# Dynamic Network Visualization in 1.5D

Lei Shi Chen Wang
IBM Research - China
19 Zhongguancun Software Park
Beijing, 100193 China
{shllsh,wangcwc}@cn.ibm.com

Zhen Wen
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10578 USA
zhenwen@us.ibm.com

## Abstract

The dynamic network visualization has been a challenging topic due to the complexity introduced by the extra time dimension. Existing solutions to this problem are usually good for the overview and presentation, but not for the interactive analysis. We propose in this paper a new approach which only considers the dynamic network central to a focus node (aka dynamic ego network). The navigation of the entire network is achieved by switching the focus node with user interactions. With this approach, the complexity of the compressed dynamic network is greatly reduced without sacrificing the network and time affinity central to the focus node. As a result, we are able to present each dynamic ego network in a single static view, well supporting user analysis on temporal network patterns. We describe our general framework including the network data pre-processing, 1.5D network and trend visualization design, layout algorithms, as well as several customized interactions. In addition, we show that our approach can also be extended to visualize the event-based and multimodal dynamic networks. Finally, we demonstrate, through two practical case studies, the effectiveness of our solution in support of visual evidence and pattern discovery.

## 1 Introduction

As information about networks keeps on growing, researchers and information analysts are posed with a new challenge: how to deal with and gain insight into a network that evolves over time. Traditionally, visualization plays a key role in the network analysis, such as social and telecommunication pattern mining. The visualization of evolving networks, known as dynamic network visualization, remains challenging to the community in at least three aspects:

- **Visual metaphor:** With the additional time dimension, the dynamic network could not be mapped to the traditional 2D visualization directly, while the straightforward 3D visualization leads to the well-known occlusion issue.

- **Scalability:** When the time dimension is defined in a fine granularity, the scale of the dynamic network may increase 1∼2 magnitudes from each of its static snapshots. The resulting graph is usually huge, for which the visualization remains problematic due to the significant visual clutters and the high layout computation complexity.

- **Suitability for analysis:** Although several visualizations have been proposed for the dynamic network, none of them provide adequate visual assistance for human analysis towards temporal network patterns. Manual comparisons among networks at each snapshot are generally required.

State-of-the-art dynamic network visualization techniques can be summarized into two categories: the first class employs a straightforward method which presents the dynamic network as a movie [25]. The network at each specific time is presented as the user proceeds to the corresponding frame. This approach decouples the network data by time and results in a simpler snapshot in each view. Criticisms over this approach concern its viability for analysis because the user of the visualization is offered little freedom when the movie plays by itself. Some researchers even argue that the movie approach is only useful for presentation and not for analysis [30]. The other kind of solutions follows the principle of small multiples display [30]. They juxtapose the network snapshots of each time frame into the same view, in the same order with the time line. This approach decouples the data spatially, hence relieves the analyst from scrolling back and forth on the time line. However, it only provides an overview of the evolving networks, still offering little visual assistance for the network analysis tasks. The analysts need manual comparisons among the networks. In addition, each static network is shown in a smaller size, inversely proportional to the time granularity.

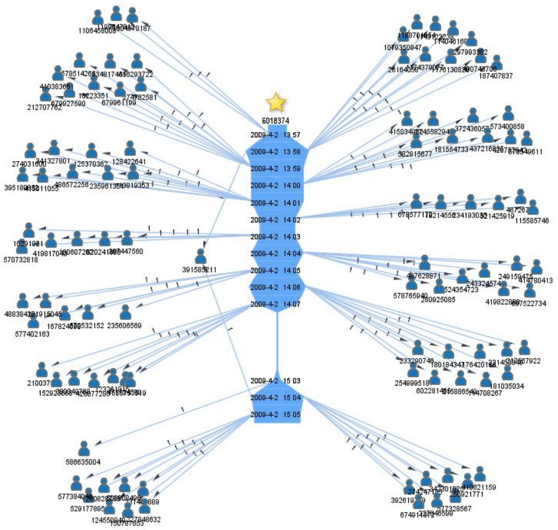


Figure 1: The dynamic short-message communication network central to a mobile phone spammer.

In practice, rather than showing the complicated dynamic network as a whole, a big class of network analysis tasks is only concerned with the dynamic network central to a focus node, such as analyzing a person's communication pattern or a company's collaboration pattern with other companies. For these analysis tasks, it is highly desirable to see the visual pattern of the focus node's dynamic network in a single glance. Figure 1 presents a visual display of the one central to a mobile spammer. As found in the graph, the spammer broadcasts advertisements in two continuous time periods with almost constant rate. Except for one mobile user receiving two messages, all the others just receive one during the analyzed time. Moreover, there is no message heading for the central spammer or

IEEE Pacific Visualisation Symposium 2011
1 - 4 March, Hong Kong, China
978-1-61284-933-1/11/$26.00 ©2011 IEEE

between the receivers.

