

# Generalizing Geometric Graphs<sup>\*</sup>

Edith Brunel, Andreas Gemsa, Marcus Krug, Ignaz Rutter, and Dorothea Wagner

Faculty of Informatics, Karlsruhe Institute of Technology (KIT), Germany  
firstname.lastname@kit.edu

**Abstract.** Network visualization is essential for understanding the data obtained from huge real-world networks such as flight-networks, the AS-network or social networks. Although we can compute layouts for these networks reasonably fast, even the most recent display media are not capable of displaying these layouts in an adequate way. Moreover, the human viewer may be overwhelmed by the displayed level of detail. The increasing amount of data therefore requires techniques aiming at a sensible reduction of the visual complexity of huge layouts.

We consider the problem of computing a generalization of a given layout reducing the complexity of the drawing to an amount that can be displayed without clutter and handled by a human viewer. We take a first step at formulating graph generalization within a mathematical model and we consider the resulting problems from an algorithmic point of view. Although these problems are NP-hard in general, we provide efficient approximation algorithms as well as efficient and effective heuristics. At the end of the paper we showcase some sample generalizations.

## 1 Introduction

As a natural consequence of the increasing amount of available data we are facing large and even huge networks such as road and flight networks, the AS-network and social networks with millions of vertices. Visualization of these networks is a key to assessing the inherent graph-based information. There are several methods for computing layouts of huge graphs with millions of vertices within a few minutes [21,23,19].

But, how do we display such layouts? Modern HD displays feature roughly 2 Mio pixels and a standard A4 page allows roughly 8.7 Mio dots at a resolution of 300 pixels per inch. Even if we require only a minimal distance of 10 pixels or dots between the vertices of the graph, then we can display only several thousand vertices, and not too many edges. If we additionally seek to display graph structure and keep visual clutter low, the number of vertices we can display degrades even further.

Even worse, the human perception is not capable of extracting detailed information from huge layouts with millions of vertices. Since, by a simple counting argument, there are incompressible adjacency matrices, a graph with only 1 Mio vertices may encode incompressible information of up to 125 Gigabytes. This exceeds by a factor of 3.6 the average daily information consumption of an American, estimated at 34 (highly compressible) Gigabytes [4]. It is thus apparent that even an aesthetically pleasing layout of a huge graph may not be suited for displaying information to a human viewer.

---

<sup>\*</sup> Research was partially supported by EUROGIGA project GraDR 10-EuroGIGA-OP-003.

*Related Work.* Known approaches to coping with the huge amount of data by allowing for some kind of abstraction can be categorized into *structural* and *geometric* methods. While structural methods create a new layout for the data typically using a clustering of the graph, geometric methods are applied to a given layout maintaining the user's mental map [26].

Eades and Feng [13] describe a multilevel visualization method for clustered graphs. A force-directed layout algorithm based on a hierarchical decomposition of the graph is given by Quigley and Eades [28]. This method allows for visualizing the graph at different levels of abstraction. Abello et al. [2] discuss graph sketches for very large graphs based on mapping clusters of the graph to certain regions of the screen.

Fisheye visualizations [15,30], on the other hand, apply a distortion to a given layout to emphasize the layout in the area of interest. The resolution of the drawing deteriorates towards the boundary of the drawing and parts of the drawing in this area are usually densely cluttered. Abello et al. [1] study the visualization of large graphs with compound-fisheye views and treemaps, employing hierarchical clustering and a treemap representation of this clustering. Edge Bundling techniques [31,22] aim at reducing the complexity of layouts by bundling similar edges.

*Generalization* has received considerable attention in cartography [25]. Mackaness and Bear [24] highlight the potential of graph theory for map generalization. Saalfeld states the map generalization problem as a straight-line graph drawing problem [29] and formulates a number of challenges resulting from this perspective. Among others, he asks for a rigorous mathematical model for graph-based generalizations and provable guarantees. We are not aware of any work aiming at assessing this problem in general.

*Our Contribution.* We take a first step towards establishing a mathematical model for the problem of generalizing geometric graphs. Our model is based on the fact that vertices have a fixed size and edges have a fixed width on the screen. *Visual clutter* refers to an agglomeration of overlapping visual features in a limited area resulting in indistinguishable features. Our goal is to either avoid or reduce visual clutter. We identify three types of clutter. *Vertex-Clutter* occurs when two or more vertices are too close to each other. It may render the drawing unusable due to hidden edge information; see Fig. 1. *Edge-Clutter* occurs when too many edges cross a limited area. Even if vertices are far enough apart, edge clutter may lead to indistinguishable edge information; see Fig. 2. *Vertex-Edge-Clutter* occurs when a vertex is too close to an edge. In this case, we are unable to tell, whether the vertex is incident to the edge or not; see Fig. 3.

We devise a framework that allows for assessing all types of clutter in an incremental way by modeling the elimination or reduction of each type of clutter as an optimization problem, which we analyze in terms of complexity. We show that these problems are NP-hard in general and we provide approximation algorithms as well as effective and efficient heuristics that can be applied to huge graphs within reasonable time.

