A Hybrid Model for Drawing Dynamic and Evolving Graphs[*]

Marco Gaertler and Dorothea Wagner

Universität Karlsruhe (TH), Faculty of Informatics, 76128 Karlsruhe, Germany.
{gaertler,dwagner}@informatik.uni-karlsruhe.de

Abstract. Dynamic processes frequently occur in many applications. Visualizations of dynamically evolving data, for example as part of the data analysis, are typically restricted to a cumulative static view or an animation/sequential view. Both methods have their benefits and are often complementary in their use. In this article, we present a hybrid model that combines the two techniques. This is accomplished by 2.5D drawings which are calculated in an incremental way. The method has been evaluated on collaboration networks.

1 Introduction

Dynamic graphs occur in many applications such as software visualization, animation of graph algorithms or social network analysis. Most of the time a dynamic graph is given by a sequence of graphs that each are snapshots of an ongoing process. While the visualization of individual points in time helps to understand the current situation, a visualization of the whole sequence can reveal information about the evolution in general. So far most visual representations use either a static cumulative view of the sequence or a dynamic animation.

We describe a new hybrid model for dynamic graph drawing that allows a simultaneous representation of both, a cumulative and an animated view. Both views are integrated in such a way that the hybrid layout reveals each of them by changing the perspective or adjusting visual effects, like color or transparency. It is assumed that not only the graph structure but also weights of nodes and edges change over time. A benefit of our approach is the integration of the past evolution of weights by incorporating a cumulative as well as a regressive change, i.e., the weights of nodes and edges reflected in the drawing can also decrease over time. Our approach uses $2.5D$ drawings where time is represented by the third dimension. However, the technique can be generalized to $d.5D$ drawings for arbitrary dimensions d.

Multidimensional visualizations where one or more axes are fixed have been proposed frequently for network data from various applications. Related methods use the third dimension to display structural information [3], a hierarchy [4, 8, 1], or an evolution over time [2]. Other visualization techniques for dynamics that are based on conventional 2D or 3D drawings or animations are [9, 11, 5]. See also [6] for an overview and [10] for a more recent work.

The paper is organized as follows. Section 2 introduces our model and the corresponding layout technique. It also includes a short discussion of its benefits and potential drawbacks. The special case of evolving graphs and updating dynamic layouts are topic in Section 3. The results are presented in Section 4. For illustrative purpose, data from the DBLP[1] are used. Finally, Section 5 gives the conclusion.

2 Hybrid Model

In this section, the basic hybrid model is introduced. Section 2.2 and 2.3 provide the description on the model and the algorithmic realization. This is followed by a short discussion of accumulating weights over time while preserving the mental map.

2.1 Notation

A dynamic graph $\mathcal{G}$ is given by a mapping of a time interval $\mathcal{T}$ into the set of weighted graphs. In the following, we assume that there are only finitely many different images of $\mathcal{G}$ and $\mathcal{G}(t) = (V(t), E(t))$ denotes the graph at time $t \in \mathcal{T}$. Without loss of generality we assume that $\mathcal{T}$ can be covered by left-closed and right-open intervals $[t, t'[$ such that $\mathcal{G}$ is fixed on each such interval, and changes on subsequent intervals. For any given point in time $t \in \mathcal{T}$, we denote the earliest time of the left-adjacent corresponding interval with $\mathrm{pred}(t)$ and the earliest time of the right-adjacent corresponding interval with $\mathrm{succ}(t)$. Let $\mathcal{G}(t) = (V(t), E(t))$ be the graph at time $t \in [t_1, t_2[$, then the nodes that have not been in any previous graph are denoted by $V_{\mathrm{new}}(t) := \{v \in V(t) \mid \forall t' < t_1 : v \notin V(t')\}$ and corresponds to 'new' nodes. Similarly $V_{\mathrm{old}}(t) := V(t) \setminus V_{\mathrm{new}}(t)$ denotes the 'old' nodes. Let ω_t denote the weight of a node respectively edge at time t, i.e., $\omega_t \colon V(t) \cup E(t) \to \mathbb{R}_0^+$.

