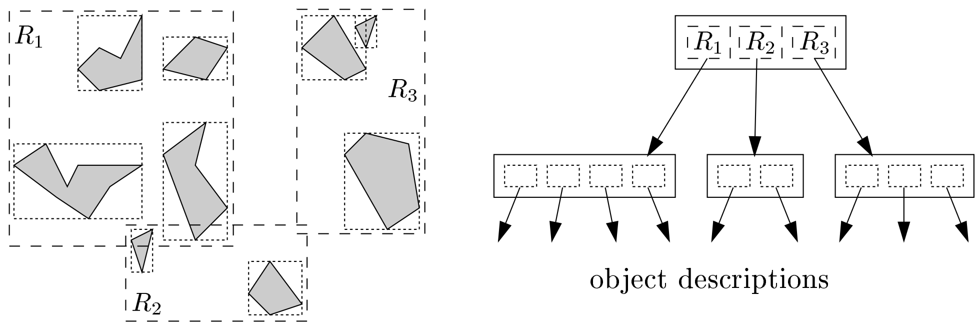
59.1.1 RASTER AND VECTOR STRUCTURES

Geographic data is composed of geometry, topology, and attributes. The attributes contain the semantics of a geographic feature. There are two essentially different ways to represent the geometric part of geographic data: raster and vector. This distinction is the same as representation in image space and object space in computer graphics (Chapter 52).

Data acquisition and input into a GIS often cause error and imprecision in the data, which must be corrected either manually or in an automated way. Also, the digitizing of paper maps yields unstructured collections of polygonal lines in vector format, to which topological structure is usually added using the GIS.

The topological vector structure obtained could be represented as, for example, a doubly-connected edge list or a quad-edge structure. But for maps with administrative boundaries, where long polygonal boundary lines occur where all vertices have degree 2, such a representation is space-inefficient. It is undesirable to have a separate object for every vertex and edge, with pointers to the incident features. The following variation gives better efficiency. Group maximal chains of degree-2 vertices into single objects, and treat them like an edge in the doubly-connected edge list. More explicitly, a chain stores pointers to the origin vertex (junction or endpoint), the destination vertex (junction or endpoint), the left face, the right face, and the next and previous chains in the two cycles of the faces incident to this

FIGURE 59.1.1
A set of polygons and an example of an R-tree for it.



chain. Such a representation allows retrieval of k adjacent faces of a face with m vertices to be reported in $O(k)$ time rather than in $O(m)$ time.

Advances in geographic data modeling and representation include multi-scale models, temporal and spatio-temporal models, fuzzy models, and qualitative representations of location.

59.1.2 R-TREES

The most widely used spatial data structure in GIS is the R-tree of Guttmann [Gut84]. It is a type of box-tree (see [BCG+96]) that has high (but constant) degree internal nodes, with all leaves at the same depth. It permits any type of object to be stored, and supports several types of queries, such as windowing, point location, and intersection. Insertions and deletions are both supported. The definition of R-trees does not specify which objects go in which subtree, and different heuristics for grouping give rise to different versions [BKSS90, KF94, LLE97].

R-trees generally do not have nontrivial worst-case query time bounds, so different versions must be compared experimentally. Only the versions of Agarwal et al. [ABG+02] and Arge et al. [ABHY08] have a query time bound better than linear (close to $O(\sqrt{n})$) when the stored objects are rectangles. These structures are also I/O-efficient.

59.2 SPATIAL ANALYSIS

Spatial analysis is the process of discovering information implicitly present in one or more spatial data sets. It includes common GIS operations such as map overlay, buffer computation, and shortest paths on road networks, but also geostatistical and spatial data analysis functions such as cluster detection, spatial interpolation, and spatial modeling. We discuss the most common forms and results in this section. For cluster analysis and classification, see Chapter 32.

GLOSSARY

Map overlay: The operation of combining two thematic map layers of the same region in order to obtain one new map layer, often with a refinement of the subdivisions used for the input map layers.

Buffer: The region of the plane within a certain specified distance to a geographic feature.

Neighborhood analysis: The study of how relations between geographic features depend on the distance.

Network analysis: The study of distance, reachability, travel time, and similar geographic functions that can be defined for network data (graphs with a geographic meaning).

Cluster analysis: The study of the grouping in sets of geographic features by proximity.

Trend analysis: The study of time-dependent patterns in geographic data.

Spatial interpolation: The derivation of values at locations based on values at other (nearby) locations.

Geostatistics: Statistics for data associated with locations in the plane.

59.2.1 MAP OVERLAY

With map overlay, two or more thematic map layers are combined into one. For example, if one map layer contains elevation contours and another map layer (of the same region) forest types, then their overlay reveals which types of forest occur at which elevations. One layer can also serve as a mask for the other layer. Overlay is essential to locating a region that has various properties that appear in different thematic map layers. In the spatial database literature, map overlay is also called ***spatial join***.

Map overlay is commonly solved using a plane sweep like the Bentley-Ottmann algorithm [BO79] for line segment intersection. This leads to an $O((n+k)\log n)$ time algorithm for two planar subdivisions of complexity $O(n)$, and output complexity $O(k)$. However, map overlay of two thematic map layers is essentially an extension of a red-blue line segment intersection problem, and can therefore be solved in optimal $O(n \log n + k)$ time [CEGS94, Cha94, PS94]. In case each subdivision is simply connected, the running time can be improved to $O(n+k)$ [FH95].

Map overlay in GIS must handle imprecise data as well, and therefore overlay methods that include sliver removal have been suggested. Essentially, boundaries that are closer than some pre-specified value are identified in the overlay. This is also called ***epsilon filter*** or ***fuzzy tolerance*** [Chr97]. The idea of using an epsilon band around a cartographic line is due to Perkal [Per66].

Since R-trees can also be used as access structures for subdivisions, map overlay can also be performed efficiently using R-trees [BKS93, KBS91, Oos94].

59.2.2 BUFFER COMPUTATION

The buffer of a geographic feature of width ϵ is the same as the Minkowski sum of that feature with a disk of radius ϵ, centered at the origin; see Chapter 50.

Computation can be done with the algorithms mentioned in that chapter.

Buffer computation and map overlay are two main ingredients for urban planning. As an example, three requirements for a new factory may be the proximity of a river, at least some distance to houses, and not in nature preserves. A map with suitable locations is obtained after computing buffers for two of the thematic map layers, and then combining these with each other and the third layer.

59.2.3 SPATIAL INTERPOLATION

Spatial interpolation is one of the main operations in geostatistics. It is the operation of defining values at locations when only values at other locations are known. For example, when ground measurements are taken at various locations, we only know values at a finite set of points, but we would like to know the values everywhere. Several methods exist for this version of the spatial interpolation problem, including triangulation, moving windows, natural neighbors, and Kriging. Triangulation is discussed in a later section. Moving windows is a form of weighted averaging of known (or observed) values within a window around the point with unknown value.

Natural neighbor interpolation is a method based on Voronoi diagrams [Sib81, SBM95]. Suppose the Voronoi diagram of the points with known values is given, and we want to obtain an interpolated value at another location p_0. We determine what Voronoi cell p_0 would "own" if it were inserted in the point set defining the Voronoi diagram. Let A be the area of the Voronoi cell of p_0, and let $A_1, \ldots, A_k$ be the areas removed from the Voronoi cells of the points $p_1, \ldots, p_k$, due to the insertion of p_0. Then, by natural neighbor interpolation, the interpolated value at p_0 is $\sum_{i=1}^{k}(A_i/A)\cdot V(p_i)$, where $V(p_i)$ denotes the known or observed value at p_i. The bivariate function obtained is continuous everywhere, and differentiable except at the points with known values.

Kriging is an interesting method that also applies weighted linear combinations of the known (or observed) values, that is, $V(p_0) = \sum_{i=1}^{n} \lambda_i \cdot V(p_i)$. The λ_i are the weights, and $\sum_{i=1}^{n} \lambda_i = 1$. Furthermore, the weights are chosen so that the estimation variance is less than for any other linear combination of known values. One additional advantage of Kriging is that it provides an estimation error as well [BM98].

Splines, discussed extensively in Chapter 56, can also be used for interpolation. A version used in GIS are the thin-plate splines. They do not necessarily pass through the known values of the points, and can therefore reduce artifacts. The spline function minimizes the sum of two components, one representing the smoothness and the other representing the proximity to the known values of the points [BM98].

59.3 VISUALIZATION OF SPATIAL DATA

Various tasks traditionally performed manually by cartographers can be automated. This not only leads to a more efficient map production process, it may also be necessary during the use of a GIS. A user of a GIS will typically select a several thematic map layers for display, and exactly which layers are selected determines which labels are needed and where they are placed best. Hence, pre-computation of label positions can often not be done off-line or by hand.