*Preliminaries.* A *geometric graph* is a pair  $G = (P, E)$  such that  $P \subseteq \mathbb{R}^2$  is a finite set of  $n$  points in the plane and  $E$  is a set of  $m$  straight-line segments with endpoints in  $P$ . If not otherwise stated, graph refers to a geometric graph throughout this paper. For  $p \in P$  and a non-negative number  $r \in \mathbb{R}_0^+$ , we denote by  $B(p, r)$  the disk with center  $p$  and radius  $r$ . We model the finite resolution of a screen by assuming that each point  $p$  occupies the



Fig. 1. Vertex-Clutter

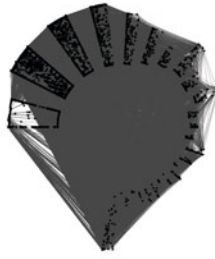


Fig. 2. Edge-Clutter

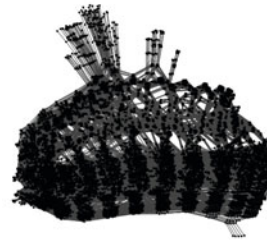


Fig. 3. Vertex-Edge-Clutter

locus of points whose distance to  $p$  is bounded by  $s \in \mathbb{R}_0^+$  and, similarly, each edge  $e$  occupies the locus of points whose distance to  $e$  is bounded by  $w \in \mathbb{R}_0^+$ .

A *generalization* of  $G$  is a pair  $(H, \varphi)$  where  $H = (Q, F)$  is a geometric graph with  $Q \subseteq P$  such that  $\varphi: P \rightarrow Q$  maps vertices of  $G$  to vertices of  $H$  and  $F$  is a subset of edges resulting from a contraction of  $G$  according to  $\varphi$ . Since the subgraph induced by  $\varphi^{-1}(q)$  is contracted into a single vertex, we call this subgraph the *cluster* of  $q$ , denoted by  $C_q$ . Given  $Q \subseteq P$ , we denote by  $v: P \rightarrow Q$  the *Voronoi mapping* which maps  $p \in P$  to its closest neighbor in  $Q$  with respect to the Euclidean metric. We call the corresponding clusters *Voronoi clusters*. Throughout the paper distance refers to the Euclidean metric.

*Organization of the Paper.* In Section 2 we consider the problem of eliminating vertex-clutter. We discuss our model for the generalization of the vertex set and show NP-hardness of the corresponding optimization problem. We further show that the size of the generalized pointset can be approximated and we devise an efficient heuristic for further optimization. In Section 3 we study the reduction of edge-clutter. We show that it is in general NP-hard to find a sparse or short subset of the edges maintaining monotone tendencies. When the original graph is complete, however, or if we are not restricted to use edges of the original graph, we can efficiently compute a sparse graph approximately representing monotone tendencies of the edges. In Section 4 we model the problem of reducing vertex-edge clutter and we show how to compute a drawing that allows for unambiguously deciding whether an edge is incident to a vertex or not, thus effectively eliminating vertex-edge clutter. We showcase some sample generalizations and conclude with a short discussion as well as open problems in Section 5. We omit some details due to space constraints; for full proofs and additional sample generalizations we refer the reader to the long version of this paper [6].

## 2 Generalizing the Vertex Set without Vertex-Clutter

In this section we consider the problem of computing a generalization  $(H, \varphi)$  without vertex clutter for a geometric graph  $G = (V, E)$ , where  $H = (Q, F)$ . We focus on the case that  $\varphi$  is the Voronoi-mapping assigning each vertex in  $P$  to its nearest neighbor in  $Q$ . In order to avoid vertex-clutter we require a minimal distance  $r \in \mathbb{R}_0^+$  between the vertices of a generalized geometric graph. Hence for each vertex  $p \in Q$  in the generalized graph the disk  $B(p, \rho(p))$  with  $\rho: P \rightarrow \mathbb{R}_0^+$  and  $\rho(p) \geq r$  may not contain any other point from  $Q$ . We call a pointset  $Q$  with this property a  $\rho$ -set of  $P$ . This prerequisite, however,

must be balanced with additional quality measures such as the size of the  $\rho$ -set, the clustering induced by  $\varphi$  and the distribution of the points in  $Q$  in order to avoid trivial solutions such as a single vertex.

Choosing  $\rho \equiv r$  uniformly for all points  $p \in P$  may have a severe effect on the distribution of the points when maximizing the size of a  $\rho$ -set, since the distances to the nearest neighbors in an inclusion-maximal  $\rho$ -set tend to be uniformly distributed regardless of the original distribution. However, it may be more appropriate to approximate the distribution of the original pointset. In order to approximate this distribution by an inclusion-maximal  $\rho$ -set we can choose  $\rho$  as follows. Let  $p_0$  be the point that maximizes the number of points in  $B(p, r) \cap P$  over all  $p \in P$  and let  $k = |B(p_0, r) \cap P| - 1$ . For each  $p \in P$  let  $d_k(p) \geq r$  denote  $p$ 's distance to its  $k$ -nearest neighbor in  $P$ . By choosing  $\rho(p) = d_k(p) \geq r$  any inclusion-maximal pointset will have approximately the same distribution as the original pointset since for each point in the generalized pointset we discarded the same amount of points from the original graph.

Since, in general, it is not clear which behavior is more appropriate, we introduce a parameter  $\alpha \in [0, 1]$  and let the user decide by setting  $\rho(p) := \max\{r, \alpha d_k(p)\}$ . That is, the user can choose between retaining as many points in areas with low clutter as possible ( $\alpha = 0$ ) and approximating the distribution of the pointset ( $\alpha = 1$ ) as well as interpolations between the two extremes.

