# Selected Open Problems in Graph Drawing

Franz Brandenburg*, David Eppstein**, Michael T. Goodrich***,
Stephen Kobourov†, Giuseppe Liotta‡, and Petra Mutzel§

**Abstract.** In this manuscript, we present several challenging and interesting open problems in graph drawing. The goal of the listing in this paper is to stimulate future research in graph drawing.

## 1 Introduction

This paper is devoted to exploring some interesting and challenging open problems in graph drawing. We have specifically chosen topics motivated from research themes of current interest in graph drawing, including proximity drawings, 2D and 3D straight-line drawings, 2D and 3D orthogonal representations, and graph drawing checkers. Our choice of open problems and themes should not be seen as exhaustive, however, as graph drawing is a rich area of research with many open problems and research themes. For example, there are interesting open problems related to the routing of multiple curves in the plane, dynamic graph drawing, drawing of hypergraphs and Venn diagrams, applied graph drawing, and label placement in graph drawings, which we do not specifically address in this paper.

In any survey of open problems, there is naturally going to be a certain emphasis that reflects the perspective of the authors, and this paper is no exception. Nevertheless, we have striven to include open problems motivated by topics we believe are of general interest. That is, we feel that the solution to any of the open problems we include in this paper would be of wide interest in the graph drawing community.

## 2 Proximity Drawability Problems

Recently, much attention has been devoted to the study of the combinatorial properties of different types of proximity graphs. Proximity graphs, originally defined in the context of computational geometry and pattern recognition, are typically used to describe the shape of a set of points. In the survey by Toussaint [59], such graphs are classified by using the notion of *proximity* between

* Univ. of Passau, Germany. brandenburg@informatik.uni-passau.de
** Univ. of California-Irvine, USA. eppstein@ics.uci.edu
*** Univ. of California-Irvine, USA. goodrich@acm.org
† Univ. of Arizona, USA. kobourov@cs.arizona.edu
‡ Univ. of Perugia, Italy. liotta@diei.unipg.it
§ Vienna Univ. of Technology, Austria. mutzel@ads.tuwien.ac.at

points: In a proximity graph, two points are connected by an edge if and only if they are deemed *close* by some proximity measure. Examples of proximity graphs include well-known geometric graphs such as minimum spanning trees, Gabriel graphs, minimum weight triangulations, rectangle of influence graphs, visibility graphs, and Delaunay diagrams [82].

Proximity graphs can be regarded as straight-line drawings that satisfy some additional geometric constraints. Thus the problem of analyzing the combinatorial properties of a given type of proximity graph naturally raises the question of the characterization of those graphs admitting the given type of drawing. This, in turn, leads to the investigation of the design of efficient algorithms for computing such a drawing when one exists. These questions are far from being resolved in general, and only partial answers have appeared in the literature [35, 37, 59, 65, 67, 72, 73, 75].

### 2.1 Realizability of minimum spanning trees in 3D space

Among the most challenging questions in this family of graph drawing problems it has to be mentioned one related with drawing trees as minimum spanning trees in 3D. A *minimum spanning tree* of a set $P$ of points is a connected, straight-line drawing that has $P$ as vertex set and minimizes the total edge length. A tree $T$ can be drawn as a minimum spanning tree if there exists a set $P$ of points such that the minimum spanning tree of $P$ is isomorphic to $T$.

The problem of testing whether a tree can be drawn as a Euclidean minimum spanning tree in the plane is essentially solved. Monma and Suri [80] proved that each tree with maximum vertex degree at most five can be drawn as a minimum spanning tree of some set of vertices by providing a linear time (real RAM) algorithm. In the same paper it is shown that no tree having at least one vertex with degree greater than six can be drawn as a minimum spanning tree. As for trees having maximum degree equal to six, Eades and Whitesides [41] showed that it is NP-hard [47] to decide whether such trees can be drawn as minimum spanning trees. The 3-dimensional counterpart of the problem is however not yet solved. In [71] it is shown that no trees having at least one vertex with degree greater than twelve can be drawn as a Euclidean minimum spanning tree in 3D while all trees with vertex degree at most nine are drawable.

**Problem 1:** *Let $T$ be a tree with maximum vertex degree at most twelve. Is there a polynomial time algorithm to decide whether $T$ can be drawn as a Euclidean minimum spanning tree in 3D-space and, if so, compute such a drawing?*

### 2.2 Drawing minimum weight triangulations

A *minimum weight drawing* of a planar triangulated graph $G$ is a straight-line drawing $\Gamma$ of $G$ with the additional property that $\Gamma$ is a minimum weight triangulation of the points representing the vertices. If a graph admits a minimum weight drawing it is called *minimum weight drawable*, else it is called *minimum weight forbidden.*

Little is known about the problem of constructing a minimum weight drawing of a planar triangulation. Moreover, it is still not known whether computing a minimum weight triangulation of a set of points in the plane is NP-hard. Several papers have been published on this last problem, either providing partial solutions, or giving efficient approximation heuristics. A limited list of references includes the work by Meijer and Rappaport [83], Lingas [70], Keil [63], Dickerson et al. [33], Kirkpatrick [64], Aichholzer et al. [2], Cheng and Xu [14], Dickerson and Montague [34], and Levcopoulos and Krznaric [68, 69].

In [65] Lenhart and Liotta show that all maximal outerplanar triangulations are minimum weight drawable and a linear time (real RAM) drawing algorithm for computing a minimum weight drawing of these graphs was given. In [67] Lenhart and Liotta examine the *endoskeleton*—or *skeleton*, for short—of a triangulation: that is, the subgraph induced by the internal vertices of the triangulation. They construct skeletons that cannot appear in any minimum weight drawable triangulation; skeletons that do appear in minimum weight drawable triangulations; and skeletons that guarantee minimum weight drawability.

Besides the natural (and ambitious) goal of characterizing those planar graphs that are minimum weight drawable, we recall here two open problems whose solution would represent an interesting contribution to the investigation of the combinatorial structure of minimum weight triangulations.

In [67] it is shown that any forest can be the skeleton of a minimum weight drawable triangulation. On the other hand, Wang, Chin, and Yang [97] also focus on the minimum weight drawability of triangulations with acyclic skeletons and show examples of triangulations of this type that do not admit a minimum weight drawing. These two results naturally lead to the following problem.

**Problem 2:** *Characterize the class of triangulations with acyclic skeleton that admit a minimum weight drawing.*

Lenhart and Liotta [67] also show that there exist an infinite class of minimum weight drawable triangulations that cannot be realized as Delaunay triangulations (that is, for any triangulation $T$ of the class it does not exist a set $P$ of points such that the Delaunay triangulation of $P$ is isomorphic to $T$). It is worth remarking that the study of the geometric differences between the minimum-weight and Delaunay triangulations of a given set of points in order to compute good approximations of the former has a long tradition (see, e.g., [64, 68, 77]), yet little is known about the combinatorial difference between Delaunay triangulations and minimum-weight triangulations.

