Map schematization with circular arcs

Map Schematization with Circular Arcs*

Thomas C. van Dijk[1], Arthur van Goethem[2], Jan-Henrik Haunert[3]
Wouter Meulemans[2], and Bettina Speckmann[2]

[1] Universität Würzburg, Würzburg, Germany.
thomas.van.dijk@uni-wuerzburg.de
[2] Technical University Eindhoven, Eindhoven, The Netherlands.
[a.i.v.goethem|w.meulemans|b.speckmann]@tue.nl
[3] Universität Osnabrück, Osnabrück, Germany.
janhenrik.haunert@uni-osnabrueck.de

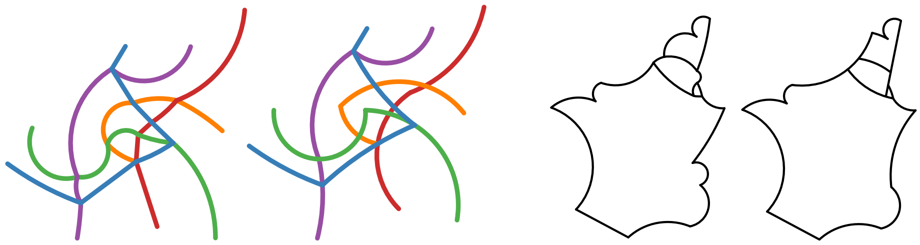
Abstract. We present an algorithm to compute schematic maps with circular arcs. Our algorithm iteratively replaces two consecutive arcs with a single arc to reduce the complexity of the output map and thus to increase its level of abstraction. Our main contribution is a method for replacing arcs that meet at high-degree vertices. This allows us to greatly reduce the output complexity, even for dense networks.
We experimentally evaluate the effectiveness of our algorithm in three scenarios: territorial outlines, road networks, and metro maps. For the latter, we combine our approach with an algorithm to more evenly distribute stations. Our experiments show that our algorithm produces high-quality results for territorial outlines and metro maps. However, the lack of caricature (exaggeration of typical features) makes it less useful for road networks.

1 Introduction

Maps are a common and intuitive way of communicating and exploring information with a geographic component. In many cases exact geographic details are not required to convey the primary information. For thematic maps exact details in the base map may even distract from or obscure the thematic overlay. Consequently, there has been a continuous interest in schematic maps (e.g., [3, 5, 18, 19]). A schematic map is typically highly abstract and stylized, maintaining only those features that support the message of the map. There exist a wide variety of schematic maps, including metro maps and chorematic diagrams [1].

Most automated methods to create schematic maps have focused on straight-line schematization, often with an orientation restriction [2, 20, 25] (e.g., admitting only horizontal, vertical and diagonal lines). In contrast, manually drawn schematic maps often use curves. It can be desirable to have a good continuation [15] of line features, to strengthen their representation. For example, it may

* A. van Goethem and B. Speckmann are supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 612.001.102 (AvG) and no. 639.023.208 (BS). T. van Dijk is supported by DFG-grant Ha5451/3-1. Collaboration started at Dagstuhl seminar 13151 “Drawing graphs and maps with curves”.



(a) Continuation of metro lines (b) Continuation of shorelines

Fig. 1. At points where multiple curves meet, continuity may improve schematization.

be desirable for metro lines to continue smoothly at interchanges or for shorelines to span multiple countries (see Fig. 1).

Results and organization. In Section 2 we present a new iterative and topologically correct schematization algorithm using circular arcs. This algorithm maintains good geometric correlation with the geographic input. It allows vertices of degree three or higher to be shifted, creating arcs that continue "through" such vertices. We also show how to tailor our algorithm specifically to metro maps. In Section 3 we present experimental results for three different scenarios: territorial outlines, road networks and metro networks. Our approach yields high-quality results for both territorial outlines and metro maps. It appears less suitable for road networks. A lack of *caricature* (exaggeration of typical features) interferes with the subconscious link between road type and shape. We also discuss the effect of the different algorithmic features on the maximal complexity reduction. In Section 4 we reflect on the implications of our design decisions.

Related work. Automated schematization has mostly restricted itself to representations with orientation-restricted line segments. There is a large number of results for the schematization of networks (e.g. [3, 18, 20, 25]) or even single lines (e.g. [6, 11, 19]). In contrast, only a few recent algorithms exist that explicitly aim to schematize outlines [2, 4]. The orientation-restricted style can enhance the visual clarity of a map as it promotes parallelism and the continuation of edges at high-degree vertices. Ti and Li [27] discuss the use of strokes and network distortion to further improve the usability of the schematization. However, Roberts [23] recently showed that manually drawn metro maps with curves are more efficient and effective than the long-standing octilinear designs.

Curves are important in manual cartography [22]. There are also several automated approaches to schematize an outline [9, 14] or a subdivision of outlines [12, 13] with circular arcs. However, both methods for subdivisions cannot move vertices of degree three or higher, although doing so is beneficial for subdivisions and crucial in dense metro networks. Fink *et al.* [10] use Bézier curves to draw metro maps. They are able to move high-degree vertices and aim to prevent abrupt turns of metro lines. However, Bézier curves inherently admit more freedom than circular arcs, resulting in a less strict schematic style (similar to the difference between simplification and orientation-restricted schematization).

2 Curved schematization algorithm

Our schematization algorithm iteratively replaces two neighboring circular arcs with a single arc while ensuring correct topology. This approach is similar to the one proposed by Van Goethem *et al.* [13] for territorial outlines. However, we present some significant improvements which make our algorithm more suitable for generic networks. Most importantly, we show how to move vertices of degree three and higher, to reduce the number of arcs of a schematization while improving the overall quality. Furthermore, in Section 2.4 we introduce some specific improvements geared towards metro maps.