In this way a dynamic graph corresponds to the observations of the (dynamic) process and approximates it with a step function. The changes need not be homogeneously distributed over time and additional observations could be created artifically using interpolation. Furthermore, it reflects the realized changes in the dynamic graph drawing setting.

2.2 Paradigm

The original dynamic graph drawing problem has two realizations: First, the *cumulative view*, which consists of one static layout that emphasizes major trends

---
[1] http://www.informatik.uni-trier.de/~ley/db/

during the evolution but hides sporadic fluctuations in the graph structure. Secondly, the *animated* or *sequential view*, which requires a static layout for each graph that highlights current changes while preserving the general mental map. Both have in common that they are based on the whole sequence. We address a more general problem: Given a dynamic graph $\mathcal{G}$ and its time interval $\mathcal{T} := [0, T[$. For any subinterval $\mathcal{T}' = [t_1, t_2[$ of $\mathcal{T}$ construct suitable layouts that represent the evolution during $\mathcal{T}'$ based on the history of the interval $[0, t_1[$.

Our hybrid model consists of one $2D$ layout for each graph of a sequence embedded layer-wise in $3D$ where the additional dimension represents the time. A sketch of this situation is given in Figure 1. To be more precise, we use one layer for the history $[0, t_1[$ and one layer for each different graph in the interval $[t_1, t_2[$. By tuning the perspective and the individual properties of the layers, the original views, i.e., cumulative and sequential, are obtained: Looking along the time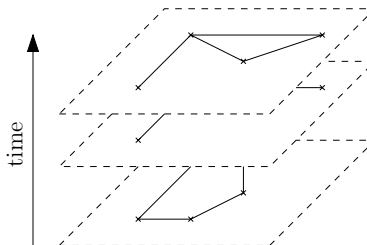 axes (in its negative direction) yields the cumulative view while showing only one layer at a time results in the sequential view. A third kind of view is obtained when identical nodes in different time slots are connected and the perspective is parallel to the layers. It shows the nodes' changes over time. The model is realized by an incremental layout algorithm. First, the history-layer is initialized with a suitable layout obtained by some established algorithm. For every additional layer the nodes are split into two groups, old and new nodes. The old nodes can be easily placed respecting their former positions. The new nodes first need a good initial placement, before the whole layout can be optimized to meet esthetic criteria as well as preserving the mental map.

Fig. 1. Hybrid model

2.3 Algorithmic Framework

The incremental algorithm, which is associated with the hybrid model, is given in pseudo code in Algorithm 1. It has a large degree of freedom that allows to derive several versions which are optimized for running time, achieved quality, or dependency of temporal knowledge. Especially the last issue also provides means to layout dynamic graphs where only partial information is available during the process. However, a fundamental problem might occur through lack of (future) knowledge, i.e., the position of a connected component which is completely contained in $V_{\text{new}}(t)$ has great influence on the overall quality of a drawing. But the component cannot be properly placed without using information about its future role. Section 3.1 discusses these aspects for dynamic graphs with complete information (evolving graphs).

In the following, some simple methods for Step 1 and 2 are stated. For the initial placement, we suggest a two step approach which combines a barycentric layout with a localized force-directed relaxation. In this way, new nodes are close to their older "anchor" nodes and 'uniformly' spread. This requires that

every connected component of $\mathcal{G}(t)$ has at least one node in $V_{\text{old}}(t)$. As for the general optimization step, a modified force-directed approach works well if no further layout properties have to be ensured. The modifications mainly target the incorporation of node and edge weights, i.e., large/heavy nodes should be well distributed, thick/heavy edges should be short.