**Problem 3:** *Further investigate the combinatorial relationship between minimum weight and Delaunay drawable triangulations. Are there any Delaunay drawable and minimum weight forbidden triangulations?*

### 2.3 Gabriel drawability

Let $P$ be a set of $n$ distinct points in the plane. The Gabriel graph of $P$ is a geometric graph whose vertices are the points of $P$ such that there exists an edge $(u, v)$ if and only if the disk whose antipodal points are $u$ and $v$ does not contain any points of $P$ except $u$ and $v$ (the disk is assumed to be a closed set). Gabriel graphs have been first studied in the context of pattern recognition and computational morphology, where one is given a set of points on the plane and is asked to display the underlying *shape* of the set by constructing a graph whose vertices are the points and whose edges are segments connecting pairs of points. For references on this topics see, e.g., the survey by Toussaint [59].

More recently, the problem of computing straight-line drawings that have the additional property of being Gabriel graphs of their vertices has been considered. In [7], the problem of characterizing Gabriel-drawable trees has been addressed and an algorithm to compute Gabriel drawings of trees in the plane is given. Lubiw and Sleumer [75] proved that maximal outerplanar graphs admit both relative neighborhood drawings and Gabriel drawings. This result has been extended in [66] to all biconnected outerplanar graphs. Several problems about Gabriel drawings remain open. Among the most fascinating and challenging, we mention the following.

**Problem 4:** *Characterize the family of Gabriel drawable triangulations, that is, the family of those triangulations that admit a straight-line drawing where the angles of each triangular face are less than* $\pi/2$.

## 3 Straight-line Drawability in 2D and 3D

A classical area of investigation deals with crossing-free straight-line drawings of planar graphs in two and three dimensions. Given a graph $G$, the vertices in a drawing of $G$ are constrained to be located at integer grid points and the optimization goal is that of computing drawings whose area/volume is small. The area/volume of a drawing $\Gamma$ is measured as the number of grid points contained in a *bounding box* of $\Gamma$, i.e., the smallest axis-aligned box enclosing $\Gamma$.

### 3.1 Compact straight-line drawings in 2D-space

A rich body of literature has been published on computing straight-line drawings of graphs, such that the vertices are the intersection points of an integer 2D grid and the overall area of the drawing is kept small. Typically, papers that deal with this subject focus on lower bounds on the area required by straight-line drawings of specific classes of graphs and on the design of algorithms that possibly match these lower bounds. Schnyder [86, 87] and de Fraysseix, Pach, and Pollack [20] independently showed that every $n$-vertex triangulated planar graph has a crossing-free straight-line $O(n) \times O(n)$ grid drawing, and that this is worst case optimal. This seminal contribution was followed by related work by Kant [60, 61], Chrobak and Kant [17], Schnyder and Trotter [88], Felsner [45], and Chrobak, Goodrich, and Tamassia [15]. Results on classes of drawings of trees include the results by Garg, Goodrich and Tamassia [50], by Chan [10], and by Garg and Rusu [52]. Summarizing tables and more references can be found in the book by Di Battista, Eades, Tamassia, and Tollis [22].

The following problem is motived by the observation that trees admit straight-line grid drawings with linear or almost-linear area, while triangulations may require a grid of quadratic size.

**Problem 5:** *Find nontrivial classes of graphs with $n$ vertices richer than trees that admit straight-line planar drawings on an integer 2D grid of size $o(n^2)$.*

Some evidence that Problem 5 might be solvable for families of graphs other than trees was given in a recent paper by Biedl [5], where she shows that all outerplanar graphs admit an $O(n \log n)$ area drawing; however, the algorithm presented in [5] does not compute straight-line drawings. Additional problems in this area include the following well-known problem, which has been popularized by Xin He:

**Problem 6:** *Given an $n$-vertex plane triangular graph $G$, what is the smallest area straight-line planar drawing of $G$ such that vertices are drawn at integer grid points?*

This problem was first posed by Rosenstiehl and Tarjan [84]. A well-known upper bound is $(n-1) \times (n-1)$ (many such algorithms achieve this), which was recently improved by Zhang and He [100] to $(n - \Delta - 1) \times (n - \Delta - 1)$, where $0 \leq \Delta \leq \lfloor (n-1)/2 \rfloor$ is a value derived from the cycle structure of $G$. A known lower bound for this problem is $2n/3 \times 2n/3$ (the nested triangles example). Specifically, the nested triangles example requires $2n/3 \times 2n/3$ area to draw if a combinatorial embedding and outer face are given [20, 76]. Because of this input restriction, the following problem is also of interest.

**Problem 7:** *What is the smallest area grid drawing for the nested triangles graph (where a combinatorial embedding and outer face are not given)?*

Many planar graph drawing algorithms incrementally construct a drawing based on inserting vertices according to a particular ordering, such as an st-numbering or canonical ordering of the graph's vertices [61]. Such drawings place vertices in a *numbering-upward* fashion, so that listing the vertices according to the ordering lists the vertices by nondecreasing $y$-coordinates.

**Problem 8:** *Is there a polynomial algorithm for numbering and embedding the vertices of a given planar graph $G$ so as to minimize the area of a numbering-upward planar drawing of $G$ (taken over all possible numberings of $G$)?*

### 3.2 Extensibility and universality for planar graphs

Consider the following problem: given a planar graph in which some vertices have already been placed in the plane (i.e., a *partial embedding*), place the remaining vertices to form a straight line embedding of the graph.

**Problem 9:** *Is there a polynomial time algorithm for extending a partial straight-line embedding?*

This problem also comes up in mesh generation, where the already-placed vertices can be assumed to form a simple polygon and the graph can be assumed to have all interior faces triangles; do these assumptions simplify the problem?

A related problem is the following.

**Problem 10:** *(Drawing with fixed vertex positions) Suppose you are given as input an unlabeled planar graph with $n$ vertices, and a set of $n$ points in the plane. You wish to assign vertices to points to create a planar straight line drawing. What is the computational complexity of this problem?*

Concerning the problem above, Bose [6] conjectured it to be NP-complete and solved some special cases in polynomial time. Kaufmann and Wiese [62] show NP-completeness of a related problem in which some bends are allowed.

In general, a set $S$ of points is called a *universal set* for a family of a graphs $\mathcal{F}$ if every graph in $\mathcal{F}$ can be drawn crossing-free using the points from $S$ for vertex locations.