2.1 Preliminaries

A *network* is a planar straight-line embedding of a graph in $\mathbb{R}^2$ which may represent various types of information such as metro lines or territorial outlines (subdivisions). The edges of a network are circular arcs (line segments are degenerate circular arcs). The edges meet at vertices. The *degree* of a vertex is its number of incident edges. We refer to vertices of degree three or higher as *junctions*. The *complexity* of a network N is its number of edges.

We require that the schematization N is *topologically equivalent* to the input network I. A schematization is topologically equivalent if there is a continuous function transforming I to N where at all times edges intersect only at vertices. This implies that N is planar, the order of incident edges around each vertex is maintained and that adjacencies are preserved.

2.2 Main algorithm

We describe an algorithm that computes a circular-arc schematization for a given network I. To this end, it maintains a network N; initially, N is a copy of I and consists only of straight edges (line segments). To create the schematization, two edges in N are replaced by a single edge (an *operation*). This reduces complexity and introduces circular arcs. We maintain as invariant that N is topologically equivalent to the input network I. Below, we provide details for the various steps of our algorithm; an overview is given in Algorithm 1.

Stroke partition. The main innovation of our algorithm is its ability to deal with junctions. We allow the conceptual removal of a junction, joining two incident edges into a single edge. The junction is then implicitly represented as the intersection of edges. To decide which edges may be joined by such an operation, we partition the network into *strokes*. A stroke is a "natural" path through the network, continuing relatively smoothly at junctions. Strokes may correspond to through roads (e.g., [26]) or to coastlines spanning multiple territories.

To compute a stroke partition, we proceed as follows. We first assign each edge to a unique stroke. Let E_v be the set of incoming edges for a vertex v of degree two or higher. For any pair of edges $e, f \in E_v$ we compute the angular deviation. The angular deviation at v equals 180 degrees minus the minimum

Algorithm 1 COMPUTECURVEDSCHEMATIZATION(N, k)

Input: Network N and desired complexity k
Output: Topologically equivalent circular-arc schematization of N

Partition N into strokes
Compute all operations O
while N has more than k edges and O contains an admissible operation **do**
    Execute the admissible operation $o \in O$ with the lowest cost
    Remove all operations involving an arc replaced by o
    Update admissibility of remaining operations
    Create new operations involving the edge introduced by o
return N

angle between e and f. We repeatedly combine the strokes of the pair of edges in E_v with the lowest angular deviation and remove them from E_v. The stroke partition can be computed in $O(d_i^2 \log d_i)$, where d_i is the maximum vertex-degree in the network.

Operations. We now define a set of *operations* O that can be executed on the network. An operation removes some vertex v at which two consecutive edges of a single stroke S meet. For now, assume that v has degree 2: no other strokes pass through v. Let u and w be the other endpoints of the two edges in S that are incident to v. To maintain topology, a new edge should be inserted connecting u and w. Thus, an operation replaces two consecutive edge with a single edge.

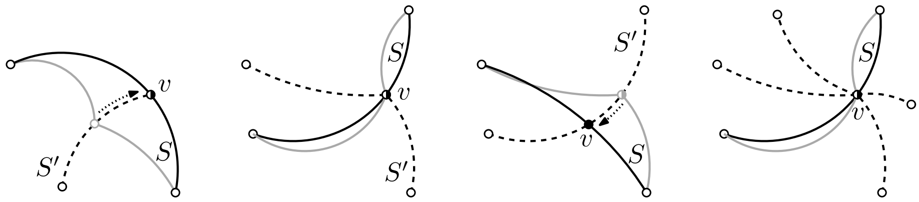
We call an operation *admissible* if it maintains the correct topology. To ensure a topologically equivalent result, the algorithm performs only admissible operations. As it depends on how we deal with junctions, we further discuss admissibility later in this section.

The *cost* of a replacement indicates the dissimilarity between the replacement edge and the represented input edges. We quantify this dissimilarity with the *Fréchet distance* [24]. For a single edge in N that represents n edges of the input, this measure can be computed in $O(n \log n)$ time [24]. By using this measure to weigh the operations, we maintain a geometric correlation between the resulting schematization and the geographic input.

The replacement with lowest cost for a given pair of edges may be inadmissible. To allow for more flexibility, we add three replacements to the set of operations O. To this end, we create a discrete set of candidate replacements using arcs of different radii and add the three replacements with lowest score. We generate these candidate replacements using angles of $i \cdot \frac{\pi}{k}$ with respect to line $\overline{uw}$ with $-k < i < k$ for some parameter k; we used $k = 20$ in our experiments.

Junctions. Above, we assumed that operations remove only vertices of degree 2. We now introduce operations for two edges that meet at a junction. The examples in Fig. 1, illustrating the gain of schematizing across junctions, have been generated with our algorithm without and with these additional operations.

Let v be a junction on stroke S. When replacing the edges in S incident to v, we must ensure that the junction remains. This constrains the replacement



(a) Degree 3: extension (b) Degree 4: restriction (c) Degree 4: reposition (d) Degree 5: restriction

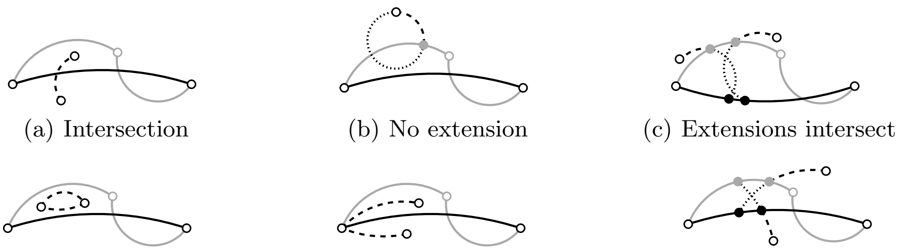
Fig. 2. Dealing with junctions. Regular vertices are white dots, virtual vertices are black. Replaced edges are indicated in gray, edges of other strokes are dashed.

edge. We keep track of these constraints by marking v as a *virtual vertex* on the replacement edge. Note that v is a virtual vertex only for strokes in which is has been "removed"; for other strokes, it remains a *regular vertex.* There are four cases that constrain the possible replacement arc. As v is a junction, it is included in at least two strokes; let S' denote a second stroke that includes v. Refer to Fig. 2.