2.4 Adjustments for Weights and Position

As mentioned in the introduction, the hybrid model is capable of dealing with decreasing weights. In fact, we propose an updating of the weights of nodes and edges to incorporate both their current and their accumulated weight. Basically, there is a tradeoff between old, heavy, and inactive nodes versus young, light, and extremely active nodes. For every interval $[t_2, t_3[$ and its left-adjacent interval $[t_1, t_2[$, we define a semi-cumulative version $\widetilde{\omega}$ of ω_t as shown in Equation (1).

$$
\begin{aligned}
\widetilde{\omega_c}(v, t_2) &:= \frac{\widetilde{\omega}(v, t_1)}{s} + \frac{\omega_{t_2}(v)}{t_2 - t_1} \qquad \text{continuous version} \\
\widetilde{\omega_d}(v, t_2) &:= \frac{\widetilde{\omega}(v, t_1)}{s^{t_2 - t_1}} + \omega_{t_2}(v) \qquad \text{discrete version.}
\end{aligned}
\tag{1}
$$

If there is no left-adjacent interval, then $\widetilde{\omega}(v, t) := \omega_t(v)$. The function $\widetilde{\omega}$ can be extended to a continous function via interpolation. Depending on a scaling parameter s, different behavior is favored, i.e., for $s = 1$ it is the standard cumulative version, for $s > 1$ young and active nodes are preferred over old and inactive nodes while it is vice versa for $0 < s < 1$. The difference between the two versions is the interpretation of time, i.e., the continuous version assumes that the weight $\omega_t(v)$ has been accumulated since the last observation t_1 while the discrete version interpretes the weight $\omega_t(v)$ as an instantaneous impulse at time t and that no other impulse has occurred since time t_1. Both models can be justified, the continuous weighting reflects steady growth in contrast to singleton events during an elementary time window that is imitated in the discrete

---

Algorithm 1: Generic hybrid layouter

Input: dynamic graph $\mathcal{G}$ with time interval $[0, T[$ and a subinterval $[t_1, t_2[\subseteq [0, T[$
initialize $\mathcal{G}([0, t_1[)$ with a suitable layout
$t \leftarrow t_1$
while $t \leq t_2$ **do**
    adjust weights on nodes and edges
    for $v \in V_{old}(t)$ **do**
        initialize v with its last used position
1   initialize $V_{\text{new}}(t)$
2   optimize $\mathcal{G}(t)$ to meet esthetic criteria while preserving the mental map
    $t \leftarrow \text{succ}(t)$
project each $G(t)$ to the t.th layer

---

version. In collaboration networks which are restricted to certain publications, like certain conference publications only, one would prefer the discrete version over the continuous one because of the time dependency.

Similar to other approaches, we introduce additional forces to ensure that the movement of old nodes stays uniform. There are several different ways to anchor a node to its copies in different snapshots. The simplest approach introduces an edge between two identical nodes in consecutive snapshots of ideal length zero. Thus the copies of a node are connected via a path. By introducing additional edges the movement of a node can be further restricted. In the extreme case, all copies of a node are connected to a clique. This type of connection ensures best to preserve the mental map but might slow down the actual layout computation. However, since only one (time) layer is active at a point in time t, there are only T additional positions at which a node might be anchored, where T is the number of previous intervals. Thus even a clique-like connection between identical nodes results in only $\mathcal{O}(T \cdot |V(t)|)$ additional active edges which does not slow down the computation too much. Actually, most of the known techniques to control the movement of nodes can be directly integrated in the hybrid model and its incremental layout method.

3 Extensions of the Hybrid Model

After the basic hybrid model has been introduced in Section 2, two adjustments for specific tasks are presented. First, the modification for evolving graph, i. e., dynamic graph where the whole function is given, and second, a dynamic version of the dynamic graph drawing problem, i. e., given a dynamic graph with a layout, find an extention of this layout if additional time layers are introduced.

3.1 Adjustments for Evolving Graphs

As already mentioned in Section 2.3, incremental layouts cannot find a good position for connected components consisting of only new nodes. Algorithms that are based on the whole sequence avoid this problem through their 'future' dependencies. For example, in [5, 10] identical nodes in consecutive (time-)layers are connected with an edge. During the minimization of the overall forces, a good position of the connected components in early layers is ensured by the position in subsequent layers in which the component has been connected to an already placed part. Thus the relative placement is propagated back in time. A similar scheme can be integrated in the hybrid model: First, the earliest succeeding time is calculated in which the component is connected to some already placed nodes. This layer is then used to estimate the relative position of the component to its anchor nodes and projected back. The optimization step (Algorithm 1, Step 2) treats the components independently and uses the relative placement to ensure that none of them are interfering with each other. If a component has no anchoring nodes, its placement is independent from the remaining graph and can be done arbitrarily. This additional step can be done in linear time plus

the time for finding the relative position, which depends on the involved layout technique (Step 1 and 2).