**Problem 11:** *Does there exist a small universal point set for planar graphs? That is, given $n$, is there a set of $O(n)$ points in the plane, so that every planar graph of $n$ vertices can be drawn with straight-lines and no crossings on this point set? If not, what is the smallest $f(n)$ such that there exists a set of $f(n)$ points such that every $n$-vertex graph can be drawn using these points?*

There are several algorithms that show $f(n) \leq n^2$. Chrobak and Karloff [18] show that $f(n) \geq 1.098n$, for sufficiently large $n$. Likewise, the following bi-universal set question is also of interest.

**Problem 12:** *Given two planar graphs each on $n$ vertices, can one always find a set of $n$ points, so that each of the two graphs can be embedded with straight-lines and no crossings on this set?*

### 3.3 Universal Sets in 3D

While the problem of computing small-sized crossing-free straight-line drawings in the plane has a long tradition, its 3D counterpart has become the subject of much attention only in recent years. Chrobak, Goodrich, and Tamassia [15] gave an algorithm for constructing 3D convex drawings of triconnected planar graphs with $O(n)$ volume and non-integer coordinates. Cohen, Eades, Lin and Ruskey [19] showed that every graph admits a straight-line crossing-free 3D drawing on an integer grid of $O(n^3)$ volume, and proved that this is asymptotically optimum. Calamoneri and Sterbini [8] showed that all 2-, 3-, and 4-colorable graphs can be drawn in a 3D grid of $O(n^2)$ volume with $O(n)$ aspect ratio and proved a lower bound of $\Omega(n^{1.5})$ on the volume of such graphs. For $r$-colorable graphs, Pach, Thiele and Tóth [81] showed a bound of $\theta(n^2)$ on the volume. Garg, Tamassia, and Vocca [55] showed that all 4-colorable graphs (and hence all planar graphs) can be drawn in $O(n^{1.5})$ volume and with $O(1)$ aspect ratio but using a grid model where the coordinates of the vertices may not be integer.

In a recent paper, Felsner, Liotta, and Wismath [46] approached the drawing problem with the following point of view: Instead of "squeezing" a drawing onto a small portion of a grid of unbounded dimensions, it is assumed that a grid of *specified* dimensions (involving a function of $n$) is given and we consider what the graphs are whose drawings fit that restricted grid. For example, it is well-known that there are families of graphs that require $\Omega(n^2)$ area to be drawn in the plane, the canonical example being a sequence of $n/3$ nested triangles (see [20, 16, 87]), as mentioned above. Such graphs can be drawn on the surface of a three dimensional triangular prism of linear volume and using integer coordinates. Thus a natural question is whether there exist specific restrictions of the 3D integer grid of linear volume that can support straight-line crossing-free drawings of meaningful families of graphs. The following open problem is mentioned in the paper by Felsner, Liotta, and Wismath [46].

**Problem 13:** *Does there exist a 3D universal grid set of linear volume that supports all planar graphs?*

Recent partial results on Problem 13 are as follows. In [46] it is shown that all outerplanar graphs can be drawn in a restricted integer 3D grid of linear volume, called *prism*. A prism is a subset of the integer 3D grid consisting of three parallel lines through the points $(0, 1, 0)$, $(0, 0, 0)$, and $(0, 0, 1)$. In the same paper it is also shown that a prism does not support all planar graphs and that even adding a fourth parallel line does not suffice in general. Dujmovic, Morin, and Wood [38] present $O(n \log^2 n)$ volume drawings of graphs with bounded tree-width and $O(n)$ volume for graphs with bounded path-width. Wood [99] shows that also graphs with bounded queue number have 3D straight-line grid drawings of $O(n)$ volume. A recent result by Dujmovic and Wood [40, 39] shows that linear volume can also be achieved for graphs with bounded tree-width; they show 3D straight-line grid drawings of volume $c \times n$ for these graphs, where $c$ is a constant whose value exponentially depends on the tree-width. Di Giacomo, Liotta, and Wismath [31, 28] show $4 \times n$ and $32 \times n$ volume for two subclasses of series-parallel graphs. Work-in-progress about linear-volume 3D drawings of graphs includes [29, 32].

## 4 Orthogonal Representations

An *orthogonal drawing* of a graph maps its vertices to points on an integer grid and its edges to a sequence of alternating axis-parallel segments. The study of the orthogonal drawing convention has a very long tradition in the graph drawing literature, because of the several applications of this convention in a variety of fields. For an introduction on orthogonal drawings and their several applications, see [22]. It is immediate to see that in order for an orthogonal drawing to exist, the degree of the vertices must not exceed four in the plane and six in three dimensions.

### 4.1 Minimizing the number of bends in a 2D orthogonal representation

The *topology-shape-metrics approach* [22] for constructing a planar orthogonal drawing of a planar graph in $2D$ consists of three main steps, called planarization, orthogonalization, and compaction. The planarization step determines an embedding, i.e., the face cycles, for the graph in the plane. The orthogonalization step determines an *orthogonal representation* of the input graph, i.e., a labeling for each edge $(u, v)$ of the graph that defines the shape of $(u, v)$ in the final drawing. For example, $(u, v)$ could be labeled $RLLLR$, which would say "starting from $u$ first turn right, then turn left three times, then right." Finally, the compaction step computes the drawing, giving coordinates to vertices and bends while preserving the shape of the edges determined in the orthogonalization step. For each step of the approach, different optimization problems (for example minimizing the number of bends, minimizing the area, minimizing the maximum edge length) have been studied, and papers providing optimal algorithms and effective heuristics have been presented.

A *bend-minimum* planar orthogonal drawing of a plane graph $G$ is one which has the minimum number of bends along the edges among all possible planar orthogonal drawings of $G$. The problem of computing bend-minimum planar orthogonal drawings is among the most famous in the graph drawing literature and has been studied both in the *fixed embedding setting* and in the *variable embedding setting*.

In the fixed embedding setting, the input is a planar graph $G$ together with a planar embedding (i.e., a circular ordering of the edges incident on each vertex of $G$); the output is a planar orthogonal drawing of $G$ that has the minimum number of bends among all drawings which maintain the given embedding of $G$. A well-known seminal paper by Tamassia [93] shows that such drawing of $G$ can be computed in $O(n^2 \log n)$ time by mapping the problem to computing a flow of minimum cost on a suitable network. The time complexity bound is further improved by Garg and Tamassia [53] who present an $O(n^{1.75} \log n)$ time algorithm. One of the most famous and long standing open problems in graph drawing is the following.