GLOSSARY

Choropleth map: Map in which the regions of an administrative subdivision are shown using a particular color scheme to represent a geographic variable.

Isoline map: Map for a continuous spatial phenomenon where curves of equal value for that phenomenon are displayed.

Cartogram: Map in which the area of the regions of an administrative subdivision represent a geographic variable (also called ***value-by-area map***).

Rectangular cartogram: Map in which administrative regions are shown as rectangles and the area of the rectangles represent a geographic variable.

Linear cartogram: Map in which travel times between locations are represented by distance on the map.

Schematic map: Map where important locations and connections between them (direct transportation) are shown highly stylized, and where location is preserved only approximately.

Flow map: Map that displays arrows of varying thickness that represent flows between geographic features. Arrows are often bundled at shared departure features or arrival features to enhance the visual representation.

Dot map: Map that shows dots where a dot represents the presence of a phenomenon or of a certain quantity of entities.

Label placement problem: The problem of placing text to annotate features on a map, according to various constraints and optimization criteria.

Line simplification problem: The problem of computing a polygonal line with fewer vertices from another polygonal line, while satisfying given constraints of distance.

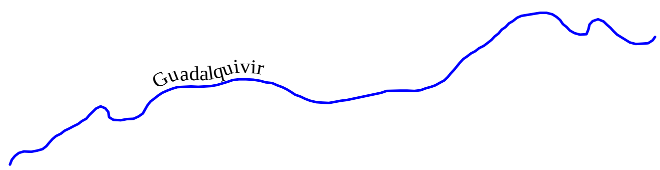
Cartographic generalization problem: The problem of computing a map at a coarser (smaller) scale from a data set whose detail would be appropriate for a map at a finer (larger) scale.

59.3.1 LABEL PLACEMENT

Automated label placement has been the topic of considerable research, both within cartography and within the field of algorithms. One can distinguish three types of labels: labels for point objects, labels for line objects, and labels for polygonal objects. Imhof [Imh75] provides many examples of well-placed and poorly-placed labels, demonstrating the many different requirements for practical, high-quality label placement.

The point-label placement problem is the following optimization problem. Given a set of points, each with a specified label (name or other text), place as many labels as possible adjacent to their point, but without overlap between any two labels. One can extend the problem by restricting, or not allowing, overlap with other map features, avoiding ambiguity, and so on. Another version of point labeling is to maximize label size under the condition that all points be labeled. Label

FIGURE 59.3.1
River labeling due to Wolff et al. [WKK+00].



placement for point objects is usually approached as follows. A label is modeled by an axis-parallel rectangle, the bounding box of the text. For each point, define a restricted set of positions considered for its label, the candidates. Typically, the four positions where a corner of the label coincides with the point are chosen. In the label number maximization problem, the problem is abstracted to maximum independent set in a graph where edges represent intersections of two candidate label positions. Tables 59.3.1 and 59.3.2 contain selected results.

TABLE 59.3.1 Point-label placement: size maximization; selected results.

TYPE OF LABEL	POSITIONS	APPROX. FACTOR	TIME	SOURCE
Equal-size square	4	2	$O(n \log n)$	[FW91, WW97]
Equal-size square	2	1	$O(n \log n)$	[FW91]
Arbitrary rectangle	2	1	$O(n \log^2 n)$	[FW91]
Arbitrary rectangle	4	2	$O(n^2 \log n)$	[JC04]
Equal-size disk	touching	$2.98 + \epsilon$	$O(n \log n) + n(1/\epsilon)^{O(1/\epsilon^2)}$	[JBQZ04].

TABLE 59.3.2 Point-label placement: number maximization; selected results.

TYPE OF LABEL	POSITIONS	APPROX. FACTOR	TIME	SOURCE
Rectangle	constant	$O(\log n)$	$O(n \log n)$	[AKS98]
Fixed-height rectangle	constant	2 (or PTAS)	$O(n \log n)$	[AKS98, Cha04]
Fixed-height rectangle	touching	2 (or PTAS)	$O(n \log n)$	[KSW99, PSS+03]
Disk, disk-like	constant	PTAS	$n^{O(1/\epsilon^2)}$	[EJS05]

Combinatorial optimization approaches have also been applied frequently to the point-label placement problem [CMS95, DTB99]. However, experiments show that simple heuristics work well in practice [CMS95, WWKS01].

We next discuss the labeling of line features. Here we distinguish in streets and rivers. The labeling of street patterns yields a combinatorial optimization problem similar to point labeling [NW00, Str01]. River labeling is quite different, because there are several different criteria that constitute a good river label placement. The label should be close to the river, it should follow the shape of the river, it should not have too high curvature, it should be as horizontal as possible, and it should have few inflection points. The algorithm of Wolff et al. [WKK$^+$00] includes all of these criteria; Figure 59.3.1.

The labeling of polygonal features appears for instance when placing the name of a country or lake inside that feature. It is common to either choose horizontal and straight placement, or let the shape follow the main shape of the polygonal feature. In the first case, one can place the label in the middle of the largest scaled copy of the label that fits inside the region. In the second case, one can use the medial axis to retrieve the main shape and place the label along it.

59.3.2 CARTOGRAPHIC GENERALIZATION

Cartographic generalization is the process of transforming and displaying cartographic data with less detail and information (i.e., on a coarser, smaller scale) than the input data contains. Examples include omitting small towns and minor roads, using only one color for nature regions rather than a distinction in forest, heath, moor, etc., aggregating several buildings into one block, and exaggerating the width of a road on a small-scale map. Generalization is a very important research topic in automated cartography [MLW95, MRS07].

The changes to the map data for generalization are done by *generalization operators*. They include selection, aggregation, typification, reclassification, smoothing, displacement, exaggeration, symbolization, collapse, and many others. Detecting a need for generalization is accomplished by computing certain geometric measures on distance, density, and detail. This will trigger the generalization operators to perform transformations. It may happen that one operator causes the need of a change somewhere else on the map, possibly leading to a domino effect. For example, one of the common generalization operators is displacement, to ensure a certain distance between two map features that should not appear adjacent. Moving one feature may cause the need to move another, leading to iterative displacement algorithms [Høj98, LJ01, MP01].

The problem of (polygonal) line simplification (cf. Section 50.3) is often considered a cartographic generalization problem, too. However, if the motivation for line simplification is only data reduction, then line simplification cannot be considered generalization. But since line simplification methods automatically reduce detail in polygonal lines, we will discuss some methods here.

The best-known cartographic line simplification method is due to Douglas and Peucker [DP73]. Starting with a line segment between the endpoints of the polygonal line, it selects the most distant vertex to be added to the simplification, and then continues recursively on the two parts that appear. This process continues until the most distant vertex is closer than some chosen threshold value to the approximating line segment. Theoretically, the method is highly unsatisfactory because it can create self-intersections in the output, requires quadratic time in the worst case, and may need many more segments in the approximation than the optimal approximation. However, it is very simple and usually works well in practice. Hershberger and Snoeyink devised a different algorithm to compute the same approximation which runs in $O(n \log^* n)$ time [HS98].

Weibel [Wei97] and van der Poorten and Jones [PJ99] demonstrate that many aspects are involved in practical line simplification for GIS, and that many different criteria may be used. Buchin et al. [BMS11b] show that local improvements are very effective and can also be used for other generalization operators. The GIS

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

literature contains several more practical approaches.

Guibas et al. [GHMS93] and Estkowski and Michell [EM01] show that minimum vertex simplification is NP-hard when self-intersections are not allowed. Selected algorithmic results are listed in Table 59.3.3.

TABLE 59.3.3 Polygonal line simplification: selected results.

OUTPUT VERTICES	COMPLEXITY	ERROR CRITERION	NOTES	SOURCE
From input	$O(n^2)$	distance	min. link, self-inter.	[CC96]
From input	$O(n^{4/3+\delta})$	vertical distance	min. link, self-inter.	[AV00]
From input	$O(n^2 \log n)$	distance	respects context	[BKS98]
From input	$O(n^3)$	distance	subdivision	[EM01]
From input	$O(n^2)$	distance along path		[GNS07]
From input	$O(n^3), O(n^4)$	area displacement	min. area measures	[BCC+06]
From input	$O(n^2)$	distance	angle constraints	[CDH+05]
Arbitrary	$O(n^2 \log n)$	distance	subdivision	[GHMS93]

59.3.3 THEMATIC MAPS

Topographic maps are general-purpose maps that display a variety of themes of general interest together, like roads, towns, forests, and elevation contours. Thematic maps, on the other hand, concentrate on a particular theme, and may use alternative methods of visualizing the information. A choropleth map could, e.g., show the population densities of the states of the U.S. by coloring each with a color from a well-chosen set of colors, for instance, five saturation values of red. The geographic theme of population density can be seen as a scalar function. Here the points of the plane are aggregated by state.