As illustrated by Figure 1, we propose a third approach to the dynamic network visualization problem, i.e. to decouple the data in network/social domain. We compose a dynamic network central to one focus node by combining the static ego networks in each time point together, also known as dynamic ego network. With this approach, we are able to address the three challenges mentioned above: First, we encode ”time” into one dimension of the view space but do not impose strict mapping for the non-focus nodes. In other words, 1.5D freedom is provided for a visually aesthetic network layout (which gives the name to our solution). As a result, both the network and temporal affinity are properly accommodated in the traditional 2D view space for the ease of perception. Second, the complexity of the dynamic ego network is greatly reduced from the entire network. The visual clutters could be alleviated and graph layout is computed more efficiently. In some extreme cases, we adopt a customized edge bundling technique to further relieve the visual clutter. We also introduce a circular layout model to scale to the ego networks with up to a thousand nodes. Third, as the aggregated network is presented in a single static view, the analysis tasks for both the network and temporal pattern discovery are better supported. Also to assist the users in the analysis, we develop several effective interactions, such as the time-dimension navigation and focus node expand/collapse. This is demonstrated through two case studies in separate scenarios.

The major trade-off of our approach is to only show the ego networks evolving over time, which brings the most pros over the previous visualizations. Nevertheless, through our observations in industry, a large number of our customers' network-related analysis tasks can be satisfied with the ego-centric views. This paper does not aim at proposing a grand solution for all the problems, but to present an effective technology addressing a portion of specific tasks with many real-life cases. Moreover, our solution can be extended to handle dynamic networks of broader scope, including the event-based and multimodal dynamic networks.

The remainder of this paper is organized as follows. Section 2 summarizes related works. Section 3 introduces the data preprocessing stage that translates the raw network data into consumable format for the visualization. Section 4 details the 1.5D dynamic network visualization solution proposed in this paper, encompassing the visualization design, graph layout algorithms and user interactions for analytic tasks. Section 5 validates our solution through two case studies and reports insights gained with our approach. Section 6 discusses the future works. Finally in Section 7, we conclude the paper.

## 2 Related Work

General approaches to the dynamic network visualization fall into two categories. The first one animates the network with either fixed or dynamic node layout, also known as flip books and network movies [25]; the second category presents the evolving network spanning a period of time in a single static view by either trace visualization or small multiples display [30] juxtaposing networks during short time frames or at discrete time points. In this stage, it's still under debate which methodology performs better for the users. While some initial user studies [2] [18] do support the animated approaches, some others [30] argue that the animated approach is only appealing for presentation tasks and less efficient for analysis compared with the small multiples display. Also, the effect of change blindness [28] poses additional challenges to the animation solution.

In reality, most of the literatures take the first path. The key to their choices is that the animations with dynamic layout algorithms preserve the user's mental map [10][24]. These approaches can be further partitioned into two sub-classes. The first sub-class adopts an online approach that computes the network layout of one time frame only from the layout of the previous time and the delta change. They are often extended from the layout models of the static graph [21][15]. A practical approach adds an extra stability term to the Kamada-Kawai energy model [21] and calculates the layout result by balancing the layout aesthetics and stability [4]. The kind of online algorithms also incorporates techniques of incremental graph layout [26][27][8][14], where the primary goal is to stabilize the layout upon graph editing and browsing. On the other hand, there are offline approaches that take all the previous networks along the timeline into consideration [5][6][7]. They try to optimize the stability of the network movie in a global manner, at the cost of a much larger computation complexity. One piece of work in this sub-class proposes to encode the changes of network by a multi-layer presentation [5]. Besides the online/offline taxonomy, there are several other relevant layout algorithms for the dynamic drawing of special-purpose graphs, including the hierarchical [26][27], orthogonal [29][17], clustered [20][13][23] and radial [32] graphs, as well as trees [3].

Another collection of related works focus on the creation of animations for the network movie. Ref. [12] gives a set of basic criteria and a general framework to compose these animations. It is suggested that each intermediate graph layout during the animation should at best follow the typical graph drawing aesthetics. Further in [18], staged animations are recommended based on controlled experiments.

The dynamic network visualization can be considered as a special case of graph drawing problem, where numerous solutions [1][9][19] have been proposed. The most related works to this one are the force-directed and circular layout models. The force-directed solution emulates physics systems (either spring [9] or electrostatic [15] one) to achieve a best layout in system equilibrium. Optimization methods such as stress majorization [16] and multi-dimensional scaling [22] are often applied to compute the layout from the quantitative models [21][15]. In this paper, we adopt the Kamada-Kawai layout model with stress majorization solver [16] to handle graphs with medium size. For dynamic graphs in larger scales, we applied the circular (radial) layout model [32], which is originally proposed for arranging tree graphs.

## 3 Data Pre-processing

In this section, we describe how we process the data into more consumable format for our visualization. The raw data input is that of the dynamic network, formally defined by the time-evolving graph $G(t)$ where $t \in [0,T)$. $G(t)$ at each time point is further constituted of the vertex (node) set $V(t)$ and edge set $E(t)$. Note that the time series could be defined at any granularity. Each edge in $E(t)$ indicates the relationship that happens just in time $t$, not an accumulated one.

### 3.1 Dynamic Ego Network

To show the dynamic ego network in a single view, we process the data into one graph, denoted as $D(\Omega)$, where $\Omega$ denotes the focus node set (the single focus node is a special case of this model). It is generated in three steps.