**Problem 14:** *Let $G$ be a 4-planar graph (i.e., a planar graph whose maximum vertex degree is at most four) with a given planar embedding $\Phi$. Is there an algorithm that computes a bend-minimum orthogonal representation of $G$ preserving $\Phi$ and has time complexity $o(n^{1.75} \log n)$?*

Notice that the choice of a planar embedding of $G$ can deeply affect the resulting number of bends in the orthogonal representation. For example, the algorithm in [93] can give rise to orthogonal representations of the same graph that, depending on the choice of its planar embedding, differ in the number of bends by even a linear factor [26]. Thus, there is natural interest in algorithms that work in a *variable embedding setting*, i.e., algorithms that are allowed to change the planar embedding of the input graph in order to minimize the bends.

The variable embedding version of the bend-minimization problem is unfortunately harder than the fixed embedding one. Namely, Garg and Tamassia [54] have proved that computing a bend-minimum planar orthogonal drawing in the variable embedding setting is NP-hard. Di Battista, Liotta, and Vargiu [26] proved that the problem can be solved in $O(n^3)$ time if the input graph is a series-parallel graph and that it can be solved in $O(n^5 \log n)$ time if the input is a 3-planar graph (i.e., a graph that has vertex degree at most three). Garg and Liotta [51] further reduced this time complexity, but at the expenses of a few extra bends in the computed representation. Namely, they present an $O(n^2)$-time algorithm that receives as input a 3-planar graph $G$ with $n$ vertices and computes an orthogonal representation of $G$ with at most 3 bends more than the minimum number of bends.

The natural counterpart of Problem 14 in the variable embedding setting is therefore the following.

**Problem 15:** *Let $G$ be a planar graph whose vertices have degree at most three. Is there an algorithm to compute a planar bend-minimum orthogonal drawing of $G$ in $o(n^5 \log n)$ time?*

### 4.2 Compact Orthogonal Layouts

As in other drawing problems, there is considerable interest in the exact area bounds for various orthogonal drawing problems.

**Problem 16:** *What are exact bounds for the area of binary tree layouts in each of the following scenarios:*

1. *straight-line drawings on the grid*
2. *rectangular drawings on the grid (H-layouts)*
3. *straight-line upwards drawings on the grid*
4. *rectangular upwards drawings on the grid (T-layouts)*

**Problem 17:** *Does every binary tree have a drawing with straight-line edges in $O(n)$ area such that all nodes are placed on grid points?*

In the rectangular case, the straight-line edges are either horizontal or vertical. In the upwards case, the edges are $y$-monotone or go in directions west, east, and north (south is excluded). Known results are $O(n \log \log n)$ area for straight-line drawings, $O(n \log \log n)$ for straight-line upwards drawings [50, 89], $O(n \log \log n)$ for rectangular drawings [11], and $O(n \log n)$ for upwards rectangular (from hv drawings).

**Problem 18:** *What are exact bounds for the area of upwards (strictly upwards) layouts of binary trees. Specifically, what is the complexity of determining, given a binary tree $T$ and a bound $K$, whether there a rectangular upwards (strictly upwards, hv) layout with area at most $K$? Are these problems NP-hard?*

Known results: NP-hardness is known for "H"-tree layouts (all 4 directions) [4] and improved to binary trees by Gregori [57], for upwards drawings of ternary trees by Edler [42], and for rectilinear layouts by Garg and Tamassia [54].

### 4.3 Orthogonal representations in 3D space

An essential prerequisite of the topology-shape-metrics approach to drawing graphs in 2D is a characterization of 2D orthogonal representation, i.e., a characterization of those graphs with edges labeled by orthogonal directions that can be drawn without crossings, while respecting the desired shapes for the edges. This problem has been studied in several papers, including [93, 95], and has also been generalized to non-orthogonal polygons and graphs in [96, 27, 49]. However, while the literature on 3D orthogonal drawings is quite rich, the extension of the topology-shape-metrics approach to 3D has so far remained an elusive target. A major difficulty is that in 3D, there is no counterpart to the 2D characterization of orthogonal representations.

A *direction label* is a label in the set $\{E, W, N, S, U, D\}$, where each label specifies a direction $East$, $West$, $North$, $South$, $Up$, or $Down$, respectively. Let $G$ be a graph such that each edge $(u, v)$ of $G$ is associated with two opposite orientations called *darts* of $(u, v)$. A *3D shape graph* is a labeling $\sigma$ of the darts of $G$ such that:

- Each dart is associated with exactly one direction label.
- For each edge $e$ of $G$ the two opposite darts of $e$ have labels which specify opposite directions.
- Each vertex does not have two entering darts with the same label.

Shape graph $\gamma$ is said to be *a 3D orthogonal representation* if there exists an orthogonal drawing $\Gamma$ of $G$ so that $\Gamma$ is *simple* (i.e., no two edges of $\Gamma$ share any points except common endpoints) and satisfies the direction constraints on its edges as specified by $\sigma$.

**Problem 19:** *Characterize those shape graphs that are 3D orthogonal representations.*

Only very preliminary results toward finding such characterization have so far been discovered. Di Battista, Liotta, Lubiw, and Whitesides [25] solve the problem for paths (with the additional constraint the one endpoint must be drawn at the origin and the other in a given octant) and for shape cycles [24]. As for structurally more complex graphs, Di Giacomo, Liotta, and Patrignani [30] show that the characterization for shape cycles does not extend to general graphs, even for apparently simple structures such as theta graphs (a theta graph consists of three cycles). In the same paper, the authors present a sufficient condition under which a shape theta graph is a 3D orthogonal representation. A consequence of the work in [30] is the following question, whose answer may be an important intermediate step toward the ambitious goal of solving Problem 19.

**Problem 20:** *Characterize those shape theta graphs that are 3D orthogonal representations.*

## 5 Graph Drawing Checkers

The intrinsic structural complexity of the implementation of geometric algorithms makes the problem of formally proving the correctness of the code unfeasible in most of the cases. This has been motivating the research on *checkers*. A checker is an algorithm that receives as input a geometric structure and a predicate stating a property that should hold for the structure. The task of the checker is to verify whether the structure satisfies or not the given property. Here, the expectation is that it is often easier to evaluate the quality of the output than the correctness of the software that produces it. Different papers (see, e.g., [21, 79]) have agreed on the basic features that a "good" checker should have:

**Correctness:** The checker should be correct beyond any reasonable doubt. Otherwise, one would incur in the problem of checking the checker.
**Simplicity:** The implementation should be straightforward.
**Efficiency:** The expectation is to have a checker that is more efficient than the algorithm that produces the geometric structure.
**Robustness:** The checker should be able to handle degenerate configurations of the input and should not be affected by errors in the flow of control due to round-off approximations.