There are other ways of visualizing scalar functions cartographically, including isoline maps, dot maps, and cartograms. The latter again applies to aggregated regions of the plane. Flow maps visualize a presence and quantity of flow from one (aggregated) region to another. Schematic maps visualize connections between locations, such as subway maps. Dent et al. [DTH09] provide a good overview of several thematic map types.

Cartograms show values for regions by shrinking and expanding those regions, so that the area of each region corresponds to the value represented. The most important usage is the population cartogram, where a region A with a population twice that of region B will be shown twice as large as B. Necessarily, cartograms show a distortion of the geographic space. To keep the regions more or less recognizable, they should keep their shape, location, and adjacency as much as possible.

Several algorithms have been proposed to construct cartograms, given an administrative subdivision and a value for each region. Tobler [Tob86] simply uses scaling on the x- and y-coordinates, which may prevent regions from being shown at the correct size. Dougenick et al. [DCN85] compute a centroid for each region, which is assigned a repelling force if the region should grow and an attracting force if the region should shrink. The forces of all centroids on all boundaries of the map then result in new positions of these boundaries. This is used in an iterative algorithm. Kocmoud and House's approach [KH98] is constraint programming. They

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

FIGURE 59.3.2
Left, regions with a K_4 as adjacency graph cannot be represented by adjacent rectangles. Right, four rectangles inside a rectangular outline and with specified adjacencies cannot realize the specified areas without changing the adjacencies.

also attempt to preserve the main orientations of the boundaries. Edelsbrunner and Waupotitsch [EW97] give a cartogram construction algorithm based on simplicial complexes in the plane, where paths of triangles are used to define deformations that let one region grow at the expense of the size of another.

The first algorithmic study of rectangular cartograms, where all regions are rectangles, was done by van Kreveld and Speckmann [KS07]; extensions and refinements were presented by Buchin et al. [BSV12]. Broadly speaking, there are two reasons why correct rectangle sizes and correct adjacencies are not always possible, see Figure 59.3.2. To overcome these problems, rectilinear cartograms were introduced, where regions can have more than four corners. De Berg et al. [BMS09] showed that only constantly many corners are needed in rectilinear cartograms. Their bound on the number of corners was improved by Alam et al. [ABF$^+$13] to eight corners per region, which is worst-case optimal.

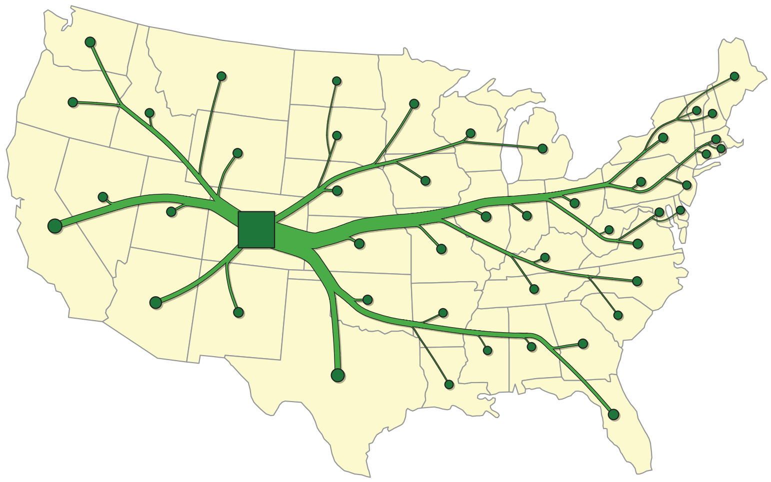
In linear cartograms, distances are distorted rather than areas. This is useful when a map serves to represent travel time [BK12, KWFP10, SI09]. Alternatively, one can use edge length to represent travel time [BGH$^+$14].

Flow maps show the movement of objects between geographic locations on a map using thick arrows, see Figure 59.3.3. Edge bundling is commonly used to avoid visual clutter. Using a modification of Steiner minimal trees, Buchin et al. [BSV15] modeled this problem and solved it with an approximation algorithm, since the general problem is NP-hard.

Dot maps show values by dots, where one dot represents, e.g., ten thousand people. This allows the distribution of the population to be shown better than in cartograms, but the relative populations for two regions are more difficult to compare. De Berg et al. [BBCM02] show the connection between dot maps and discrepancy, and compare various heuristics to construct dot maps.

Schematic maps are commonly used for public transportation systems. Direct lines, or connections between major stations, are shown with a polygonal line that is highly abstracted: it has only a few segments and often, these segments are restricted to be horizontal, vertical, or have slope $+1$ or -1. Cabello et al. [CBK05] place the connections incrementally in a pre-computed order, leading to an $O(n \log n)$ time algorithm. Neyer [Ney99] views the problem as a line simplification problem and approximates each connection with the minimum number of segments in the specified orientations. Nöllenburg and Wolff [NW11] give an integral approach to the problem that takes multiple constraints into account and solve it with mixed-integer programming. Brandes and Wagner [BW98] show connections between stations by circular arcs and address the visualization problem as a graph layout problem (Chapter 55).

FIGURE 59.3.3
Flow map showing migration from Colorado (from Buchin et al. [BSV11]*).*



59.3.4 DYNAMIC AND ANIMATED MAPS

Besides computations for traditional paper maps, a more recent trend is to study dynamic maps, animated maps, interactive maps, Web maps, and multimedia maps [KB01, CPG99]. This area leads to a number of new computational issues, where efficiency is very important and quality is less critical. At the same time, the computational problems involved may become harder. For example, where label maximization on static maps is generally NP-hard, a related version on dynamic maps is PSPACE-hard [BG14]. Several other results on dynamic and interactive map labeling exist [BFH01, PPH99].

Zooming out on a map also makes real-time cartographic generalization necessary. The problem is that not only the size of features must be changed, but also the way of visualization. On large-scale maps, cities are shown by polygonal outlines, but on small-scale maps, they are shown by point symbols. The process is called ***dynamic*** or ***on-the-fly generalization*** [BNPW10, MG99, GNN13, GNR16, Oos95, SHWZ14]. Ideally, the changes made during zooming should be made in a continuous manner, with no major, sudden changes on the map [Kre01]. In static generalization, the objective is to compute a new representation, but in dynamic generalization, the problem is the computation of the transition.

59.4 DIGITAL ELEVATION MODELS

We have concentrated on types of data based on subdivisions with well-marked boundaries. Another important type of data is the scalar function in two variables. The most common example from geography is elevation above sea level, also called terrain. Three other examples are annual precipitation, nitrate concentration per cubic meter, and average noise level.

There are two common representations for elevation: the regular square grid, or ***elevation matrix***, which is a raster representation, and the triangular irregular network (TIN), which is a vector representation. For the latter representation, the Delaunay triangulation is often used.

GLOSSARY

Digital elevation model (DEM): Representation of a scalar function in two variables. Sometimes specifically used for the raster-based representation.

Triangular irregular network (TIN): Vector-based representation of a digital elevation model defined by a triangulation of a point set. Also called polyhedral terrain.

Drainage network: Collection of linear features that represent the locations where water on a terrain has formed rivers.

Viewshed analysis: The study of visibility on a terrain.

59.4.1 CONSTRUCTION AND SIMPLIFICATION OF TINS

The problem of simplifying a digital elevation model, or performing raster to vector conversion for a digital elevation model, is a higher-dimensional version of line simplification. The best algorithm known is similar in approach to the Douglas-Peucker algorithm for line simplification given in Section 59.3.2. Assume that the outer boundary of the DEM is rectangular, a set of points with their elevation is given (e.g., based on a regular square grid), and assume that a maximum allowed vertical error $\epsilon > 0$ is specified. An initial coarse simplification of the TIN is a triangulation of the four corners of the rectangle. If that simplification is vertically within a distance ϵ from all points, then it is accepted. Otherwise, the point with largest vertical distance is selected and added to the triangulation, which is restored by flipping to the Delaunay triangulation. The process is then repeated.

The method requires quadratic time in the worst case, but an implementation can be given which, under natural assumptions, takes $O(n \log n)$ time in practice [Hel90, Fjä91, HG95].

Agarwal and Suri [AS98] show that a corresponding optimization problem is NP-hard, and give an approximation algorithm that requires $O(n^8)$ time. If m is the size of the optimal piecewise linear ϵ-approximation of the n given points, then the computed approximation has size $O(m \log m)$. Agarwal and Desikan [AD97] give a cubic time ϵ-approximation algorithm with a worse size bound on the approximation, but with some assumptions the approximation has the same size asymptotically and runs in near-quadratic time.