**Slotting:** The first step is to discretize the dynamic networks. Given the ordered time series $[t_0, t_1, \cdots, t_k]$ where $t_0 = 0, t_k = T$, the slotted dynamic network graphs $G_S(t_i) = (V_S(t_i), E_S(t_i))$ is computed as

$$V_S(t_i) = \bigcup_{t \in [t_i, t_{i+1})} V(t), \quad i = 0, 1, \cdots, k-1 \tag{1}$$

$$E_S(t_i) = \bigcup_{t \in [t_i, t_{i+1})} E(t), \quad i = 0, 1, \cdots, k-1 \tag{2}$$

**Extraction:** The second step is to extract egocentric graph at each time slot, denoted as $H_S(\Omega, t_i)$. Its node set $V_S(\Omega, t_i)$ is exactly

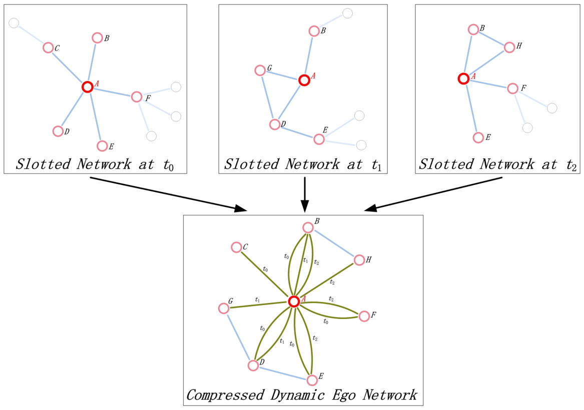


Figure 2: A example of the dynamic ego network compression step. The dynamic network encompasses three time slots $t_0$, $t_1$ and $t_2$. The network is compressed based on the focus node set $\Omega = \{A\}$, highlighted in red. In the generated graph, blue edges indicate time-independent edges and green edges indicate time-dependent edges where the label tells the timeslot attached to each edge.

the union of $\Omega$ and the nodes in $G_S(t_i)$ adjacent to at least one node in $\Omega$. The edge set $E_S(\Omega, t_i)$ contains all the edges in $G_S(t_i)$ that have both endpoints in $V_S(\Omega, t_i)$. $H_S(\Omega, t_i)$ is essentially an induced subgraph of $G_S(t_i)$, formally generated by

$$V_S(\Omega, t_i) = \Omega \cup \{v | v \in V_S(t_i) \wedge (\exists u, (v,u) \in E_S(t_i) \vee (u,v) \in E_S(t_i))\} \tag{3}$$

$$E_S(\Omega, t_i) = \{e = (v_1, v_2) | e \in E_S(t_i) \wedge v_1 \in V_S(\Omega, t_i) \wedge v_2 \in V_S(\Omega, t_i)\} \tag{4}$$

**Compression:** The third step is to compress the slotted egocentric graphs along the time line into a single dynamic egocentric graph $D(\Omega)$. The node set, denoted as $V_D(\Omega)$, is the union of node set $V_S(\Omega, t_i)$ at each time slot. The edge set, denoted as $E_D(\Omega)$, is constituted of two subsets. One is the time-independent edge set computed as the union of edges not incident to any nodes in the focus node set. The other is the time-dependent edge set containing all the edges incident to the nodes in the focus node set. The latter is a little different from the standard definition in that the edges with the same source and destination but happen in separate time is considered as different edges. The dynamic egocentric graph $D(\Omega)$ is formally generated by

$$V_D(\Omega) = \bigcup_{i \in [0, k-1]} V_S(\Omega, t_i) \tag{5}$$

$$E_D(\Omega) = \overline{E_D}(\Omega) \cup \widetilde{E_D}(\Omega) \tag{6}$$

where

$$\overline{E_D}(\Omega) = \{e = (v_1, v_2) | v_1 \notin \Omega \wedge v_2 \notin \Omega \wedge (\exists t_i, e \in E_S(\Omega, t_i))\} \tag{7}$$

$$\widetilde{E_D}(\Omega) = \{e = (v_1, v_2, t_i) | (v_1 \in \Omega \vee v_2 \in \Omega) \wedge (v_1, v_2) \in E_S(\Omega, t_i)\} \tag{8}$$

Figure 2 gives an illustrative example of the compression step.

### 3.2 Event-based Dynamic Networks

Our data model can be extended to the event-based dynamic networks. In such a network, each edge indicates that the event involving the two endpoints has occurred. Examples include the paper co-authorship network of a conference where each event represents a paper publication in this conference. For the effective visualization, we process the dynamic network data into event-centric graph $D(\Gamma)$, where $\Gamma$ denotes the kind of event we want to visualize. Each single event is denoted as $(\Gamma, \psi)$, where $\psi$ denotes the event ID for decoupling purpose. The edge $e$ representing event $(\Gamma, \psi)$ is denoted as $e \sim (\Gamma, \psi)$.

