

Area-preserving subdivision schematization

Citation for published version (APA):

Meulemans, W., Renssen, van, A. M., & Speckmann, B. (2010). Area-preserving subdivision schematization. In 6th International Conference on Geographic Information Science (GIScience) (pp. 160-174). (Lecture Notes in Computer Science; Vol. 6292). Berlin: Springer. DOI: 10.1007/978-3-642-15300-6_12

DOI:

[10.1007/978-3-642-15300-6_12](https://doi.org/10.1007/978-3-642-15300-6_12)

Document status and date:

Published: 01/01/2010

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Area-Preserving Subdivision Schematization*

Wouter Meulemans, André van Renssen, and Bettina Speckmann

Dep. of Mathematics and Computer Science, TU Eindhoven, The Netherlands.

w.meulemans@student.tue.nl a.m.v.rensen@student.tue.nl
speckman@win.tue.nl

Abstract. We describe an area-preserving subdivision schematization algorithm: the area of each region in the input equals the area of the corresponding region in the output. Our schematization is axis-aligned, the final output is a rectilinear subdivision. We first describe how to convert a given subdivision into an area-equivalent rectilinear subdivision. Then we define two area-preserving contraction operations and prove that at least one of these operations can always be applied to any given simple rectilinear polygon. We extend this approach to subdivisions and showcase experimental results. Finally, we give examples for standard distance metrics (symmetric difference, Hausdorff- and Fréchet-distance) that show that better schematizations might result in worse shapes.

Keywords: Schematization, polygonal subdivisions.

1 Introduction

A schematic map displays a set of nodes and their connections—for example, highway, train, or metro networks—in a highly simplified form to communicate the connectivity information as effective as possible. Connections are usually drawn as polygonal paths using few links and few orientations; the orientations are restricted to be axis-parallel or to adhere to the four main orientations.

Although most previous efforts are concentrated on the schematization of networks, it is of course also possible, and often desirable, to schematize the boundaries of regions or even complete subdivisions. This is particularly useful in conjunction with schematic networks. Consider, for example, a schematized railway network which is displayed on top of a geographic base map. A detailed depiction of the region’s boundary distracts from the schematic map, whereas a schematized version supports the schematic map (see Fig. 1). Schematized regions are also used when depicting fare zone boundaries. Generally, whenever exact boundaries are not needed it is preferable



Fig. 1. Danish rail network [<http://www.eurail.com/>].

* B. Speckmann is supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.022.707.

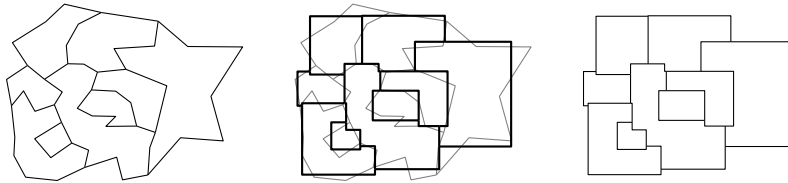


Fig. 2. A subdivision and an area-preserving schematization.

to replace them by schematic ones, to reduce visual clutter and indicate that the purpose of the map in question is not a (purely) geographic one.

A schematized subdivision of high visual quality satisfies at least the following criteria. Regions are approximated using few links and few orientations. There are no self-intersections and the region adjacencies of the input map are maintained. Finally, the output visually resembles the input, that is, region shapes and sizes are preserved as well as possible. It is comparatively easy to avoid self-intersections and to ensure proper adjacencies. However, it is less clear how to create regions of the “best” shape.

Results. We focus on area-preserving schematization, that is, the area of each region in the input subdivision equals the area of the corresponding region in the output. In particular, we present a schematization algorithm which is based on two simple area-preserving contraction operations. Our schematization is axis-aligned, the final output is a rectilinear subdivision (see Fig. 2). Our contraction operations are defined for rectilinear polygons. Hence our first step is to convert a given input subdivision into an area-equivalent rectilinear subdivision, see Section 2 for details. Experiments show that our rectilinearization approach increases the number of edges only by a small constant factor.

In Section 3 we discuss our contraction operations in detail. We prove that any given rectilinear polygon with 6 or more edges can be simplified in an area-preserving manner. That is, at least one of our operations can always be executed. We then extend our approach to subdivisions and show how our operations can be adapted to vertices of degree 3. For simple polygons we can guarantee any desired output complexity; this is not the case for subdivisions. However, our experiments show that also subdivisions can be schematized using few edges.

For a given input subdivision there are clearly many area-preserving schematizations of equal complexity (number of edges). It would seem natural to choose the best of those with respect to any standard distance metric, such as the symmetric difference, the Hausdorff-distance, or the Fréchet-distance. However, in Section 4 we show examples for each of these distance functions where better approximations result in worse shapes. Hence our algorithm does not try to minimize either of these distance functions but instead greedily contracts those parts of the subdivision where the contraction results in the smallest symmetric difference. Finally, in Section 5 we showcase some results of our algorithm.

