GraphAEL: Graph Animations with Evolving Layouts⋆
(System Demo)

C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler, and G. Yee

Department of Computer Science
University of Arizona
{cesim,harding,kobourov,wamplerk,gyee}@cs.arizona.edu

Abstract. `GraphAEL` extracts three types of evolving graphs from the Graph Drawing literature and creates 2D and 3D animations of the evolutions. We study citation graphs, topic graphs, and collaboration graphs. We also create difference graphs which capture the nature of change between two given time periods. `GraphAEL` can be accessed online at `http://graphael.cs.arizona.edu`.

1 Introduction

While static graphs arise in many applications, dynamic processes give rise to graphs that evolve through time. Such dynamic processes can be found in software engineering, internet/telecommunications traffic, and social networks, among others. Typically, the underlying graph structures are large, and depending on the time-granularity, may contain from a few to a large number of timeslices. The graph representing a particular timeslice may contain fewer/more vertices/edges than the one representing the previous timeslice, or it may differ in some other graph attributes, such as node-weights, edge-weights, or labels.

We describe algorithms and techniques for visualization of dynamic graph processes, using the evolution of the graph drawing literature as an example of such a process. `GraphAEL` is composed of three distinct components:

1. a relational database that stores and catalogs the data;
2. a graphical user interface (GUI) for querying the database and for graph creation;
3. a toolkit for interactive visualization of the resulting graphs.

We built a MySQL database and populated it with all the papers from the graph drawing proceedings in the period 1994-02. The database can be queried online via a GUI. In addition to standard queries and responses, we can generate graphs that evolve through time. In particular, we generate *citation graphs*, *co-authorship graphs*, and *topic graphs* at varying time-granularity. Each of these types comes in two flavors, *individual graphs* and *cumulative graphs*. To study the underlying graph processes, we also extract *difference graphs* which facilitate visual discovery of new trends and patterns. The evolution of the underlying graphs and the difference graphs can be studied via 2D and 3D animations.

2 Related Work

The visualization techniques for graphs that evolve through time are related to previous work in dynamic graph visualization. A number of papers use modifications of static graph

---

drawing algorithms [4, 13, 19]. Incremental graph drawing is used in DynaDag [18]. Bayesian decision theory, together with a force-directed method are used in [3] to model dynamic graphs. Sequences of graphs are considered in [7] in order to create smoother transitions. Special classes of graphs such as trees, series-parallel graphs and *st*-graphs have also been studied in dynamic models [5, 16].

More recently, several papers consider larger graphs that evolve through time. In [2] a system for visualizing network evolution is presented, in which each modification is shown in a separate layer of 3D representation with vertices common to two layers represented as columns connecting the layers. Along these lines, Collberg *et al* [6] describe a graph-based system for visualization of software evolution, which uses a modification a force-directed algorithm for visualization of large graphs [10]. Using a preliminary version of our system we analyzed the ACM digital library database, focusing on collaboration graphs and category graphs but the lack of citation data prevented us from extracting citation graphs [8].

3 The Relational Database

We gathered data about the graph drawing community, more specifically the nine years of published proceedings from the International Symposium on Graph Drawing. This data came from three sources: an XML file with information from the DBLP Computer Science Bibliography, a GML file used in the 2001 graph drawing contest [1], and the proceedings themselves. We designed a schema that maps the data to all necessary relations for creation of collaboration, topic, and citations graphs; see Fig. 1. The database schema consists of seven tables storing information about articles, authors, proceedings, and publishers. Rectangles represent entities, circles represent attributes, and diamonds represent binary relationships. The primary key for each entity is underlined. Each relationship is also subscripted to in oder to indicate whether the two entities share a one-to-one, one-to-many, or many-to-many relationship.
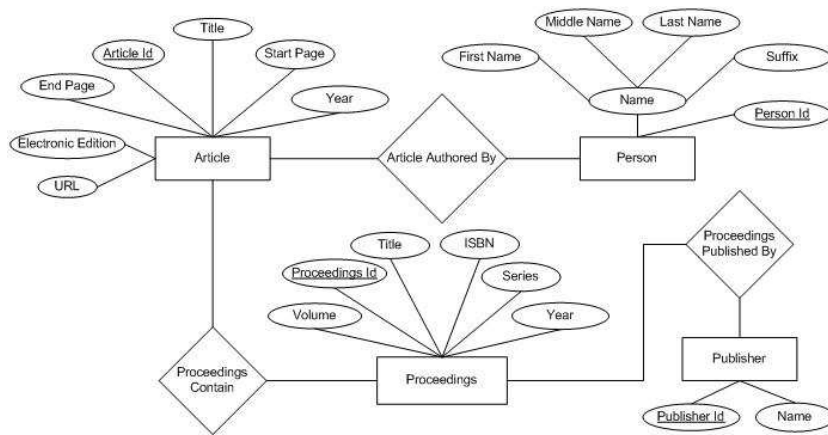

Fig. 1. The bibliographical database schema in Extended Entity-Relationship format.