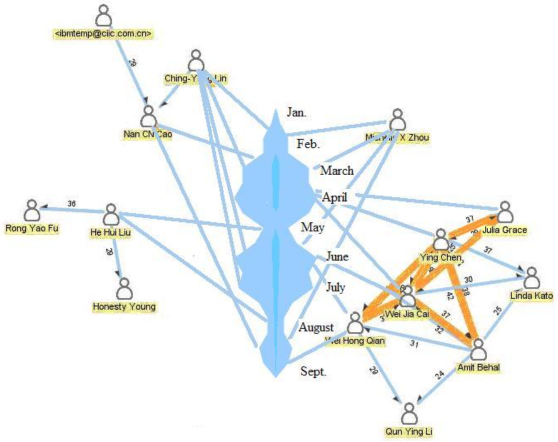


Figure 3: Dynamic network visualization in 1.5D: the sketch view.

The graph generation process still includes the three steps in Section 3.1, except that we add a graph insertion step in the beginning.

**Insertion:** For the graph $G(t)$ at each time point, $\Gamma$ is added to the node set as an abstract node. For the edge set, edges are added for each node in $V(t)$ such that it connects to the edges representing events in $\Gamma$. The graph insertion step is formally defined as

$$V(t) = V(t) \cup \{\Gamma\} \tag{9}$$

$$\begin{aligned} E(t) = E(t) \cup \{(v, \Gamma, \psi) | v \in V(t) \\ \wedge (\exists u, (u,v) \sim (\Gamma, \psi) \vee (v,u) \sim (\Gamma, \psi))\} \end{aligned} \tag{10}$$

In the following slotting, extraction and compression steps, $\{\Gamma\}$ is selected as the focus node set. There is one difference from Section 3.1, two edges $(v, \Gamma, \psi_1)$ and $(v, \Gamma, \psi_2)$ are merged together only if $\psi_1 = \psi_2$.

### 3.3 Multimodal Networks

Another extension of our model is the multimodal network. Relationships in such network can be classified into several categories according to one selected dimension $\Lambda$, such as the communication type in people's social network. In this case, the network could be decoupled into sub-networks connected by emails, phone calls, and instant messages respectively. To visualize such multimodal networks, we replace the default time dimension in the 1.5D visualization with the selected dimension $\Lambda$.

## 4 1.5D Dynamic Network Visualization

### 4.1 Visualization Design

A sketch of our visualization is given in Figure 3, showing a dynamic email network with one focus person. It follows the traditional network visualization paradigm with nodes and edges. Compared with the static network visualization, the major differences lie in the focus nodes and the time-dependent edges rendering.

The focus nodes in our network are not drawn in simple icons or glyphs. Instead, we introduce a two-sided trend visualization metaphor to present their temporal data. The trend is shown in a symmetric way such that the non-focus nodes could be placed in either side of the central trend. By default, the width of the trend at a time slot represents the total strength of the edges connected to the focus node at this time slot. Further, the visualization supports stacked trend view which presents multiple temporal data. E.g., upon highlighting an inner red stack could correspond to the sending mail trend while the outer green stack shows the receiving mail trend. Apart from the vertical trend metaphor, there are several other alternatives to show the trends of the focus node, e.g. the horizontal trend and the spiral glyph.

The rendering of the time-independent edges in our approach is the same with the previous network visualizations. However, the time-dependent edges are drawn separately. Each such edge connects the non-focus node to the specific brim of the central focus node representing the exact time slot this edge establishes. Bidirectional edges are distinguished with the orange color, and upon a highlight operation, the edge further splits into two edges, with the red ones indicating the sending edges and the green ones indicating the receiving edges.

## 4.2 Graph Layout

In this part, we propose the layout algorithms for the dynamic ego networks with only one focus node. By default, the central trend is placed vertically at the center of the view, partially mapping the Y axis to the time dimension in the data. The goal for the layout of the non-focus nodes is to represent both its temporal affinity to the focus node and the topological characteristic within the network. We introduce a mixed force-directed layout model to satisfy these requirements.

### 4.2.1 Force-directed Layout Model

In this layout model, the focus node $p$ is virtually split by the time slot to several sub-nodes $\{p_0, p_1, \cdots, p_{k-1}\}$, so that the time-dependent edges could be converted into time-independent ones. Consider the dynamic graph $D(\Omega) = (V_D(\Omega), E_D(\Omega))$ where $\Omega = \{p\}$, the new graph for layout model, denoted as $L = (V_L, E_L)$, is generated by

$$V_L = \{v | v \in D(\Omega) \wedge v \neq p\} \cup \{p_0, p_1, \cdots, p_{k-1}\} \tag{11}$$

$$\begin{aligned} E_L = \overline{E_D}(\Omega) &\cup \{(v, p_i) | v \neq p \wedge (v, p, t_i) \in \widetilde{E_D}(\Omega)\} \\ &\cup \{(p_i, v) | v \neq p \wedge (p, v, t_i) \in \widetilde{E_D}(\Omega)\} \end{aligned} \tag{12}$$