(a) v has degree 3. Hence, v is an endpoint of S' and we extend or shorten the edge of S' such that its endpoint lies on the replacement edge.
(b) v has degree 4 and is a regular vertex on S'. To maintain the degree-4 vertex, we constrain the replacement edge to pass through v: only a single replacement candidate remains.
(c) v has degree 4 and is a virtual vertex on S'. We can reposition the virtual vertex of S' along its arc. This admits flexibility in the replacement edge and thus we use the same approach as for a degree-2 vertex.
(d) v has degree 5 or more. As in case (b), we constrain the replacement edge to pass through v.

An edge e in the network may have any number of virtual vertices along its boundary. These virtual vertices can constrain further replacements involving e. Let z be a virtual vertex on e. If z originates from a replacement of case (a) or (c), then its incident arc can be extended or z can be repositioned. Otherwise, any operation replacing e must maintain the position of z, thus limiting the possible replacements. If an operation is limited in this way by two or more virtual vertices, a replacement is only possible if these vertices are cocircular with the endpoints of the replacement arc.

Computing admissibility. An operation is admissible if it maintains the correct topology. To decide on admissibility, we proceed as follows. Let e_1 and e_2 be the two replaced edges and let v be the vertex at which these edges meet. Let u and w be the other vertex of e_1 and e_2 respectively. Finally, let e_r denote the replacement edge. An *involved virtual vertex* is a virtual vertex along e_1 or e_2. The *involved vertices* are all involved virtual vertices as well as u, v and w. A *virtually involved edge* is an edge that is incident to an involved virtual vertex;



(a) Intersection (b) No extension (c) Extensions intersect

(d) Vertex in region enclosed by e_1, e_2, e_r (e) Order of incident edges changed (f) Order of virtual vertices changed

Fig. 3. Examples of topology violations of an inadmissible operation. e_r given in black; e_1 and e_2 in gray. Other edges are dashed; extensions are dotted.

this does not include e_1 and e_2. The set of *uninvolved edges* contains all edges that are not e_1 or e_2 or an involved virtual edge. To compute admissibility, we check the following conditions; if any of these is not satisfied, the operation is not admissible. These conditions are illustrated in Fig. 3.

(a) There are no intersections between e_r and any uninvolved edge.
(b) Each involved virtual edge connects to or can be extended to e_r.
(c) Extensions of involved virtual edges do not intersect.
(d) There are no vertices in the regions enclosed by e_1, e_2 and e_r.
(e) The order of incident edges around each involved vertex is maintained.
(f) The order of virtual vertices along e_r and along the virtually involved edges is maintained.

To quickly update admissibility, we maintain for each operation a set of edges that cause inadmissibility: an operation is admissible if this set is empty.

Cycles to circles. When the start and end vertex of an operation are the same, the definitions for the operations can become degenerate. We introduce an additional operation for this case. Let v be the vertex being removed by the operation and u be the other endpoint of both edges. We define a set A of *anchor points* that the replacement circle needs to intersect. A consists of all junctions along both edges, possibly including u or v, that cannot be moved. If A has three or more points all points are required to be cocircular, uniquely defining a replacement circle; otherwise no replacement is possible. If A has two or less points, we extend A by up to two elements by adding u and, if required, v. We regularly sample tangents angles around the connecting chord defined by these two points, defining a set of possible replacement circles.

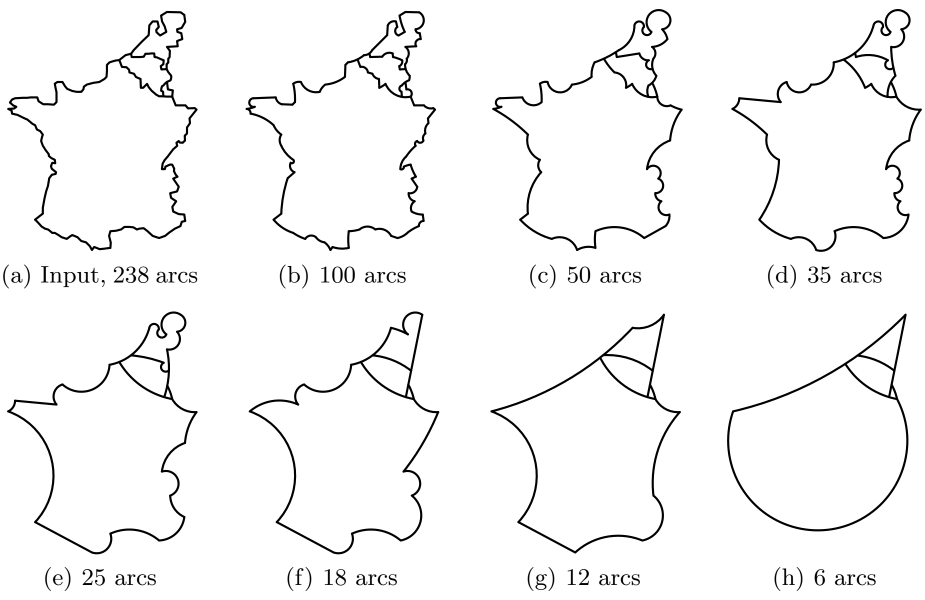
2.3 Analysis

We now analyze the asymptotic execution time of our schematization algorithm. Let n denote the number of vertices in the input and h the number of junctions.