We used MySQL version 4.0.12, a powerful open-source database system, to implement this schema. The XML file obtained from DBLP contains bibliographical information for
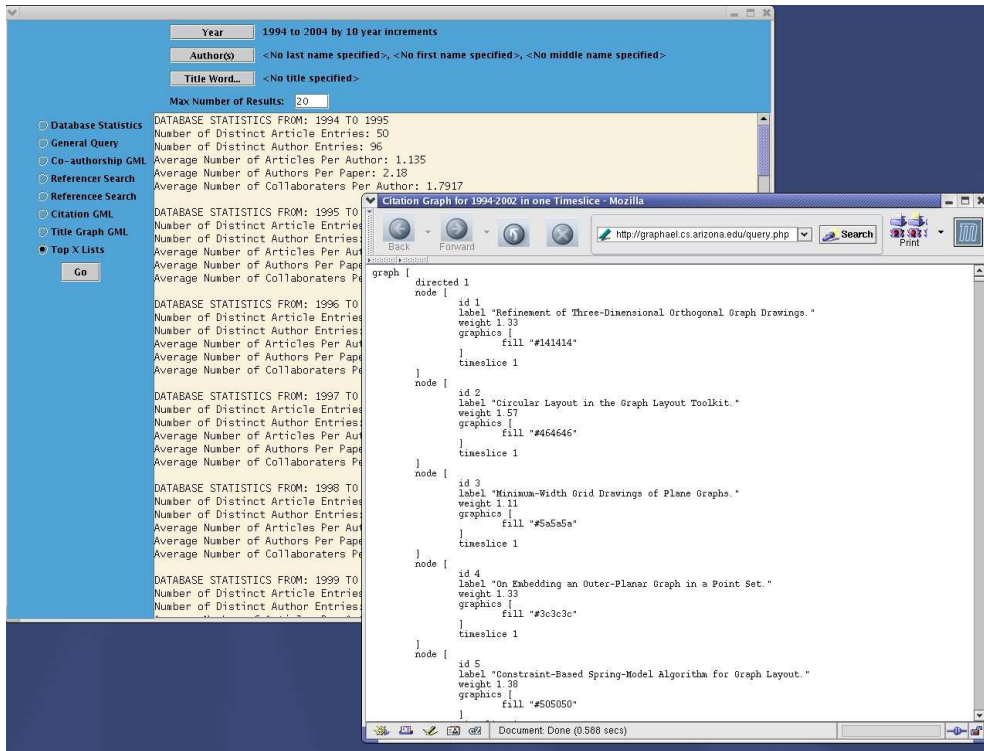
Fig. 2. The Graphical User Interface, shown here displaying a GML file.

every paper published in the GD proceedings. It does not, however, contain citation information. We built an XML parser to extract the relevant bibliographical data from DBLP. The citation data for the period 1994-00 was gathered from the the 2001 graph drawing contest GML file. Citation data for 2001 and 2002, was manually entered directly from the proceedings. We intend to update the database with new entries from the latest proceedings of the symposium on graph drawing, on an yearly basis. The database can be accessed and queried online at `http://graphael.cs.arizona.edu`.

4 Graphical User Interface

To interact with the database we created a graphical user interface; see Fig. 2. The GUI, written in Java, is available as both a web-based Java applet and a downloadable Java application. The applet version makes use of a PHP script as a servlet to maintain the security of the database server. The GUI allows users to query general dataset statistics, search for specific authors and papers, and query citation data by referencer and referencee. In addition, users can generate collaboration, topic, and citation graphs on the fly in the form of GML files with varying time-granularity. The applet opens a new browser window displaying the GML file, which can be saved locally. All of these queries can be limited in scope by date and the amount of results requested.

The top panel of the GUI contains buttons that allow the user to change the querying parameters. The left panel of the GUI contains buttons representing the different types of

3

results and GML files that can be obtained. The 'Go' button, activates a servlet that will then query the database located on the server. The server returns the results to the servlet which then hands them to the applet where they are displayed. The following queries are supported by the GUI:

- Database Statistics: lists data statistics, including number of article and authors;
- General Query: allows for search by author name, or title words;
- Co-authorship GML: creates a co-authorship graph, for the given parameters;
- Referencer Search: finds all papers that references a particular paper;
- Referencee Search: finds all papers that are referenced from a particular paper;
- Citation GML: creates a citation graph, for the given parameters;
- Topic Graph GML: creates a topic graph, for the given parameters;
- Top X Lists: lists the most productive authors, most collaborative authors, etc.;

While many of our results are obtained by directly querying the database, some of our more complex queries and the generation of GML files require the use of additional programs. These are written in Icon.

5 Interactive Visualization of Evolving Graphs

`GraphAEL` calculates the layout for evolving graphs quickly. The underlying graph drawing algorithm is a modification of the `GRIP` algorithm [10, 11]. Given a graph $G = (V, E)$, `GRIP` computes a hierarchical sequence of vertex filtrations given by a sequence of m subsets of V. Let V_i, V_j be two filtrations such that $i < j$, then $V_i \subset V_j$ and $V_m = V$. The graphs defined by this filtration sequence are laid out in order, with the layout of the graph at each level i providing a basis to guide the layout of the graph at level $i + 1$. In this way large-scale structures of the graph are quickly captured by the filtration levels with low indices, and the small-scale details are refined in the layout of the levels with high indices.