We apply a stable layout algorithm to compute the node placement. This algorithm defines energy functions over the graph. By solving the energy minimization problem, the final layouts are optimized, which maximizes the layout aesthetic. Different from the classical force-directed model, our energy function consists of two terms. The first term implements the Kamada-Kawai energy function [21] and the second term is the stable function encodes the temporal constraint to the nodes. It is formally written as

$$\begin{aligned} F = (1-\alpha) \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} (\|X_i - X_j\| - d_{ij})^2 \\ + \frac{(n-1)\alpha}{2} \sum_{i=1}^{n} \mu_i \|X_i - X_i'\|^2 \end{aligned} \tag{13}$$

where $X_i$ denotes the position of $i$th node in graph $L$, $d_{ij}$ defines the optimal distance between $i$th node and $j$th node, $\omega_{ij}$ and $\mu_i$ are the parameters controlling the weight of each node (pair), $X_i'$ denotes the desired position of $i$th node to represent temporal information, $\alpha$ denotes the stability factor balancing the network and temporal graph aesthetics.

In our case, the position of sub-nodes split from the focus node is fixed. Hence we remove them from the energy function by setting

$$\omega_{ij} = \begin{cases} 0 & i\text{th and } j\text{th nodes are both sub-nodes of } p \\ d_{ij}^{-2} & \text{otherwise} \end{cases}$$

$$\mu_i = \begin{cases} 0 & i\text{th node is sub-node of } p \\ \|X_i' - \overline{X'}\|^{-2} & \text{otherwise} \end{cases} \tag{14}$$

where $\overline{X'} = (\bar{x}, \bar{y})$ denotes the center of the trend glyph.

The desired position of each non-focus node is set on the circumference with the center of trend glyph as circle centre. For unweighted graph $L$, the angular position is mapped from the average time value of edges connected to the focus node $p$, and the radius is inversely proportional to the total number of edges to $p$. Given the non-focus node $v_i$ in graph $L$, $X_i' = (x_i, y_i)$ is computed by

$$\begin{aligned} x_i &= \bar{x} + syn(v_i)\rho \cos\theta \\ y_i &= \bar{y} + \rho \sin\theta \\ \rho &= \frac{\rho_0}{\|\{j | (v_i, p_j) or (p_j, v_i) \in E_L\}\|} \\ \theta &= \theta_0 + \frac{(\theta_{k-1} - \theta_0)(s_i - t_0)}{t_{k-1} - t_0} \\ s_i &= \overline{t_j}, \forall j, (v_i, p_j) or (p_j, v_i) \in E_L \end{aligned} \tag{15}$$

where $syn(v_i)$ is the signal function indicating whether the node is on the left (-1) or on the right (1), $\rho_0$ denotes the maximal node distance from the centre, $\theta_0$ and $\theta_{k-1}$ denote the two boundary angles from the centre, set to $\frac{\pi}{2}$ and $-\frac{\pi}{2}$ respectively. We implement a uniform graph bisection algorithm to set the signal function for each node. The case with weighted graph is handled slightly differently by adding weights to the computation of (15).

### 4.2.2 Circular Layout Model

The force-directed model could support the layout of ego graph up to a hundred nodes with little computation time. However, in case the dynamic graph scales or the time granularity is set to rather fine level, the generated ego graph could be too complicated to layout with force-directed model. Also, the visualization may become too messy for human to perceive. Here we propose another layout model based on circular node placement for the ego graph of large scale and also introduce edge bundling method to alleviate the visual clutters.

This model determines the layout in polar coordinate only from the temporal relationship with the focus node. The centre of the polar coordinate is still set as the center of the trend glyph. The radius of each node is inversely proportional to the number of edges connecting to the focus node. The computation of node angle involves three steps.

**Partition:** All the non-focus nodes are divided into two subsets placed in the left and right region of the central trend. The default partition method separates the nodes with only one edge to the focus node from the others with multiple edges to the focus. Other partition methods could also be applied for customized comparison purpose.

**Sort:** The average time of each non-focus node is calculated as (15). Then for each subset generated in last step, the nodes are sorted according to this average time with rank starting from zero.

**Assign:** The node $v_i$ with rank $r_i$ in subset $S$ is assigned the angle value $\theta_i$ by

$$\theta_i = \theta_0 + \frac{r_i(\theta_{k-1} - \theta_0)}{\|S\| - 1} \tag{16}$$

A layout example using the circular model is shown in Figure 6 where the temporal patterns are shown clearly. To further alleviate the visual clutter caused by the huge amount of edges in the generated dynamic network, we introduce bundled curve edge drawing to highlight the connection trends. Figure 6(b) shows the effect of edge bundling.

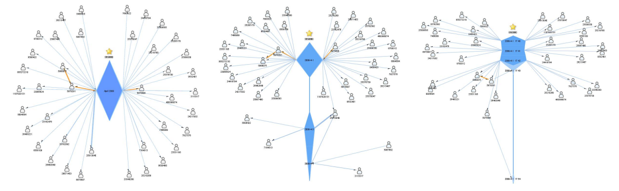
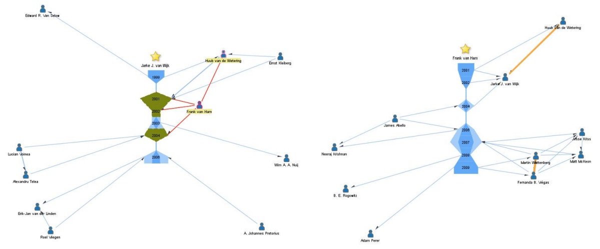

(a) (b) (c)

Figure 4: 1.5D Network Visualization with different timelines: (a) Slotted by month; (b) By day; (c) By minute.