We first determine the time required to compute a single candidate operation. To compute the cost, we compute the Fréchet distance between a single arc and a polygonal line of at most $O(n)$ vertices in $O(n \log n)$ time [24]. For admissibility, we test whether the candidate arc intersects any other arc, requiring $O(n)$ time. If the arcs being replaced contained virtual vertices, then the connected arcs might need to be extended and could start to intersect. There are at most $O(h)$ of these arcs, which are ordered along the replaced arcs. We need to test only intersections between arcs that are adjacent in this order which takes $O(h)$ time. In addition, for up to four arcs (the first and last on either side of the replacement pair), we check whether they intersect any other arc, taking $O(n)$ time. Thus, computing the cost and admissibility of a single operation takes $O(n \log n)$ time.

At initialization of the algorithm we compute all operations. Since each candidate represents only two edges, we compute the Fréchet distance in $O(1)$ time and thus the initialization takes $O(n^2)$ time. Performing an operation may change the geometry or admissibility of other operations. The geometry of at most $O(h)$ edges changes. Hence, we compute at most $O(h)$ new operations in $O(hn \log n)$ total time. Moreover, we remove the old edges from the sets of edges causing inadmissibility and insert the new edges where necessary. This also takes $O(hn \log n)$ time. Thus, the complete algorithm runs in $O(n^2 h \log n)$ time.

Solution tree. A sample of the solutions generated by our algorithm for different complexities is shown in Fig. 4. While the complexity of the map in Fig. 4(b) is less than half of the input complexity, the effect of schematization is not very



(a) Input, 238 arcs (b) 100 arcs (c) 50 arcs (d) 35 arcs

(e) 25 arcs (f) 18 arcs (g) 12 arcs (h) 6 arcs

Fig. 4. Sample of the possible schematizations for a network that represents Belgium, France, Luxembourg and the Netherlands.

noticeable. On the other hand, the map in Fig. 4(h) is highly schematized, but is also geometrically heavily distorted. The results in between make a trade-off between schematization and geometric accuracy. The optimal trade-off for a schematic map depends on its size, content and purpose.

Since the optimal number of arcs may not be clear a priori, it is desirable to be able to interactively explore schematizations with different complexity. To this end, we maintain not only the current schematic network N, but also some additional structure that allows us to efficiently recover other intermediate solutions. Two simple options for this structure are an operation list or an operation tree. Both approaches use $O(n)$ space and take $O(nh)$ time to recover an intermediate solution. We use a tree as the expected runtime is lower for schematizations of low complexity. Original arcs are stored in leaves and parent nodes contain changes made by an operation. With every node the current complexity is stored.

2.4 Extensions for metro maps

We describe four extensions which are geared towards metro-map construction.

Distributing stations. A typical metro network consists of a city center that has many highly-connected stations and some lines that go to suburban areas with fewer stations. Keeping stations near their geographic position causes problems due to the high-density city center. We want to distribute the stations more evenly across the drawing, which increases the scale of the center and decreases the scale of the suburbs: this improves readability [17] and helps the schematization. Distributing the stations necessarily distorts the geography of the network; to retain as much geography as possible, we use minimum-distortion focus maps [7]. We set the desired scale factor of a station v to $1 + c \cdot k_v$, where c is a constant and k_v is the number of stations within a disc of some radius d around v. We chose c and d such that the assigned scale factors range approximately up to 2. Note that this sets all scale factors to at least 1.

Stroke partition revisited. For metro maps, it is desirable for a single line to continue smoothly at an interchange, even if the angular deviation is high in the original geography. Hence, we change the stroke partition to a two-step process. In the first step, we aggregate adjacent metro connections having the same set of metro lines into preliminary strokes. If a metro line has multiple branches, we aggregate based on the smallest angular deviation. In the second step, we aggregate the preliminary strokes using angular deviation as before.

Interchanges. In the final metro map, interchanges are drawn as circles with some given radius r. Hence, the lines at the interchange do not have to intersect in a single unique point: we may admit some leeway depending on r. In particular, we maintain for an interchange a smallest enclosing disk of the intersections of the incident lines. Topological errors within the disk are allowed, since drawing the disk hides them. We constrain this flexibility by maintaining the order of the incident lines around the boundary of the disk. Moreover, we bound the maximum radius of the smallest enclosing disk by r.

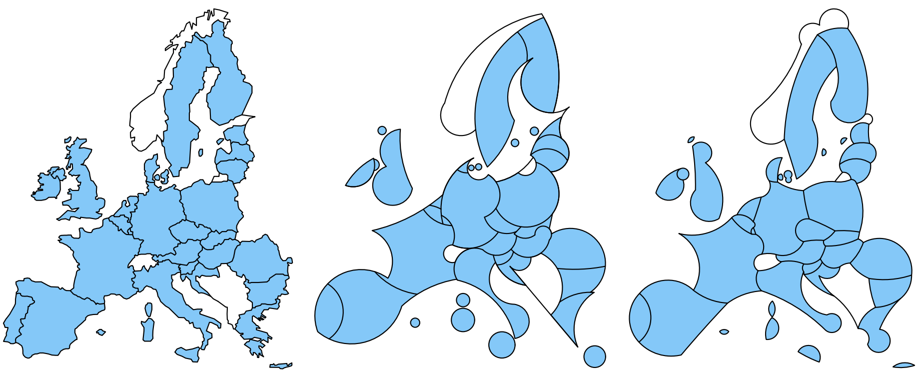
Rendering. To visualize a schematic metro map, we draw the metro lines in appropriate colors. Where multiple metro lines connect two stations, we draw parallel metro lines. The use of circular arcs makes this comparatively simple: we use concentric arcs in such a situation, with slightly varying radius. The ordering of such parallel connections is a problem in itself (see e.g. [21]), one we do not consider here. Though algorithms exist, we manually set the order in our results. Aside from visualizing the metro lines, a metro map should also have vertices for each station. During rendering, we hence reinsert the degree-2 vertices removed during schematization by distributing them evenly along the appropriate edge.