We consider two measures to assess the quality of a  $\rho$ -set  $Q$ . While the size of  $Q$  is a measure of the amount of data that is retained, the quality of the clustering induced by  $\varphi$  is a measure for the amount of data that is lost due to the contraction of the vertices. There are several established ways of assessing the quality of clusterings, such as coverage, performance, conductance [16], and modularity [5]. We consider a measure similar to coverage, which we adapt to our purpose as follows. For each cluster  $C_q$  let  $n_q$  denote the number of vertices and  $m_q$  denote the number of edges in  $C_q$ , respectively. We define the *local coverage* of a cluster  $C_q$  by  $\text{lcov}(C_q) = 2m_q / (n_q(n_q - 1))$ , i.e., as the amount of intra-cluster coherence that is explained by the intra-cluster edges. The local coverage of the generalization is defined as  $\text{lcov}(H, \varphi) = \min_{q \in Q} \text{lcov}(\varphi^{-1}(q))$ .

We consider the following multi-objective optimization problem. Given a geometric graph  $G = (P, E)$ , a non-negative radius  $r \in \mathbb{R}_0^+$  and  $\alpha \in [0, 1]$  the LOCAL COVERAGE CLUSTER PACKING (LCCP) problem is to compute a  $\rho$ -set  $Q \subseteq P$  and a mapping  $\varphi: P \rightarrow Q$  that maximizes both  $|Q|$  and  $\text{lcov}(H, \varphi)$ .

## 2.1 Complexity

The problem of computing a  $\rho$ -set of maximum size for  $\alpha = 0$  can be reduced to the problem of computing a maximum independent set in the intersection graph of the disks with radius  $r/2$  centered at the points in  $P$ . Clark et al. [9] prove that this problem is NP-hard in unit-disk graphs, even if the disk representation of the graph is given.

**Corollary 1.** *Maximizing the size of a  $\rho$ -set is NP-hard for  $\alpha = 0$ .*

Next, we show that it is also NP-hard to maximize the local coverage in the induced clusters of a  $\rho$ -set as well as the total size of the generalization obtained by choosing a  $\rho$ -set if the clustering is obtained by the Voronoi mapping induced by the points in  $Q$ .

**Theorem 1.** *Maximizing  $\text{lconv}(H, \nu)$  of a generalization  $(H, \nu)$  is NP-hard for  $\alpha = 0$ .*

The proof is by reduction from the NP-hard problem PLANAR MONOTONE 3-SAT [11]. Given an instance of this problem we construct a geometric graph  $G$  composed of gadgets acting as variables, literals and clauses, respectively, such that for constant  $\rho$   $G$  contains a  $\rho$ -set with local coverage 1 if and only if the corresponding planar monotone 3-sat-formula is satisfiable. A full proof can be found in [6].

## 2.2 Approximating the Maximum Size of a Generalization

Although it is unlikely that we can efficiently compute a  $\rho$ -set with maximum size, we show that we can approximate the size of a maximum  $\rho$ -set.

**Theorem 2.** *Let  $G$  be a geometric graph and let  $r \in \mathbb{R}_0^+$  and  $\alpha \in [0, 1]$  be given. In  $O(kn + n \log^5 n (\log \log n)^2)$  time we can compute a generalization  $\mathcal{H}$  of  $G$  that approximates the maximum number of vertices of a generalization by a factor of  $(7k + 2)/3$ , where  $k = \max_{p \in P} |B(p, \rho(p)) \cap P| - 1$ .*

In order to prove Theorem 2 we use the following auxiliary lemma, whose proof can be found in [6].

**Lemma 1.** *Let  $p_0$  be a point in the plane and let  $k \in \mathbb{N}$ . Then there are at most  $6k$  points  $Q$  such that  $p_0$  is among the  $k$  closest points for each of the points  $q \in Q$ .*

*Proof (Proof of Theorem 2).* Let  $H$  be the graph on the set of points such that  $pq$  is a (directed) edge if and only if  $q \in B(p, \rho(p))$ . The graph  $H$  contains an independent set of size  $s$  if and only if  $G$  contains a  $\rho$ -set of this size. Each independent set in  $H$  corresponds to a  $\rho$ -set in  $G$  since each point in  $H$  is connected to all points that are closer than  $\rho(p)$  and it is connected to all points  $q$  such that  $p$  is in the  $\rho(q)$  disk around  $q$ . On the other hand, each  $\rho$ -set in  $G$  induces an independent set due to this construction.

By choice of  $\rho$ , each vertex has out-degree bounded by  $k = \max_{p \in P} |B(p, r) \cap P| - 1$  for any value of  $\alpha$ . There is an ingoing edge from  $q$  into  $p$  if and only if  $p$  is among the  $k$  closest neighbors of  $q$ . By Lemma 1 there are at most  $6k$  points such that  $p$  is among the closest  $k$  points for each of these points. Hence, the in-degree of each vertex is bounded by  $6k$ . In total, each vertex has degree at most  $7k$ . Hence, by a result due to Halldórsson and Radhakrishnan [20] we can approximate the maximum size of an independent set by a factor of  $(7k + 2)/3$ . The algorithm greedily chooses the minimum degree vertex in each step and can be implemented to run in time  $O(kn)$ , given the graph  $H$ .

In order to compute  $H$  we locate the points in a closed disk by a circular range query in  $O(\log n + k)$  time using  $O(n \log^5 n (\log \log n)^2)$  preprocessing time [7]. Hence, the total running time is  $O(kn + n \log^5 n (\log \log n)^2)$ .  $\square$

Based on this approximation, we heuristically compute a  $\rho$ -set  $Q$  balancing both the size of  $Q$  and the local coverage of the Voronoi clustering induced by  $Q$ . For  $p \in P$  let  $\tilde{m}(p)$  be the number of edges whose endpoints are both contained in  $B(p, \rho(p)/2)$  and let  $\tilde{n}(p)$  be the number of points in  $B(p, \rho(p))$ . We show the following.