(a) (b)

Figure 5: Consecutive views during egocentric graph navigation in Infovis co-author network: (a) Network central to Prof. Jark van Wijk with Dr. Frank van Ham moused over; (b) Switch to the network central to Dr. Frank van Ham by clicking the node.

### 4.3 User Interactions for Analysis

In addition to the common interactions in the static network visualizations, such as zoom&pan, node selection, highlighting, dragging and node/edge filtering, we also develop a few customized interactions to assist the user's dynamic network analysis tasks.

**Timeline Navigation:** Since the dynamic graph data is processed into time slots, changing the slotting granularity will result in quite different views of the same data. Inspired from the geometric zoom-in/zoom-out interaction, we develop the timeline zoom navigation which allows the user to select an interested timeline area and zoom-in to show the network with a finer time slotting value for analysis. Oppositely, the user could zoom-out to a high-level timeline with a coarser slotting value for overview purpose.

Figure 4 gives a detailed example. In Figure 4(a), the network is shown by month where only one month data is available. As the user zooms to the day level, it is quickly discovered that the behavior of the focus person can be divided into two periods: April 1st and April 3rd $\sim$ 4th. As he further selects the April 1st and zooms to the minute level, the pattern of a constant burst within three minutes are finally located.

**Egocentric Graph Navigation:**

The major trade-off of our solution is to only show the egocentric dynamic network. Also, the temporal information associated with the time-independent edges can not be visualized in the approach. These limitations are mitigated by allowing the user to navigate across many egocentric graphs through interactions. Upon double-click of one node outside the central trends, the dynamic network view switches smoothly to the new network central to the selected focus node, as illustrated by Figure 5. We provide another kind of navigation by double-selecting an extra node (ctrl+double-click). After that, the new focus node will be the cluster consisting of both the old and new focus nodes. The rubber-band selection is also supported to choose the focus cluster.

**Graph Overlap Highlighting:**

We provide graph comparison functionality through this interaction. By default setting, the neighboring nodes of a selected node will be highlighted. Further, the overlapped neighbors of several selected nodes will be drawn with larger size, differentiating the overlapped subgraph with the others. This allows the graph evolvement analysis across timeline. Visual examples can be found in Figure 9.

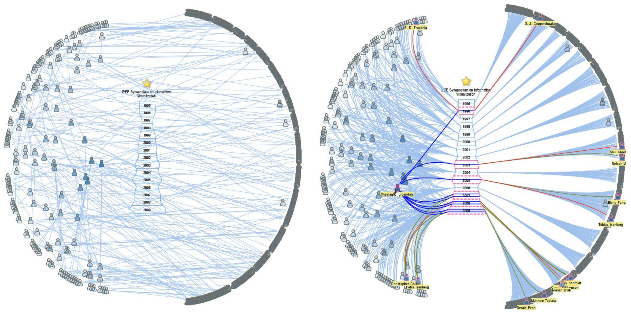

(a) (b)

Figure 6: Event-based network visualization with only time-independent edges: (a) Without bundling; (b) After bundling.

**Event-based Edge Bundling:**

This interaction is specially designed for the event-based dynamic network. In the default view, the edges are drawn in a straight-line manner, hence introduces significant visual clutters, as shown in Figure 6(a). Given the fact that, in the event-based dynamic network, each edge is associated with an event happening at a specific time. Upon this interaction, we could bundle the edges with events happening in the same time slot together by letting them go through the center of the trend glyph at this time slot. Moreover, when the user mouse-overs one node for its connections, the incident edges bundled together with different events are decoupled in the view to reflect the details. In the example shown by Figure 6(b), the overall visual clutter is alleviated and the Prof. Carpendale's connection patterns are highlighted in detail. She published 9 papers with 13 co-authors during the history of Infovis conference.

