Advances in Graph Drawing*

Ashim Garg and Roberto Tamassia

Department of Computer Science
Brown University
Providence, RI 02912-1910
{ag,rt}@cs.brown.edu

Abstract. Graph drawing addresses the problem of constructing geometric representations of abstract graphs and networks. It is an emerging area of research that combines flavors of topological graph theory and computational geometry. The automatic generation of drawings of graphs has important applications in key computer technologies such as software engineering, database design, visual interfaces, and computer-aided-design. This paper surveys recent results of the authors on graph drawing and overviews various research trends in the area.

1 Introduction

Graph drawing addresses the problem of constructing geometric representations of abstract graphs and networks. It is an emerging area of research that combines flavors of topological graph theory and computational geometry. The automatic generation of drawings of graphs has important applications in key computer technologies such as software engineering, database design, visual interfaces, and computer-aided-design.

Research on graph drawing has been especially active in the last decade. The latest version of the annotated bibliography on graph drawing algorithms, by G. Di Battista, P. Eades, R. Tamassia and I.G. Tollis [18] lists more than 300 papers, and the *Graph Drawing '93* workshop [17] gathered 80 scientists from 19 countries.

In this paper we survey recent results of the authors on graph drawing and overview various research trends in the area. We do not aim at a comprehensive coverage of recent literature on the subject (see [18] for an up-to-date bibliography). Rather, we focus on specific problems related to the authors' own research, which are of theoretical and practical interest, and on which substantial progress has been recently made.

The rest of this paper is organized as follows. Basic definitions are given in Section 2. Section 3 discusses trade-offs between various properties of drawings. Section 4 deals with upward planarity testing. Techniques for dynamic graph drawing are presented in Section 5. Finally, work in progress on a new visual approach to graph drawing is discussed in Section 6.

* Research supported in part by the National Science Foundation under grant CCR-9007851, by the U.S. Army Research Office under grant DAAL03-91-G-0035, and by the Office of Naval Research and the Advanced Research Projects Agency under contract N00014-91-J-4052, ARPA order 8225.

2 Definitions

In this section we review basic graph drawing terminology. Various graphic standards have been proposed for the representation of graphs in the plane. Usually, vertices are represented by points, and each edge (u, v) is represented by a simple open Jordan curve joining the points associated with the vertices u and v. A drawing such that each edge is represented by a polygonal chain is a *polyline* drawing. There are two common special cases of this standard. A *straight-line* drawing maps each edge into a straight-line segment. An *orthogonal* drawing maps each edge into a chain of horizontal and vertical segments. A polyline drawing is a *grid* drawing if the vertices and the bends of the edges have integer coordinates. The *angular resolution* of a straight-line drawing is the smallest angle formed by two edges incident on the same vertex. This definition can be extended to polyline drawings by considering also the angles formed by the bends.

A drawing is *planar* if no two edges intersect. A drawing of a digraph is *upward* if every edge is monotonically nondecreasing in the y-direction. A digraph is *upward planar* if it admits a planar upward drawing. Planarity and acyclicity are necessary but not sufficient conditions for upward planarity.

A graph drawing algorithm reads as input a combinatorial description of a graph G, and produces as output a drawing of G according to a given graphic standard.

3 Trade-Offs in Graph Drawing

Many graph drawing problems can be formalized as multi-objective optimization problems. E.g., construct a drawing with minimum area and minimum number of crossings. Hence, trade-offs are inherent in graph drawing. Several interesting trade-offs have been recently discovered relating important properties of drawings, such as area, angular resolution, number of crossings, number of bends, and upwardness (for digraphs).

3.1 Upward Planar Drawings of Digraphs

Di Battista, Tamassia, and Tollis [23] study the problem of constructing upward planar polyline drawings with the the objective of achieving small area and small number of bends. Their results, summarized below, show that these two properties cannot be simultaneously attained, and that the area can become exponential if a straight-line drawing (i.e., a drawing the minimum number of bends) is required:

- There exists a family G_n of planar digraphs such that G_n has $2n + 2$ vertices, and a planar straight-line upward drawing of G_n has area $\Omega(2^n)$.
- Let G be an upward planar digraph with n vertices. A planar polyline upward grid drawing of G with $O(n^2)$ area, at most one bend per edge, and at most $2n - 5$ bends in total can be constructed in $O(n)$ time.