Potential drawbacks are the overhead, if several connected components have anchoring nodes in different time layers, then the relative position for each component involves a 'whole' layout step for each corresponding time-layer. Also, the case where some connected components of a time layer get connected with each other before anchoring nodes appear is a bit problematic. But the above method can be extended to include these cases as well. A different issue is the impact on the overall quality, i.e., because only a relative placement is estimated, certain areas of the layout can become wide-stretched while others are too condensed. This usually happens if the components are rather sparsely connected. By manually adjusting the strength of edges, one can counterweight this effect, however simultaneously diminish the relation between distance in the layout and edge weights.

3.2   Updating Layouts of Dynamic Graphs

In contrast to the connected components, the hybrid models benefits the extension or update of the dynamic graph layout when additional data become available. In other words, given two dynamic graphs $\mathcal{G}$ and $\mathcal{G}'$ such that $\mathcal{G}'$ refines $\mathcal{G}$, i.e., both graphs coincide on some parts of their time interval and differ on the remaining, and a layout $\mathcal{L}$ for $\mathcal{G}$, find a new layout $\mathcal{L}'$ for $\mathcal{G}'$ such that if $\mathcal{G}(t) = \mathcal{G}'(t)$ then also $\mathcal{L}(t) = \mathcal{L}(t')$. This can be interpreted as constraint dynamic graph drawing problem.

Independent of the algorithm, one can always use interpolation of two adjacent fixed time layers for intermediate layers. Using the hybrid model, our approach is to refine the interpolation via bisection, i.e., calculating a rough estimate of the layout for an intermediate layer and using this as an auxiliary layout for the interpolation, more precisely: Let $[t_1, t_{2k}[$ be an interval on which $\mathcal{G}$ is constant and $t_1 < t_2 < \cdots < t_{2k}$ be a subdivision such that $\mathcal{G}'$ is constant on $[t_i, t_{i+1}[$ for $1 \leq i < 2k$ and differs from $\mathcal{G}$. First, a rough placement $L(t_k)$ for time t_k is estimated and afterwards recursively applied to the interval $[t_1, t_k[$. Upon reaching time t_2, an 'exact' placement is calculated instead of the rough placement. Afterwards layouts for $t_3, \ldots, t_k$ are determined in the incremental fashion of the hybrid model. The process is then repeated on the interval $[t_k, t_{2k}[$.

Instead of the bisection approach one could only use the incremental algorithm to interpolate the interval $[t_1, t_{2k}[$, however, if the intervals have many unknown intermediate points or the layout for t_1 and t_{2k} differs a lot, then the overall quality significantly drops.

Consistency So far, the hybrid model and other dynamic visualization algorithms behaved similar to the update problem. However, there is a difference when comparing the $\mathcal{L}'$ with the layout $\mathcal{L}''$ for $\mathcal{G}'$ ignoring the constraint $\mathcal{L}$. A fully time-dependent algorithm, like the one in [10] can produce very different results for $\mathcal{L}'$ and $\mathcal{L}''$, while general incremental algorithms will give the same

partial layouts on the interval $[t_0, t_1[$ where t_0 is the earliest time and t_1 the time of the first deviation of $\mathcal{G}$ and $\mathcal{G}'$.

Also, the hybrid model will produce the same partial layout on $[t_0, t_1[$. Furthermore, if the modifications of the intermediate time slots are small or even consistent with our continual weighting (Section 2.4), then the layouts $\mathcal{L}'$ and $\mathcal{L}''$ of the hybrid model will be very similar. The following observation verifies this claim: If the introduced modifications are small, then both graphs, the original and the modified one, should have similar high-quality layouts. Moreover using a local optimum of force-direction layout of the original graph as initialization for the modified one will quickly convert to a close local optimum.