When a TIN is constructed for modeling terrains, various geometric computation problems arise. When the input is a set of (digitized) contour lines, a triangulation between the contour lines such as the constrained Delaunay triangulation can be used [DP89, Sch98]. Care must be taken that no triangle with all three vertices on the same contour line is created, as this gives undesirable artifacts. Thibault and Gold [TG00] provide a solution that avoids flat triangles by adding vertices on the medial axis or skeleton, which are given intermediate elevations (see also [GD02]). If information on rivers is present too, then these can be included as edges of the TIN using a constrained Delaunay triangulation [MS99]. Other approaches of interest are those by Silva et al. [SMK95], Little and Shi [LS01], and van Kreveld and Silveira [KS11]. These methods concern the construction of a TIN that can integrate or preserve important features like valleys and ridges of the terrain.

Gudmundsson et al. [GHK02] define a class of well-shaped triangulations called higher-order Delaunay triangulations, and use them to create TINs with fewer local minima in the terrain, because such minima generally do not occur. The idea was developed further by de Kok et al. [KKL07].

Multi-resolution terrain modeling has been studied extensively, see Puppo and Scopigno [PS97] for an overview. The approach of de Berg and Dobrindt [BD98] allows multiple levels of detail, and even the combination of multiple levels of detail in one terrain, but still uses a Delaunay triangulation in all cases. See [Ros04] for more information on surface simplification and multi-resolution representations.

59.4.2 VISUALIZATION OF DEMS

Digital elevation models may be visualized in several ways. A traditional way is by contour maps, and the process of deriving a contour map is called ***contouring*** [Wat92].

A perspective view of a digital elevation model can be obtained by back-to-front rendering of the grid elements or triangles. If a vector representation of a perspective view is needed, an algorithm of Katz et al. [KOS92] achieves this for a TIN in $O((n+k)\log n \log\log n)$ time, where n is the number of triangles and k is the complexity of the visibility map.

59.4.3 DERIVED MAPS AND PRODUCTS

In the analysis of terrain—e.g., for land suitability studies—slope and aspect maps are important derived products of a digital elevation model. They are straightforward to compute. Similarly, the plan and profile curvature can be of importance, for example for waterflow and erosion modeling.

The computation of the drainage network, based on the shape of the terrain, has been frequently studied, most often for grid data. Besides the drainage network, watersheds also provide important terrain information. A surprising combinatorial result on TINs of de Berg et al. [BBD$^+$96] is that if water always follows the direction of steepest descent, and the drainage network consists of all points that receive flow from some region with positive area, then triangulations exist with n triangles for which the drainage network has complexity $\Theta(n^3)$, see also de Berg et al. [BHT11].

Viewshed analysis is the study of visibility in the terrain [BCK08, BHT10, HLM$^+$14, KOS92]. Viewshed analysis has applications in urban and touristical planning, and for telecommunication, for example, to place a small number of antennas so that every point on the terrain has direct visibility to at least one antenna [BMM$^+$04, Fra02]. Other visibility results for terrains may be found in Section 33.8.3.

The computation of shortest paths between two points on a terrain is a problem of both theoretical and practical interest. The approach of Alexandrov et al. [AMS05, LMS97] is significant both theoretically and practically, because it also deals with the weighted version. The main idea is to place Steiner points on edges

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

to convert the problem into a graph problem, and then apply Dijkstra's algorithm. This gives a simple approximation algorithm for least-cost paths.

59.5 TRAJECTORY DATA

Trajectory data is a type of geographic data that has become of increasing importance. Due to high-quality GPS and RFID technology, large data sets of trajectories have been collected and stored, with applications in pedestrian and vehicle movement analysis, weather forecasting, behavioral ecology, security, and more. The processing and analysis of such data gives rise to algorithmic problems that have been studied in the areas of GIS and computational geometry [ZZ11]. Geometrically, a trajectory can be seen as a polygonal line, but the time component implies that different methods are needed from those for polygonal line processing. Most algorithmic research models a trajectory by a sequence of time-stamped points (the vertices) and assumes linear interpolation in both space and time in between these vertices. This makes location change over time in a piecewise-linear manner and velocity in a piecewise-constant manner.

Since processing and analyzing trajectories may be time-consuming, trajectory simplification can be important as a preprocessing step. One could consider any algorithm for polygonal line simplification to simplify trajectories, but it is better to use a dedicated trajectory simplification method [CWT06, GKM$^+$09, LK11].

59.5.1 TRAJECTORY SEGMENTATION

Just like images can be segmented into semantically meaningful parts, so can trajectories. A segmentation of a trajectory partitions it into parts where each part corresponds to a type of motion that could indicate a type of behavior, a mode of transport, an environment in which movement takes place, or a reason for movement. The parts of a segmentation are subtrajectories, but confusingly, they are also referred to as segments.

The formal, algorithmic study of trajectory segmentation was initiated by Buchin et al. [BDKS11], who distinguish attribute values associated to each time instance of the trajectory (location at time t, speed at time t, etc.) and criteria relating to these attributes that specify when two time instances may not be in the same segment of the segmentation. The objective is a minimum-segment segmentation. For a class of criteria, an optimal segmentation can be computed in $O(n \log n)$ time. There are, however, criteria that are important in practice which are not in this class. Examples are some relaxed criteria that allow some small deviations to their strict versions. Follow-up research has identified wider classes of criteria that also allow polynomial-time optimal segmentation [ADK$^+$13, ABB$^+$14].

59.5.2 TRAJECTORY SIMILARITY

Determining the similarity two trajectories is important, also for clustering of trajectories. A similarity measure can be based on the geometry of the trajectories only, in which case the Hausdorff distance or Fréchet distance may be used. If the

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

time aspect should influence similarity then other similarity measures are needed. One of the most widely used measures is called *dynamic time warping* [BC94]. Unfortunately, dynamic time warping does not satisfy the triangle inequality, it requires quadratic time to compute, and it uses only the distances between vertices. A large body of follow-up research has tried to overcome these problems.

When time cannot be conceptually stretched to get a better match, simpler similarity measures can be used [CW99, NP06]. These can be used for subtrajectory similarity computation: find the subtrajectories of two given trajectories of at least a given length whose distance is minimum [BBKL11]. Other research allows rotation and scaling for computing the similarity of two complete trajectories [TDS+07]. Yet another direction is to incorporate the context when computing similarity [BDS14].

59.5.3 PATTERNS IN TRAJECTORY DATA

When a collection of trajectories is given, analysis involves finding patterns in these trajectories. One of the most common data analysis tasks is clustering, and indeed, trajectory clustering has been studied extensively [HLT10, NP06]. However, most patterns in trajectory data concern parts of trajectories, for instance trajectories that move together for a certain length of time (and not necessarily the full duration of the trajectory). This pattern is referred to as a flock. For a flocking model that limits the involved trajectories to the ones that can fit together in a fixed-size disk, computing a maximum-duration flock is NP-hard [GK06], but approximation algorithms exist that do not fix the size of the disk exactly. Various other models for flocks and accompanying algorithms have been presented [BGHW08, GKS07, KMB05]. Besides finding flocks, it is also possible the study the changes in the grouping structure. The Reeb graph has been identified as a useful structure for this [BBK+13].

Patterns may involve identifying individuals that take on a specific role. The most common pattern of this type is the leadership pattern [AGLW08, GKS07]. A model for leadership describes a leader as an entity such that there are other entities that move in the direction of the leader and are close by. A related pattern is the single-file movement pattern [BBG08].

Patterns may involve the discovery of frequently visited places or hotspots [BDGW10, DGPW11, GKS13]. In these patterns, the region of interest may be given, or to be determined from the trajectories while only the size and shape is given. In some versions, we are interested in the longest consecutive visit, in other versions we are interested in the longest total visit.

59.6 OTHER GEOMETRIC ALGORITHMS IN GIS

The previous sections gave a brief overview of the main topics in which geometric algorithms play a role. There are a number of other topics, the most important of which are mentioned in this section.

GPS data from cars have been used as a source to construct road networks. This requires large-scale averaging, because many traces represent the same road [AW12, AKPW15, CGHS10, GSBW11]. To analyze the similarity of two road networks, an appropriate measure is needed. Here the geometric aspects of the network should be taken into account, and not just the graph structure [AFHW15]. By studying the average number of times a network is intersected by a random line, point location and ray-shooting navigation queries can be done efficiently [EGT09]. Generally, road networks are not planar graphs, and therefore many algorithmic results on planar graphs do not automatically transfer to road networks. Using other natural properties of road networks, several algorithmic results apply nevertheless [EG08] Clusters of points in networks are stretches of the network whose point density is relatively high. One can consider paths, trees, or other connected subgraphs of the network [BCG+10].

The search for I/O-efficient algorithms for GIS-related problems has led to a substantial amount of research [AVV07]. Besides the R-tree and its variants mentioned earlier, terrain analysis has been the main application area for such algorithms [AAMS08].