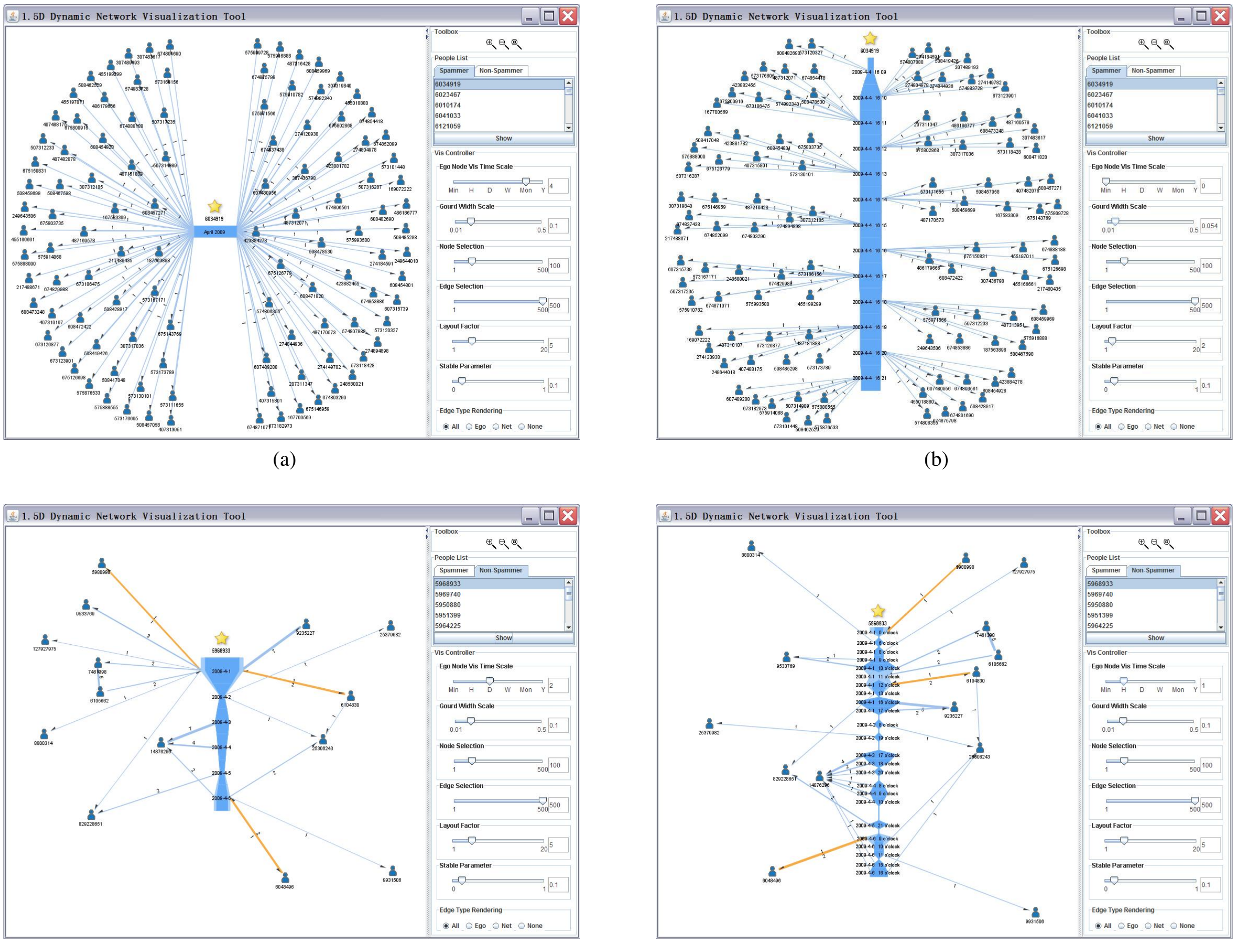
## 5 Case Studies

Visualization can help the data analysis tasks in two ways: Serving as visual evidence through which the underlying analytic methods (e.g. data mining algorithms) are validated; Enabling explorative pattern discovery through interactive data navigation. We present two case studies of our dynamic network visualization approach covering the both aspects respectively.

### 5.1 Communications Network

In this case, we visualize a communications network built from the real data set collected by telecom service providers. Each node in the network corresponds to one mobile phone user and the edges between them indicate the short messages sent/received. The resulting network is essentially a dynamic social network during the time period of the data. The background of this case is that we have developed a data mining system [31] which could capture the mobile users that spam through the characteristics of the social graph. However, it is still important for service providers and researchers to evaluate whether the system works, and in case it fails to find reasons. We herein design this dynamic visualization tool to help them.

Consider Adam, an analyst of the telecom service provider, who is checking the spammer/non-spammer classification results from the online mining system. He starts by selecting one spammer in the list and accessing its dynamic ego network with our tool. As in Figure 7(a), the ego network slotted by month shows up a simple pattern that the potential spammer sends out one message to quite a lot neighboring nodes without receiving any message from them. Also



(a) (b) (c) (d)

Figure 7: 1.5D dynamic network visualization in communications scenario: (a) A typical spammer behavior slotted by month; (b) Slotted by minute; (c) A typical non-spammer behavior slotted by day; (d) Slotted by hour.

there is no communications between the neighbors of the spammer which indicates an extremely abnormal social network. This observations correspond to the classification features used in the mining system for a spammer: high outbound degree but low inbound degree, low average outbound edge weight, high sending/receiving ratio and low clustering coefficient in the 2-hop network. Adam further uses our tool to drill-down to more details by changing the slotting granularity to minute, as shown in Figure 7(b). This time the temporal pattern of the spammer behavior is also located. The spammer tends to send messages out in a rather constant frequency within a short bursty time, 9 messages per minute for 12 consecutive minutes in this case. Also, there is no neighbor who connects to the spammer in more than one time slot. This corresponds to the temporal classification feature of bursty smooth sending.