3 Results

We consider three different use cases: territorial outlines, road networks, and metro maps. Our algorithm produces high-quality results for both territorial outlines and metro maps. For road networks, a high complexity reduction can be obtained though the resulting maps do not convey their message well.

Territorial outlines. A territorial outline represents administrative and geographic boundaries. Examples include country and province borders, or shorelines. Territorial outlines are typically low-density networks and contain a low number of junctions. Often, features can be combined through multiple junctions, see, for example, the combined shoreline of France, Belgium and the Netherlands in Fig. 5(a). Such features should be recognized and schematized as a single arc, since otherwise the saliency of junctions incorrectly increases.

Fig. 5(b) shows the result of our algorithm for a map of the European Union. Note that many of the coastline features have been replaced by single arcs. By schematizing across junctions, we reinforce the importance of the geographic features. For comparison, in Fig. 5(c) the result of the circular-arc schematization method by Van Goethem *et al.* [13] is shown. Here, junctions are fixed and no



(a) Geographic map (b) 68 circular arcs (c) 123-arc map by [13]

Fig. 5. Schematizing the borders of the European Union, augmented with shorelines.

(a) Geographic map (b) 54 circular arcs (c) Population density

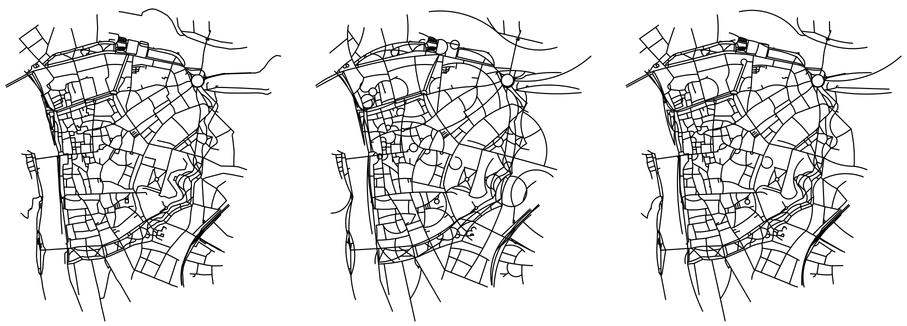
Fig. 6. Schematizing the province boundaries in Netherlands.

operations are performed on edges that meet at such a vertex. As a consequence, the junctions become more important in the final map, which is often undesirable. It also prevents the schematization from reaching the same low complexity, as vertices that are close to each other require small details to be maintained.

In contrast to our method, the algorithm by Van Goethem *et al.* [13] maintains the area of each country. This ensures that countries maintain their relative size, which is not guaranteed in our method (e.g. compare Luxembourg, Denmark and the Netherlands). Also, due to the shifting of junctions, some borders can become very short, such as the boundary between Switzerland and Austria. Though in theory the result is topologically equivalent, drawing the actual geometry with nonzero-width lines causes very small elements to be obscured. We can extend the admissibility of operations to include a check for a minimum distance between affected boundaries. The output then adheres to these minimal distances, assuming the input also does.

Fig. 6 shows a schematization of the provinces of the Netherlands. The ability to schematize across junctions helps to capture both the western coastline feature and the typical shape of the north-east border. Even though this is a highly schematized version of the Netherlands, all boundaries are still geometrically reasonably accurate. As most borders between provinces are represented by a single or a few arcs, the result gives a very stylized impression.

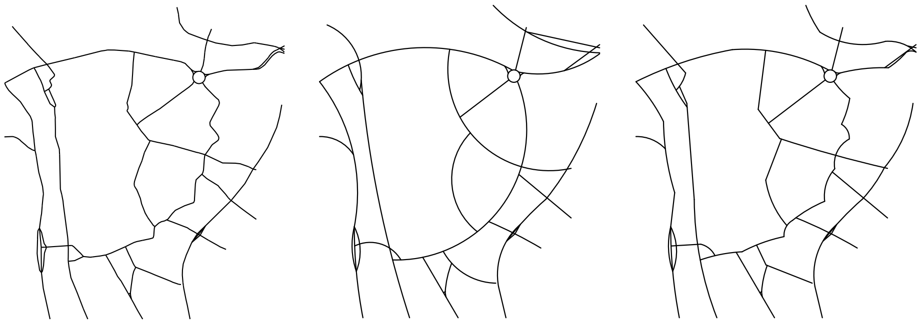
Road networks. Road networks are typically very densely connected with many junctions. In contrast to territorial outlines and metro maps, however, the geometry of a road implicitly correlated to the type of road. A wiggly mountain road that has been "schematized" to a smooth curve could easily be misinterpreted as a highway. To maintain this implicit correlation, a form of caricature is often required when schematizing roads [16]. By exaggerating the features of roads, the association to their respective road types is maintained. Due to our geometric approach, our algorithm is unlikely to produce a road map that has a



(a) Geographic map with 2965 edges

(b) "Schematization" with 540 circular arcs

(c) Without operations on junctions: 1566 arcs



(d) Generalized map with 648 edges

(e) Schematization with 35 circular arcs

(f) Schematization with 60 circular arcs

Fig. 7. Schematizing the road network of Würzburg.

"schematic appearance" in this sense. Regardless, we investigate road networks to evaluate the possible complexity reduction in very dense networks.

The road network around Würzburg, Germany, is shown in Fig. 7(a). Our algorithm is able to reduce the complexity by roughly 82%, from 2965 to 540 circular arcs (Fig. 7(b)). The ability to schematize junctions allows for a significantly higher complexity reduction: without this addition, no admissible operations exist at 1566 edges stopping further progress (Fig. 7(c)).