Related Work. There is an ample body of work on map schematization and metro map construction. For example, Cabello *et al.* [1] give an algorithm that

schematizes a given network using two or three links per path, if that is possible. Nöllenburg and Wolff [2] use a method based on mixed-integer programming to generate metro maps using one edge per path. Swan *et al.* [3] give an extensive overview of existing schematization algorithms and study their applicability to automated schematic map construction for web services. Algorithms for map schematization can be used to schematize subdivisions. However, they usually do not take criteria such as shape and size preservation into account.

Cartographic generalization is a very active research field, with a multitude of new results appearing each year. Of particular relevance to this paper is the generalization of urban data, specifically building generalization [4–7]. Building generalization typically involves several consecutive actions or operators. Among those, building wall squaring [8–10] and outline simplification [9, 11, 12] are most closely related to our work. Algorithms for these generalization tasks can also be used for subdivision schematization and vice versa; in Section 5 we show a few examples of building outlines generalized with our algorithm.

Line simplification has been a prominent topic in the GIS literature for many years and various quality criteria have been proposed. One of the possible criteria is areal displacement; see the work by Bose *et al.* [13] for an area-preserving approach. When simplifying subdivisions it is generally not advisable to simplify each chain of the subdivision in isolation. There are some approaches, developed in computational geometry, that preserve the topology of the input subdivision. De Berg *et al.* [14, 15] describe a method that simplifies a polygonal subdivision without introducing intersections or passing over special input points. Estkowski and Mitchell [16] give a heuristic for simplifying parallel lines, such as elevation contours. Van de Kraats *et al.* [17] discuss the special case where the subdivision to be simplified is a printed circuit board. Unfortunately many subdivision simplification problems that minimize the output complexity are NP-complete [18].

2 Rectilinearization

Here we describe how to turn a simple subdivision S into a simple *area-equivalent* rectilinear subdivision R . That is, R is a simple rectilinear subdivision, the regions of R correspond one-to-one to the regions of S , the adjacencies between regions are maintained, and the area of each region in R equals the area of its corresponding region in S . We assume that the input subdivision S has vertex degree at most three. The *complexity* (number of edges) of the output subdivision R depends on the minimal distance δ between a vertex and an edge of S .

The quality of the schematization improves if R does not contain “long” edges. We use a small constant fraction $\alpha \approx 0.002$ of the diameter of S as an upper bound for the edge length and split all edges of S which are longer.

We associate four axis-aligned *quadrants* with each vertex v of S . We call a vertex v of an edge (u, v) *sharp* if there is another edge (u', v) such that u and u' lie in the same quadrant of v (see Fig. 3). Let $e = (u, v)$ be an edge of S . We assign

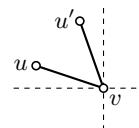


Fig. 3. v is sharp.

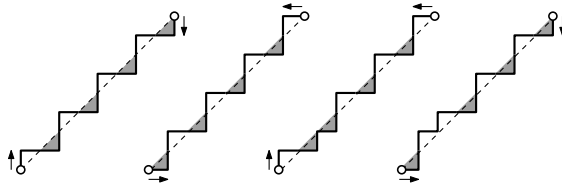


Fig. 4. An edge is rectilinearized using 8 steps.

an axis-aligned direction (up, down, left, or right) to each pair $\langle e, u \rangle$ and $\langle e, v \rangle$ independently. Since each vertex has degree at most three we can easily find an assignment of directions to its outgoing edges which ensures that (i) no two edges are assigned the same direction, and (ii) the total angular deviation is minimized. We assume for ease of explanation that not all three outgoing edges of a vertex lie in the same quadrant. Our approach can be adapted to also deal with this case (and with vertices of degree four), but at a substantial increase of cases to be considered.

We now show how to rectilinearize each edge within its axis-aligned bounding box in an area-preserving manner while avoiding intersections. We first consider an edge e without a sharp vertex. Let $d(e)$ denote the minimal distance between e and any edge that intersects the axis-aligned bounding box of e . We ensure that the maximal distance between e and its rectilinearization is at most $d(e)/2$, which implies that we do not introduce intersections. We rectilinearize e by making $s(e)$ many *steps*: a step starts on e , then goes horizontally (or vertically) away from e and returns vertically (or horizontally) to e . An area-preserving approximation must take as many vertical as horizontal steps, hence $s(e)$ needs to be even. A step can cover at most a distance $d(e)$ along e , otherwise the distance between e and its rectilinearization exceeds $d(e)/2$. The number of steps hence equals at least the length of e divided by $d(e)$ and rounded up to the next even number. We alternate steps depending on the directions assigned to e at its two vertices, see Fig. 4 for some examples.

Next we consider edges that have a sharp vertex. If an edge e has two sharp vertices, we split it. Let e now be an edge with a sharp vertex v and let e' be the second edge that has v as a vertex and lies in the same quadrant of v . In principle we treat e as before, with two exceptions. The first difference is the

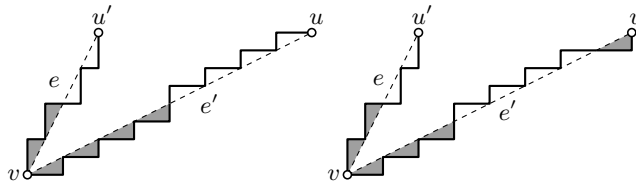


Fig. 5. Rectilinearizing edges with sharp vertices.