Checking is especially relevant in the graph drawing context. Indeed, graph drawing algorithms are among the most sophisticated of the entire computational geometry field, and their goal is to construct complex geometric structures with specific properties. Also, because of their immediate impact on application areas, graph drawing algorithms are usually implemented right after they have been devised. Of course, the checking problem becomes crucial when the drawing algorithm deals with very large data sets, when a simple complete visual inspection of the drawing is difficult or unfeasible.

Devising graph drawing checkers involves answering only apparently innocent questions like: "is this drawing planar?" or "is this drawing upward?" or "are the faces convex polygons?".

The problem of checking the planarity of a subdivision has been independently studied by Mehlhorn et al. [79] and by Devillers et al. [21]. In these papers linear time algorithms are given to check the planarity of a subdivision composed by convex faces. The inputs are the subdivision plus its topological embedding in terms of the ordered adjacency lists of the edges. Unfortunately, extending the above techniques to checking the planarity of a subdivision whose faces are not constrained to be convex, relies on the usage of algorithms for testing the simplicity of a polygon. The only general linear time algorithm known for this problem is the well-known and fairly complex algorithm by Chazelle [12]. Hence, devising a checker based on such algorithm would not satisfy the simplicity requirement. The algorithm in [12] tests the simplicity of a polygon by means of an intermediate triangulation step. Alternative algorithms that can triangulate in linear time special classes of polygons have been devised. See e.g. [43, 48, 98]. Other almost optimal algorithms can be found in [13, 58, 94].

**Problem 21:** *Let $\Gamma$ be a connected straight-line drawing of a graph $G$ with $n$ vertices such that for each vertex $v$ of $\Gamma$ the circular ordering of the edges incident on $v$ is given. Devise a simple, robust, and efficient checker for testing the planarity of $\Gamma$.*

An example of checker of the type described by Problem 21 can be found in a paper by Di Battista and Liotta [23] who check the *upward planarity* of straight-line oriented drawings whose faces may not be convex polygons. They introduce *regular* upward planar embeddings and show that such embeddings coincide with those having a "unique" including planar *st*-digraph. The concept of regularity is exploited to investigate the relationships between topology and geometry of upward planar drawings. In particular, it is shown that an upward drawing of a regular planar upward embedding satisfies strong constraints on the left-to-right ordering of the edges. Based upon the above results and under the assumption of regularity, a simple and robust linear time checker is presented that tests whether a given drawing $\Gamma$ is upward planar without using any polygon triangulation routine.

Besides planarity, several other checking problems remain open in graph drawing. Indeed, all graph drawing algorithms guarantee certain geometric properties for the drawings they produce. Such properties are usually called "graphic standards" or "drawing conventions". Some of them appear to be easy to check, while others like checking proximity drawings seem to be more challenging. For example, consider the following problem.

**Problem 22:** *Let $\Gamma$ be a straight-line drawing of a tree (or even a simple chain). Is there a robust and simple algorithm to check in $o(n \log n)$ time whether $\Gamma$ is a Euclidean minimum spanning tree of the set of its vertices?*

## 6 Visualizing Graph Properties

There are also several interesting open problems that are related to the visualization of various graph properties or additional information associated with a graph.

### 6.1 Layered Graphs

Some graphs are labeled so as to partition the set of vertices into layers. Drawing algorithm that deal with this additional information usually seek to assign the vertices of each layer to a shared $y$-coordinate that is greater than that of the previous layer. Algorithms for drawing layered graphs usually are based on an approach that is commonly referred to as the Sugiyama algorithm [90–92] (even though the number of such algorithms is now quite large [22]). When such graphs attempt to minimize crossings, they invariably do so iteratively, processing layers two at a time.

**Problem 23:** *(Multilayer (or global) crossing minimization for the Sugiyama algorithm.) Design an effective heuristic which can perform crossing minimization for a layered graph over more than two layers at a time.*

Another issue regarding layered drawings is the representational complexity of describing a layered drawing.

**Problem 24:** *If a multi-layer drawing of a graph is made in which all the vertices have integer coordinates, how big do those integers need to be? (For planar graphs, it is known that all coordinates can be $O(n)$.)*

Also of interest are methods for speeding up Sugiyama-type algorithms.

**Problem 25:** *Is it possible to count the number of crossings in a bilayered graph faster than in $\Theta(V \log V)$ time? (this is equivalent to count the inversions in a sequence of length $n = V$).*

### 6.2 Clustered Graph Drawing

Say $C = (G, T)$ is a clustered graph with underlying graph $G$ and cluster tree $T$. A *horizon* of $C$ is a graph $H = (VH, EH)$ whose

- vertex set $VH$ is a subset of the node set of $T$ that "covers" the leaves of $T$, that is
  - every node of $G$ is a decsendent in $T$ of some node in $VH$
  - no node in in $VH$ is a descendent of another node in $VH$
- edge set $EH$ consists of the "implied edges" between nodes in $VH$,

that is, if $u$ and $v$ are in $VH$ then $(u, v)$ is an edge in $EH$ iff there is an edge $(a, b)$ in $G$ such that $a$ is a descendent of $u$ and $b$ is a descendent of $v$. Note that $G$ is a horizon of $C$; say that $G$ is the "level 0" horizon of $C$. If $i > 0$, then the level $i$ horinzon $H(i)$ has node set all nodes of $T$ whose children in $T$ are in the level $i - 1$ horizon of $C$.

**Problem 26:** *A drawing problem: say $C$ is c-planar. Can draw $C$ be drawn in 3D such that all $H(i)$ are drawn as straight-line planar, as follows:*

- *$H(i)$ is on the plane $z = i$ the projection of a node $u$ in $H(i)$ onto the plane $z = (i - 1)$ lies within the convex hull of the children of $u$*
- *All edges (including implied edges) are straight lines*
- *no edges cross*

**Problem 27:** *A topological problem: say $H(3)$ is empty (i.e., there are only 2 levels) and $H(2)$ is a path. Can c-planarity be tested in polynomial time?*

This is something like testing layered graph planarity, so maybe it is possible in linear time.

### 6.3 Crossing minimization and related problems

The *crossing number* of a graph $G$ is the minimum number of edge crossings in any drawing of $G$, and the *pairwise crossing number* is the minimum number of pairs of edges that cross at least once.

**Problem 28:** *Is the crossing number equal to the pairwise crossing number?*

The *odd crossing number* is the minimum number of pairs of edges that cross an odd number of times.