**Lemma 2.** *Let  $Q$  be an inclusion-maximal  $\rho$ -set and let  $\alpha = 0$ . Further, let  $H = (Q, F)$  be the generalization obtained from  $G = (P, E)$  by the Voronoi-mapping  $\nu$ . Then the value  $\min_{q \in Q} 2\tilde{m}(q)/(\tilde{n}(q)(\tilde{n}(q) - 1))$  is a lower bound for  $\text{lconv}(H, \nu)$ .*

*Proof.* For  $\alpha = 0$  we have  $\rho \equiv r$ . Whenever  $p$  is chosen as a cluster center in  $Q$ , the points in  $B(p, r/2)$  are closer to  $p$  than to any other point in  $Q$ , since the closest point to  $p$  in  $Q$  has distance to  $p$  at least  $r$ . Hence, the edges in  $B(p, r/2)$  are intra-cluster edges of  $C_p$ . On the other hand, the number of points in each of the clusters is bounded by  $\tilde{n}(p)$  whenever  $\alpha = 0$  and  $Q$  is an inclusion-maximal  $\rho$ -set. To see this, consider any vertex  $q$  that is not contained in  $B(p, r)$ , but closer to  $p$  than to any other cluster center. Then  $q$  is contained in none of the disks centered in the cluster centers and, thus,  $q$  must be a cluster center itself, since  $Q$  is inclusion-maximal. Hence, the claim holds.  $\square$

Based on Lemma 2 we propose a heuristic, called GREEDY WEIGHT HEURISTIC, that operates as follows. First we compute an estimate of  $2\tilde{m}(q)/(\tilde{n}(q)(\tilde{n}(q) - 1))$  for each  $p \in P$ . Subsequently we sort the points according to these estimates in  $O(n \log n)$  time and iteratively consider the points in this order. If the current vertex is not covered by a previous vertex, then it is chosen for the  $\rho$ -set, otherwise it is discarded.

Instead of computing  $\tilde{m}(p)$  and  $\tilde{n}(p)$  exactly, we estimate these numbers by counting the number of vertices and edges in the bounding boxes of the disks  $B(p, \rho(p)/2)$ . To count the number of edges we use a 4-dimensional range searching query on a data structure containing tuples of points corresponding to edges in  $E$  with query time  $O(\log^3 m)$  [8]. We use the 2-dimensional counterpart to locate points. Further, we use a data structure for dynamic nearest neighbor queries with  $O(\log^2 n)$  query time [3], into which we insert the selected points to decide whether the current point is covered by a previously selected point. The total running time is  $O((n + m) \log^3 m + n \log^2 n)$ .

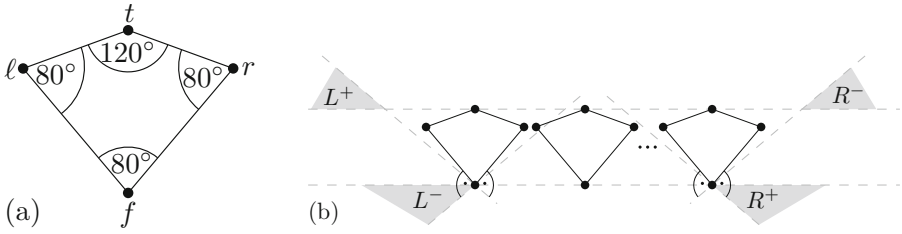
### 3 Minimizing Edge-Clutter

In order to reduce the clutter resulting from an excess of edges in certain areas we must filter out some of the edges without destroying the visual appearance of the graph. The total length of the edges seems to be a good measure for the clutteredness of the graph since it is proportional to the ink used for the drawing. While a minimum spanning tree will minimize this quantity, it is unlikely to preserve the visual appearance of the graph. We therefore require that monotone tendencies of the edges are preserved in order to best maintain the mental map of the adjacencies between vertices of the graph.

Let  $\ell$  be a line in the plane and let  $S = (p_1, \dots, p_k)$  be a sequence of points. We say that  $S$  is  $\ell$ -monotone if the order of the orthogonal projections of  $p_1, \dots, p_k$  onto  $\ell$  is the same as the order of the points in  $S$ . Let  $G = (P, E)$  be a geometric graph and let  $(H, \varphi)$  be a generalization of  $G$  such that  $H = (P, F)$ , i.e.,  $F \subseteq E$ . We say that  $H$  is a *monotone generalization of  $G$*  if for every edge  $e \in E$  with endpoints  $p$  and  $q$  there is a  $p$ - $q$ -path  $\pi_e$  in  $H$  such that  $\pi_e$  is  $\ell_e$ -monotone, where  $\ell_e$  is the line defined by the endpoints of  $e$ . Given  $G = (P, E)$  the SHORTEST GEODESIC SUBGRAPH (SGS) problem asks for a monotone generalization  $H$  of  $G$  minimizing the total length of  $H$ .