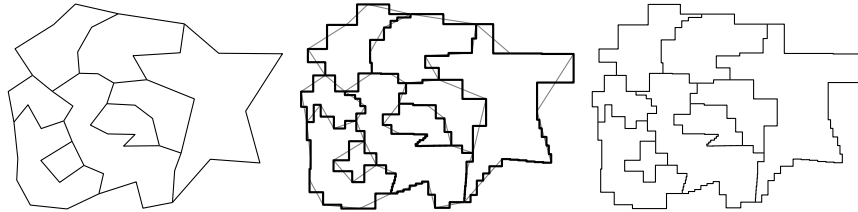


Fig. 6. A subdivision and an area-preserving rectilinearization (with $\alpha = 0.1$).

computation of $d(e)$. For this, we ignore the first quarter of e starting at v , that is, we compute $d(e)$ as the distance to a shorter edge \hat{e} that coincides with e but is missing the first quarter. Secondly, when rectilinearizing e we use “evasive” behavior along the first quarter: all steps lie on the “other” side (see Fig. 5).

The procedure as outlined above turns a simple subdivision S into a simple area-equivalent rectilinear subdivision R in $O(n^2 + m)$ time, where n is the complexity of S and m is the complexity of R . See Fig. 6 for an example. In our experiments the number of edges of the rectilinearized subdivision was always within a constant factor (usually around three, never more than eight) of the number of edges of the input subdivision.

3 Schematization

In this section we describe how to simplify a given rectilinear polygon or subdivision with the help of two area-preserving contraction operations. In Subsection 3.1 we first introduce our operations and then prove that at least one of them can be applied to any simple rectilinear polygon with at least six vertices. In Subsection 3.2 we extend our approach to subdivisions and sketch how to adapt our operations to vertices of degree three.

3.1 Simple polygons

Assume that we are given a simple rectilinear polygon R . Our area-preserving contraction operations work with configurations of three consecutive edges along the boundary of R . We distinguish *S-configurations* (a left turn followed by a right turn or vice versa) and *C-configurations* (two left turns or two right turns).

The first operation is the *S-contraction*: we replace an S-configuration with the weighted average of its two outer edges and connect this new edge to the previous neighbors of the S-configuration (see Fig. 7). This operation is clearly area-preserving. An S-contraction reduces the complexity of the polygon by at least two. The *contraction area* of an S-configuration is the symmetric difference between the polygon before and after the corresponding S-contraction (indicated in gray in Fig. 7). An S-configuration

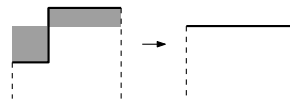


Fig. 7. An S-contraction.

is *feasible* if its contraction area is empty. Not surprisingly, an arbitrary rectilinear polygon might not have any feasible S-configurations. In this case we need our second operation, the *C-contraction*, which is based on two complementary C-configurations.

We distinguish two types of C-configurations: *inner C-configurations* (the interior of the polygon lies on the same side of the middle edge as the two outer edges) and *outer C-configurations* (the interior of the polygon lies on the other side of the middle edge as the two outer edges). A C-contraction requires both an inner and an outer C-configuration. We move the middle edges of both C-configurations simultaneously, until the length of one of the outer edges is reduced to zero. A simple calculation shows that we can choose the speed with which to move each edge in such a way that the operation is area-preserving (see Fig. 8). Just as an S-contraction, a C-contraction reduces the complexity of the polygon by at least two. The contraction area of a C-configuration is the rectangle defined by its middle edge and the shorter of its two outer edges. As before, a C-configuration is feasible if its contraction area is empty.

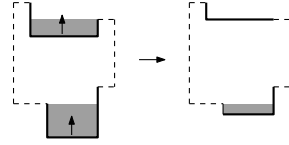


Fig. 8. A C-contraction.

We now prove through a sequence of lemmas that every simple rectilinear polygon R with at least six edges has either a feasible S-configuration or two complimentary feasible C-configurations. That is, at least one of our operations can be applied to R as long as R is not a rectangle.

Lemma 1. *Every rectilinear polygon R with at least six edges has a feasible inner C-configuration.*

Proof. Let e_h be the highest horizontal edge of R . The two vertical neighbors of e_h must necessarily be directed down and hence e_h is the middle edge of an inner C-configuration C_h . If C_h is feasible, then we are done. Otherwise let e'_h be the highest horizontal edge in the contraction area of C_h . We distinguish two cases: (i) e'_h is connected to a neighbor of e_h , and (ii) e'_h is not connected to a neighbor of e_h . If e'_h is connected to a neighbor e_i of e_h , then e_h , e_i and e'_h form an inner C-configuration (see Fig. 9 (i)). This C-configuration must be feasible since e'_h is the highest horizontal edge in the contraction area of C_h .