Note that if we relax the upwardness requirement, then area $O(n^2)$ is asymptotically optimal in the worst case, and an $O(n^2)$-area planar straight-line drawing can be constructed in $O(n)$ time [8, 14, 15, 48, 33].

3.2 Upward Planar Drawings of Trees

Garg, Goodrich, and Tamassia [27] study the aforementioned problem of constructing upward planar polyline drawings for the class of rooted trees (where the edges are directed towards the root). Again, they aim at achieving small area and small number of bends. Their results, summarized below, show that the area of the drawing depends on the restrictions on the number and type of bends, and on the possibility of changing the left-to-right order of the children.

- A bounded-degree rooted tree with n vertices admits a planar polyline upward grid drawing with $O(n)$ area, $O(n^{\alpha})$ width, and $O(n^{1-\alpha})$ height (for any pre-specified constant α such that $0 < \alpha < 1$) that can be constructed in $O(n)$ time. Also, the above area bound is asymptotically optimal in the worst case.
- A binary tree with n vertices admits a planar orthogonal upward grid drawing with area $O(n \log\log n)$ that can be constructed in $O(n)$ time. Also, the above area bound is asymptotically optimal in the worst case.
- An ordered bounded-degree rooted tree with n vertices admits an $O(n \log n)$-area planar polyline upward grid drawing that preserves the left-to-right ordering of the children of each vertex. The drawing can be constructed in $O(n)$ time. Also, the above area bound is asymptotically optimal in the worst case.

Previously, Crescenzi, Di Battista, and Piperno [10] considered *strictly upward* grid drawings, where the vertices have integer coordinates, and the parent of a vertex has y-coordinate strictly greater than the ones of its children. They show that a rooted tree with n vertices, admits a strictly upward planar straight-line grid drawing with $O(n \log n)$ area, $O(n)$ width, and $O(\log n)$ height that can be constructed in $O(n)$ time. Also, they show that a complete binary tree or a Fibonacci tree with n vertices admits a planar straight-line strictly upward grid drawing with $O(n)$ area, and $O(\sqrt{n})$ width and height, that can be constructed in $O(n)$ time. The above area bounds are asymptotically optimal in the worst case.

Other interesting trade-offs arise if we relax the upwardness requirement. As shown in [39, 53], a binary tree with n vertices admits an $O(n)$-area planar orthogonal grid drawing. However, Brent and Kung [7] show that if the leaves of a complete binary tree with n vertices are constrained to be on the convex hull of the drawing, then the drawing needs $\Omega(n \log n)$ area.

3.3 Angular Resolution

Formann, Hagerup, Haralambides, Kaufmann, Leighton, Simvonis, Welzl, and Woeginger [26] study the angular resolution of (generally nonplanar) straight-line drawings of various classes of graphs. In particular, they show that a degree-d graph with n vertices has a straight-line drawing with angular resolution $\Omega(1/d^2)$ and area $O(d^6 n)$, and every degree-d planar graph with n vertices has a straight-line grid drawing with angular resolution $\Omega(1/d)$ and area $O(d^3 n)$.

Further work has been recently done on the angular resolution of planar drawings. Malitz and Papakostas [41] show that a degree-d planar graph admits a planar straight line drawing with angular resolution $\Omega(1/7^d)$. Kant [34] shows that a degree-d triconnected planar graph with n vertices admits a planar polyline grid drawing with angular resolution $\Omega(1/d)$, $O(n^2)$ area and $O(n)$ bends, and that testing whether a biconnected planar graph admits a planar straight-line drawing with angular resolution greater than or equal to a given constant is NP-hard. Di Battista and Vismara [24] give an $O(n)$-time algorithm for testing whether a given assignment of angles to a maximal planar graph can be achieved in a planar straight-line drawing.