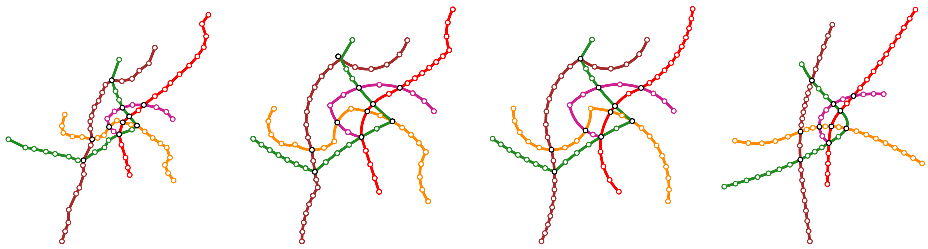
The high complexity of the input map in Fig. 7(a) limits the schematized look attainable even though complexity is greatly reduced. This high density is, however, not the main problem we encounter when schematizing road networks. We also apply our algorithm on a generalized version of the same road network (see Fig. 7(d)). While the schematization is more pronounced in these maps (see Fig. 7(e)), local roads in the input map have been schematized into single, smooth arcs. As a consequence it is impossible to distinguish large through roads from local roads without additional knowledge. Schematizations using a higher complexity may give more reasonable results (see Fig. 7(f)), but they require a lower complexity reduction limiting the effectiveness of the schematization.

Metro maps. The networks of metro maps are usually heavily schematized and have a matching rendering style. As metro maps contain mainly connectivity information, caricature of actual lines is not important. Thus, we do not expect similar recognition problem as with road networks. We computed our results using the extensions described in Section 2.4.

Our results for Vienna are shown in Fig. 8(a)–(c). Distorting the input geometry creates extra space in the center of the map, thus allowing for a better schematization of the network. The circular arcs create a sense of continuity along the metro lines, while giving a very stylized appearance. Fig. 8(d) shows the same network drawn by the algorithm of Fink *et al.* [10]. Their use of Bézier curves leads to a smooth drawing, but has less of a stylized appearance.

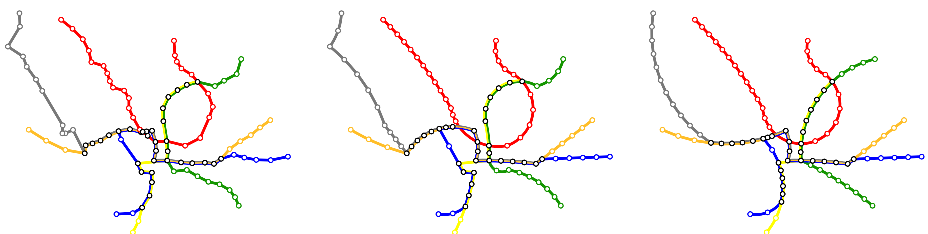
In Washington (Fig. 9) multiple metro lines connect the same stations. The stroke partition merges connections where the same lines run in parallel and determines continuity of lines at the junctions (interchanges). This increases the continuity of metro lines at junctions.

As a final example, we show the metro map of London computed by our algorithm (Fig. 10). Despite its complexity, the algorithm is able to schematize this network using a comparatively low number of arcs. A number of stations are placed very close to one another and overlap when they are drawn as circles. This reduces the legibility of the map. Future work may investigate ways of appropriately dealing with distance constraints between stations and lines.



(a) Geographic map (b) 19 circular arcs (c) 13 circular arcs (d) Vienna by [10]

Fig. 8. Schematizing the metro network of Vienna.



(a) Geographic map (b) 23 circular arcs (c) 14 circular arcs

Fig. 9. Schematizing the metro network of Washington, DC.

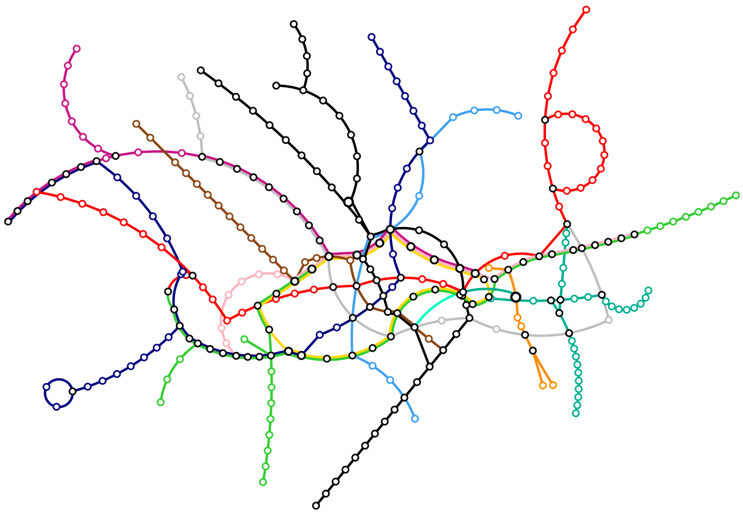


Fig. 10. Schematization of the metro network of London with 70 circular arcs.

Complexity reduction. Our algorithm has different features, but it is unclear how they affect the minimum complexity that can be obtained. So far we mainly focussed on the ability to schematize across junctions. For metro maps we also introduced automated distortion and the ability to treat interchanges as discs instead of points. Here we briefly investigate the effects on the minimum complexity. Table 1 presents a summary. The additions are cumulative from left to right: the last three columns also include operations on junctions; the last two columns also includes distortion. While the lowest complexity is not necessarily desirable, it gives an indication of the ability of our algorithm to obtain abstract low-complexity schematizations. Many of the figures in this paper use a few arcs more than the minimum, since the lowest attainable complexity may cause undesirable deformations (see Fig. 4(h)).

Table 1. Minimum complexity of schematization achievable with our various additions. The last two additions are used only for metro maps.