**Problem 29:** *Is the crossing number equal to the odd-crossing number?*

**Problem 30:** *In a straight-line drawing of the complete graph $K_n$, how large a set of mutually crossing edges must there be?*

Aronov et al. [3] showed an $\Omega(\sqrt{n})$ bound, and asked whether a linear sized crossing family always exists. No nontrivial upper bound seems to be known. In addition, the following is of interest:

**Problem 31:** *What is the rectilinear crossing number of $K_n$? I.e., in a straight line drawing of the complete graph $K_n$, how many crossing pairs of edges must exist [74, 85]?*

**Problem 32:** *Finding the largest cardinality planar subgraph of a graph is NP-hard, and even MAXSNP-hard, so one can't expect to approximate it arbitrarily well, but there may be room for reducing the best known approximation ratio. For a long time the best approximation was the trivial one given by finding a spanning tree of the input graph, but this has now been improved to 2.25 [9]. Can this be further improved?*

### 6.4 Geometric Thickness

The *geometric thickness* of a graph is the minimum number of subgraphs into which the graph can be partitioned, in such a way that all subgraphs have planar straight line drawings with the same vertex positions in each drawing. Useful references for the problems that follow are [36, 78].

**Problem 33:** *The complete graphs $K_n$ should require a number of layers of the form $c\,n$, for some constant $c$. What is $c$? Similarly, what is the asymptotic number of layers needed for $K_{n,n}$?*

For arbitrary graphs, computing the thickness is known to be NP-complete.

**Problem 34:** *Is it equally hard to compute the geometric thickness? Can geometric thickness be approximated efficiently?*

**Problem 35:** *Which minor-closed graph families have bounded geometric thickness? Do bounded-degree graphs have bounded geometric thickness?*

### 6.5 Alternative graph representations

There are many additional research themes in graph drawing that consider alternative representations the the one that assigns vertices to points and edges to curves joining those points.

**Problem 36:** *Is there a polynomial algorithm that takes as input a graph and outputs a representation in the form of a closed simple polygon, with the graph's vertices placed at polygon corners, such that an edge is present in the graph if and only if the line segment between the vertices is contained in the polygon?*

This is the famous *visibility graph recognition* problem. Also of interest is the incidence graphs of line segments.

**Problem 37:** *(Incidence graph recognition) Which graphs can be represented as incidence graphs of line segments in the plane? Does every planar graph have such a representation*[1]*?*

This problem is related to the following.

**Problem 38:** *(Conway's thrackle problem) Can a graph with more than $n$ edges be drawn in such a way that each pair of edges has a single point of intersection (including the shared endpoint of coincident edges as an intersection)?*

This is known to be true for straight line drawings, but is open for curved edges[2].

Finally, there is a class of problems for finding small representations of dense graphs. For example, given a simple polygon $P$, the *visibility graph* $G$ for $P$ is the graph having vertex set the same as $P$ and such that there is an edge $(v, w)$ in $G$ if and only if the line segment joining $v$ and $w$ never crosses the boundary of $P$. The following problem[3] is now classic [1, 44, 56].

**Problem 39:** *(Visibility graph recognition) Is there a polynomial algorithm for determining, given a graph $G$ and Hamiltonian cycle $C$ in $G$, if there a polygon $P$ with $C$ as its boundary and having $G$ as its visibility graph?*


### Acknowledgements

We wish to thank Emilio Di Giacomo for useful discussions and comments while preparing this collection of open problems. We also thank Peter Eades for communicating Problems 26 and 27.


[1] http://www.cs.unt.edu/~cbms/CBMS

[2] http://cs.smith.edu/ orourke/TOPP/P30.html#Problem.30

[3] http://cs.smith.edu/ orourke/TOPP/P30.html#Problem.17

## References


1. J. Abello and K. Kumar. Visibility graphs and oriented matroids. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes Comput. Sci.*, pages 147–158. Springer-Verlag, 1995.
2. O. Aichholzer, F. Aurenhammer, S.-W. Chen, N. Katoh, M. Taschwer, G. Rote, and Y.-F. Xu. Triangulations intersect nicely. *Discrete Comput. Geom.*, 16:339–359, 1996.
3. B. Aronov, P. Erdős, W. Goddard, D. J. Kleitman, M. Klugerman, J. Pach, and L. J. Schulman. Crossing families. *Combinatorica*, 14:127–134, 1994.
4. S. Bhatt and S. Cosmadakis. The complexity of minimizing wire lengths in VLSI layouts. *Inform. Process. Lett.*, 25:263–267, 1987.
5. T. Biedl. Drawing outer-planar graphs in $O(n \log n)$ area. In M. Goodrich, editor, *Graph Drawing (Proc. GD'02)*, volume 2528 of *Lecture Notes Comput. Sci.*, pages 54–65. Springer-Verlag, 2002.
6. P. Bose. On embedding an outer-planar graph in a point set. *Comput. Geom.*, 23(3):303–312, 2002.
7. P. Bose, W. Lenhart, and G. Liotta. Characterizing proximity trees. *Algorithmica*, 16:83–110, 1996.
8. T. Calamoneri and A. Sterbini. 3D straight-line grid drawing of 4-colorable graphs. *Inform. Process. Lett.*, 63(2):97–102, 1997.
9. G. Calinescu, C. G. Fernandes, U. Finkler, and H. Karloff. A better approximation algorithm for finding planar subgraphs. In *Proc. 7th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 16–25, 1996.
10. T. Chan. A near-linear area bound for drawing binary trees. *Algorithmica*, 34:1–13, 2001.
11. T. M. Chan, M. T. Goodrich, S. R. Kosaraju, and R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 63–75. Springer-Verlag, 1997.
12. B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.
13. B. Chazelle and J. Incerpi. Triangulation and shape-complexity. *ACM Trans. Graph.*, 3(2):135–152, 1984.
14. S.-W. Cheng and Y.-F. Xu. On $\beta$-skeleton as a subgraph of the minimum weight triangulation. *Theoretical Computer Science*, 262(1-2):459–471, 2001.
15. M. Chrobak, M. T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 319–328, 1996.
16. M. Chrobak and S. ichi Nakano. Minimum-width grid drawings of plane graphs. *Comput. Geom. Theory Appl.*, 11:29–54, 1998.
17. M. Chrobak and G. Kant. Convex grid drawings of 3-connected planar graphs. *Internat. J. Comput. Geom. Appl.*, 7(3):211–223, 1997.
18. M. Chrobak and H. Karloff. A lower bound on the size of universal sets for planar graphs. *SIGACT News*, 20, 1989.
19. R. F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. *Algorithmica*, 17:199–208, 1997.
20. H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.