Motivated by geographic information retrieval, the Internet can be used as a data source for vernacular regions. The delineation of such regions involves finding a reasonable boundary for a given set of inliers and outliers [BMS11a, RBK+08].

Privacy considerations play a major role in location-based services. With this in mind, data-oblivious geometric algorithms can be devised which allow secure multi-party computation protocols [EGT10].

59.7 ALGORITHMIC CHALLENGES IN GIS

The application area GIS is the source of a number of interesting research problems. Many of these are simply-stated algorithmic problems, such as finding the most efficient algorithm for a well-defined problem, or finding the best approximation factor for some computationally hard problem. But from the application perspective, the study of relatively simple solutions for problems in which a number of different requirements must be satisfied or optimized simultaneously is more important. For example, label placement with high cartographic quality has to be achieved with no overlap between different labels, no or little overlap of a label with other map features, clear association between a feature and its label, and avoidance of areas that are too dense with text. There is no simple problem statement with optimization that can capture this. As a second example, in realistic terrain reconstruction, seven constraints have been listed [Sch98]. Such constraints cannot be formulated straightforwardly as algorithmic optimization. It is usually more important which requirements can be handled simultaneously and effectively, than how efficient a solution is.

Challenges for algorithms research on GIS problems include developing methods that deal with multiple criteria simultaneously, either as a whole solution or as part of an optimization approach such as genetic or evolutionary algorithms. The appropriate formulation of the GIS problem itself, and comparison of results based on different formulations, are also issues of major importance.

A more specific research direction is the study of movement patterns where entities are not modeled as single points, like a model for a moving glacier [SBL+12]. In extension to this, the computations involved in the modeling of geographic processes is a good source of problems for computational geometers. Another research direction is the handling of data imprecision in a computational model that is valid, relevant, and simple enough to be accepted in GIS.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

59.8 SOURCES AND RELATED MATERIAL

BOOKS

[Sam06]: Book on data structures for GIS and other applications.

[HCC11, LGMR11]: Two general GIS books.

[WD04]: A GIS book with a computing focus.

[DTH09, RMM+95]: Two books that focus on cartography aspects of GIS.

[Lau14, ZZ11]: Two books on movement and trajectories.

OTHER

Other surveys: computational geometry and GIS [DMP00], terrain modeling [HT14], patterns in trajectories [GLW08].

Journals: Journal of Spatial Information Science, ACM Transactions on Spatial Algorithms and Systems, GeoInformatica, International Journal of Geographical Information Science (IJGIS), Transactions in GIS, Cartography & GIS.

Conference proceedings: SIGSPATIAL Conference, GIScience, Symposium on Spatial and Temporal Databases (SSTD), International Symposium on Spatial Data Handling (SDH), Auto-Carto, International Cartographic Conference (ICC), Conference on Spatial Information Theory (COSIT).

RELATED CHAPTERS

Chapter 23: Computational topology of graphs on surfaces
Chapter 27: Voronoi diagrams and Delaunay triangulations
Chapter 32: Proximity algorithms
Chapter 51: Robotics
Chapter 52: Computer graphics
Chapter 54: Pattern recognition

REFERENCES

[AAMS08] P.K. Agarwal, L. Arge, T. Mølhave, and B. Sadri. I/O-efficient efficient algorithms for computing contours on a terrain. In *Proc. 24th Sympos. Comput. Geom.*, pages 129–138, ACM Press, 2008.

[ABB+14] S.P.A. Alewijnse, K. Buchin, M. Buchin, A. Kölzsch, H. Kruckenberg, and M.A. Westenberg. A framework for trajectory segmentation by stable criteria. In *Proc. 22nd ACM SIGSPATIAL Int. Conf. Advances Geographic Inform. Syst.*, pages 351–360, 2014.

[ABF+13] M.J. Alam, T. Biedl, S. Felsner, M. Kaufmann, S.G. Kobourov, and T. Ueckerdt.

Computing cartograms with optimal complexity. *Discrete Comput. Geom.*, 50:784–810, 2013.

[ABG+02] P. Agarwal, M. de Berg, J. Gudmundsson, M. Hammar, and H.J. Haverkort. Box-trees and R-trees with near-optimal query time. *Discrete Comput. Geom.*, 28:291–312, 2002.

[ABHY08] L. Arge, M. de Berg, H.J. Haverkort, and K. Yi. The priority R-tree: A practically efficient and worst-case optimal R-tree. *ACM Trans. Algorithms*, 4:9, 2008.

[AD97] P.K. Agarwal and P.K. Desikan. An efficient algorithm for terrain simplification. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 139–147, 1997.

[ADK+13] B. Aronov, A. Driemel, M. van Kreveld, M. Löffler, and F. Staals. Segmentation of trajectories for non-monotone criteria. In *Proc. 24th ACM-SIAM Sympos. Discrete Algorithms*, pages 1897–1911, 2013.

[AFHW15] M. Ahmed, B.T. Fasy, K.S. Hickmann, and C. Wenk. A path-based distance for street map comparison. *ACM Trans. Spatial Algorithms Systems*, 1:3, 2015.

[AGLW08] M. Andersson, J. Gudmundsson, P. Laube, and T. Wolle. Reporting leaders and followers among trajectories of moving point objects. *GeoInformatica*, 12:497–528, 2008.

[AKPW15] M. Ahmed, S. Karagiorgou, D. Pfoser, and C. Wenk. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica*, 19:601–632, 2015.

[AKS98] P.K. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Comput. Geom.*, 11:209–218, 1998.

[AMS05] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Determining approximate shortest paths on weighted polyhedral surfaces. *J. ACM*, 52:25–53, 2005.

[AS98] P.K. Agarwal and S. Suri. Surface approximation and geometric partitions. *SIAM J. Comput.*, 27:1016–1035, 1998.

[AV00] P.K. Agarwal and K.R. Varadarajan. Efficient algorithms for approximating polygonal chains. *Discrete Comput. Geom.*, 23:273–291, 2000.

[AVV07] L. Arge, D.E. Vengroff, and J.S. Vitter. External-memory algorithms for processing line segments in geographic information systems. *Algorithmica*, 47:1–25, 2007.

[AW12] M. Ahmed and C. Wenk. Constructing street networks from GPS trajectories. In *Proc. 20th Eur. Sympos. Algorithms*, vol. 7501 of *LNCS*, pages 60–71, Springer, Berlin, 2012.

[BBCM02] M. de Berg, J. Bose, O. Cheong, and P. Morin. On simplifying dot maps. In *Proc. 18th Eur. Workshop Comput. Geom.*, pages 96–100, 2002.

[BBD+96] M. de Berg, P. Bose, K. Dobrint, M. van Kreveld, M.H. Overmars, M. de Groot, T. Roos, J. Snoeyink, and S. Yu. The complexity of rivers in triangulated terrains. In *Proc. 8th Canad. Conf. Comput. Geom.*, pages 325–330, 1996.

[BBG08] K. Buchin, M. Buchin, and J. Gudmundsson. Detecting single file movement. In *Proc. 16th ACM SIGSPATIAL Int. Sympos. Advances Geographic Inform. Syst.*, page 33, 2008.

[BBK+13] K. Buchin, M. Buchin, M. van Kreveld, B. Speckmann, and F. Staals. Trajectory grouping structure. In *Proc. 13th Algorithms Data Structures Sympos.*, vol. 8037 of *LNCS*, pages 219–230, Springer, Berlin, 2013.

[BBKL11] K. Buchin, M. Buchin, M. van Kreveld, and J. Luo. Finding long and similar parts of trajectories. *Comput. Geom.*, 44:465–476, 2011.

[BC94] D.J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop. Technical Report WS-94-03*, pages 359–370, 1994.

[BCC+06] P. Bose, S. Cabello, O. Cheong, J. Gudmundsson, M. van Kreveld, and B. Speckmann. Area-preserving approximations of polygonal paths. *J. Discrete Algorithms*, 4:554–566, 2006.

[BCG+96] G. Barequet, B. Chazelle, L. Guibas, J. Mitchell, and A. Tal. BOXTREE: A hierarchical representation for surfaces in 3D. *Computer Graphics Forum*, 15:387396, 1996.

[BCG+10] K. Buchin, S. Cabello, J. Gudmundsson, M. Löffler, J. Luo, G. Rote, R.I. Silveira, B. Speckmann, and T. Wolle. Finding the most relevant fragments in networks. *J. Graph Algorithms Appl.*, 14:307–336, 2010.

[BCK08] B. Ben-Moshe, P. Carmi, and M.J. Katz. Approximating the visible region of a point on a terrain. *GeoInformatica*, 12:21–36, 2008.