If e'_h is not connected to a neighbor of e_h then its vertical neighbors must be directed down (see Fig. 9 (ii)). There is only one way—topologically speaking—to

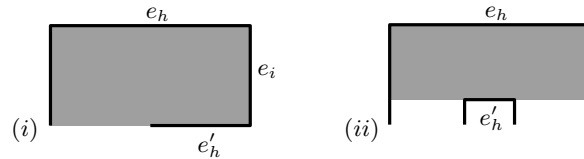


Fig. 9. C_h is not feasible: (i) e'_h is connected to a neighbor of e_h and (ii) e'_h is not connected to a neighbor of e_h .

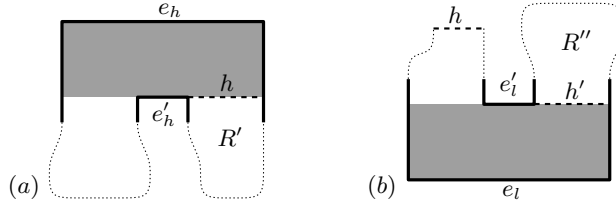


Fig. 10. An example of a virtual edge v completing the right subpart.

connect the neighbors of e_h to the neighbors of e'_h while keeping the polygon simple, ensuring that e_h is the highest horizontal edge, and ensuring that e'_h is the highest horizontal edge in the contraction area of C_h (see Fig. 10 (a)). We connect e'_h horizontally to a neighbor of e_h with a *virtual edge* h (see Fig. 10 (a)). The virtual edge splits R into two sub-polygons, both of which have strictly smaller complexity than R . We recurse on the sub-polygon R' that does not contain e_h . We will apply the same reasoning as before, but this time using the lowest horizontal e_l for our argument to avoid h .

Let e_l hence be the lowest horizontal edge of R' . By construction e_l cannot be h . Let C_l be the C-configuration with middle edge e_l . If C_l is feasible, then we are done. Otherwise, if R' is a rectangle, then the only edge inside the contraction area of C_l is h and hence C_l is feasible. It remains to consider the case where R' is not a rectangle and C_l is not feasible. We distinguish two cases (i) and (ii) as above. If we are in case (ii) then we split R' with a virtual edge h' that connects the lowest horizontal edge e'_l inside C_l to one of the neighbors of e_l . Now, and in all further recursions, we have to carefully choose to which neighbor of e_l to connect: we need to create a new sub-polygon R'' that contains neither e_l nor h (see Fig. 10 (b)). There is always exactly one possibility for h' . In R'' we continue again with the highest horizontal edge, and so on. Since the complexity of the polygon under consideration strictly decreases with each iteration, and since the lowest (or highest, depending on parity) C-configuration is always feasible if we recurse down to a rectangle, the lemma follows. \square

The *slab* of an edge e , denoted by $\text{slab}(e)$, is the region bounded by e and two half-lines orthogonal to e starting at the two endpoints of e , such that the interior of the polygon R does not intersect $\text{slab}(e)$ in the immediate neighborhood of e .

Lemma 2. *If there is an edge e such that $\text{slab}(e)$ contains a point of the boundary of R which is not directly connected to e , then R has an outer C-configuration.*

Proof. Without loss of generality assume that e is horizontal and that $\text{slab}(e)$ lies above e . Let p be the point in $\text{slab}(e)$ which is closest to e and not connected to e by a single vertical edge. Further, let q be the closest point to p on e . We connect p and q with a virtual edge h which splits the outside of R into two parts, one bounded and one unbounded (see Fig. 11). Some special care has to be taken if p lies on the boundary of $\text{slab}(e)$: if the neighbor e' of e below p is directed upwards, then h connects to the upper vertex of e' .

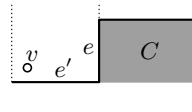
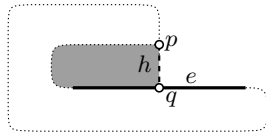


Fig. 11. Finding an outer C-configuration. **Fig. 12.** S-configuration next to C .

Denote the bounded part of the outside of R by R' and assume that R' lies to the left of h as depicted in Fig. 11. Let e_l be a leftmost vertical edge of R' . Since e_l is leftmost it must be the middle edge of an outer C-configuration of R . By construction, the virtual edge h cannot be part of this C-configuration. If R' lies to the right of h the argument is symmetric. \square

Lemma 3. *Every rectilinear polygon R with at least six edges has either a feasible S-configuration or an outer C-configuration.*

Proof. R has a feasible inner C-configuration C by Lemma 1. Let e be the shorter of the two outer edges of C . Since C is feasible, e is the middle edge of an S-configuration S . If S is feasible, then we are done. Otherwise, denote with e' the neighbor of e that is not part of C . Since S is not feasible, there must be a vertex v of R inside its contraction area. Vertex v cannot lie inside the contraction area of C and must hence lie in the slab of e' (see Fig. 12). Lemma 2 now implies an outer C-configuration. \square