The layout of each filtration level is calculated with the force-directed placement method of Kamada-Kawai [14]. For a vertex v in G, the displacement of v is calculated by:

$$\boldsymbol{F}_{KK}(v) = \sum_{u \in N_i(v)} \left(\frac{\|pos[u] - pos[v]\|^2}{dist_G(u,v)^2 \cdot edgeLength^2} - 1 \right) (pos[u] - pos[v])$$

This equation is convenient because it calculates the displacement on each vertex based only upon its physical and graph distances from other vertices in the graph. Since in general, vertices in a particular filtration level are not adjacent to any other vertex in the level, but are instead connected to other vertices in the higher levels through a sequence of edges, using graph distances is a reasonable approach. Only in the last level of of filtration does the adjacency make sense, since the last level is the same as the initial graph. Thus at this level, it is possible to use force directed placement methods which depend on a notion of adjacency. The Fruchterman-Reingold method [9] is for calculating the displacement of a vertex v, $\boldsymbol{F}_{FR}(v) = \boldsymbol{F}_{a,FR} + \boldsymbol{F}_{r,FR}$, where the attractive and repulsive force vectors are given by:

$$\boldsymbol{F}_{a,FR} = \sum_{u \in Adj(v)} \frac{\|pos[u] - pos[v]\|^2}{edgeLength^2} (pos[u] - pos[v])$$

$$\boldsymbol{F}_{r,FR} = \sum_{u \in N_i(v)} s \frac{edgeLength^2}{\|pos[u] - pos[v]\|^2} (pos[v] - pos[u])$$

5.1 GraphAEL's Layout Algorithm

While `GRIP` does a decent job of laying out large static graphs, there is no appropriate mechanism to assign the vertices and edges of a graph attributes, such as weights, which might affect the layout. This is a major drawback in visualizing certain relations and even more so in visualizing evolving graphs. For instance, in visualizing the GD citation graph, it is useful to have a notion of both the weight of a vertex, (which might represent the number of times a paper is cited), and a temporal notion (which might represent the year the paper was written). In contrast, `GraphAEL` supports the layout of graphs with node-weights and edge-weights, as well as the notion of a *timeslice* which is used to visualize graphs with a temporal component (evolving graphs).

`GraphAEL` tries to place heavy nodes well away from each other and to place vertices connected by heavy edges, closer to each other. Appropriate modifications to the force-directed algorithm were made to accommodate these characteristics. Each edge in a graph is given an ideal length that it will attempt to attain, although this is not always possible. An edge e of weight w_e connecting vertices u, v of weight w_u, w_v respectively is given an ideal length of $\sqrt{w_u \cdot w_v}/w_e$.

Since the Kamada-Kawai method is used to determine parts of the layout, it is necessary to come up with a modified graph distance for weighted graphs. Because of the computational or space requirements of calculating the effects of all paths between two vertices, or of computing the shortest weighted path between them, `GraphAEL` uses an approximation. Let $p_1, p_2, \ldots, p_n$ be the sequence of vertices in the shortest unweighted path in G connecting two vertices, u and v. The modified Kamada-Kawai vector is given by $F_{KK}(v) =$

$$\sum_{u \in N_i(v)} \left(\frac{2\|pos[u] - pos[v]\|^2}{optDist_G(u,v)^2 \cdot edgeLength^2 + \|pos[u] - pos[v]\|^2} - 1 \right) (pos[u] - pos[v]),$$

where $optDist_G(u,v)$ is defined by:

$$optDist_G(u,v) = \sum_{i=2}^{n} \frac{\sqrt{w_{p_i} \cdot w_{p_{i-1}}}}{w_{e_{p_i p_{i-1}}}}.$$

To achieve an aesthetically pleasing layout of the graph, it is also necessary to modify the Fruchterman-Reingold vectors. As in `GRIP`, the Kamada-Kawai calculation results in a good approximate layout so that the Fruchterman-Reingold calculations can quickly "tidy up" the layout. The modifications needed to support weighted graphs with the ideal edge lengths described earlier are simple:

$$\boldsymbol{F}_{a,w,FR} = \sum_{u \in Adj(v)} \frac{w_e \cdot \|pos[u] - pos[v]\|^2}{edgeLength^2} (pos[u] - pos[v])$$

$$\boldsymbol{F}_{r,w,FR} = \sum_{u \in N_i(v)} s \frac{edgeLength^2 \cdot \sqrt{w_u \cdot w_v}}{\|pos[u] - pos[v]\|^2} (pos[v] - pos[u])$$

5.2 Timeslices and Animation

To visualize a series of graphs, embodying the evolution of a set of relationships over time, we associate another attribute, called a *timeslice*, with each vertex. The timeslice of a vertex is simply a label associated with a vertex. We use the timeslice attribute to partition the vertices of a graph into groups by time. Several simple modifications to the layout algorithm are needed to accommodate timeslice information. Repulsive forces should only exist between vertices in the same timeslice, and thus the optimal distance between vertices in different timeslices which are connected by an edge is zero. In the Kamada-Kawai vectors, the only alteration required is that the the function $optDist_G(u, v)$ be redefined so that for two vertices u, v with timeslice indices of t_u and t_v respectively:

$$optDist_G(u, v) = \delta_{t_u t_v} \cdot \frac{\sqrt{w_u \cdot w_v}}{w_e},$$

where $\delta_{t_u t_v}$ is 1 if $t_u = t_v$ and 0 otherwise. The modifications needed for the Fruchterman-Reingold calculations are similar. Repulsive forces are simply eliminated between vertices in different timeslices, $\boldsymbol{F}_{r,w,t,FR} = \delta_{t_u t_v} \cdot \boldsymbol{F}_{r,w,FR}$ while the attractive forces remain unchanged, $\boldsymbol{F}_{a,w,t,FR} = \boldsymbol{F}_{a,w,FR}$.