Thus the similarity of $\mathcal{L}'$ and $\mathcal{L}''$ on a refined interval $[t, t'[$ highly depends on the similarity or consistency of the intermediate graphs and the impact on previous modifications but not on succeeding ones. An extreme case would be consistent refinement together with a large modification at the end of the sequence. Traditional algorithms that use the whole available information would produce very different layouts $\mathcal{L}'$ and $\mathcal{L}''$ while incremental and especially the hybrid model would result in very different layouts upto the heavy modification. However, this is paid in terms of achieved overall layout quality.

4 Results

We illustrate some results of our hybrid model for citation networks extracted for the DBLP which is a well-maintained database with approximately 500,000 articles in the area of computer science.

4.1 Data Sets

DBLP maintains information of certain publications. We extracted the overall collaboration graph, i.e., nodes are people and edges connect to nodes if they have common publications. Because the publication activity varies a lot and a single publication can have up to 36 authors, we weighted the edges correspondingly. The weight of a single publication is reciprocal to the number of authors and the weight of all publications in a year is the sum of the individual weights. The weight of a node is the sum of the weights of its incident edges (for a given year).

4.2 Visualizations

In the following, we present several drawings of collaboration networks. In each visualization, there are the following correspondences: node size and cumulative publication weight, node color and time, edge thickness and publication weight, edge color and time. If an edge has a checked pattern, it connects two identical nodes in consecutive snapshots. When speaking of cumulative weight, we always refer to the continuous version shown in Equation (1).
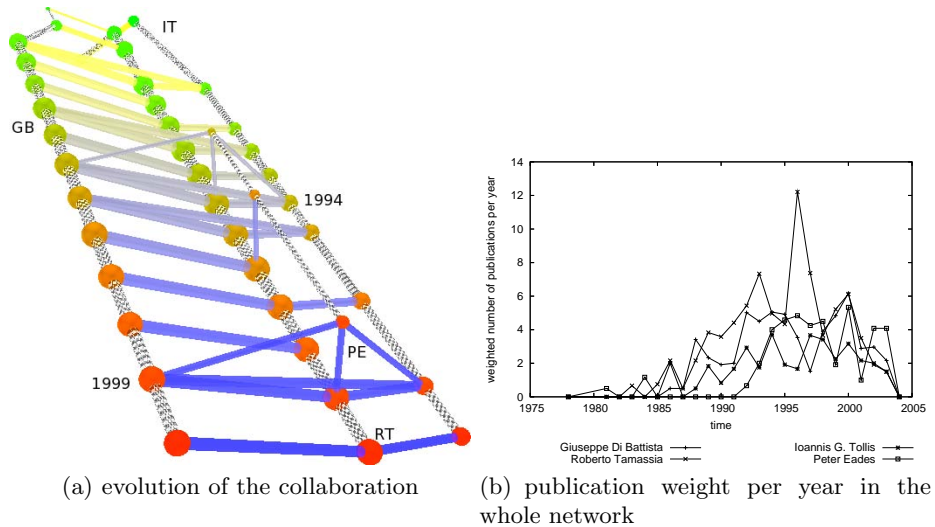
7

(a) evolution of the collaboration

(b) publication weight per year in the whole network

Fig. 2. Collaboration between Giuseppe Di Battista (GB), Ioannis Tollis (IT), Peter Eades (PE) and Roberto Tamassia (RT) between 1986 and 2000.

A first example is the authorship of [7], which is one of the first books about graph drawing. Figure 2(a) shows the evolution of their collaboration between 1986 and 2000. It is clearly visible when common publications have occurred, although individual publications are not identifiable. For example the first collaboration between the four authors happened in 1994. Also the node size reflects the continuity of cooperations between authors. However, the visualization in Figure 2(a) is limited to the four authors and only their collaboration. Figure 2(b) shows the publication weight of each author within the whole collaboration network. Also, Figure 2(a) illustrates the effect of consistent modifications (Section 3.2) quite well. Between 1991 and 2000, the modifications have been very small, i. e., some reweighting on the nodes and edges, one node appeared while another node disappeared for some time, but the overall layout has been quite stable.