Lemma 4. *If a rectilinear polygon R has an outer C-configuration, then it also has a feasible outer C-configuration.*

Proof. The proof of Lemma 4 is very similar to the proof of Lemma 1, so we only sketch the argument. Let C be an outer C-configuration of R . If C is feasible, then we are done. Otherwise let e be the middle edge of C and assume without loss of generality that e is horizontal. Further, assume that R lies locally below e . Let e' be the lowest horizontal edge in the contraction area of C . As before, either e' is directly connected to e and forms a feasible outer C-configuration with their joint neighbor, or e' is not directly connected to e and we can split the outside of R with a virtual edge h into two parts, one bounded and one unbounded (see Fig. 13). We now turn the problem inside out and find a feasible inner C-configuration inside the bounded part of the outside of R . \square

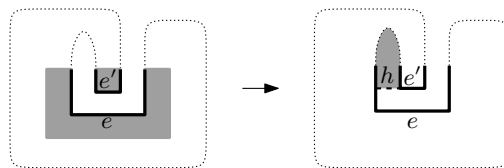


Fig. 13. Finding an inner C-configuration on the outside of R .

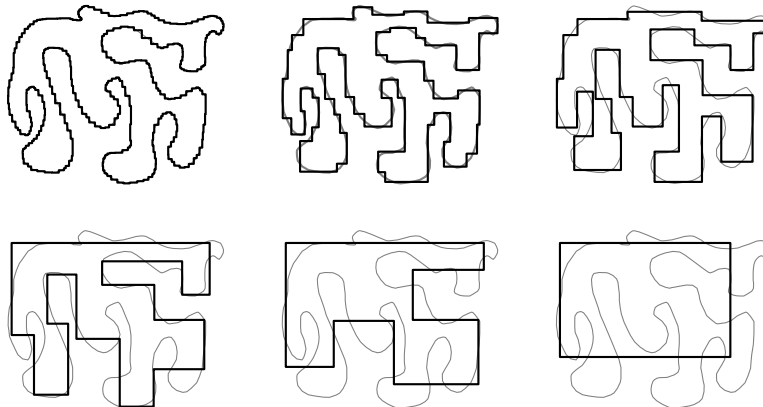


Fig. 14. Schematizing a “Matisse leaf”: rectilinearization (with $\alpha = 0.1$), 100, 50, 24, 12, and 4 edges.

Theorem 1. *Given a rectilinear polygon R with n edges and an integer k with $4 \leq k \leq n$, an area-preserving schematization of R with at most k edges can be generated using only S - and C -contractions.*

Theorem 1 guarantees that we can always find a feasible contraction, as long as the polygon is not a rectangle. Often we even have a choice out of several feasible operations. Our algorithm then contracts those parts of the subdivision where the contraction results in the smallest symmetric difference with respect to the current schematization (see Fig. 14). C -contractions are inherently not local, the inner and outer C -configurations might lie in completely different parts of the polygon. This can pose a problem with symmetric polygons, insofar that we do not necessarily create symmetric output. We therefore try to find complementary C -configurations which are “close”.

To increase the number of feasible contractions, we use a weakened definition of “feasible” for C -contractions. We do not require the contraction area of the larger C -configuration to be empty, it is sufficient if a contraction up to the area of the smaller C -configuration is possible. The bookkeeping necessary for an efficient implementation of our algorithm can be done with standard data structures. Each contraction (re)moves only a constant number of vertices and changes the contraction area of a constant number of edges. This leads to a total running time of $O(n^2)$. Similar to [14] we can easily extend our algorithm to support *landmarks*: special points that lie inside a particular face of the input, must remain in this face, and cannot be moved.

3.2 Subdivisions

Our area-preserving contraction operations can be adapted to schematize subdivisions. If the input subdivision is not rectilinear, then we first use the approach

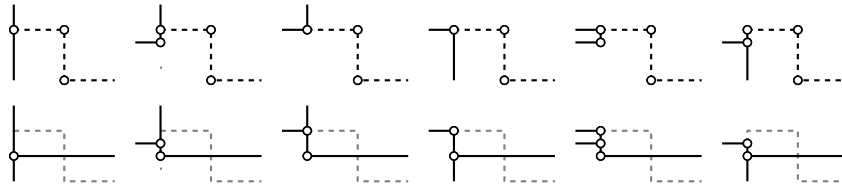


Fig. 15. Several cases for an S-contraction close to a vertex of degree three.

described in Section 2 to turn the input into a rectilinear subdivision. As before, we assume that we have only vertices of degree two and three. For subdivisions we cannot guarantee that the schematization can proceed to remove edges until only one edge per polygonal chain remains. However, experimental results show that we can reduce the complexity of the subdivision significantly.

Our ability to use C-contractions in subdivisions is very restricted, we require the middle edges of two complementary C-configurations to be adjacent to the same face of the subdivision. To give the algorithm sufficient flexibility, we allow S-contractions to change edge orientations around vertices of degree three (see Fig. 15). These special S-contractions might remove only one or even zero edges, but are necessary to make further progress afterwards.