Garg and Tamassia [28] show a continuous trade-off between the area and the angular resolution of planar drawings:

– There exists a constant $c > 1$ and a family G_n^d of planar graphs such that G_n^d has n vertices and degree d, and any planar straight-line drawing of G_n^d with angular resolution ρ has area $\Omega(c^{\rho n})$.

In particular, the above result implies that there exist bounded-degree planar graphs that require exponential area in any planar straight-line drawing with bounded resolution.

3.4 Visibility Representations of Trees

The concept of *visibility* (see, e.g., [45, 43, 49]) plays an important role in computational geometry, and arises in art gallery problems, motion planning, and graphics. Given a set of disjoint objects in the plane (e.g., points, lines, rectangles), two objects of are said to be *1-visible* if they can be joined by a vertical segment that does not intersect any other object. A *1-visibility representation* of a graph maps the vertices to disjoint horizontal segments such that any two adjacent vertices are 1-visible. It is easy to show that a graph admits a 1-visibility representation only if it is planar. A 1-visibility representation of a digraph is said to be upward if for every edge (u, v), the vertex-segment of u is placed below the vertex-segment of v.

We distinguish three types of visibility representations. In a *weak* 1-visibility representation, adjacent vertices are associated with visible vertex-segments. However, visible vertex-segments are not necessarily associated with adjacent vertices. Tamassia and Tollis [51] and Rosenstiehl and Tarjan [47] show that every planar graph admits a weak 1-visibility representation. In *strong* and ϵ-1-visibility representations, two vertex-segments are 1-visible *if and only if* their associated vertices are adjacent. The two representation differ in that an ϵ-1-visibility representation allows open or semi-open vertex-segments (i.e., which may or may not contain one or both endpoints), while a strong 1-visibility representation allows only closed vertex-segments (i.e., with both endpoints). Tamassia and Tollis [51] and Wismath [54] characterize the class of graphs that admit an ϵ-1-visibility representation, and provide a linear-time algorithm for testing membership in this class. Tamassia and Tollis [51] also show that every 4-connected planar graph has a strong 1-visibility representation. However, Andreae [1] proves that deciding whether a graph admits a 1-strong visibility representation is NP-complete.

Algorithms for constructing in $O(n)$ time an $O(n^2)$-area weak 1-visibility representation of an n-vertex planar graph are given by Tamassia and Tollis [51] and by Rosenstiehl and Tarjan [47]. Kant [34] shows that a weak 1-visibility representation of an n-vertex planar graph with area at most $(\lfloor \frac{3}{2}n \rfloor - 3) \times (n-1)$ can be constructed in $O(n)$ time.

In *2-visibility representations* of graphs the vertices are represented by disjoint isothetic rectangles in the plane, and adjacent vertices are associated with rectangles that are visible in the vertical or horizontal direction. As for the 1-visibility case, we distinguish *weak*, *strong*, and ϵ-2-visibility representations. Kirkpatrick and Wismath [38] prove that every planar graph admits an ϵ-2-visibility representation.

Kant, Liotta, Tamassia, and Tollis [35] study the area requirement of 1- and 2-visibility representations of trees. Their results show how the area requirement varies with the specific type of visibility representation.

The following definitions are needed: Let T be a free tree, and let v be a vertex of T. A *subtree* of v in T is a subtree of the rooted tree obtained by rooting T at v. Let $T_1, T_2, \cdots, T_d$ be the subtrees of v sorted by nonincreasing height (i.e., $h(T_1) \geq h(T_2) \geq \cdots \geq h(T_d)$). We call $h(T_k)$ the *k-th height* of v. A subtree of v with height equal to the k-th height of v is called a *k-tallest subtree* of v. If T has degree at least 3, we call *critical height* of T the maximum third-height of any vertex of T.