The timeslice information alone is not enough to nicely layout evolving graphs; it is also to necessary arrange edges between the timeslices so that the resulting layouts can be used for animation. Perhaps the most straightforward and common case is when one has a series of "snapshots" of a graph taken at some interval over a period of time. When laying out such a graph there are normally two types of constraints which need to be met: Each timeslice should have a pleasing layout, and the layout of consecutive timeslices should be similar, that is, the mental map should be preserved. Another way of formulating this latter constraint is that vertices on one timeslice should tend toward their positions in adjacent timeslices. To meet these constraints the timeslices are combined into a single graph and edges are added between vertices with the same labels in adjacent timeslices.

Since vertices in different timeslices have no repulsive forces between them (but the edges between them retain their attractive forces) these additional edges attract each vertex toward the vertices associated with it in adjacent timeslices. The trade-off between layout within a timeslice and mental map preservation can easily be adjusted by altering the weights of the inter-timeslice edges. Heavier weights lead to better mental map preservation and lighter weights improves the layout of each timeslice. For the visualization of the graphs in this paper the weight of an inter-timeslice edge connecting two vertices of weights w_u and w_v is given a weight of $(w_u + w_v)/2$, causing heavier vertices to stay in the same position over multiple timeslices.

One disadvantage of this method of adding edges is that although the movements of a vertex from one timeslice to the next are minimized, it is still possible that over a span of many timeslices vertices will drift far from their initial positions. If this is an undesirable effect it can be prevented by adding additional edges between vertices in distant timeslices. A straight-forward approach is to create a clique of all the occurrences of each vertex over all timeslices. `GraphAEL` has both options implemented.

Once the layout of the evolving graph has been computed, several visualization options can be used. A static view, capturing all timeslices can be displayed. Each timeslice can be restricted to its own 2D plane or can be drawn in 3D and the individual graphs arranged on top of each other. Edges connecting different timeslices can be shown or hidden. Alternatively, the evolution can be shown via an animation between the timeslices in 2D or 3D. The animation uses an interpolation between timeslices and fading in/out
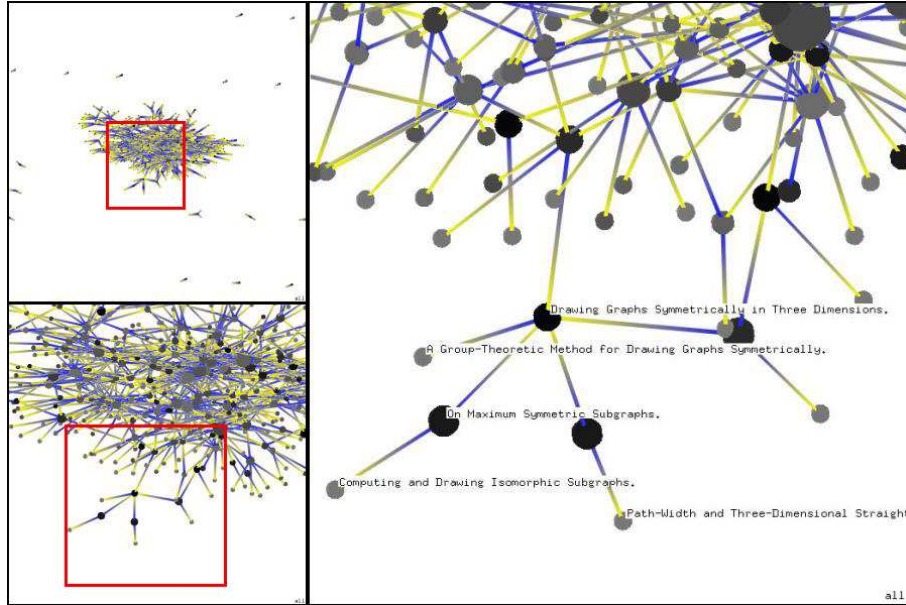
Fig. 3. Top left: cumulative citation graph for the period 1994-02. **Bottom Left:** view of the area enclosed in the red rectangle from top left. **Right:** Labeled vertices in a "symmetry" cluster. Directed edges go from light/source to dark/target. The color of the nodes corresponds to the year it was published: the lighter the node, the older the paper.

of appearing/disappearing vertices and edges. Edge and vertex coloring is used to convey additional information. For example, in the citation graphs we indicate the direction of an edge by gradually varying the color from light(source) to dark(target) and the vertex color indicates the relative age of a particular paper.

Finally, **GraphAEL** can generate and display *difference graphs*. The difference graph between two adjacent timeslices captures the difference between the two underlying graphs. For example, when visualizing the topic graph we may miss significant growth in an area with a few papers if the corresponding vertices are relatively small. With this in mind, the difference graph captures the percentage change between levels. We create a series of difference graphs and at it is an evolving graph we can use all the tools for evolving graphs.

6 Interactive Visualization of the Graph Drawing Literature

In this section we consider three types of evolving graphs extracted from the MySQL database: citation graphs, topic graphs, and collaboration graphs. We also calculated some general statistics about the data and we include gnuplot charts and tables for the data in the Appendix.