**Theorem 3.** SHORTEST GEODESIC SUBGRAPH is NP-hard.

*Proof.* We reduce from monotone 3-SAT, an NP-complete variant of 3-SAT where each clause contains either only positive or only negative literals [17]. Let  $\varphi$  be an instance of monotone 3-SAT with variables  $x_1, \dots, x_n$  and clauses  $C_1, \dots, C_m$ . We construct the



**Fig. 4.** Overview of the reduction from 3-SAT to SHORTEST GEODESIC SUBGRAPH. A kite with foot point  $f$ , top point  $t$  and left and right points  $r$  and  $\ell$  (a), and the arrangement of the kites in the reduction with the corresponding regions for clause vertices (b).

following instance  $G_\varphi$  of SHORTEST GEODESIC SUBGRAPH. For each variable  $x$  we create a kite as shown in Figure 4. Note that the angles at  $f$ ,  $\ell$  and  $r$  are strictly less than  $90^\circ$ , and the angle at  $t$  is strictly greater than  $90^\circ$ . The two edges incident to the top vertex  $t$  are the *top edges*, the edges  $f\ell$  and  $fr$  are the left and right *side edges*, respectively. We place the kites so that their foot points  $f$  lie evenly spaced on the  $x$ -axis and the kites are disjoint. The region  $R^+$  (resp.  $L^+$ ) is the region below the  $x$ -axis (resp. above the horizontal line defined by the top points of the kites) and to the right (resp. left) of the line through the bottom point of the rightmost (resp. leftmost) kite that is perpendicular to the right sides of the kites. We define  $R^-$  and  $L^-$  analogously, but with lines orthogonal to the left sides of the kites.

It follows immediately from the construction that a path that is monotone in the direction from a point in  $R^+$  to a point in  $L^+$  may not contain any right edge of a kite as this would mean a turn of more than  $90^\circ$ , which is not monotone. Analogously, monotone paths from  $L^-$  to  $R^-$  may not contain left edges of kites. In our reduction the kites will act as variables, and edges from  $R^+$  to  $L^+$  (from  $L^-$  to  $R^-$ ) will act as clauses with only positive (only negative) literals.

For each clause  $C_i$  consisting of only positive literals, we add a *clause vertex*  $c_i^1$  into  $R^+$  and a clause vertex  $c_i^2$  in  $L^+$ . We add *connector edges* that connect  $c_i^1$  to the foot points of all kites that correspond to variables that occur in  $C_i$  and that connect  $c_i^2$  to all the left points of kites that correspond to variables that occur in  $C_i$ . Finally, we add the *clause edge*  $c_i^1c_i^2$ . We treat the clauses consisting of only negative literals analogously, except that we place the new vertices in  $L^-$  and  $R^-$ , respectively, and we connect the new vertices in  $R^-$  to the right kite points instead to the left.

This completes our construction, and we claim that an optimal solution of this instance allows us to decide whether the initial formula  $\varphi$  was satisfiable. We will make this more precise in the following. A subset of edges of  $G_\varphi$  is called *tight* if it contains both top edges of each kite, all connector edges, and exactly one of the two side edges of each kite. The proof relies on two claims; full proofs are in [6].

*Claim.* Any feasible solution contains a tight edge set.

*Claim.* There exists a tight set that is feasible if and only if  $\varphi$  is satisfiable.

Note that the total length  $L$  is the same for all tight edge sets. The first claim shows that any geodesic subgraph has length at least  $L$ . And thus, the second claim implies that  $\varphi$

is satisfiable if and only if  $G_\varphi$  admits a geodesic subgraph of length at most  $L$ . Since the construction can easily be performed in polynomial time this concludes the proof.  $\square$

As we have seen, the restriction to edges from the input graph makes it difficult to construct short monotone subgraphs. One possibility is thus to drop this constraint and to allow arbitrary edges. Additionally, we would like to control the distance of the monotone path  $\pi_e$  and the edge it is approximating in terms of monotonicity. This is motivated by the observation that the shortest monotone generalization of a clique whose vertices are arranged equidistantly on a circle is the convex hull of the pointset. Given a line segment  $s$  with length  $\ell_s$  and a point  $p$  with distance  $d_p$  from  $s$  we call the ratio  $d_p/\ell_s$  the *drift of  $p$  from  $s$* . The *drift of a path  $\pi_e$*  with endpoints  $pq$  is defined as the maximum drift of any point on  $\pi_e$  from the segment  $pq$ . Given a geometric graph  $G = (P, E)$  and a non-negative real number  $\delta \in \mathbb{R}_0^+$  the SPARSE GEODESIC NETWORK (SGN) problem asks for a geometric graph  $H = (P, F)$  with minimum total length such that for each edge  $e$  in  $E$  there is an  $\ell_e$ -monotone path  $\pi_e$  with drift at most  $\delta$ , where  $\ell_e$  denotes the line defined by the endpoints of  $e$ . We show the following.

**Lemma 3.** *Given a (complete) geometric graph  $G = (P, E)$ , the Delaunay graph  $\mathcal{D}(P)$  contains for each edge  $e \in E$  an  $\ell_e$ -monotone path  $\pi_e$  with drift at most  $1/2$ .*

*Proof.* Let  $P$  be a set of points and let  $p, q \in P$ . Without loss of generality we assume that  $p$  and  $q$  are on the  $x$ -axis such that  $x(p) < x(q)$ . According to Dobkin et al. [12] we can construct an  $x$ -monotone path in the Delaunay graph  $\mathcal{D}(P)$  of  $P$  as follows. Let  $\mathcal{V}(P)$  denote the Voronoi diagram of  $P$  and let  $p_1, \dots, p_k$  be the ordered points corresponding to the Voronoi cells that are traversed when following the line from  $p$  to  $q$ . Then the path  $p, p_1, \dots, p_k, q$  is an  $x$ -monotone path in the Delaunay graph. Further, all points  $p_i$  are contained within the disk with radius  $d(p, q)/2$  centered in the midpoint of the segment  $pq$ . Hence, the drift is at most  $1/2$ .  $\square$