- Let T be a rooted tree with n vertices, ℓ leaves and height h. The area required by an upward 1-visibility representation of T is $\Omega(\ell \cdot h)$. Also, an upward 1-visibility representation of T with $O(\ell \cdot h)$ area can be computed in $O(n)$ time.
- Let T be a free tree with n vertices, ℓ leaves, and critical height h^*. The area required by a strong or ϵ-1-visibility representation of T is $\Omega(n + \ell \cdot h^*)$. Also, a strong or ϵ-1-visibility representation of T with $O(n+\ell \cdot h^*)$ area can be computed in $O(n)$ time.
- Let T be a free tree with n vertices and ℓ leaves. The area required by a strong or ϵ-2-visibility representation of T is $\Omega(n \cdot \ell)$. Also, a strong or ϵ-2-visibility representation of T with $O(n \cdot \ell)$ area can be computed in $O(n)$ time.

4 Upward Planarity Testing

Testing upward planarity and constructing upward planar drawings of digraphs is important for displaying hierarchical structures, such order diagrams, subroutine-call graphs, is-a hierarchies, and organization charts. While planarity testing can be done in linear time [30, 40, 6, 16], the complexity of upward planarity testing has been until very recently an open problem.

Combinatorial results on upward planarity of covering digraphs of lattices were first given in [37, 44]. The interplay between upward planarity and ordered sets is surveyed in [46]. Lempel, Even, and Cederbaum [40] relate the planarity of biconnected undirected graphs to the upward planarity of *st*-digraphs.

A combinatorial characterization of upward planar digraphs is provided in [36, 22]: namely, a digraph is upward planar if and only if it is a subgraph of a planar *st-digraph*. Di Battista, Liu, and Rival [21] shown that every planar bipartite digraph is upward planar. Bertolazzi, Di Battista, Liotta, and Mannino [3, 4] give a polynomial-time algorithm for testing upward planarity of triconnected digraphs and digraphs with a fixed embedding. Concerning single-source digraphs, Thomassen [52] characterizes upward planarity in terms of forbidden circuits. Hutton and Lubiw [31] combine Thomassen's characterization with a decomposition scheme to test upward planarity of an n-vertex single-source digraph in $O(n^2)$ time.

Bertolazzi, Di Battista, Mannino, and Tamassia [5] give optimal algorithms for upward planarity testing of single-source digraphs:

- Upward planarity testing of a single-source digraph with n vertices can be done in $O(n)$ time, and in $O(\log n)$ time on a CRCW PRAM with $n \log\log n/\log n$ processors.

Garg and Tamassia [29] have recently settled the open question of the complexity of upward planarity testing:

- Upward planarity testing is NP-complete.

5 Dynamic Graph Drawing

The motivation for investigating dynamic graph drawing algorithms arises when very large graphs need to be visualized in a dynamic environment, where vertices and edges are inserted and deleted, and subgraphs are displayed. Several graph manipulation systems allow the user to interactively modify a graph; hence, techniques that support fast restructuring of the drawing are very useful. Also, it is important that the dynamic drawing algorithm does not alter drastically the structure of the drawing after a local modification of the graph. In fact, human interaction requires a "smooth" evolution of the drawing.

Cohen, Di Battista, Tamassia, Tollis, and Bertolazzi [9] present dynamic algorithms for drawing planar graphs under a variety of drawing standards. They consider straight-line, polyline, grid, upward, and visibility drawings together with aesthetic criteria that are important for readability, such as the display of planarity, symmetry, and reachability. Also, they provide techniques that are especially tailored for important subclasses of planar graphs such as trees and series-parallel digraphs. Their dynamic drawing algorithms have the important property of performing "smooth updates" of the drawing. Their contributions are summarized as follows:

- A model for dynamic graph drawing is presented that uses a "query/update" setting, where an *implicit* representation of the drawing of a graph is maintained such that the following operations can be efficiently performed: *Drawing queries* that return the drawing of a subgraph consistent with the overall drawing of the graph. *Window queries* that return the portion of the drawing inside a query rectangle. *Point-location queries* in the subdivision of the plane induced by the drawing. *Update operations*, e.g., insertion and deletion of vertices and edges or replacement of an edge by a graph, which modify the implicit representation of the drawing accordingly.
- There exists a dynamic algorithm for maintaining a planar upward straight-line $O(n^2)$-area grid drawing of a rooted tree with n vertices. The memory space used is $O(n)$. Updates and point-location queries take $O(\log n)$ time. Drawing queries take time $O(k + \log n)$ for a subtree and $O(k \log n)$ for an arbitrary subgraph, and window queries take time $O(k \log n)$, where k is the output size.