6.1 Citation Graphs

The citation graph for a given time period is a simple vertex-weighted directed graph in which the vertices correspond to distinct articles. A directed edge connects two articles, with the article that cites as the source and the cited article as the target. The weight of a
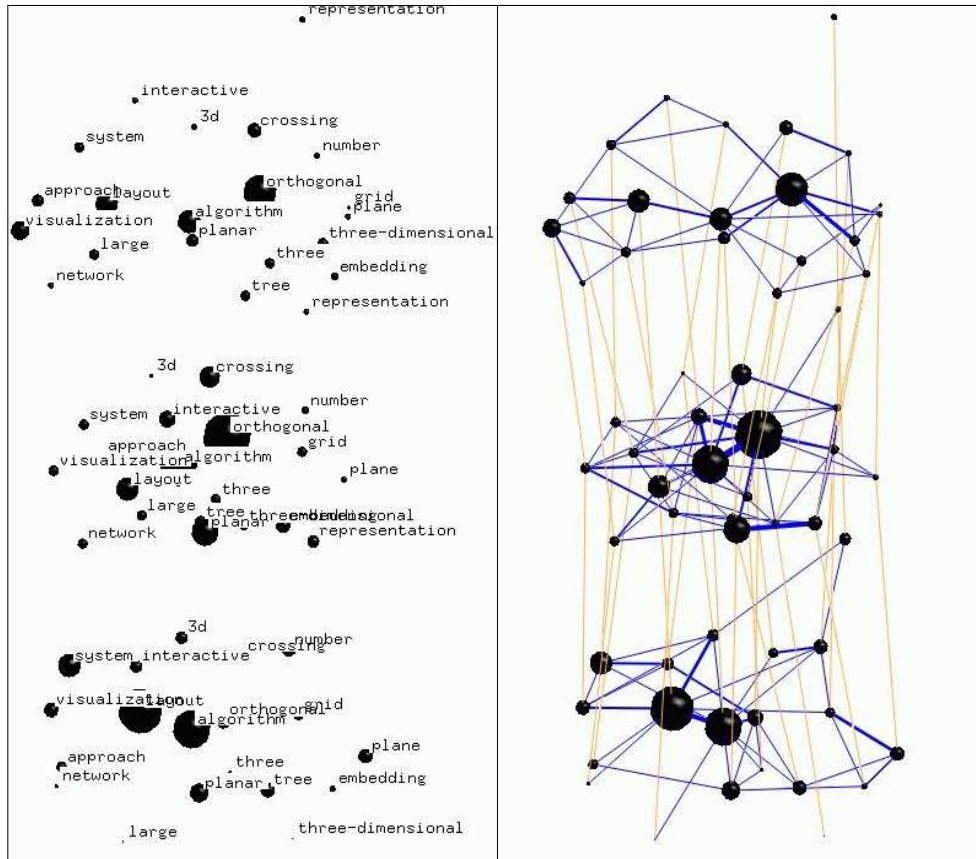
7

Fig. 4. Individual topic graphs for the period 1994-02, restricted to the top 20 title words, shown in timeslices of 3 consecutive years each: 1994-96 on the bottom, 1997-99 in the middle, and 2000-2002 on the top. On the left are all the labeled nodes in each timeslice with invisible edges.

vertex is determined by the number of citations an article received, divided by the number of years since its publication. The citation graphs can reveal information about influential papers in the field, and their relation with the rest of the graph. Fig 3 shows several views of the citation graph. The size of a vertex reflects its weight, i.e. a large vertex corresponds to an article that is cited many times relative to its age. The vertex color indicates the age of the article, the darker a vertex the more recent the corresponding article is. The edge color changes from yellow to blue to indicate direction, where the blue end of the edge is the target vertex.

6.2 Topic Graphs

The topic graph for a given time period is a simple vertex-weighted and edge-weighted undirected graph in which vertices correspond to title words and edges are placed between title words that co-occur in research papers. The weight of a vertex in the topic graph is proportional to the number of papers that contain the corresponding word in their titles. Similarly, the edge weight is proportional to the number of papers in which the
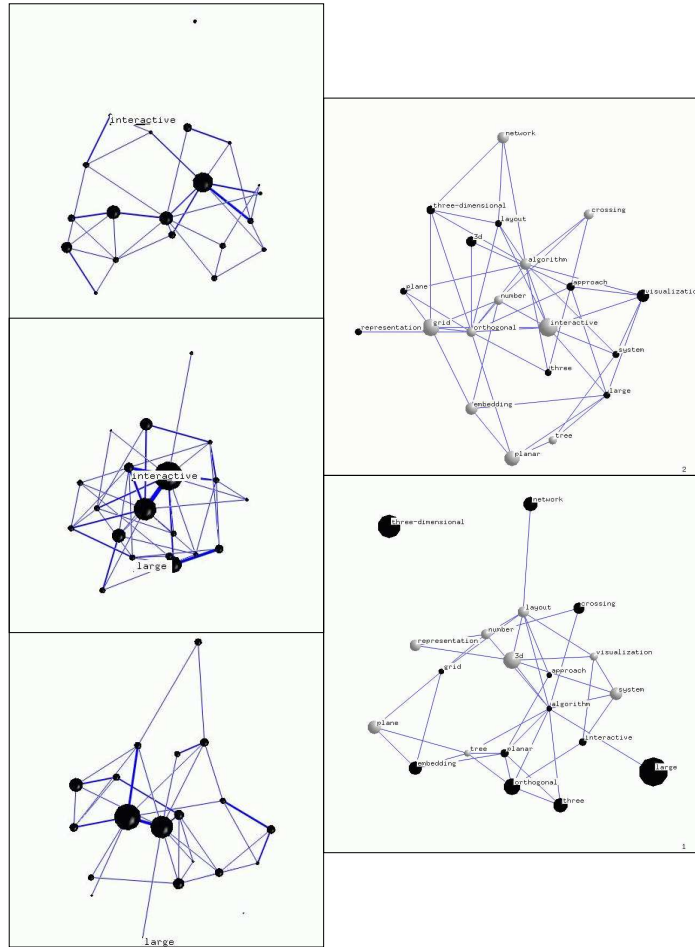
Fig. 5. The three graphs on the left represent 2D views of each timeslice from Fig. 4. The two graphs on the right represent the individual*difference graphs* capturing the change from 1994-96 to 1997-99 (bottom) and 1997-99 to 2000-02 (top). Dark nodes indicate growth and light nodes indicate decline. Note the percentage increase in "large" and decline in "interactive".