The second example is the collaboration between Ulrik Brandes, Dorothea Wagner and their direct coauthors. Both have published several articles in the graph drawing area and others. Figure 3(a) shows the evolution for 2001 and 2002. The whole network is rather dense, a static cumulative view (without edge weights) is given in Figure 3(c). The static graph has 50 nodes and 161 edges while the time-expanded graph has 206 nodes and 434 edges. Some individual layers are presented in Figure 3(d) and 4 and present the collaboration at specific points in time. In every layer only those nodes that have published something are shown. Again Figure 4(a) and 4(b) clearly indicate that the node size is only relative to the selected network, i. e., nodes like Peter Eades, Joe Marks or Michael Kaufmann who have a large weight in the whole network (see for
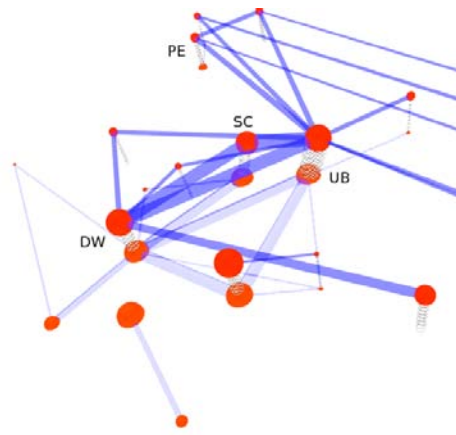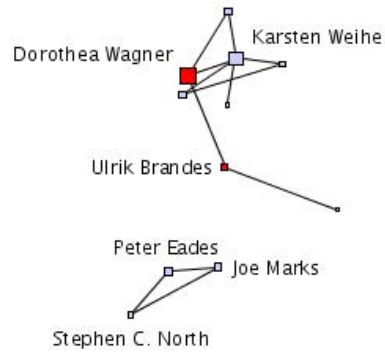
(a) evolution between 2000 and 2001
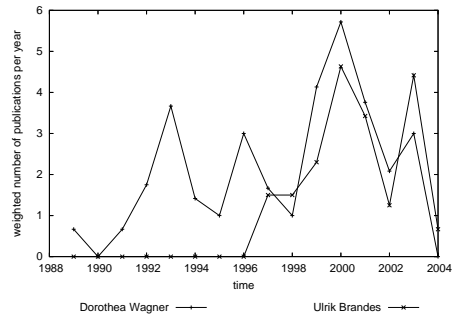
(b) publication weights per year
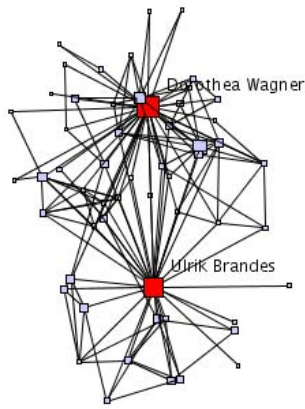
(c) cumulative static view