[BD98] M. de Berg and K. Dobrindt. On levels of detail in terrains. *Graphical Models Image Processing*, 60:1–12, 1998.

[BDGW10] M. Benkert, B. Djordjevic, J. Gudmundsson, and T. Wolle. Finding popular places. *Internat. J. Comput. Geom. Appl.*, 20:19–42, 2010.

[BDKS11] M. Buchin, A. Driemel, M. van Kreveld, and V. Sacristán. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *J. Spatial Inform. Sci.*, 3:33–63, 2011.

[BDS14] M. Buchin, S. Dodge, and B. Speckmann. Similarity of trajectories taking into account geographic context. *J. Spatial Inform. Sci.*, 9:101–124, 2014.

[BFH01] B. Bell, S. Feiner, and T. Höllerer. View management for virtual and augmented reality. In *Proc. ACM Sympos. User Interface Software Tech.*, pages 101–110, 2001.

[BG14] K. Buchin and D.H.P. Gerrits. Dynamic point labeling is strongly PSPACE-complete. *Internat. J. Comput. Geom. Appl.*, 24:373, 2014.

[BGH+14] K. Buchin, A. van Goethem, M. Hoffmann, M. van Kreveld, and B. Speckmann. Travel-time maps: Linear cartograms with fixed vertex locations. In *Proc. 8th Conf. Geographic Inform. Sci.*, vol. 8728 of *LNCS*, pages 18–33, Springer, Berlin, 2014.

[BGHW08] M. Benkert, J. Gudmundsson, F. Hübner, and T. Wolle. Reporting flock patterns. *Comput. Geom.*, 41:111–125, 2008.

[BHT10] M. de Berg, H.J. Haverkort, and C.P. Tsirogiannis. Visibility maps of realistic terrains have linear smoothed complexity. *J. Comput. Geom.*, 1:57–71, 2010.

[BHT11] M. de Berg, H.J. Haverkort, and C.P. Tsirogiannis. Implicit flow routing on terrains with applications to surface networks and drainage structures. In *Proc. 22nd ACM-SIAM Sympos. Discrete Algorithms*, pages 285–296, 2011.

[BK12] S. Bies and M. van Kreveld. Time-space maps from triangulations. In *Proc. 20th Sympos. Graph Drawing*, vol. 7704 of *LNCS*, pages 511–516, Springer, Berlin, 2012.

[BKS93] T. Brinkhoff, H.-P. Kriegel, and B. Seeger. Efficient processing of spatial joins using R-trees. In *Proc. ACM SIGMOD Conf. Management Data*, pages 237–246, 1993.

[BKS98] M. de Berg, M. van Kreveld, and S. Schirra. Topologically correct subdivision simplification using the bandwidth criterion. *Cartography and GIS*, 25:243–257, 1998.

[BKSS90] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proc. ACM SIGMOD Conf. Management Data*, pages 322–331, 1990.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[BM98] P.A. Burrouh and R.A. McDonnell. *Principles of Geographical Information Systems*. Oxford University Press, 1998.

[BMM+04] B. Ben-Moshe, J.S.B. Mitchell, M.J. Katz, and Y. Nir. Visibility preserving terrain simplification—an experimental study. *Comput. Geom.*, 28:175–190, 2004.

[BMS09] M. de Berg, E. Mumford, and B. Speckmann. On rectilinear duals for vertex-weighted plane graphs. *Discrete Math.*, 309:1794–1812, 2009.

[BMS11a] M. de Berg, W. Meulemans, and B. Speckmann. Delineating imprecise regions via shortest-path graphs. In *Proc. 19th ACM SIGSPATIAL Int. Sympos. Advances Geographic Inform. Syst.*, pages 271–280, 2011.

[BMS11b] K. Buchin, W. Meulemans, and B. Speckmann. A new method for subdivision simplification with applications to urban-area generalization. In *Proc. 19th ACM SIGSPATIAL Int. Sympos. Advances Geographic Inform. Syst.*, pages 261–270, 2011.

[BNPW10] K. Been, M. Nöllenburg, S.-H. Poon, and A. Wolff. Optimizing active ranges for consistent dynamic map labeling. *Comput. Geom.*, 43:312–328, 2010.

[BO79] J.L. Bentley and T.A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28:643–647, 1979.

[BSV11] K. Buchin, B. Speckmann, and K. Verbeek. Flow map layout via spiral trees. *IEEE Trans. Vis. Comp. Graph.*, 17:2536–2544, 2011.

[BSV12] K. Buchin, B. Speckmann, and S. Verdonschot. Evolution strategies for optimizing rectangular cartograms. In *Proc. 7th Conf. Geographic Inform. Sci.*, pages 29–42, 2012.

[BSV15] K. Buchin, B. Speckmann, and K. Verbeek. Angle-restricted steiner arborescences for flow map layout. *Algorithmica*, 72:656–685, 2015.

[BW98] U. Brandes and D. Wagner. Using graph layout to visualize train interconnection data. In *Proc. 6th Sympos. Graph Drawing*, vol. 1547 in *LNCS*, pages 44–56, Springer, Berlin, 1998.

[CBK05] S. Cabello, M. de Berg, and M. van Kreveld. Schematization of networks. *Comput. Geom.*, 30:223–228, 2005.

[CC96] W.S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *Internat. J. Comput. Geom. Appl.*, 6:59–77, 1996.

[CDH+05] D.Z. Chen, O. Daescu, J. Hershberger, P.M. Kogge, N. Mi, and J. Snoeyink. Polygonal path simplification with angle constraints. *Comput. Geom.*, 32:173–187, 2005.

[CEGS94] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. Algorithms for bichromatic line segment problems and polyhedral terrains. *Algorithmica*, 11:116–132, 1994.

[CGHS10] D. Chen, L.J. Guibas, J. Hershberger, and J. Sun. Road network reconstruction for organizing paths. In *Proc. 21st ACM-SIAM Sympos. Discrete Algorithms*, pages 1309–1320, 2010.

[Cha94] T.M. Chan. A simple trapezoid sweep algorithm for reporting red/blue segment intersections. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 263–268, 1994.

[Cha04] T.M. Chan. A note on maximum independent sets in rectangle intersection graphs. *Inform. Process. Lett.*, 89:19–23, 2004.

[Chr97] N. Chrisman. *Exploring Geographic Information Systems*. Wiley, New York, 1997; second edition, 2002.

[CMS95] J. Christensen, J. Marks, and S. Shieber. An empirical study of algorithms for point-feature label placement. *ACM Trans. Graph.*, 14:203–232, 1995.

[CPG99] W. Cartwright, M. Peterson, and G. Gartner, editors. *Multimedia Cartography.* Springer, Berlin, 1999.

[CW99] K.K.W. Chu and M.H. Wong. Fast time-series searching with scaling and shifting. In *Proc. 18th ACM Sympos. Principles Database Syst.*, pages 237–248, 1999.

[CWT06] H. Cao, O. Wolfson, and G. Trajcevski. Spatio-temporal data reduction with deterministic error bounds. *VLDB J.*, 15:211–228, 2006.

[DCN85] J.A. Dougenik, N.R. Chrisman, and D.R. Niemeyer. An algorithm to construct continuous area cartograms. *The Professional Geographer*, 37:75–81, 1985.

[DGPW11] B. Djordjevic, J. Gudmundsson, A. Pham, and T. Wolle. Detecting regular visit patterns. *Algorithmica*, 60:829–852, 2011.

[DMP00] L. De Floriani, P. Magillo, and E. Puppo. Applications of computational geometry to geographic information systems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 333–388, Elsevier, Amsterdam, 2000.

[DP73] 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, 1973.

[DP89] L. De Floriani and E. Puppo. A survey of constrained Delaunay triangulation algorithms for surface representation. In G.G. Pieroni, editor, *Issues on Machine Vision*, pages 95–104, Springer, Vienna, 1989.

[DTB99] S. van Dijk, D. Thierens, and M. de Berg. On the design of genetic algorithms for geographical applications. In *Proc. Genetic Evol. Comput. Conf.*, pages 188–195, Morgan Kaufmann, San Francisco, 1999.

[DTH09] B.D. Dent, J. Torguson, and T.W. Hodler. *Cartography: Thematic Map Design*, 6th edition. McGraw Hill, New York, 2009.

[EG08] D. Eppstein and M.T. Goodrich. Studying (non-planar) road networks through an algorithmic lens. In *Proc. 16th ACM SIGSPATIAL Int. Sympos. Advances Geographic Inform. Syst.*, page 16, 2008.

[EGT09] D. Eppstein, M.T. Goodrich, and L. Trott. Going off-road: Transversal complexity in road networks. In *Proc. 17th ACM SIGSPATIAL Int. Sympos. Advances Geographic Inform. Syst.*, pages 23–32, 2009.