two corresponding words co-occur in the title. Topic graphs can reveal information about related topics, the concentration of research on a specific topic and the trends as they evolve through time. Fig. 4 contains several visualizations of the topic graph. The edges connecting same vertices in adjacent timeslices help with mental map preservation and are used to determine the vertex locations in the animation between timeslices.

We also construct topic *difference graphs* in which the weight of a vertex is the percentage of its weight change between adjacent timeslices. The edge weight is proportional to the corresponding percent-change between two time periods. Together with the topic graphs, the topic-change graphs can be effectively used to visualize the evolution of GD research focus through time. Fig. 5 shows the topic-change graph corresponding to the topic graph in Fig. 4.
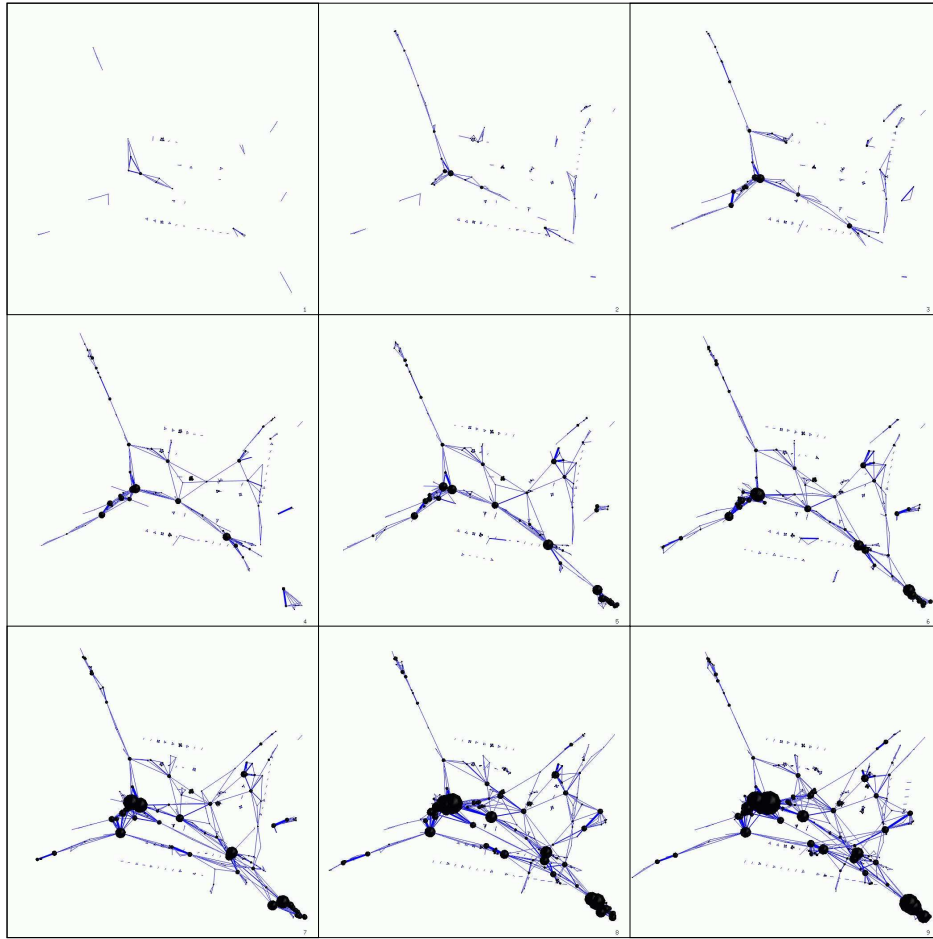
Fig. 6. Screenshots from the animation of the cumulative collaboration graph in the period 1994-02. Each timeslice represents a one year period. A movie of the actual animation can be seen at the `GraphAEL` webpage.

