Report on the International Work Meeting on Graph Drawing

Marino (Rome), Italy, June 4–5 1992

Giuseppe Di Battista[*] *Roberto Tamassia*[†]

dibattista@iasi.rm.cnr.it rt@cs.brown.edu

Research on graph drawing has received increasing attention in the last years. This area combines flavors of topological graph theory, graph algorithms, and computational geometry, and has a wide range of applications including computer aided software engineering, project management, network design, and visual interfaces.

A work meeting on graph drawing has taken place on June 4–5 at the Hotel Helio Cabala in Marino, near Rome. The goals of this meeting have been: (1) bringing together researchers in graph drawing to assess the state of the art in the area by means of survey talks; and (2) developing further research activities, including workshops, schools, and conferences. While the survey talks did not cover all aspects of graph drawing, they identified fundamental combinatorial, algorithmic, and practical issues of the area. Due to the nature of the meeting, the number of participants was kept small.

The main financial support was provided by the *Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo* of the Italian National Research Council. Additional support came from the *Istituto di Analisi dei Sistemi ed Informatica* of the Italian National Research Council, by the *Association pour the Developpement de l'Informatique dans le Sciences de l'Homme*, and by the *Basic Research Action* of the EC (ESPRIT II ALCOM).

The organizing committee consisted of Giuseppe Di Battista (Univ. Rome, Italy), Peter Eades (Univ. Newcastle, Australia), Pierre Rosenstiehl (EHESS, France), and Roberto Tamassia (Brown Univ., USA). Local arrangements were handled by Paola Bertolazzi (IASI-CNR, Italy).

The complete program of the meeting and the abstracts of the talks are given below:

[*]Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, 00198 Rome, Italy.

[†]Department of Computer Science, Brown University, Providence, R.I. 02912–1910.

Program

Thursday, June 4

9:00–9:15	*Welcome Address*
9:15–10:15	*Applications of Graph Drawing to Software Engineering*, Carlo Batini (Univ. Rome, Italy)
10:15–11:15	*Algorithms for Drawing Trees*, Peter Eades (Univ. Newcastle, Australia)
11:15–11:45	**Coffee Break**
11:45–12:45	*Extremal Problems in Graph Drawing*, Janos Pach (Hungarian Academy of Sciences)
12:45–15:00	**Lunch**
15:00–16:00	*Visibility Representations of Planar Graphs*, Ioannis G. Tollis (Univ. of Texas at Dallas, USA)
16:00–16:30	**Coffee Break**
16:30–18:30	*Partial Orders for Planarity and Drawings*, Hubert de Fraysseix, Patrice de Mendez, and Pierre Rosenstiehl (EHESS, France)
19:30	**Dinner**
21:30	Business Meeting

Friday, June 5

9:00–10:00	*Drawing Compound Digraphs and its Application to an Idea Organizer*, Kozo Sugiyama (Fujitsu, Japan)
10:00–11:00	*Area Requirements*, Giuseppe Di Battista (Univ. Rome, Italy)
11:00–11:30	**Coffee Break**
11:30–12:30	*Angular Resolution of Straight-Line Drawings*, Michael Kaufmann (Max Plank Inst., Germany)
12:30–15:00	**Lunch**
15:00–16:00	*Algorithms for Orthogonal Drawings*, Roberto Tamassia (Brown Univ., USA)
16:00–16:30	**Coffee Break**
16:30–17:15	*Circle Packing Representation in the Plane and Other Surfaces*, Bojan Mohar, (Univ. of Ljubjana, Slovenia)
17:15–17:45	*Dynamic Graph Drawing*, Robert F. Cohen (Brown Univ., USA)
17:45–18:15	*A New Method for Planar Graph Drawings on a Grid*, Goos Kant (Utrecht Univ., The Netherlands)
18:15–18:45	*An Automatic Layout Facility*, Giuseppe Liotta (Univ. Rome, Italy)
19:30	**Dinner**

Abstracts

Applications of Graph Drawing to Software Engineering
Carlo Batini (Univ. Rome, Italy)

In several application areas of software engineering intermediate and final products of the design activity are represented by means of diagrams. Diagrams present several advantages over other representations, since they are an aid to clear thinking (a diagram is better that a thousand words). Diagrams are external representations of organization charts, data and function schemas, structure of procedure calls, pieces of code, etc. A diagram is characterized by a meaning (the "piece of reality" it draws), and a syntax, expressed by: a graphic grammar, that defines the types of symbols and connections allowed, aesthetic criteria adopted in drawing the diagram. an underlying diagram model (e.g. hierarchical, planar, connected, etc). In software engineering tools, diagram are manipulated by several functions, such as beautification, modification, refinement, merging of several diagrams, animation, etc. Each one of such functions is characterized by specific algorithms needed for the final activity of diagram drawing. E.g. when we beautify a diagram we are interested to achieve some aesthetic criteria that were previously not respected; when we refine a diagram into a more detailed one, we are interested to maintain monotonicity of shape. Producing incrementally such a library by dynamically tailoring and merging paradigmatic algorithms, results in a new and interesting research area.