4 Distance measures

In this section we consider the quality of area-preserving approximations of simple rectilinear polygons with respect to three standard distance metrics, namely the symmetric difference, the Hausdorff-distance, and the Fréchet-distance. We show examples for each of these distance measures where among two approximations with equal complexity the one with smaller distance has a worse shape compared to the input polygon. Hence, while it is in principle desirable that an approximation or schematization has a small distance to the input, it is not true that the approximation that minimizes this distance preserves the shape best.

Symmetric difference. The *symmetric difference* between two polygons is defined as the total area that is covered by one polygon but not by the other: it is

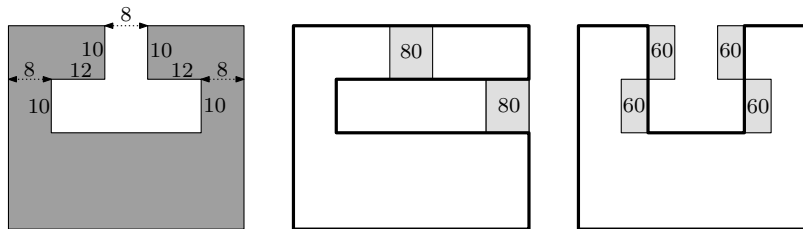


Fig. 16. A polygon and two area-preserving approximations: with minimal symmetric difference (middle), and with better shape (right).

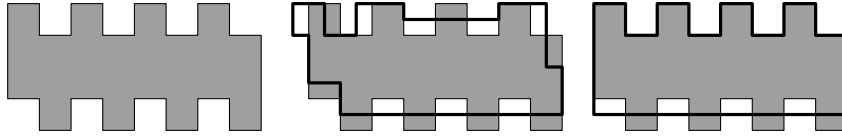


Fig. 17. A polygon and two area-preserving approximations with the same Hausdorff-distance, but significantly different shapes.

exactly the area in which they differ from each other. Consider the example in Fig. 16. The input is a 12-sided rectilinear polygon which we would like to approximate in an area-preserving manner with an 8-sided rectilinear polygon. The solution with minimal symmetric difference loses the vertical axis of symmetry and converts the polygon from a U-shape to a C-shape.

Hausdorff-distance. The *Hausdorff-distance* $d_H(X, Y)$ measures the distance between two subsets X and Y of the plane. It finds for each point in X the closest point in Y , and vice versa, and then takes the maximum of these distances. The example in Fig. 16 also shows that an area-preserving approximation with the smallest Hausdorff-distance can have a worse shape than an approximation with a slightly larger distance. Furthermore, since the Hausdorff-distance is determined by a maximum value, the quality of the shape of two area-preserving approximations with the same complexity and Hausdorff-distance can differ greatly (see Fig. 17).

Fréchet-distance. The *Fréchet-distance* measures the similarity of two curves by measuring the minimal maximal difference when “walking” along the two curves without moving backwards. Although the Fréchet-distance is defined on continuous curves it can also be applied to polygons. The Fréchet-distance is very sensitive to outliers. Consider the example in Fig. 18 where an 8-sided polygon is approximated by an area-equivalent rectangle. The solution with the smallest Fréchet-distance tries to approximate the thin part, while ignoring most of the polygon.

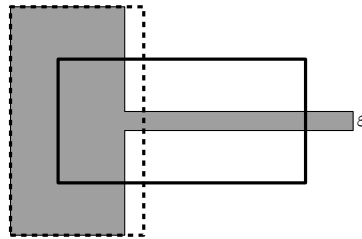


Fig. 18. A polygon and two area-preserving approximations: with minimal Fréchet-distance (solid), and with a better shape (dashed).

5 Experimental results

We have implemented our area-preserving subdivision schematization algorithm and we have generated schematized versions of various polygons and polygonal subdivisions. Fig. 2, 6, and 14 in earlier sections of this paper have all been created by our program. In this section we showcase some additional results.

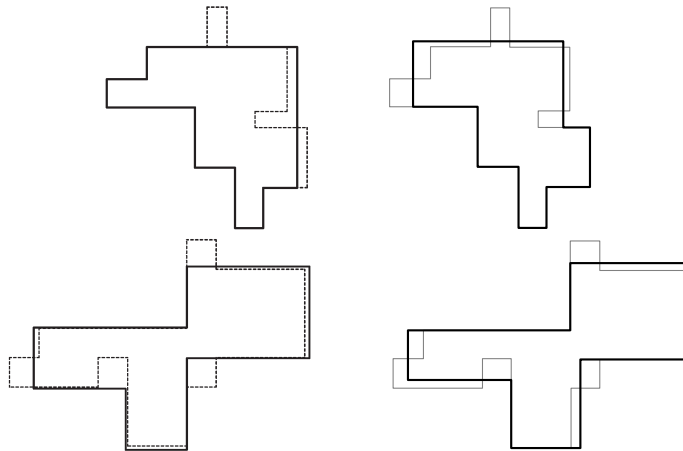


Fig. 19. Two examples from [8] and our schematizations of the same buildings (right).