Map	*Input*	*Basic*	*Junctions*	*Distortion*	*Interchanges as disks*
Europe	1669	105	56		
Netherlands	494	54	42		
Würzburg	2965	1566	540		
Vienna	90	25	12	12	11
Washington	99	22	12	12	12
London	339	128	72	69	67

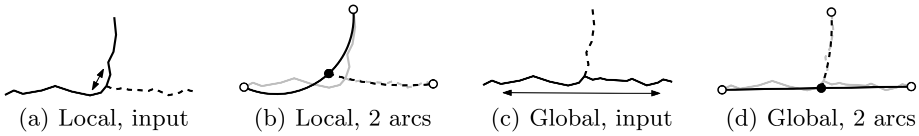
As expected, the ability to schematize across junctions has a large effect on the complexity. In contrast, both the distortion of the input map and the leeway at interchanges, appear to have only a minor effect on the minimum complexity (if any). However, the complexity of the schematization does not capture all aspects. Distorting the input network, for example, greatly enhances the overall spacing, avoiding visual clutter and increasing legibility.

4 Discussion and future work

The experimental results in Section 3 appear satisfactory in most cases, though room for improvement remains. In this section, we review and discuss some of the steps we took to design our algorithm.

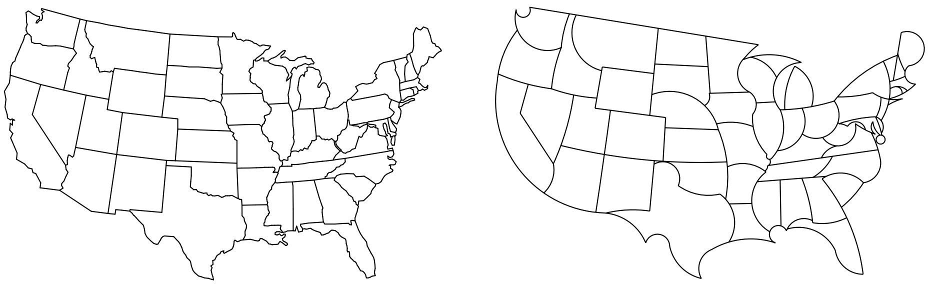
We use a stroke partition to decide how edges are combined across junctions. This use of strokes has both benefits and drawbacks. Advantages of our partitioning method are its efficiency and simplicity. The main drawback is that the criterion is local and this may negatively impact schematization: local continuity need not correspond to continuity from a global perspective (see Fig. 11). Without a stroke partition we could determine continuation on-the-fly by computing the operations for any pair of edges meeting at a junction. The problem is that once continuation across a junction has been decided it is irreversible. As small operations are executed in order of similarity, it is likely that local geometry still prevails. An advantage of an explicit stroke partition is that we may improve on this independently. Moreover, the continuation may carry information in some networks (e.g. metro networks).

Our algorithm allows for easy integration of both hard constraints and soft constraints. For hard constraints, the admissibility of operations should be modified accordingly. We enforced correct topology, but in addition we could also consider requiring some minimal distance between different strokes. Note that, whereas correct topology is initially guaranteed, this need not be the case for such a minimal distance. In such cases, the algorithm cannot guarantee that the constraint holds for the result; it may even prevent any complexity reduction in areas that violate the constraint initially. This would, for example, also occur when requiring strict smooth continuity at each vertex. To include additional soft constraints, the weighting of operations can be modified. For example, weights can be modified based on the resulting circular arc (e.g. [13]) or on the location of the operation (similar to [8]). We must be careful when combining different soft constraints, to avoid an average solution that is worse that either solution



(a) Local, input (b) Local, 2 arcs (c) Global, input (d) Global, 2 arcs

Fig. 11. Different local and global continuity depending on the strokes detected.



(a) Geographic map (b) 103 circular arcs

Fig. 12. Schematization of the USA with 103 circular arcs. The complexity of geographic shapes may affect the perception of schematization.

separately. This may occur, for example, if we were to combine a measure of parallelism to other arcs with geometric similarity: the resulting geometry does not exhibit parallelism nor does it represent the geographic situation well.

Our results show that road networks appear unsuitable for purely geometric schematization. This is in line with previous work, stating that road networks require caricature [16]. That this does not cause similar problems for territorial outlines is likely due to the inherent expectations of such geographic boundaries. Country and province borders are mostly not smooth, low-complexity curves. Therefore, observing a low number of circular arcs is a strong indication that the map is schematized. A smooth road on the other hand is not unusual and may cause associations with different road types. Interestingly enough, this assumption for territorial outlines is not always valid: for example, many of the state boundaries in the US are of very low complexity. This raises the question if a higher level of schematization is needed to attain the same perceived level of schematization (see Fig. 12).

Schematization with circular arcs appears effective in metro maps. Metro maps mainly focus on connectivity and the abstract style of circular arcs fits this purpose well. Maintaining continuity across junctions helps reinforce the structure of the network. The preliminary results from this paper should, however, be further validated in future work. Research into the usability of curved metro maps was recently started [23], but a further study of the effects is required, also for different map types. For metro maps it would also be interesting to see if the property of continuity at junctions can be exploited further. Strokes might be required to always continue smoothly at junctions. This would improve legibility at these positions, be it at the cost of maintaining more degree-2 vertices. The increase in complexity may, however, detract from the visual clarity obtained through extensive schematization. Moreover, as an edge may represent multiple metro lines, it may be infeasible for sufficient lines to continue smoothly at these vertices.

Lastly, we discuss the effect of applying an iterative algorithm. Iteratively selecting the best operation ensures a fast and comparatively simple algorithm. However, it does not guarantee that the obtained result is optimal. Also, it may be rather unstable: a minor change in the input may greatly affect the output. Stability is important for networks that change over time. Though the network changes, the schematic maps should remain similar if possible. Ideally, one would design a stable algorithm that computes an optimal schematic map, e.g. the map with highest similarity given some maximum complexity. However, it is likely that such an algorithm has significantly higher computation time.