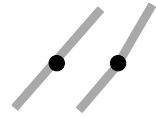
Although the Delaunay graph seems to be well suited to represent monotone tendencies, this result also shows the limitations of allowing arbitrary edges. In the following we therefore focus on subgraphs of the original graph and describe a greedy heuristic for computing a monotone generalization with bounded drift  $\delta$  and short total length, which we call MONOTONE DRIFT HEURISTIC. Given a geometric graph  $G = (P, E)$  and a maximal drift  $\delta$  we sort the edges of  $G$  with respect to increasing length in  $O(m \log m)$  time. Then we consider the edges  $e_1, \dots, e_m$  in this order and iteratively construct a sequence of graphs  $H_0, \dots, H_m$ , where  $H_0 = (P, \emptyset)$ . We insert the edge  $e_i$  into  $H_{i-1}$  whenever there is no  $\ell_{e_i}$ -monotone path with drift at most  $\delta$  in  $H_{i-1}$ . This can be tested by performing a modified depth-first search exploring only monotone subpaths in  $O(n + m)$  time. Hence, the total running time of this approach is  $O(nm + m^2)$ .

## 4 Vertex-Edge-Clutter

Vertex-edge-clutter is the most complicated type of clutter since it involves both vertices and edges and the selection of these features cannot be handled independently as in the previous sections. On the other hand, this type of clutter may be considered



as the least annoying type of clutter. While vertex-edge clutter is caused by edges that are close to a vertex resulting in the difficulty to determine correct incidences, the human perception is rather good at determining whether a line passes a disk through the center or not. For instance, it is easy to see that the leftmost line in Fig. 5 is not incident to the vertex although it crosses the vertex. Additionally, the human perception is also good at determining whether a line has a bend or not, which is illustrated in Fig. 5.



**Fig. 5.** Line perception

Hence, as long as there is neither vertex-clutter nor edge-clutter and as long as no pair of edges incident to a common vertex form a  $180^\circ$ -angle, we will be able to unambiguously tell whether an edge is incident to a vertex or not. In order to attack vertex-edge clutter we therefore propose the following optimization problem. For a pair of edges incident to a common vertex  $p$  we define the *angular straight-line deviation* as the smaller of the two angles that is enclosed by the lines defined by the two edges, respectively. The angular straight-line deviation of  $p$  is then defined as the minimum angular straight-line deviation over all pairs of edges incident to  $p$  and the angular straight-line deviation of a geometric graph  $G$  is the minimum angular straight-line deviation over all vertices of  $G$ . Note, that the angular straight-line deviation is maximized if all angles are close to a right angle. Given a geometric graph  $G = (P, E)$  and a non-negative value  $r \in \mathbb{R}^+$ , the OPTIMAL ANGLE ADJUSTMENT problem is to find a new position for each vertex  $p$  inside  $B(p, r)$  minimizing the angular straight-line deviation of the resulting geometric graph.

We tackle this problem by maximizing the vertices' distances from the lines defined by the edges incident to their neighbors. Let  $G = (P, E)$  be a geometric graph and let  $v \in P$  be a vertex. Let  $N(v)$  denote its neighbors in  $G$ . Further, let  $E(v)$  denote the edges incident to  $v$  and let  $F(v)$  denote the set of edges incident to the vertices in  $N(v)$  but not to  $v$ . By moving  $v$  we change the angles formed by pairs of edges in  $E(v)$  as well as the angles formed by pairs of edges  $(e, f)$  such that  $e \in E$  and  $f \in F$ , respectively. Let  $L_F(v)$  be the set of lines defined by the edges in  $F(v)$  and let  $L_E(v)$  be the set of lines defined by all pairs of vertices in  $N(v)$ . Note, that there will be an angle of 180 degrees involving an edge incident to  $v$  if and only if  $v$  is placed on one of the lines in  $L_E(v) \cup L_F(v)$ . Given  $p \in \mathbb{R}^2$  we denote by  $\mu_v(p)$  the minimum distance of  $p$  to the lines in  $L_E(v) \cup L_F(v)$ . We prove the following.

**Theorem 4.** *Given a graph  $G = (P, E)$ , a vertex  $v \in P$  and a positive radius  $r \in \mathbb{R}^+$  we can compute a new position  $p^*$  for  $v$  in  $B(v, r)$  such that  $\mu_v(p^*) > 0$  and such that  $p^*$  maximizes  $\mu_v(p)$  over all  $p \in B(v, r)$  in  $O(t^3 \alpha(t))$  time where  $t = \min\{\Delta^2, m\}$ ,  $\Delta$  denotes the maximum degree of  $G$  and  $\alpha(\cdot)$  denotes the inverse Ackermann function.*