- There exists a dynamic algorithm for maintaining a planar upward straight-line $O(n^2)$-area grid drawing a of an n-vertex series-parallel digraph. The memory space used is $O(n)$, updates take $O(\log n)$ time, drawing queries take time $O(k + \log n)$ for a series-parallel subgraph and $O(k \log n)$ for an arbitrary subgraph, point location queries take $O(\log n)$ time, and window queries take $O(k \log^2 n)$ time, where k is the output size.
- There exist dynamic algorithms for maintaining various types of planar drawings of a planar st-digraph with n vertices, including polyline upward drawings, and visibility representations. The drawings occupy $O(n^2)$ area. The algorithms use $O(n)$ memory space and support updates in $O(\log n)$ time.
- There exist semi-dynamic algorithms for maintaining various types of planar drawings of a biconnected planar graph with n vertices, including polyline drawings and visibility representations. The algorithms uses $O(n)$ memory space and support insertions in $O(\log n)$ amortized time (worst-case for insertions that preserve the embedding). Drawing queries take $O(k \log n)$ time, where k is the output size.

6 A Visual Approach to Graph Drawing

Work in graph drawing has traditionally focused on *algorithmic* approaches, where the drawing of the graph is generated according to a prespecified set of general rules or aesthetic criteria (such as planarity or area minimization) that are embodied in an algorithm. Perhaps the most sophisticated graph drawing system based on the algorithmic approach is *Diagram Server* [2, 20, 19], which maintains a large database of graph drawing algorithms and is able to select the one best suited to the needs of the user.

The algorithmic approach is computationally efficient, however, it does not naturally support *constraints*, i.e., requirements that the user may want to impose on the drawing of a *specific* graph (e.g., clustering or aligning a given set of vertices). Previous work by Tamassia *et al.* [50] has shown the importance of satisfying constraints in graph drawing systems, and has demonstrated that a limited constraint satisfaction capability can be added to an existing drawing algorithm.

Recently, several attempts have been made at developing languages for the specification of constraints and at devising techniques for graph drawing based on the resolution of systems of constraints (see, e.g., [32, 42]). Current constraint-based systems have three major drawbacks: (*i*) The specification of constraints is made through a detailed enumeration of facts from a fixed set of predicates, expressed in Prolog or with a set-theoretic notation. (*ii*) Natural requirements, such as planarity, need complicated constraints to be expressed. (*iii*) General constraint-solving systems are computationally inefficient. The above discussion indicates the need for a language to specify constraints that reconciles expressiveness with efficiency.

Cruz, Tamassia and Van Hentenryck [13] are developing a new technique for the visual specification of constraints in graph drawing systems. Visual languages can provide a natural and user-friendly way to express the layout of a graph. For this purpose, they plan to design a variation of DOODLE, a visual language for the specification of the display of facts in an object-oriented database [11, 12].

Recent work by Eades and Lin [25] has similar objectives, but is not based on a visual specification.