[EGT10] D. Eppstein, M.T. Goodrich, and R. Tamassia. Privacy-preserving data-oblivious geometric algorithms for geographic data. In *Proc. 18th ACM SIGSPATIAL Int. Sympos. Advances Geographic Inform. Syst.*, pages 13–22, 2010.

[EJS05] T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM J. Comput.*, 34:1302–1323, 2005.

[EM01] R. Estkowski and J.S.B. Mitchell. Symplifying a polygonal subdivision while keeping it simple. In *Proc. 17th Sympos. Comput. Geom.*, pages 40–49, ACM press, 2001.

[EW97] H. Edelsbrunner and R. Waupotitsch. A combinatorial approach to cartograms. *Comput. Geom.*, 7:343–360, 1997.

[FH95] U. Finke and K.H. Hinrichs. Overlaying simply connected planar subdivisions in linear time. In *Proc. 11th Sympos. Comput. Geom.*, pages 119–126, ACM Press, 1995.

[Fjä91] P.-O. Fjällström. Polyhedral approximation of bivariate functions. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 187–190, 1991.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[Fra02] W.R. Franklin. Siting observers on terrain. In *Proc. 10th Sympos. Spatial Data Handling*, pages 109–120, Springer, Berlin, 2002.

[FW91] M. Formann and F. Wagner. A packing problem with applications to lettering of maps. In *Proc. 7th Sympos. Comput. Geom.*, pages 281–288, ACM Press, 1991.

[GD02] C. Gold and M. Dakowicz. Terrain modelling based on contours and slopes. In *Proc. 10th Sympos. Advances Spatial Data Handling*, pages 95–107, Springer, Berlin, 2002.

[GHK02] J. Gudmundsson, M. Hammar, and M. van Kreveld. Higher order Delaunay triangulations. *Comput. Geom.*, 23:85–98, 2002.

[GHMS93] L.J. Guibas, J.E. Hershberger, J.S.B. Mitchell, and J.S. Snoeyink. Approximating polygons and subdivisions with minimum link paths. *Internat. J. Comput. Geom. Appl.*, 3:383–415, 1993.

[GK06] J. Gudmundsson and M. van Kreveld. Computing longest duration flocks in trajectory data. In *14th ACM Int. Sympos. Geographic Inform. Syst.*, pages 35–42, 2006.

[GKM$^+$09] J. Gudmundsson, J. Katajainen, D. Merrick, C. Ong, and T. Wolle. Compressing spatio-temporal trajectories. *Comput. Geom.*, 42:825–841, 2009.

[GKS07] J. Gudmundsson, M. van Kreveld, and B. Speckmann. Efficient detection of patterns in 2D trajectories of moving points. *GeoInformatica*, 11:195–215, 2007.

[GKS13] J. Gudmundsson, M. van Kreveld, and F. Staals. Algorithms for hotspot computation on trajectory data. In *SIGSPATIAL 2013 Int. Conf. Advances Geographic Inform. Syst.*, pages 134–143, 2013.

[GLW08] J. Gudmundsson, P. Laube, and T. Wolle. Movement patterns in spatio-temporal data. In S. Shekhar and H. Xiong, editors, *Encyclopedia of GIS*, pages 726–732, Springer, Berlin, 2008.

[GNN13] A. Gemsa, B. Niedermann, and M. Nöllenburg. Trajectory-based dynamic map labeling. In *Proc. 24th Int. Sympos. Algorithms Comput.*, vol. 8283 of *LNCS*, pages 413–423, Springer, Berlin, 2013.

[GNR16] A. Gemsa, M. Nöllenburg, and I. Rutter. Consistent labeling of rotating maps. *J. Comput. Geom.*, 7:308—331, 2016.

[GNS07] J. Gudmundsson, G. Narasimhan, and M. Smid. Distance-preserving approximations of polygonal paths. *Comput. Geom.*, 36:183–196, 2007.

[GSBW11] X. Ge, I. Safa, M. Belkin, and Y. Wang. Data skeletonization via Reeb graphs. In *Proc. 24th Int. Conf. Neural Inform. Process. Syst.*, pages 837–845, Curran Associates, Red Hook, 2011.

[Gut84] A. Guttmann. R-Trees: a dynamic indexing structure for spatial searching. In *Proc. ACM-SIGMOD Int. Conf. Management of Data*, pages 47–57, 1984.

[HCC11] I. Heywood, S. Cornelius, and S. Carver. *An Introduction to Geographical Information Systems*. Prentice Hall, 4th edition, 2011.

[Hel90] M. Heller. Triangulation algorithms for adaptive terrain modeling. In *Proc. 4th Sympos. Spatial Data Handling*, pages 163–174, 1990.

[HG95] P.S. Heckbert and M. Garland. Fast polygonal approximation of terrains and height fields. Report CMU-CS-95-181, Carnegie Mellon University, 1995.

[HLM$^+$14] F. Hurtado, M. Löffler, I. Matos, V. Sacristán, M. Saumell, R.I. Silveira, and F. Staals. Terrain visibility with multiple viewpoints. *Internat. J. Comput. Geom. Appl.*, 24:275–306, 2014.

[HLT10] J. Han, Z. Li, and L.A. Tang. Mining moving object, trajectory and traffic data. In *Proc. 15th Int. Conf. Database Syst. Advanced Appl.*, pages 485–486, 2010.

[Høj98] P. Højholt. Solving local and global space conflicts in map generalization using a finite element method adapted from structural mechanics. In *Proc. 8th Int. Sympos. Spatial Data Handling*, pages 679–689, 1998.

[HS98] J. Hershberger and J. Snoeyink. Cartographic line simplification and polygon CSG formulae in $O(n \log^* n)$ time. *Comput. Geom.*, 11:175–185, 1998.

[HT14] H.J. Haverkort and L. Toma. Terrain modeling for the geosciences. In *Computing Handbook: Computer Science and Software Engineering*, 3rd edition, chap. 31, pages 1–21, Taylor & Francis, 2014.

[Imh75] E. Imhof. Positioning names on maps. *The American Cartographer*, 2:128–144, 1975.

[JBQZ04] M. Jiang, S. Bereg, Z. Qin, and B. Zhu. New bounds on map labeling with circular labels. In *Proc. 15th Int. Sympos. Algorithms Comput.*, vol. 3341 of *LNCS*, pages 606–617, Springer, Berlin, 2004.

[JC04] J.-W. Jung and K.-Y. Chwa. Labeling points with given rectangles. *Inform. Process. Lett.*, 89:115–121, 2004.

[KB01] M.-J. Kraak and A. Brown, editors. *Web Cartography: Developments and Prospects.* Taylor & Francis, 2001.

[KBS91] H.-P. Kriegel, T. Brinkhoff, and R. Schneider. The combination of spatial access methods and computational geometry in geographic database systems. In *Proc. 2nd Sympos. Advances in Spatial Databases*, vol. 525 of *LNCS*, pages 5–21, Springer, Berlin. 1991.

[KF94] I. Kamel and C. Faloutsos. Hilbert R-tree: An improved R-tree using fractals. In *Proc. 20th VLDB Conf.*, pages 500–510, 1994.

[KH98] C.J. Kocmoud and D.H. House. A constrained-based approach to constructing continuous cartograms. In *Proc. 8th Int. Sympos. Spatial Data Handling*, pages 236–246, 1998.

[KKL07] T. de Kok, M. van Kreveld, and M. Löffler. Generating realistic terrains with higher-order Delaunay triangulations. *Comput. Geom.*, 36:52–65, 2007.

[KMB05] P. Kalnis, N. Mamoulis, and S. Bakiras. On discovering moving clusters in spatio-temporal data. In *Proc. 9th Sympos. Advances Spatial Temporal Databases*, pages 364–381, 2005.

[KOS92] M.J. Katz, M.H. Overmars, and M. Sharir. Efficient hidden surface removal for objects with small union size. *Comput. Geom.*, 2:223–234, 1992.

[Kre01] M. van Kreveld. Smooth generalization for continuous zooming. In *Proc. 20th Int. Cartographic Conf.*, vol. 3, pages 2180–2185, 2001.

[KS07] M. van Kreveld and B. Speckmann. On rectangular cartograms. *Comput. Geom.*, 37:175–187, 2007.

[KS11] M. van Kreveld and R.I. Silveira. Embedding rivers in triangulated irregular networks with linear programming. *Int. J. Geographical Inform. Sci.*, 25:615–631, 2011.

[KSW99] M. van Kreveld, T. Strijk, and A. Wolff. Point labeling with sliding labels. *Comput. Geom.*, 13:21–47, 1999.

[KWFP10] C. Kaiser, F. Walsh, C.J.Q. Farmer, and A. Pozdnoukhov. User-centric time-distance representation of road networks. In *Proc. 6th Int. Conf. Geographic Inform. Sci.*, pages 85–99, 2010.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[Lau14] P. Laube. *Computational Movement Analysis*. Springer Briefs in Computer Science, Springer, Berlin, 2014.