As mentioned before, our algorithm can also be used to perform building generalization. In Fig. 19 we compare our generalizations to those obtained in [8]. Fig. 20 shows the generalized outline of a castle. In both cases it appears that the simple requirement of area preservation coupled with a greedy approach to minimize the symmetric difference enables us to capture the essential structure of the buildings. The final two figures, Fig. 21 and Fig. 22, show two large subdivisions, namely the provinces and islands of the Netherlands and the countries of Europe, including large lakes and islands. Clearly such maps will benefit from an extension of our methods to the four main orientations, but nevertheless, even the current axis-aligned approach already leads to visually pleasing results.



Fig. 20. A schematized castle.

6 Conclusions and open problems

We described an area-preserving subdivision schematization algorithm which is based on two simple area-preserving contraction operations. We proved that at least one of our operations can always be applied to any given simple rectilinear polygon with at least six vertices. We extended this approach to subdivisions and experimentally evaluated the quality of the resulting schematizations. We also gave examples for standard distance metrics that show that better schematizations might result in worse shapes.

An obvious direction for further work is an extension to the four main orientations. Also, we greedily choose the next contraction that incurs the least symme-

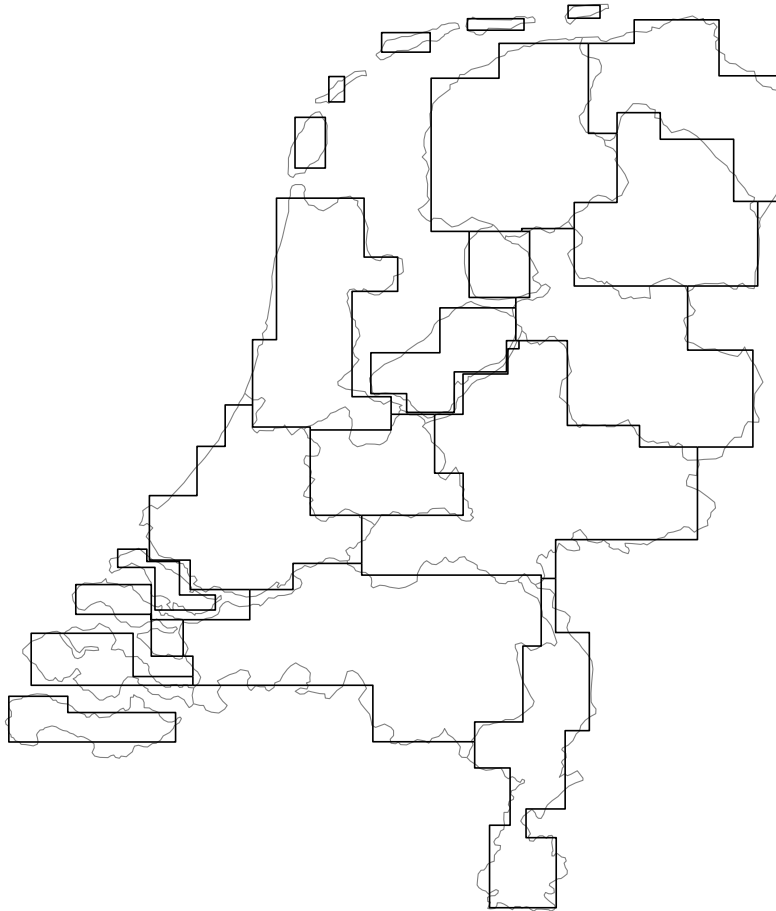


Fig. 21. The provinces and islands of the Netherlands (184 edges).

tric difference. This might lead to an asymmetric schematization of a symmetric polygon and hence other criteria might be more appropriate. Finally, the use of C-contractions is currently very restricted in subdivisions, we require the middle edges of two complementary C-configurations to be adjacent to the same face of the subdivision. Approaches where area is moved via a cycle of neighboring countries might give better results.

Acknowledgements. The authors would like to thank Marc van Kreveld for helpful discussions on the topic of this paper.

References

1. Cabello, S., de Berg, M., van Kreveld, M.: Schematization of networks. *Computational Geometry: Theory and Applications* **30**(3) (2005) 223–238