Algorithms for Drawing Trees
Peter Eades (Univ. Newcastle, Australia)

Several conventions for drawing rooted trees are discussed: classical drawings, tip-over drawings, inclusion drawings, and h-v drawings. In each convention, a number of algorithms and complexity results are available. For free trees (no specific root), algorithms are mostly drawn from folklore. Radial drawings, spring drawings, and orthogonal drawings are discussed.

Extremal Problems in Graph Drawing
Janos Pach (Hungarian Academy of Sciences)

A geometric graph (a convex geometric graph) is a pair (V, E), where V is a set of points in the plane in general position (in convex position), and E is a set of segments connecting some points in V. I survey many recent results in geometric graph theory with special emphasis on the following questions. (A) What is the maximum number of edges a geometric graph with n vertices can have without containing a given forbidden configuration (e.g., k pairwise disjoint edges o k pairwise crossing edges, etc.)? (B) What is the largest number $f(n)$ with the property that any geometric graph of n vertices contains either $f(n)$ pairwise disjoint edges or $f(n)$ pairwise crossing edges? (C) What is the maximum number of sides of a cell in a geometric graph with n vertices? Some generalizations to higher dimensions (for geometric hypergraphs) are also considered.

Visibility Representations of Planar Graphs
Ioannis G. Tollis (Univ. of Texas at Dallas, USA)

Interesting representations of graphs result from mapping vertices into horizontal segments and edges into vertical segments drawn between visible vertex-segments. (Two parallel segments of a given set are visible if they can be joined by a segment orthogonal to them, which does not intersect any other segment.) Such representations are called visibility representations. These representations find applications in VLSI layout, algorithm animation, visual languages and CASE tools. We consider three types of visibility representations: (*i*) Weak-visibility representation, if two vertices are adjacent then their corresponding segments must be visible. (*ii*) ϵ-visibility representation, where vertices can be represented by open, closed, or semiclosed horizontal segments in the plane such that two vertices are adjacent if and only if their associated segments are visible. (*iii*) Strong-visibility representation, where vertices are represented by closed horizontal segments such that two vertices are adjacent if and only if their corresponding segments are visible. We discuss various types of visibility representations of planar graphs on the plane, cylinder, and sphere.

Partial Orders for Planarity and Drawings
Hubert de Fraysseix, and Pierre Rosenstiehl (EHESS, France)

A bipolar orientation of a graph appears often in the algorithm literature as a first step for the generation of a particular drawing. Here the properties of bipolar orientations are systematically explored in terms of circuits, cocircuits, rank activities, Tutte polynomial, poset dimension, angle bipartition and max flow-min cut theorem. Efficient algorithms are described to list, generate or extend bipolar orientations for general graphs or plane ones, with or without constraints. (Joint work with Patrice de Mendez.)

Drawing Compound Digraphs and its Application to an Idea Organizer
Kozo Sugiyama (Fujitsu, Japan)

A compound digraph is a directed graph with both inclusion and adjacency edges and is widely used in diverse fields. In this talk, first, a heuristic method for automatically drawing a compound digraph is presented. Then, several graphic interface techniques which facilitate the method in organizing ideas such as direct manipulation, animation, incremental editing, fish-eyes (focusing, abridgements), and utilization of curves are discussed. Finally, it is emphasized that interface issues are important as well as algorithmic aspects of graph drawing. (Joint work with K. Misue.)

Area Requirements
Giuseppe Di Battista (Univ. Rome, Italy)

An *upward* drawing of an acyclic digraph is a planar straight-line drawing with the additional requirement that all the edges flow in the same direction, e.g., from bottom to top. The literature on the problem of constructing upward drawings of important classes of digraphs is surveyed. First, it is show that there is a family of binary trees with n vertices requiring $\Omega(n \log n)$ area for any upward drawing; moreover, that bound is tight, i.e. each binary tree with n vertices can be drawn with $O(n \log n)$ area. Second, motivated by the elegant H-tree layout algorithm for constructing non-upward drawings of complete binary trees, an algorithm is presented for constructing an upward drawing of a complete binary tree with n vertices in $O(n)$ area. This result is extended to the drawings of *Fibonacci trees*. Third, it is shown that the area requirement of upward drawings of series-parallel digraphs crucially depends on the choice of planar embedding. Also, parallel and sequential drawing algorithms are presented that are optimal with respect to both the time complexity and to the area achieved. Several results show that while series-parallel digraphs have a rather simple and well understood combinatorial structure, naive drawing strategies lead to drawings with exponential area, and clever algorithms are needed to achieve optimal area.