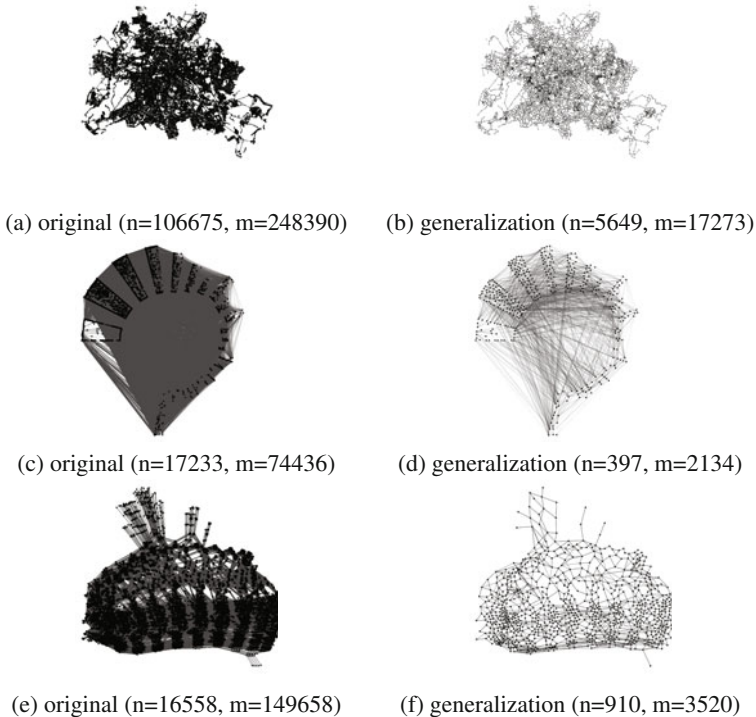
*Proof.* First, we compute the set of edges  $L_F(v)'$  incident to  $v$ 's neighbors, but not to  $v$ , that intersect  $B(v, r)$  as well as the set of lines  $L_E(v)'$  defined by all pairs of  $v$ 's neighbors intersecting  $B(v, r)$ . Let  $L = L_E(v)' \cup L_F(v)'$ . We compute the arrangement of lines in  $L$  in  $O(|L|^2)$  time. Over each of the resulting faces  $C$  we compute the lower envelope of the hyperplanes defining the distance to the boundaries of the faces and project the graph  $\mathcal{G}_C$  defined by the resulting 3-dimensional polytope onto the plane.

The lower envelope of a set of  $n$  hyperplanes can be computed in  $O(n^2 \alpha(n))$  time where  $\alpha(\cdot)$  denotes the inverse of the Ackermann function [14]. Hence the lower

envelopes can be computed in time  $O(|L|^2\alpha(|L|))$  for each face, resulting in a total complexity of  $O(|L|^3\alpha(|L|))$ . For each face  $C$  we inspect the vertices of  $\mathcal{G}_C$  in  $B(v, r)$  as well as its intersection with  $B(v, c)$  and thus compute the point  $p^*$  maximizing  $\mu_v$  in  $B(v, r)$ . Since  $L$  is bounded by  $\max\{\Delta^2, m\}$  we obtain the claimed time complexity. Further, since  $r > 0$  and therefore  $B(v, r)$  is non-degenerate, there must be a non-degenerate face in the arrangement containing a point  $p^*$  in its interior such that  $\mu(p^*) > 0$ .  $\square$

Using Theorem 4 we can incrementally compute a new position for each vertex  $v$  such that none of the edges incident to  $v$  encloses an angle of 180 degrees with any other edge. Since the angles between pairs of edges that are not incident to  $v$  are not affected by this operation, we can iteratively apply Theorem 4 to the vertices one after another to obtain a drawing with strictly positive angular straight-line deviation. At the same time this approach heuristically maximizes this deviation.

Note that we may assume that we apply the angle adjustment to a generalized graph whose complexity tends to be significantly lower than the complexity of the original graph, i.e., both  $m$  and  $\Delta$  should be considerably smaller.



**Fig. 6.** Sample Generalizations computed by implementations of GREEDY WEIGHT HEURISTIC and MONOTONE DRIFT HEURISTIC: (a)–(b) OSM Streetmap Data of Berlin [27], (c)–(d) LunarVis Layout of the AS-Graph [18], (e)–(f) Netlib PDS10 problem graph from the University of Florida sparse matrix collection [10].

## 5 Conclusion and Open Problems

We have undertaken a first step at studying the problem of generalizing geometric graphs within a rigorous mathematical model. We formalized the problem by considering an incremental framework modeling the elimination or reduction of different types of clutter as optimization problems, which we analyzed in terms of complexity. Since these problems turned out to be NP-hard in general, we also devised efficient approximation algorithms as well as efficient heuristics. We showed how to heuristically eliminate vertex-clutter in  $O((n+m)\log^3 m + n\log^2 n)$  time and how to reduce edge clutter in  $O(nm + m^2)$  time considering geometric features such as point distributions and geodesic tendencies. After the elimination of vertex-clutter and edge-clutter we can expect the graph to be much smaller than the original graph. Hence, even larger complexities may scale accordingly. Thus, even the relatively high complexity of our heuristic for reducing vertex-edge clutter may be practical.

Even without this step, however, the resulting generalizations exhibit considerably less clutter and are easier to analyze. We showcase some generalizations produced by our heuristics in Figure 6 as well as in [6]. We conclude by listing some open problems.

- Is it possible to approximate both the local coverage and the size of a  $\rho$ -set in the vertex generalization step?
- What is the complexity of the LOCAL COVERAGE CLUSTER PACKING problem for different type of mappings?
- Is it possible to approximate the size of a shortest geodesic subgraph, possibly in the presence of a limited drift?
- What is the complexity of the optimal angle adjustment problem?
- How can the generalization problem be adapted to a dynamic scenario, where consistency issues play an additional role.