21. O. Devillers, G. Liotta, F. P. Preparata, and R. Tamassia. Checking the convexity of polytopes and the planarity of subdivisions. *Comput. Geom. Theory Appl.*, 11:187–208, 1998.
22. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing.* Prentice Hall, Upper Saddle River, NJ, 1999.
23. G. Di Battista and G. Liotta. Upward planarity checking"faces are more than polygons". In S. Whitesides, editor, *Graph Drawing (Proc. GD '98)*, Lecture Notes Comput. Sci. Springer-Verlag, 1998.
24. G. Di Battista, G. Liotta, A. Lubiw, and S. Whitesides. Orthogonal drawings of cycles in 3D space. In J. Marks, editor, *Graph Drawing (Proc. GD '00)*, Lecture Notes Comput. Sci. Springer-Verlag, 2000.
25. G. Di Battista, G. Liotta, A. Lubiw, and S. Whitesides. Embedding problems for paths with direction constrained edges. *Theoretical Computer Science*, 289(2):897–917, 2002.
26. G. Di Battista, G. Liotta, and F. Vargiu. Spirality and optimal orhogonal drawings. *SIAM J. Comput.*, 27(6):1764–1811, 1998.
27. G. Di Battista and L. Vismara. Angles of planar triangular graphs. *SIAM J. Discrete Math.*, 9(3):349–359, 1996.
28. E. Di Giacomo. Drawing series-parallel graphs on restricted integer 3D grids. Mansucript submitted for publication, 2003.
29. E. Di Giacomo, G. Liotta, and H. Meijer. 3D straight-line drawings of $k$-trees. In *Graph Drawing '03*, 2003. this proceedings.
30. E. Di Giacomo, G. Liotta, and M. Patrignani. Orthogonal 3D shapes of theta graphs. In M. Goodrich, editor, *Graph Drawing (Proc. GD'02)*, volume 2528 of *Lecture Notes Comput. Sci.*, pages 142–149. Springer-Verlag, 2002.
31. E. Di Giacomo, G. Liotta, and S. Wismath. Drawing series-parallel graphs on a box. In *14th Canadian Conference On Computational Geometry (CCCG '02)*, 2002.
32. E. Di Giacomo and H. Meijer. Track drawings of graphs with constant queue number. this proceedings, 2003.
33. M. Dickerson, J. Keil, and M. Montague. A large subgraph of the minimum weight triangulation. *Discrete and Computational Geometry*, 18:289–304, 1997.
34. M. T. Dickerson and M. H. Montague. A (usually?) connected subgraph of the minimum weight triangulation. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 204–213, 1996.
35. M. B. Dillencourt. Realizability of Delaunay triangulations. *Inform. Process. Lett.*, 33(6):283–287, Feb. 1990.
36. M. B. Dillencourt, D. Eppstein, and D. S. Hirschberg. Geometric thickness of complete graphs. *J. Graph Algorithms & Applications*, 4(3):5–17, 2000.
37. M. B. Dillencourt and W. D. Smith. Graph-theoretical conditions for inscribability and Delaunay realizability. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 287–292, 1994.
38. V. Dujmović, P. Morin, and D. Wood. Pathwidth and three-dimensional straight line grid drawings of graphs. In M. T. Goodrich and S. Koburov, editors, *Graph Drawing (Proc. GD '02)*, volume 2528 of *Lecture Notes Comput. Sci.*, pages 42–53. Springer-Verlag, 2002.
39. V. Dujmovic and D. R. Wood. Tree-partitions of $k$-trees with applications in graph layout. Technical Report 02-03, Dept. Comput. Sci., Carleton Univ., Ottawa, Canada, 2002.

40. V. Dujmović and D. R. Wood. Tree-partitions of $k$-trees with application in graph layout. In *Workshop on Graph Theoretic Concepts in Computer Science (Proc. WG '03)*, Lecture Notes Comput. Sci. Springer-Verlag, to appear.
41. P. Eades and S. Whitesides. The realization problem for Euclidean minimum spanning trees is NP-hard. *Algorithmica*, 16:60–82, 1996. (special issue on Graph Drawing, edited by G. Di Battista and R. Tamassia).
42. B. Edler. Effiziente algorithmen für flächenminimale layouts von bäumen. Diplomarbeit, Universität Passau, 1999.
43. H. ElGindy, D. Avis, and G. T. Toussaint. Applications of a two-dimensional hidden-line algorithm to other geometric problems. *Computing*, 31:191–202, 1983.
44. H. Everett. Visibility graph recognition. Report 231/90, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, 1990. Ph.D. Thesis.
45. S. Felsner. Convex drawings of planar graphs and the order dimension of 3-polytopes. *Order*, 18(1):19–37, 2001.
46. S. Felsner, G.Liotta, and S. Wismath. Straight line drawings on restricted integer grids in two and three dimensions. In P. Mutzel, M. Junger, and S. Leipert, editors, *Graph Drawing (Proc. GD '01)*, volume 2265 of *Lecture Notes Comput. Sci.*, pages 328–342. Springer-Verlag, 2001.
47. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.
48. M. R. Garey, D. S. Johnson, F. P. Preparata, and R. E. Tarjan. Triangulating a simple polygon. *Inform. Process. Lett.*, 7:175–179, 1978.
49. A. Garg. New results on drawing angle graphs. *Comput. Geom. Theory Appl.*, 9(1–2):43–82, 1998.
50. A. Garg, M. T. Goodrich, and R. Tamassia. Planar upward tree drawings with optimal area. *Internat. J. Comput. Geom. Appl.*, 6:333–356, 1996.
51. A. Garg and G. Liotta. Almost bend-optimal planar orthogonal drawings of biconnected degree-3 planar graphs in quadratic time. In J. Kratochvil, editor, *Graph Drawing (Proc. GD '99)*, Lecture Notes Comput. Sci. Springer-Verlag, 1999.
52. A. Garg and A. Rusu. Straight-line drawings of binary trees with linear area and good aspect ratio. In M. Goodrich, editor, *Graph Drawing (Proc. GD'02)*, volume 2528 of *Lecture Notes Comput. Sci.*, pages 320–331. Springer-Verlag, 2002.
53. A. Garg and R. Tamassia. A new minimum cost flow algorithm with applications to graph drawing. In S. C. North, editor, *Graph Drawing (Proc. GD '96)*, Lecture Notes Comput. Sci. Springer-Verlag, 1997.
54. A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001.
55. A. Garg, R. Tamassia, and P. Vocca. Drawing with colors. In *Proc. 4th Annu. European Sympos. Algorithms*, volume 1136 of *Lecture Notes Comput. Sci.*, pages 12–26. Springer-Verlag, 1996.
56. S. K. Ghosh. On recognizing and characterizing visibility graphs of simple polygons. *Discrete Comput. Geom.*, 17:143–162, 1997.
57. A. Gregori. Unit length embedding of binary trees on a square grid. *Inform. Process. Lett.*, 31:167–172, 1989.
58. S. Hertel and K. Mehlhorn. Fast triangulation of the plane with respect to simple polygons. *Inform. Control*, 64:52–76, 1985.
59. J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proc. IEEE*, 80(9):1502–1517, Sept. 1992.
60. G. Kant. A new method for planar graph drawings on a grid. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 101–110, 1992.