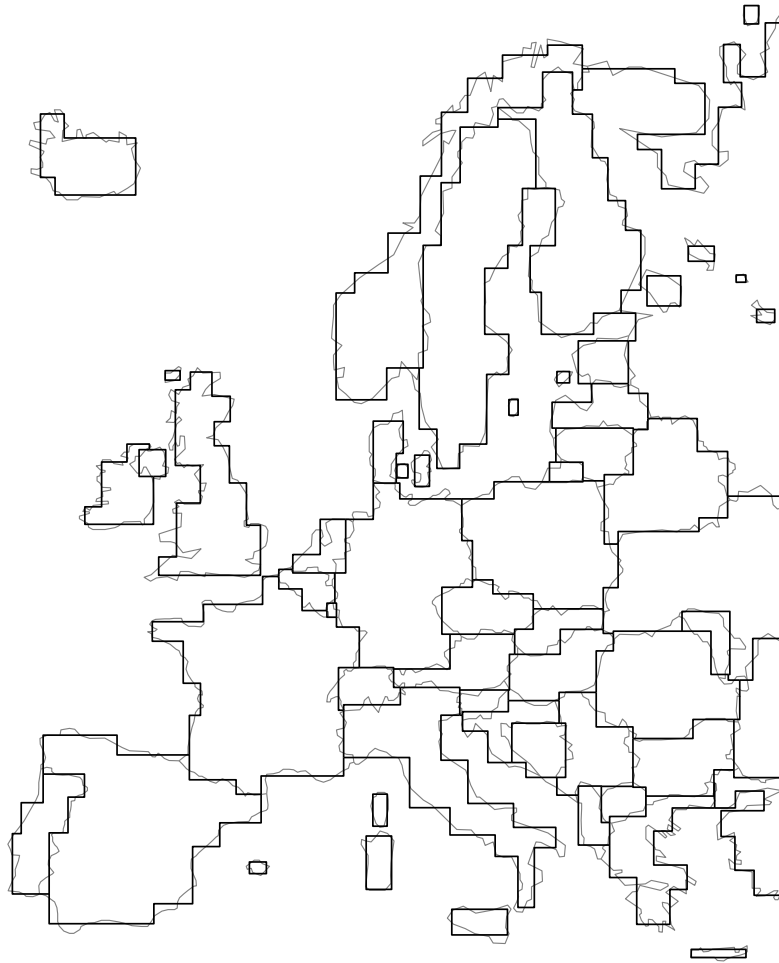


Fig. 22. The countries of Europe, including major islands and lakes (488 edges).

2. Nöllenburg, M., Wolff, A.: A mixed-integer program for drawing high-quality metro maps. In: 13th International Symposium on Graph Drawing (GD'05). Volume 3843 of LNCS. (2005) 321–333
3. Swan, J., Anand, S., Ware, M., Jackson, M.: Automated schematization for web service applications. In: Web and Wireless Geographical Information Systems. Volume 4857 of LNCS. (2007) 216–226
4. Damen, J., van Kreveld, M., Spaan, B.: High quality building generalization by extending morphological operators. In: 11th ICA Workshop on Generalization and Multiple Representation, Montpellier, France. (2008)
5. Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W., Weibel, R.: The application of agents in automated map generalisation. In: 19th International Cartographic Conference. (1999)

6. Sester, M.: Generalization based on least squares adjustment. *ISPRS - International Archives of Photogrammetry and Remote Sensing* **13** (2000) 931–938
7. Yan, H., Weibel, R., Yang, B.: A multi-parameter approach to automated building grouping and generalization. *GeoInformatica* **12** (2008) 73–89
8. Mayer, H.: Scale-spaces for generalization of 3d buildings. *International Journal of Geographical Information Science* **19** (2005) 975–997
9. Regnauld, N., Edwardes, A., Barrault, M.: Strategies in building generalisation: modelling the sequence, constraining the choice. In: *ICA Workshop on Progress in Automated Map Generalization*, Ottawa, Canada. (1999)
10. Ruas, A.: *Modèle de généralisation de données géographiques à base de contraintes et d'autonomie*. PhD thesis, Université de Marne la Vallée (1999)
11. Haunert, J.H., Wolff, A.: Optimal simplification of building ground plans. In: *Proceedings of XXIST ISPRS Congress Beijing 2008, IAPRS Vol. XXXVII (Part B2)*. (2008) 372–378
12. Sester, M.: Optimization approaches for generalization and data abstraction. *International Journal of Geographical Information Science* **19** (2005) 871–897
13. Bose, P., Cabello, S., Cheong, O., Gudmundsson, J., van Kreveld, M., Speckmann, B.: Area-preserving approximations of polygonal paths. *Journal of Discrete Algorithms* **4**(4) (2006) 554–566
14. de Berg, M., van Kreveld, M., Schirra, S.: A new approach to subdivision simplification. In: *Proceedings of Auto-Carto 12*. (1995) 79–88
15. de Berg, M., van Kreveld, M., Schirra, S.: Topologically correct subdivision simplification using the bandwidth criterion. *Cartography and Geographic Information Science* **25**(4) (1998) 243–257
16. Estkowski, R., Mitchell, J.S.B.: Simplifying a polygonal subdivision while keeping it simple. In: *17th Symposium on Computational Geometry*. (2001) 40–49
17. van de Kraats, B., van Kreveld, M., Overmars, M.: Printed circuit board simplification: simplifying subdivisions in practice. In: *11th Symposium on Computational Geometry*. (1995) 430–431
18. Guibas, L., Hershberger, J., Mitchell, J., Snoeyink, J.: Approximating polygons and subdivisions with minimum-link paths. *International Journal of Computational Geometry and Applications* **3** (1993) 383–415