5 Conclusion

We presented an algorithm for circular-arc schematization of geographic maps. Our algorithm is able to schematize across junctions (vertices of degree three or higher). Allowing junctions to be shifted makes the schematization highly flexible, which enhances its quality. We did preliminary experiments for three different use cases to test the effectiveness of our algorithm. The results obtained for both territorial outlines and metro maps appear of high quality. Extensive features can be represented with a single arc or just a few and even dense networks can be schematized effectively and efficiently. Though for road networks the complexity can be reduced significantly, the lack of caricature makes the result less effective as a schematization. We also briefly evaluated the potential complexity reduction attainable using the different features of our algorithm. From this evaluation we conclude that the ability to schematize across junctions is the most significant feature that allows for a high complexity reduction.

References

1. R. Brunet. La composition des modèles dans l'analyse spatiale. *Espace géographique*, 9(4):253–265, 1980.
2. K. Buchin, W. Meulemans, and B. Speckmann. A new method for subdivision simplification with applications to urban-area generalization. In *Proc. 19th ACM SIGSPATIAL GIS*, pages 261–270, 2011.
3. S. Cabello, M. De Berg, and M. Van Kreveld. Schematization of networks. *Comp. Geometry: Theory & Appl.*, 30(3):223–238, 2005.
4. S. Cicerone and M. Cermignani. Fast and simple approach for polygon schematization. In *Proc. 12th ICCSA*, LNCS 7333, pages 267–279, 2012.
5. D. De Chiara, V. Del Fatto, R. Laurini, M. Sebillo, and G. Vitiello. A chorem-based approach for visually analyzing spatial data. *J. of Visual Languages & Computing*, 22(3):173–193, 2011.
6. D. Delling, A. Gemsa, M. Nöllenburg, and T. Pajor. Path schematization for route sketches. In *Proc. 12th SWAT*, LNCS 6139, pages 285–296, 2010.
7. T. C. van Dijk and J.-H. Haunert. Interactive focus maps using least-squares optimization. *Int. J. of Geographical Information Science*, 2014. doi: 10.1080/13658816.2014.887718.

8. T. C. van Dijk, A. Van Goethem, J.-H. Haunert, W. Meulemans, and B. Speckmann. Accentuating focus map via partial schematization. In *Proc. 21st ACM SIGSPATIAL GIS*, pages 438–441, 2013.
9. R. Drysdale, G. Rote, and A. Sturm. Approximation of an open polygonal curve with a minimum number of circular arcs and biarcs. *Comp. Geometry: Theory & Appl.*, 41(1-2):31–47, 2008.
10. M. Fink, H. Haverkort, M. Nöllenburg, M. Roberts, J. Schuhmann, and A. Wolff. Drawing metro maps using Bézier curves. In *Proc. 20th Graph Drawing*, LNCS 7704, pages 463–474, 2013.
11. A. Gemsa, M. Nöllenburg, T. Pajor, and I. Rutter. On d-regular schematization of embedded paths. In *Proc. 37th SOFSEM*, LNCS 6543, pages 260–271, 2011.
12. A. van Goethem, W. Meulemans, A. Reimer, H. Haverkort, and B. Speckmann. Topologically safe curved schematisation. *Cartographic J.*, 50(3):276–285, 2013.
13. A. van Goethem, W. Meulemans, B. Speckmann, and J. Wood. Exploring curved schematization. In *Proc. 7th IEEE PacificVis*, pages 1–8, 2014.
14. M. Heimlich and M. Held. Biarc approximation, simplification and smoothing of polygonal curves by means of Voronoi-based tolerance bands. *Int. J. of Comp. Geometry & Appl.*, 18(3):221–250, 2008.
15. K. Koffka. *Principles of Gestalt psychology.* Routledge, 2013.
16. F. Lecordix, C. Plazanet, and J.-P. Lagrange. PlaGe: a platform for research in generalization. application to caricature. *GeoInformatica*, 1(2):161–182, 1996.
17. D. Merrick and J. Gudmundsson. Increasing the readability of graph drawings with centrality-based scaling. In *Proc. 2006 Asia-Pacific Symp. Info. Vis.*, pages 67–76, 2006.
18. D. Merrick and J. Gudmundsson. Path simplification for metro map layout. In *Proc. 14th Graph Drawing*, LNCS 4372, pages 258–269, 2007.
19. G. Neyer. Line simplification with restricted orientations. In *Proc. 6th WADS*, LNCS 1663, pages 13–24, 1999.
20. M. Nöllenburg and A. Wolff. A mixed-integer program for drawing high-quality metro maps. In *Proc. 13th Graph Drawing*, LNCS 3843, pages 321–333, 2006.
21. M. Nöllenburg. An improved algorithm for the metro-line crossing minimization problem. In *Proc. 17th Graph Drawing*, LNCS 5849, pages 381–392, 2010.
22. A. W. Reimer. Understanding chorematic diagrams: Towards a taxonomy. *Cartographic J.*, 47(4):330–350, 2010.
23. M. Roberts, E. Newton, F. Lagattolla, S. Hughes, and M. Hasler. Objective versus subjective measures of Paris metro map usability: Investigating traditional octolinear versus all-curves schematic maps. *Int. J. of Human Computer Studies*, 71:363–386, 2013.
24. G. Rote. Computing the Fréchet distance between piecewise smooth curves. *Comp. Geometry: Theory & Appl.*, 37(3):162–174, 2007.
25. J. Stott, P. Rodgers, J. Martinez-Ovando, and S. Walker. Automatic metro map layout using multicriteria optimization. *IEEE TVCG*, 17(1):101–114, 2011.
26. R. Thomson and R. Brooks. Exploiting perceptual grouping for map analysis, understanding and generalization: The case of road and river networks. In *Graphics Recognition: Algorithms & Appl.*, LNCS 2390, pages 148–157, 2002.
27. P. Ti and Z. Li. Generation of schematic network maps with automated detection and enlargement of congested areas. *Int. J. of Geographical Information Science*, 28(3):521–540, 2014.