61. G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996. (special issue on Graph Drawing, edited by G. Di Battista and R. Tamassia).
62. M. Kaufmann and R. Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *Journal of Graph Algorithms and Applications*, 6(1):115–129, 2002.
63. M. Keil. Computing a subgraph of the minimum weight triangulation. *Comput. Geom. Theory Appl.*, 4:13–26, 1994.
64. D. G. Kirkpatrick. A note on Delaunay and optimal triangulations. *Inform. Process. Lett.*, 10(3):127–128, 1980.
65. W. Lenhart and G. Liotta. Drawing outerplanar minimum weight triangulations. *Inform. Process. Lett.*, 57(5):253–260, 1996.
66. W. Lenhart and G. Liotta. Proximity drawings of outerplanar graphs. In *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1997.
67. W. Lenhart and G. Liotta. The drawability problem for minimum weight triangulations. *Theoret. Comp. Sci.*, 270:261–286, 2002.
68. C. Levcopoulos and D. Krznaric. Tight lower bounds for minimum weight triangulation heuristic. *Inform. Process. Lett.*, 57:129–135, 1996.
69. C. Levcopoulos and D. Krznaric. A linear-time approximation schema for minimum weight triangulation of convex polygons. *Algorithmica*, 21:285–311, 1998.
70. A. Lingas. A new heuristic for minimum weight triangulation. *SIAM J. Algebraic Discrete Methods*, 8(4):646–658, 1987.
71. G. Liotta and G. Di Battista. Computing proximity drawings of trees in the 3-dimensional space. In *Proc. 4th Workshop Algorithms Data Struct.*, volume 955 of *Lecture Notes Comput. Sci.*, pages 239–250. Springer-Verlag, 1995.
72. G. Liotta, A. Lubiw, H. Meijer, and S. H. Whitesides. The rectangle of influence drawability problem. *Comput. Geom. Theory Appl.*, 10(1):1–22, 1998.
73. G. Liotta and H. Meijer. Voronoi drawings of trees. *Comput. Geom. Theory Appl.* to appear.
74. L. Lovasz, K. Vesztergombi, U. Wagner, and E. Welzl. Convex quadrilaterals and $k$-sets. Technical Report MSR-TR-2003-006, Microsoft Research, 2003.
75. A. Lubiw and N. Sleumer. Maximal outerplanar graphs are relative neighborhood graphs. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 198–203, 1993.
76. S. Malitz and A. Papakostas. On the angular resolution of planar graphs. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 527–538, 1992.
77. G. K. Manacher and A. L. Zobrist. Neither the greedy nor the Delaunay triangulation of a planar point set approximates the optimal triangulation. *Inform. Process. Lett.*, 9:31–34, 1979.
78. A. Mansfield. Determining the thickness of a graph is NP-hard. *Math. Proc. Cambridge Philos. Soc.*, 93(9):9–23, 1983.
79. K. Mehlhorn, T. S. S. Nher, S. Schirra, M. Seel, R. Seidel, , and C. Uhrig. Checking geometric programs or verification of geometric structures. *Comput. Geom. Theory Appl.*, 12:85–113, 1999.
80. C. Monma and S. Suri. Transitions in geometric minimum spanning trees. *Discrete Comput. Geom.*, 8:265–293, 1992.
81. J. Pach, T. Thiele, and G. Tóth. Three-dimensional grid drawings of graphs. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 47–51. Springer-Verlag, 1997.
82. F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 3rd edition, Oct. 1990.

83. D. Rappaport and H. Meijer. Computing the minimum weight triangulation of a set of linearly ordered points. *Information Processing Letters*, 42:35–38, 1992.
84. P. Rosenstiehl and R. E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete Comput. Geom.*, 1(4):343–353, 1986.
85. E. R. Scheinerman and H. S. Wilf. The rectilinear crossing number of a complete graph and Sylvester's "four point problem" of geometric probability. *Amer. Math. Monthly*, 101(10):939–943, 1994.
86. W. Schnyder. Planar graphs and poset dimension. *Order*, 5:323–343, 1989.
87. W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 138–148, 1990.
88. W. Schnyder and W. T. Trotter. Convex embeddings of 3-connected plane graphs. *Abstracts of the AMS*, 13(5):502, 1992.
89. C.-S. Shin, S. K. Kim, and K.-Y. Chwa. Area-efficient algorithms for straight-line tree drawings. *Comput. Geom. Theory Appl.*, 15:175–202, 2000.
90. K. Sugiyama. A readability requirement on drawing digraphs: Level assignment and edge removal for reducing the total length of lines. Technical Report 45, Internat. Inst. for Advanced Study of Social Information Science, Mar. 1984.
91. K. Sugiyama. Achieving uniqueness requirement in drawing digraphs: Optimum code algorithm and hierarchic isomorphism. Technical Report 58, Internat. Inst. for Advanced Study of Social Information Science, July 1985.
92. K. Sugiyama. A cognitive approach for graph drawing. *Cybernetics and Systems: An International Journal*, 18:447–488, 1987.
93. R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.
94. G. Toussaint. Efficient triangulation of simple polygons. *Visual Comput.*, 7:280–295, 1991.
95. G. Vijayan and A. Wigderson. Rectilinear graphs and their embeddings. *SIAM J. Comput.*, 14:355–372, 1985.
96. V. Vijayan. Geometry of planar graphs with angles. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 116–124, 1986.
97. C. A. Wang, F. Y. Chin, and B. Yang. Triangulations without minimum weight drawing. In *Algorithms and Complexity (Proc. CIAC 2000)*, volume 1767 of *Lecture Notes Comput. Sci.*, pages 163–173. Springer-Verlag, 2000.
98. T. C. Woo and S. Y. Shin. A linear time algorithm for triangulating a point-visible polygon. *ACM Trans. Graph.*, 4(1):60–70, 1985.
99. D. R. Wood. Queue layouts, tree-width, and three-dimensional graph drawing. In *22nd Foundations of Software Technology and Theoretical Computer Science (FSTTCS '02)*, volume 2556 of *Lecture Notes in Computer Science*, pages 348–359, 2002.
100. H. Zhang and X. He. Compact visibility representations and straight-line grid embedding of plane graphs. In *Proc. 8th International Workshop on Algorithms and Data Structures (WADS '03)*, volume 2748 of *LNCS*, pages 493–504. Springer, 2003.