References

1. T. Andreae. Some results on visibility graphs. *Discrete Applied Mathematics*, 40:5–17, 1992.
2. M. Beccaria, P. Bertolazzi, G. Di Battista, and G. Liotta. A tailorable and extensible automatic layout facility. In *Proc. IEEE Workshop on Visual Languages (VL'91)*, pages 68–73, 1991.
3. P. Bertolazzi and G. Di Battista. On upward drawing testing of triconnected digraphs. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 272–280, 1991.
4. P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, to appear.
5. P. Bertolazzi, G. Di Battista, C. Mannino, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. In *1st Annual European Symposium on Algorithms (ESA '93)*, Lecture Notes in Computer Science. Springer-Verlag, 1993.
6. K. Booth and G. Lueker. Testing for the consecutive ones property interval graphs and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.*, 13:335–379, 1976.
7. R.P. Brent and H.T. Kung. On the area of binary tree layouts. *Information Processing Letters*, 11:521–534, 1980.
8. M. Chrobak and T. H. Payne. A linear time algorithm for drawing a planar graph on a grid. Technical Report UCR-CS-90-2, Dept. of Math. and Comput. Sci., Univ. California Riverside, 1990.
9. R. F. Cohen, G. Di Battista, R. Tamassia, I. G. Tollis, and P. Bertolazzi. A framework for dynamic graph drawing. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 261–270, 1992.
10. P. Crescenzi, G. Di Battista, and A. Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Computational Geometry: Theory and Applications*, 2:187–200, 1992.
11. I. F. Cruz. DOODLE: A visual language for object-oriented databases. In *Proc. ACM SIGMOD*, pages 71–80, 1992.
12. I. F. Cruz. Using a visual constraint language for data display specification. In P. C. Kanellakis, J.-L. Lassez, and V. Saraswat, editors, *First Workshop on Principles and Practice of Constraint Programming*, Newport, RI, April 1993.
13. I. F. Cruz, R. Tamassia, and P. Van Hentenryk. A visual approach to graph drawing. In *Graph Drawing '93 (Proc. ALCOM Workshop on Graph Drawing)*, Paris, France, September 1993.
14. H. de Fraysseix, J. Pach, and R. Pollack. Small sets supporting Fary embeddings of planar graphs. In *Proc. 20th Annu. ACM Sympos. Theory Comput.*, pages 426–433, 1988.
15. H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990.
16. H. de Fraysseix and P. Rosenstiehl. A depth-first-search characterization of planarity. *Annals of Discrete Mathematics*, 13:75–80, 1982.
17. G. Di Battista, P. Eades, H. de Fraysseix, P. Rosenstiehl, and R. Tamassia. *Graph Drawing '93 (Proc. ALCOM Int. Workshop on Graph Drawing)*. 1993. Available via anonymous `ftp` from `wilma.cs.brown.edu`, `/pub/papers/compgeo/gd93-v2.tex.Z`.
18. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. Preprint, Dept. Comput. Sci., Brown Univ., Providence,

RI, November 1993. To appear in *Comput. Geom. Theory Appl.* Preliminary version available via anonymous `ftp` from `wilma.cs.brown.edu`, `gdbiblio.tex.Z` and `gdbiblio.ps.Z` in `/pub/papers/compgeo`.