(d) snapshot from 1997

Fig. 3. Collaboration between Ulrik Brandes (UB), Dorothea Wagner (DW) and their direct coauthors between 1989 and 2002. (Other abbreviations are SC for Sabine Cornelsen and PE Peter Eades.)
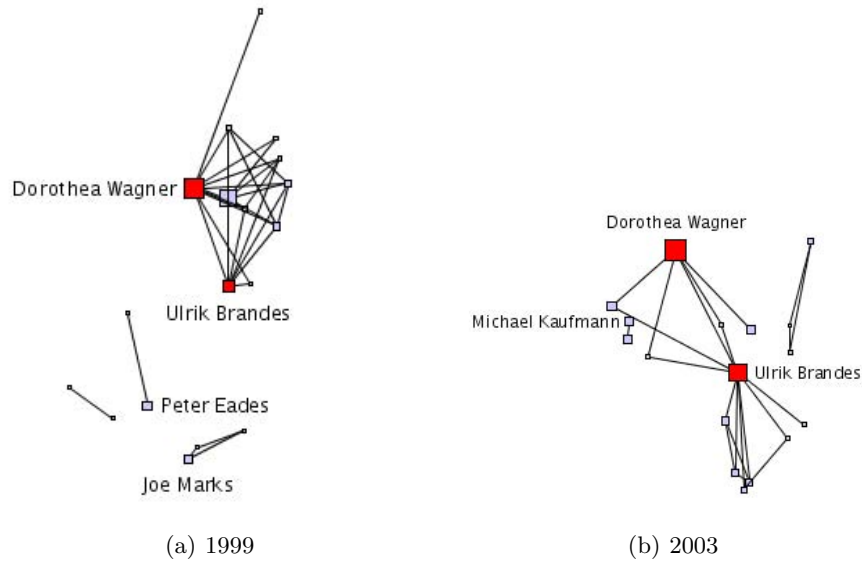
(a) 1999   (b) 2003

Fig. 4. Snapshots of the collaboration between Ulrik Brandes, Dorothea Wagner and their direct coauthors at different points in time.

example Figure 2(b)) have a rather peripheral role in this collaboration network. The balance between old and active nodes and young and active nodes is also visible in Figure 3(a). As shown in Figure 3(b), Ulrik Brandes and Dorothea Wagner have roughly the same amount of weighted publications since 1997. However, Dorothea Wagner has been active since 1989, while Ulrik Brandes started in 1997. However, in the evolutionary view both have a similar size which reflects the similar accumulated publication weight. Using a purely cumulative weight, this would not be the case. Finally, Figure 5 shows the collaboration in a broader sense, i. e., a network with increased number of intermediate coauthors. The visualization shows a sparse connection between the main part that contains both Ulrik Brandes and Dorothea Wagner and a peripheral part.

The final example is a collection of some program committee members of the International Symposium on Graph Drawing. Figure 6(a) shows a 2D project of the evolution between 1986 and 2003 which mask the time axis while a perspective view is given in Figure 6(b). This example reflects both that certain groups are formed over time that collaborate very closely, but also that occasional collaboration tend to be repeated. Some of the artifacts on the layer 2003 are due to incomplete data.
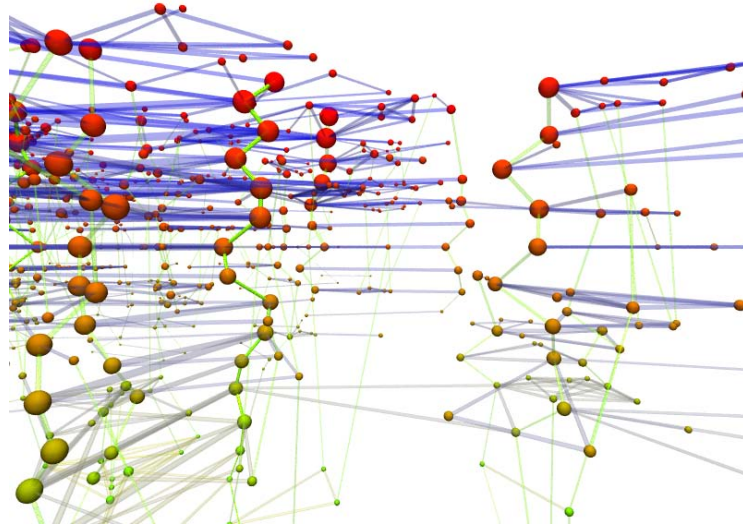
Fig. 5. Partial view of the collaboration between Ulrik Brandes and Dorothea Wagner using more intermediate coauthors.