6.3 Collaboration Graphs

Collaboration graphs are simple undirected node-weighted and edge weighted graphs. Vertices represent unique authors and there is an edge between two vertices if the respective authors have collaborated on a research paper. The weight of a vertex is determined by the author's collaborativeness and productivity. The weight of an edge represents the strength of the collaborative ties between two authors. Fig 6 shows the evolution of the cumulative collaboration graph, i.e., each timeslice adds to the previous timeslice. Once an interesting timeslice to be viewed is found it is easy to get a detailed visualization of that specific timeslice. For example, Fig 7 shows the collaboration graph for the period 1994-97. For a more detailed view we zoom in one of the areas where there is a concentration of several well-connected components.
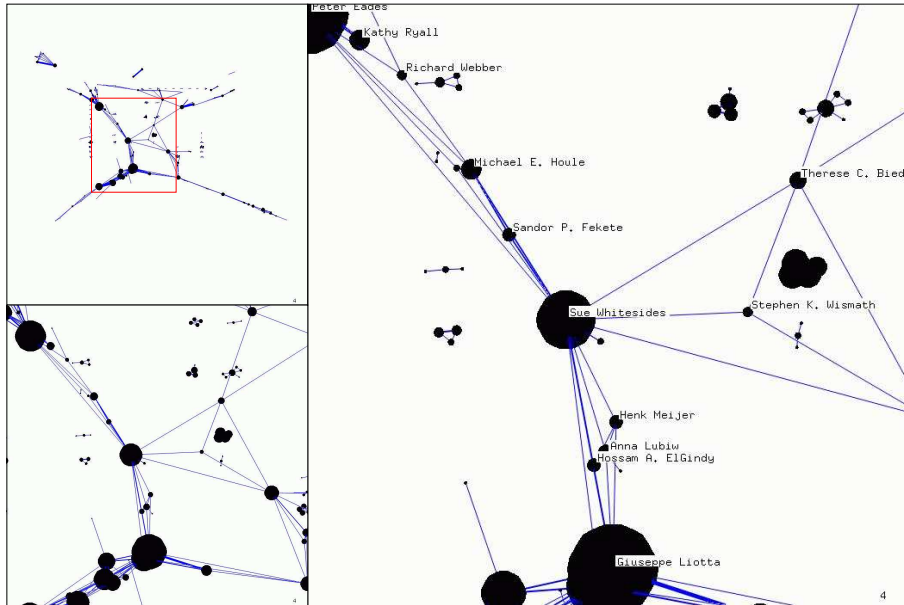
Fig. 7. Top left: collaboration graph for the period 1994-97. **Bottom Left:** view of the area enclosed in the red rectangle from top left. **Right:** closer view with more details.

6.4 Statistical of the Graph Drawing Literature

We compared the data from the GD database, with the data from two computer science databases, the ACM database [8] and the NCSTRL database [17]. Table 1 shows a summary of the overall statistics and they are examined in the detail in the appendix.

References

1. T. Biedl and F. J. Brandenburg. Graph drawing contest report. In *Proceedings of the 9th Symposium on Graph Drawing (GD)*, number 2265 in LNCS, pages 513–521, 2001.
2. U. Brandes and S. R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. In *IEEE Symposium on Information Visualization (INFOVIS '02)*, pages 145–151, 2002.
3. U. Brandes and D. Wagner. A bayesian paradigm for dynamic graph layout. In *Proceedings of the 5th Symposium on Graph Drawing (GD)*, volume 1353 of *LNCS*, pages 236–247, 1998.
4. J. Branke. Dynamic graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs: Methods and Models*, number 2025 in LNCS, chapter 9, pages 228–246. Springer-Verlag, Berlin, Germany, 2001.
5. R. F. Cohen, G. Di Battista, R. Tamassia, and I. G. Tollis. Dynamic graph drawings: Trees, series-parallel digraphs, and planar *ST*-digraphs. *SIAM J. Comput.*, 24(5):970–1001, 1995.
6. C. Collberg, S. G. Kobourov, J. Nagra, J. Pitts, and K. Wampler. A system for graph-based visualization of the evolution of software. In *ACM Symposium on Software Visualization*. To appear in 2003.
7. S. Diehl and C. Görg. Graphs, they are changing. In *Proceedings of the 10th Symposium on Graph Drawing (GD)*, pages 23–30, 2002.

General	GD-Value	ACM-value	NCSTRL-value
Total papers	413	51503	13169
Total authors	502	81279	11994
Authors per paper	2.54	2.32	2.22
Papers per author	2.09	1.80	2.55
Collaborators per author	3.74	3.36	3.59
Percentage of giant component	49	49	57.2
Percentage of 2nd component	0.02	0.11	0.004
Clustering Coefficient	0.60	0.62	0.50
Average Distance	4.33	9.26	9.7
Maximum Distance	10	30	31

Table 1. Statistics for GD, ACM [8], and NCSTRL [17].

8. C. Erten, P. J. Harding, S. Kobourov, K. Wampler, and G. Yee. Exploring the computing literature using temporal graph visualization. Technical Report TR03-04, Department of Computer Science, University of Arizona, 2003.
9. T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Softw. – Pract. Exp.*, 21(11):1129–1164, 1991.
10. P. Gajer, M. T. Goodrich, and S. G. Kobourov. A multi-dimensional approach to force-directed layouts. In *Proceedings of the 8th Symposium on Graph Drawing (GD)*, pages 211–221, 2000.
11. P. Gajer and S. G. Kobourov. GRIP: Graph dRawing with Intelligent Placement. *Journal of Graph Algorithms and Applications*, 6(3):203–224, 2002.
12. J. Grossman and P. Ion. A portion of the well-known collaboration graph. *Congressus Numerantium*, 108:129–131, 1995.
13. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.
14. T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inform. Process. Lett.*, 31:7–15, 1989.
15. H. Kautz, B. Selman, and M. Shah. Referral Web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.
16. S. Moen. Drawing dynamic trees. *IEEE Software*, 7(4):21–28, July 1990.
17. M. E. J. Newman. Who is the best connected scientist? a study of scientific coauthorship networks. *Physics Review*, E64, 2001.
18. S. C. North. Incremental layout in DynaDAG. In *Proceedings of the 4th Symposium on Graph Drawing (GD)*, pages 409–418, 1996.
19. K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *IEEE Symposium on Information Visualization (INFOVIS '01)*, pages 43–50, 2001.