**Acknowledgments.** We thank Robert Görke for the helpful discussion and for providing the LunarVis layout.

## References

1. Abello, J., Kobourov, S.G., Yusufov, R.: Visualizing Large Graphs with Compound-Fisheye Views and Treemaps. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 431–441. Springer, Heidelberg (2005)
2. Abello, J., Korn, J., Finocchi, I.: Graph sketches. In: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS 2001), p. 67. IEEE Computer Society (2001)
3. Bentley, J.L., Saxe, J.B.: Decomposable searching problems I. static-to-dynamic transformation. *Journal of Algorithms* 1(4), 301–358 (1980)
4. Bohn, R.E., Short, J.E.: How much information? 2009 Report on American consumers. Global Information Industry Center, University of California, San Diego (2009)
5. Brandes, U., Delling, D., Gaertler, M., Görke, R., Hofer, M., Nikoloski, Z., Wagner, D.: On modularity clustering. *IEEE Trans. Knowledge and Data Engineering* 20, 172–188 (2008)
6. Brunel, E., Gemsa, A., Krug, M., Rutter, I., Wagner, D.: Generalizing Geometric Graphs. Technical Report 27, Karlsruhe Institute of Technology (2011)

7. Chazelle, B., Cole, R., Preparata, F.P., Yap, C.: New upper bounds for neighbor searching. *Information and Control* 68(1-3), 105–124 (1986)
8. Chazelle, B.: Functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.* 17, 427–462 (1988)
9. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. *Discrete Mathematics* 86(1-3), 165–177 (1990)
10. Davis, T.A.: University of florida sparse matrix collection. *NA Digest* 92 (1994)
11. de Berg, M., Khosravi, A.: Optimal Binary Space Partitions in the Plane. In: Thai, M.T., Sahni, S. (eds.) *COCOON 2010*. LNCS, vol. 6196, pp. 216–225. Springer, Heidelberg (2010)
12. Dobkin, D., Friedman, S., Supowit, K.: Delaunay graphs are almost as good as complete graphs. *Discrete & Computational Geometry* 5, 399–407 (1990)
13. Eades, P., Feng, Q.-W.: Multilevel Visualization of Clustered Graphs. In: North, S.C. (ed.) *GD 1996*. LNCS, vol. 1190, pp. 101–112. Springer, Heidelberg (1997)
14. Edelsbrunner, H., Guibas, L., Sharir, M.: The upper envelope of piecewise linear functions: Algorithms and applications. *Discr. & Comp. Geometry* 4, 311–336 (1989)
15. Furnas, G.W.: Generalized fisheye views. *SIGCHI Bull.* 17, 16–23 (1986)
16. Gaertler, M.: Clustering. In: Brandes, U., Erlebach, T. (eds.) *Network Analysis*. LNCS, vol. 3418, pp. 178–215. Springer, Heidelberg (2005)
17. Garey, M.R., Johnson, D.S.: *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company (1979)
18. Görke, R., Gaertler, M., Wagner, D.: Lunarvis - Analytic Visualizations of Large Graphs. In: Hong, S.-H., Nishizeki, T., Quan, W. (eds.) *GD 2007*. LNCS, vol. 4875, pp. 352–364. Springer, Heidelberg (2008)
19. Hachul, S., Jünger, M.: Drawing Large Graphs with a Potential-Field-Based Multilevel Algorithm. In: Pach, J. (ed.) *GD 2004*. LNCS, vol. 3383, pp. 285–295. Springer, Heidelberg (2005)
20. Halldórsson, M., Radhakrishnan, J.: Greed is good: Approximating independent sets in sparse and bounded-degree graphs. *Algorithmica* 18, 145–163 (1997)
21. Harel, D., Koren, Y.: Graph Drawing by High-Dimensional Embedding. In: Goodrich, M.T., Kobourov, S.G. (eds.) *GD 2002*. LNCS, vol. 2528, pp. 207–219. Springer, Heidelberg (2002)
22. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. In: *Proc. of the 11th Eurographics/IEEE-VGTC Symp. on Vis.*, pp. 983–990 (2009)
23. Koren, Y., Carmel, L., Harel, D.: Drawing huge graphs by algebraic multigrid optimization. *Multiscale Modeling and Simulation* 1, 645–673 (2003)
24. Mackaness, W.A., Beard, K.M.: Use of graph theory to support map generalization. *Cartography and Geographic Information Science* 20, 210–221 (1993)
25. Mackaness, W.A., Ruas, A., Sarjakoski, L.T. (eds.): *Generalisation of Geographic Information. Cartographic Modelling and Applications*. Elsevier B.V. (2007)
26. Misue, K., Eades, P., Lai, W., Sugiyama, K.: Layout adjustment and the mental map. *Journal of Visual Languages & Computing* 6(2), 183–210 (1995)
27. Openstreetmap database (2011), <http://www.openstreetmap.de/>
28. Quigley, A., Eades, P.: Fade: Graph Drawing, Clustering, and Visual Abstraction. In: Marks, J. (ed.) *GD 2000*. LNCS, vol. 1984, pp. 197–210. Springer, Heidelberg (2001)
29. Saalfeld, A.: Map Generalization as a Graph Drawing Problem. In: Tamassia, R., Tollis, I.G. (eds.) *GD 1994*. LNCS, vol. 894, pp. 444–451. Springer, Heidelberg (1995)
30. Sarkar, M., Brown, M.H.: Graphical fisheye views of graphs. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1992*, pp. 83–91. ACM, New York (1992)
31. Telea, A., Ersoy, O.: Image-based edge bundles: Simplified visualization of large graphs. *Computer Graphics Forum* 29(3), 843–852 (2010)