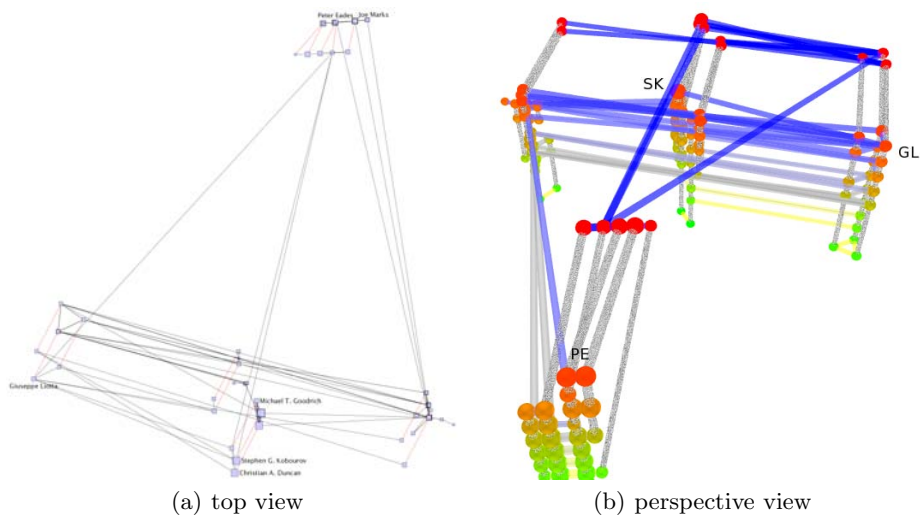(a) top view   (b) perspective view

Fig. 6. Collaboration between some PC member between 1986 and 2003. (Abbreviations are GL for Guiseppe Liotta, PE Peter Eades and SK for Stephen Kobourov.)

5 Conclusion

We introduced a hybrid model for drawing dynamic and evolving graphs based on 2.5D visualizations. It combines several aspects of static cumulative views and animated/sequential views. The obtained layout permits the general view of the evolution while integrating individual aspects of certain points in time as well as cumulative and regressive changes of weight functions. The method has been evaluated on networks modeling collaboration. In the layouts, structural important nodes were well visible and long-existent nodes did not mask younger nodes. Also sparsely connected components were spatially separated.

References

1. M. Baur, U. Brandes, M. Gaertler, and D. Wagner. Drawing the as graph in 2.5 dimensions. In *Proceedings of the 12th International Symposium on Graph Drawing (GD'04)*, Lecture Notes in Computer Science, pages 43–48, 2005.
2. U. Brandes and S. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization*, 2(1):40–50, 2003.
3. U. Brandes and S. Cornelsen. Visual ranking of link structures. *Journal of Graph Algorithms and Applications*, 7(2):181–201, 2003.
4. U. Brandes, T. Dwyer, and F. Schreiber. Visual understanding of metabolic pathways across organisms using layout in two and a half dimensions. *Journal of Integrative Bioinformatics*, 0002, 2004.
5. U. Brandes and D. Wagner. A Bayesian paradigma for dynamic graph layout. In Giuseppe Di Battista, editor, *Proceedings of the 5th Symposium on Graph Drawing (GD'97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 236–247, Rome, Italy, 1997. Springer-Verlag.
6. J. Branke. Dynamic graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs*, volume 2025 of *Lecture Notes in Computer Science*, pages 228–246. Springer-Verlag, 2001.
7. Guiseppe di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing - Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.
8. P. Eades and Q. Feng. Multilevel visualization of clustered graphs. In *Proc. of Graph Drawing*, volume 1190 of *Springer LNCS*, pages 113–128. Springer, 1996.
9. P. Eades, W. Lai, K. Misue, and K. Sugiyama. Preseving the mental map of a diagramm. In *Proceedings of Compugraphics '91*, pages 24–33, 1991.
10. C. Erten, P. Harding, S. Kobourov, K. Wampler, and G. Yee. Graphael: Graph animations with evolving layouts. In *Proceedings of the 11th International Symposium on Graph Drawing (GD'03)*, Lecture Notes in Computer Science, pages 98–110, 2004.
11. S. C. North. Incremental layout with dynadag. In *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 409–418. Springer-Verlag, 1996.