Angular Resolution of Straight-Line Drawings
Michael Kaufmann (Universität Passau and Max-Planck-Institut Saarbrücken, Germany)

The angular resolution of a drawing of graph G is the minimum angle between two incident edges that appears in the drawing. In this talk the state of the art is presented how to maximize the angular resolution. First the general technique by Formann et al. [FOCS'90] is reviewed, then a refinement for planar graphs is given. A recent paper of Malitz/Papakostas [STOC'92] shows that planar graphs can be drawn planar with minimum angle only depending on the maximum degree of the vertices. At the end, some aspect on the relationship of resolution and area are discussed.

Algorithms for Orthogonal Drawings
Roberto Tamassia (Brown Univ., USA)

An orthogonal drawing of a graph is such that the edges are represented by polygonal chains consisting of horizontal and vertical segments. The intermediate vertices of the chain (which are not vertices of the graph) are called bends. In this talk we survey algorithms for constructing planar orthogonal drawings. The main quality measures considered are the minimization of the number of bends and of the area of the drawing. The construction of planar orthogonal drawings has many important applications, including graph visualization, VLSI layout, facilities floorplanning, and communication by light or microwave. Given an embedded planar graph G with n vertices, a planar orthogonal drawing of G with the minimum number of bends can be computed in $O(n^2 \log n)$ time using network-flow techniques. Drawings with $O(n)$ bends can be constructed in $O(n)$ time using visibility representations. Also, there are families of graphs that require $\Omega(n)$ bends. Open problems include minimizing bends over all possible embeddings, and finding an efficient parallel algorithm for bend minimization.

Circle Packing Representation in the Plane and Other Surfaces
Bojan Mohar, (Univ. of Ljubjana, Slovenia)

The Andreev-Thurston circle packing theorem is generalized and improved in three ways. First, to arbitrary maps on closed surfaces. Second, we get the simultaneous circle packing of the map and its dual map so that, in the corresponding straight-line representations of the map and the dual, any two edges dual to each other cross at the right angle. The necessary and sufficient condition for a map to have such a primal-dual circle packing representation is that its universal cover is 3-connected (the map has no "planar" 2-separations). Finally, a polynomial time algorithm is obtained that given a map M and a rational number $\epsilon > 0$ finds an ϵ-approximation for the primal-dual circle packing representation of M. In particular, we get a polynomial time algorithm for geodesic convex representations of reduced maps on arbitrary surfaces.

Dynamic Graph Drawing
Robert F. Cohen (Brown Univ., USA)

Drawing graphs is an important problem that combines flavors of computational geometry and graph theory. Applications can be found in a variety of areas including circuit layout, network management, software engineering, and graphics. The main contributions of this paper can be summarized as follows: (*i*) We devise a model for dynamic graph algorithms, based on performing queries and updates on an implicit representation of the drawing, and we show its applications. (*ii*) We present several efficient dynamic drawing algorithms for trees, series-parallel digraphs, planar *st*-digraphs, and planar graphs. These algorithms adopt a variety of representations (e.g., straight-line, polyline, visibility), and update the drawing in a smooth way. (*iii*) We show that the implicit representation of the layout used by our algorithms for trees and series-parallel digraphs also supports point-location and window queries. (Joint work with P. Bertolazzi, G. Di Battista, R. Tamassia, and I.G. Tollis.)

A New Method for Planar Graph Drawings on a Grid
Goos Kant (Utrecht Univ., The Netherlands)

We introduce a new method to optimize the required area, minimum angle and number of bends of planar drawings of graphs on a grid. The main tool is a new type of ordering on the vertices and faces of triconnected planar graphs. With this method linear time and space algorithms can be designed for many graph drawing problems. (*i*) We show that every triconnected planar graph G can be drawn convexly with straight lines on an $(2n-4)\times(n-2)$ grid. (*ii*) If G has maximum degree four (three), then G can be drawn orthogonal with at most $\lfloor\frac{3n}{2}\rfloor+3$ (at most $\lfloor\frac{n}{2}\rfloor+1$) bends on an $n\times n$ grid ($\lceil\frac{n}{2}\rceil\times\lceil\frac{n}{2}\rceil$ grid, respectively). (*iii*) If G has maximum degree d, then G can be drawn planar on an $(2n-6)\times(3n-6)$ grid with minimum angle larger than $\frac{1}{d-2}$ radians and at most $5n-15$ bends. These results give in some cases considerable improvements over previous results, and give new bounds in other cases. Several other results, e.g. concerning visibility representations, are included.

An Automatic Layout Facility
Giuseppe Liotta (Univ. Rome, Italy)

An automatic layout facility is a tool that receives as input a graph-like structure and is able to produce a diagram that nicely represents such a structure. Because of the increasing number of systems that manage diagrams, automatic layout facilities and algorithms for graphs layout have been extensively studied in the last years. We present a new approach in designing an automatic layout facility. Our approach is based on a modular management of a collection of algorithms and on a tool that is able to automatically select the best algorithm for a given application. Such approach has been used for devising the automatic layout facility of Diagram Server, a network server that offers to its clients several facilities for managing diagrams. (Joint work with P. Bertolazzi and G. Di Battista.)