[LGMR11] P.A. Longley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind. *Geographic Information Systems and Science*, 3rd edition. Wiley, New York, 2011.

[LJ01] M. Lonergan and C.B. Jones. An iterative displacement method for conflict resolution in map generalization. *Algorithmica*, 21:287–301, 2001.

[LK11] W.-C. Lee and J. Krumm. Trajectory preprocessing. In *Computing with Spatial Trajectories*, pages 3–33. 2011.

[LLE97] S.T. Leutenegger, M.A. Lopez, and J. Edington. STR: A simple and efficient algorithm for R-tree packing. In *Proc. 13th IEEE Int. Conf. Data Eng.*, pages 497–506, 1997.

[LMS97] M. Lanthier, A. Maheshwari, and J.-R. Sack. Approximating weighted shortest paths on polyhedral surfaces. In *Proc. 13th Sympos. Comput. Geom.*, pages 274–283, ACM Press, 1997.

[LS01] J.J. Little and P. Shi. Structural lines, TINs and DEMs. *Algorithmica*, 21:243–263, 2001.

[MG99] W.A. Mackaness and E. Glover. The application of dynamic generalization to virtual map design. In *Proc. 19th Int. Cartographic Conf.*, 1999. CD-ROM.

[MLW95] J.-C. Müller, J.-P. Lagrange, and R. Weibel, editors. *GIS and Generalization—Methodology and Practice*, volume 1 of *GISDATA*. Taylor & Francis, 1995.

[MP01] W.A. Mackaness and R. Purves. Automated displacement for large numbers of discrete map objects. *Algorithmica*, 21:302–311, 2001.

[MRS07] W.A. Mackaness, A. Ruas, and L.T. Sarjakoski, editors. *Generalisation of Geographic Information: Cartographic Modelling and Applications*. Elsevier, Amsterdam, 2007.

[MS99] M. McAllister and J. Snoeyink. Extracting consistent watersheds from digital river and elevation data. In *Proc. ASPRS/ACSM Annual Conference*, 1999.

[Ney99] G. Neyer. Line simplification with restricted orientations. In *Proc. 6th Workshop Algorithms Data Structures*, vol. 1663 of *LNCS*, pages 13–24, Springer, 1999.

[NP06] M. Nanni and D. Pedreschi. Time-focused clustering of trajectories of moving objects. *J. Intell. Inf. Syst.*, 27:267–289, 2006.

[NW00] G. Neyer and F. Wagner. Labeling downtown. In *Proc. Italian Conf. Algorithms and Complexity*, vol. 1767 of *LNCS*, pages 113–125, Springer, Berlin, 2000.

[NW11] M. Nöllenburg and A. Wolff. Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE Trans. Vis. Comp. Graph.*, 17:626–641, 2011.

[Oos94] P. van Oosterom. An R-tree based map-overlay algorithm. In *Proc. 5th Eur. Conf. Geogr. Inform. Syst.*, pages 318–327, EGIS, 1994.

[Oos95] P. van Oosterom. The GAP-tree, an approach to "on-the-fly" map generalization of an area partitioning. In J.-C. Müller, J.-P. Lagrange, and R. Weibel, editors, *GIS and Generalization—Methodology and Practice*, number 1 in GISDATA, Taylor & Francis, 1995.

[Per66] J. Perkal. On the length of empirical curves. Discussion paper 10, Ann Arbor Michigan Inter-University Community of Mathematical Geographers, 1966.

[PJ99] P. van der Poorten and C.B. Jones. Customisable line generalisation using Delaunay triangulation. In *Proc. 19th Int. Cartographic Conf.*, 1999. CD-ROM.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[PPH99] I. Petzold, L. Plümer, and M. Heber. Label placement for dynamically generated screen maps. In *Proc. 19th Int. Cartographic Conf.*, pages 893–903, 1999.

[PS94] L. Palazzi and J. Snoeyink. Counting and reporting red/blue segment intersections. *CVGIP: Graph. Models Image Process.*, 56:304–311, 1994.

[PS97] E. Puppo and R. Scopigno. Simplification, LOD and multiresolution principles and applications. In *Proc. EUROGRAPHICS*, vol. 16. Blackwell, 1997.

[PSS+03] S.-H. Poon, C.-S. Shin, T. Strijk, T. Uno, and A. Wolff. Labeling points with weights. *Algorithmica*, 38:341–362, 2003.

[RBK+08] I. Reinbacher, M. Benkert, M. van Kreveld, J.S.B. Mitchell, J. Snoeyink, and A. Wolff. Delineating boundaries for imprecise regions. *Algorithmica*, 50:386–414, 2008.

[RMM+95] A.H. Robinson, J. Morrison, P.C. Muehrcke, A.J. Kimerling, and S.C. Guptill. *Elements of Cartography*, 6th edition. John Wiley & Sons, New York, 1995.

[Ros04] J. Rossignac. Surface simplification and 3D geometry compression. Chap. 54 in J.E. Goodman and J. O'Rourke, *Handbook of Discrete and Computational Geometry*, 2nd edition, CRC Press, Boca Raton, 2004.

[Sam06] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, San Francisco, 2006.

[SBM95] M. Sambridge, J. Braun, and H. McQueen. Geophysical parameterization and interpolation of irregular data using natural neighbours. *Geophys. J. Int.*, 122:837–857, 1995.

[SBL+12] J. Shamoun-Baranes, E. van Loon, R.S. Purves, B. Speckmann, D. Weiskopf, and C.J. Camphuysen. Analysis and visualization of animal movement. *Biology Letters*, 2012:6–9, 2012.

[Sch98] B. Schneider. Geomorphologically sound reconstruction of digital terrain surfaces from contours. In *Proc. 8th Int. Sympos. Spatial Data Handling*, pages 657–667, 1998.

[SHWZ14] N. Schwartges, J.-H. Haunert, A. Wolff, and D. Zwiebler. Point labeling with sliding labels in interactive maps. In *Connecting a Digital Europe through Location and Place*, pages 295–310, Springer, Berlin, 2014.

[SI09] E. Shimizu and R. Inoue. A new algorithm for distance cartogram construction. *Int. J. Geographical Inform. Sci.*, 23:1453–1470, 2009.

[Sib81] R. Sibson. A brief description of natural neighbour interpolation. In V. Barnet, editor, *Interpreting Multivariate Data*, pages 21–36, Wiley, New York, 1981.

[SMK95] C. Silva, J.S.B. Mitchell, and A.E. Kaufman. Automatic generation of triangular irregular networks using greedy cuts. In *Proc. IEEE Conf. Vis.*, pages 201–208, 1995.

[Str01] T. Strijk. *Geometric Algorithms for Cartographic Label Placement*. PhD thesis, Utrecht University, Department of Computer Science, 2001.

[TDS+07] G. Trajcevski, H. Ding, P. Scheuermann, R. Tamassia, and D. Vaccaro. Dynamics-aware similarity of moving objects trajectories. In *15th ACM Sympos. Geographic Inform. Syst.*, page 11, 2007.

[TG00] D. Thibault and C.M. Gold. Terrain reconstruction from contours by skeleton construction. *GeoInformatica*, 4:349–373, 2000.

[Tob86] W.R. Tobler. Pseudo-cartograms. *The American Cartographer*, 13:43–50, 1986.

[Wat92] D.F. Watson. *Contouring: A Guide to the Analysis and Display of Spatial Data*. Pergamon, Oxford, 1992.

[WD04] M. Worboys and M. Duckham. *GIS: A Computing Perspective*. Taylor & Francis, 2nd edition, 2004.

[Wei97] R. Weibel. A typology of constraints to line simplification. In M.J. Kraak and M. Molenaar, editors, *Proc. 7th Int. Symposium on Spatial Data Handling*, pages 533–546, Taylor & Francis, 1997.

[WKK$^+$00] A. Wolff, L. Knipping, M. van Kreveld, T. Strijk, and P.K. Agarwal. A simple and efficient algorithm for high-quality line labeling. In P.M. Atkinson and D.J. Martin, editors, *Innovations in GIS VII: GeoComputation*, chap. 11, pages 147–159, Taylor & Francis, 2000.

[WW97] F. Wagner and A. Wolff. A practical map labeling algorithm. *Comput. Geom.*, 7:387–404, 1997.

[WWKS01] F. Wagner, A. Wolff, V. Kapoor, and T. Strijk. Three rules suffice for good label placement. *Algorithmica*, 30:334–349, 2001.

[ZZ11] Y. Zheng and X. Zhou, editors. *Computing with Spatial Trajectories*. Springer, New York, 2011.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.