19. G. Di Battista, A. Giammarco, G. Santucci, and R. Tamassia. The architecture of diagram server. In *Proc. IEEE Workshop on Visual Languages (VL'90)*, pages 60–65, 1990.
20. G. Di Battista, G. Liotta, M. Strani, and F. Vargiu. Diagram server. In *Advanced Visual Interfaces (Proceedings of AVI '92)*, volume 36 of *World Scientific Series in Computer Science*, pages 415–417, 1992.
21. G. Di Battista, W. P. Liu, and I. Rival. Bipartite graphs upward drawings and planarity. *Inform. Process. Lett.*, 36:317–322, 1990.
22. G. Di Battista and R. Tamassia. Algorithms for plane representations of acyclic digraphs. *Theoret. Comput. Sci.*, 61:175–198, 1988.
23. G. Di Battista, R. Tamassia, and I. G. Tollis. Area requirement and symmetry display of planar upward drawings. *Discrete Comput. Geom.*, 7:381–401, 1992.
24. G. Di Battista and L. Vismara. Angles of planar triangular graphs. In *Proc. 25th Annu. ACM Sympos. Theory Comput. (STOC 93)*, pages 431–437, 1993.
25. P. Eades and T. Lin. Algorithmic and declarative approaches to aesthetic layout. In *Graph Drawing '93 (Proc. ALCOM Workshop on Graph Drawing)*, Paris, France, September 1993.
26. M. Formann,
T. Hagerup, J. Haralambides, M. Kaufmann, F. T. Leighton, A. Simvonis, E. Welzl, and G. Woeginger. Drawing graphs in the plane with high resolution. In *Proc. 31th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 86–95, 1990.
27. A. Garg, M. T. Goodrich, and R. Tamassia. Area-efficient upward tree drawings. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 359–368, 1993.
28. A. Garg and R. Tamassia. Angular resolution of planar drawings. Technical report, Brown Univ., Dept. of Computer Science, 1993.
29. A. Garg and R. Tamassia. On the complexity of upward planarity testing. Technical report, Brown Univ., Dept. of Computer Science, 1993.
30. J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.
31. M. D. Hutton and A. Lubiw. Upward planar drawing of single source acyclic digraphs. In *Proc. 2nd ACM-SIAM Sympos. Discrete Algorithms*, pages 203–211, 1991.
32. T. Kamada. *Visualizing Abstract Objects and Relations.* World Scientific Series in Computer Science, 1989.
33. G. Kant. Drawing planar graphs using the *lmc*-ordering. In *Proc. 33th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 101–110, 1992.
34. G. Kant. A more compact visibility representation. In *Proc. 19th Internat. Workshop Graph-Theoret. Concepts Comput. Sci. (WG'93)*, 1993.
35. G. Kant, G. Liotta, R. Tamassia, and I. Tollis. Area requirements of visibility representations of trees. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 192–197, Waterloo, Canada, 1993.
36. D. Kelly. Fundamentals of planar ordered sets. *Discrete Math.*, 63:197–216, 1987.
37. D. Kelly and I. Rival. Planar lattices. *Canad. J. Math.*, 27(3):636–665, 1975.
38. D. G. Kirkpatrick and S. K. Wismath. Weighted visibility graphs of bars and related flow problems. In *Proc. 1st Workshop Algorithms Data Struct.*, volume 382 of *Lecture Notes in Computer Science*, pages 325–334. Springer-Verlag, 1989.
39. C. E. Leiserson. Area-efficient graph layouts (for VLSI). In *Proc. 21st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 270–281, 1980.

40. A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. In *Theory of Graphs: Internat. Symposium (Rome 1966)*, pages 215–232, New York, 1967. Gordon and Breach.
41. S. Malitz and A. Papakostas. On the angular resolution of planar graphs. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 527–538, 1992.
42. J. Marks. A formal specification for network diagrams that facilitates automated design. *Journal of Visual Languages and Computing*, 2:395–414, 1991.
43. J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, New York, NY, 1987.
44. C. Platt. Planar lattices and planar graphs. *J. Combin. Theory Ser. B*, 21:30–39, 1976.
45. F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction.* Springer-Verlag, New York, NY, 1985.
46. I. Rival. Graphical data structures for ordered sets. In I. Rival, editor, *Algorithms and Order*, pages 3–31. Kluwer Academic Publishers, 1989.
47. P. Rosenstiehl and R. E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete Comput. Geom.*, 1(4):343–353, 1986.
48. W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 138–148, 1990.
49. T. C. Shermer. Recent results in art galleries. *Proc. IEEE*, 80(9):1384–1399, September 1992.
50. R. Tamassia, G. Di Battista, and C. Batini. Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst. Man Cybern.*, SMC-18(1):61–79, 1988.
51. R. Tamassia and I. G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.*, 1(4):321–341, 1986.
52. C. Thomassen. Planar acyclic oriented graphs. *Order*, 5(4):349–361, 1989.
53. L. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Comput.*, C-30(2):135–140, 1981.
54. S. K. Wismath. Characterizing bar line-of-sight graphs. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 147–152, 1985.

This article was processed using the LaTeX macro package with LLNCS style