Appendix A

Authors per Paper In mathematics, the average number of authors per paper has increased from about 1 per paper in 1935 to about 1.5 in 1995 [12]. Averages in medicine and physics are often higher, about 9 collaborators per paper [17]. The number of authors per paper in GD is comparable to that in ACM; see Fig. 8.

Size of Giant Component In the theory of random graphs it is known that increasing the density of the edges leads to the formation of a giant connected component. While the size of the giant component in a typical scientific collaboration graph is 80%-90% the number seems to be much smaller for the computing literature [17]. Our GD data indicates that the size of the giant connected component is about 49%. In other words, about half of the authors in the GD database are connected via a path of co-authors.

Average and Maximum Distances We can also find the shortest path from one author to another in the co-authorship graph. This information is useful in the sense that it creates a chain of references of intermediate scientists through whom contact between two authors may be established [15]. The average and maximum distances for the GD database seem to be much smaller than the other computer science databases, most likely because of the small size of the GD dataset.

Clustering Coefficient A useful measure for the strength of the ties between authors is the clustering coefficient. Let N_u denote the set of neighbors of vertex u in the collaboration graph and let E_{N_u} be the set of edges e such that both of the vertices incident to e are in N_u. The clustering coefficient for u is defined as:

$$C_u = 2 \times \frac{|E_{N_u}|}{|N_u| \times (|N_u| - 1)}$$

In other words, the clustering coefficient of u tells us how collaborative the co-authors of u are among themselves. We have found that the average clustering coefficient for the collaboration graph is 0.60, which is comparable with the clustering coefficient of the
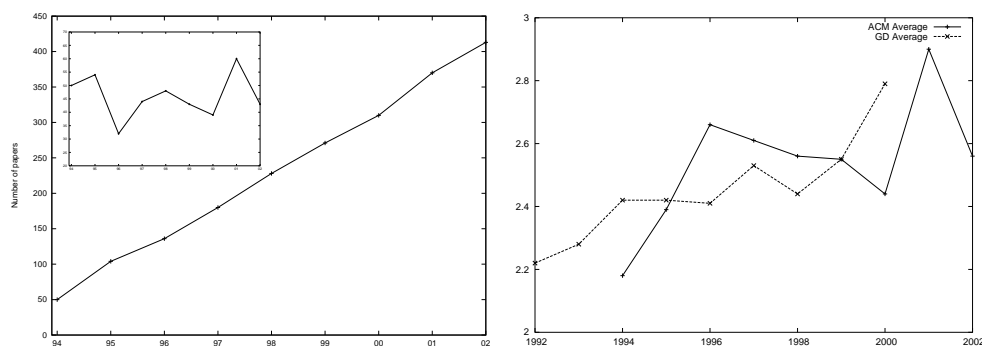

Fig. 8. Left: Cumulative number of conference papers for 1994-02 with the number of papers each year in the inset. Right: average number of authors per paper per year for GD and ACM.

Name	Num. of papers		Name	Num. of co-authors
Liotta, Giuseppe	23		Liotta, Giuseppe	35
Eades, Peter	23		Mutzel, Petra	28
Mutzel, Petra	19		Whitesides, Sue	26
Battista, Giuseppe Di	17		Battista, Giuseppe Di	25
Tollis, Ioannis G.	15		Junger, Michael	23
Junger, Michael	15		Eades, Peter	21
Tamassia, Roberto	13		Wood, David R.	18
Kaufmann, Michael	12		Brandes, Ulrik	18
Didimo, Walter	12		Marks, Joe	17
Whitesides, Sue	12		Gutwenger, Carsten	17

Table 2. Authors with highest number of papers and collaborators.

other databases. This indicates that the tendency to form strongly tied computing research groups is as high as in other research areas, despite the relatively small size of the giant connected component.

Number of Papers and Collaborators The average number of papers per author is 2.09, while the average number of collaborators is nearly double at 3.74. Table 2 shows the most productive and most collaborative authors.

Popular Title Words Parsing the paper titles allows us to look for "buzzwords" in the different years. Fig. 9 shows the most popular four words (as percent of the total) used each year starting in 1994, not including words such as *for*, *the* and the two most popular words *graph* and *drawing*. "Interactive" seems to have been popular in 1997 and 1998, while "orthogonal", and "algorithms" are consistently popular.
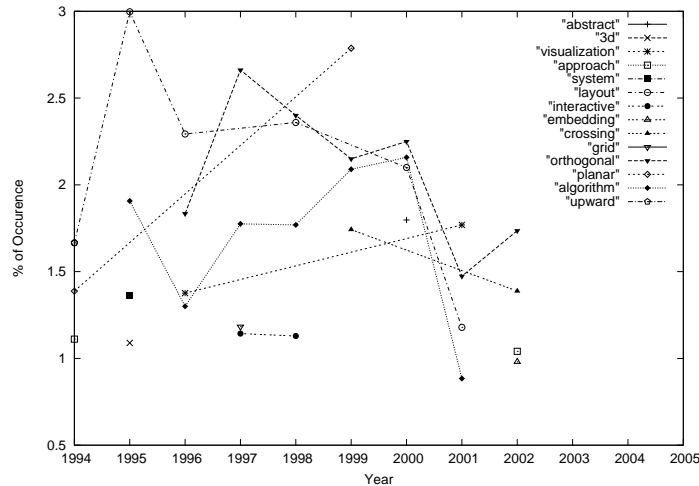

Fig. 9. Title popularity