In his second trial, one non-spammer identified by the mining system is selected. Figure 7(c) shows his ego graph slotted by day, where the orange edge indicates a bidirectional connection and the edge thickness indicates the number of messages represented by the edge (also shown as a edge label). In this graph, opposite patterns to the spammer's are discovered: between the potential non-spammer and his neighbors, there are both inbound, outbound, and bidirectional edges; the number of messages exchanged is normally larger than one; there are communications between the neighbors; the ego graphs are overlapped in different time slots; the sending/receving trend of the non-spammer has no significant temporal pattern. Drilling-down to the hour granularity, as shown in Figure 7(d), more details are revealed. Although there is few pattern to discover, more clues can be found for personal communications. E.g., some has a double-handshake like contact within an hour and some others receive a lot of messages continuously from the same person without a reply. This is more useful for scenarios such as crime network analysis.

## 5.2 Academic Co-author Network

In this part, we present the case study of our solution over paper co-authorship dynamic network. The data set is from the Infovis 2004 contest [11]. We further spend some time to clean and extend it to cover all the years of Infovis (1995∼2009). The entries not belonging to the conference are removed. Finally it involves 673 unique authors and 356 papers published.

Different from the first case where we show the visualization central to the selected node, here we present a variant of our solution which visualizes event-based dynamic networks. The entire network is organized central to the core event, the Infovis conference series. The edges between the author and conference indicate publication relationship, while the edges between the authors are still co-authorship relationship.

Figure 8 gives an overview to the 15-year history of Infovis. The central trend glyph encodes the number of authors involved each year. The nodes to the left of the central trend represent the authors

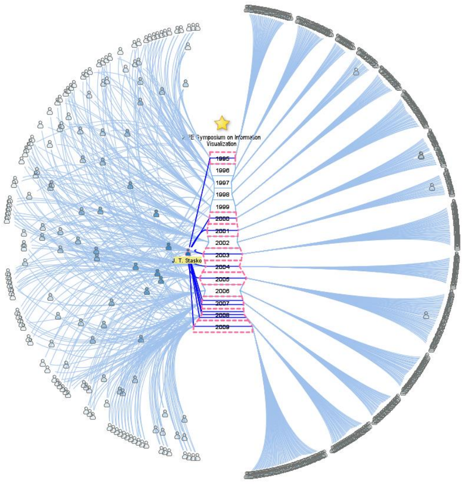


Figure 8: Infovis co-authorship network 1995∼2009. The node representing "Prof. Stasko" is highlighted, with only ego edges drawn.

that have paper in at least two separate years in Infovis, while the nodes to the right represent the authors that only publish Infovis paper in one year. The distance between the author and the center of the trend glyph encodes the number of papers he publishes in Infovis, the closer the more publications. The left-right node division could be customized in our solution, however, throughout this case we stay with the original design for the ease of analysis.

Consider John, a newcomer to the information visualization area, who wants to know more about the researchers and collaborations in this area. He starts to diagnose from the overview picture (Figure 8). He quickly finds that the node density to the right is much higher than the density to the left. It tells him that the community is friendly enough to accept papers from newcomers. Further he locates the most senior researchers from this graph, the one closest to the center, also shown in Figure 8. It's Prof. Stasko who publishes totally 12 papers in 15 years of Infovis. Significantly, he is the only one who appears in both the founding event of Infovis in 1995 and the latest one in 2009. In Figure 8, only ego edges, i.e. the edges between the central event and the authors, are drawn to reduce visual complexity. The other edges, defined as network edges, i.e. the edges between the authors, will be rendered in case the graph is switched to the network-edge mode.

John continues his discovery by navigating through the subnetwork in each Infovis conference. He finds some clues that the researchers in the early years are quite different from those still active in recent years. He tries to test this hypothesis with our subgraph highlighting interactions. He clicks on the years from 1995 to 1999 to show the subnetwork of the authors in this time period and grey out the others, the result is shown in Figure 9(a). The graph overlap highlighting mechanism works here so that the authors appearing more in this period will be drawn with larger icons and labels. John iterates the similar operations for the years 2000∼2004 and 2005∼2009, the resulting networks are shown in Figure 9(b) and Figure 9(c). It could be found that the key researchers in the three periods are significantly different, which indicates that the Infovis community is fast-evolving, with a cycle of approximately 5 years. He further validates his findings by setting the slotting value to 5 years. The graph generated is shown in Figure 9(d). The overlaps between the subnetworks of 1995∼1999 and 2005∼2009 accounts to around 20 researchers (those in larger label), which is a small portion compared to the 200 unique researchers evolved in each 5-year period.

John further leverages our tool to mine the collaboration patterns of some senior researchers. He switches to the network-edge mode to learn the relationships between authors. He first clicks on the node representing Prof. Munzner to access her personal collaboration network. He finds that Prof. Munzner likes to conduct research with newcomers (maybe her students) which are placed in the right part of the graph. Also her Infovis career may be divided into two periods. The first period is from 1996 to 1999 when she published first-author papers (maybe with her supervisors), indicated by the red edges. The second period is from 2004 to 2009 when her edges are mostly in green, which indicates that her name is often in the last few of the author list. It could be inferred that she switched to supervisor role during this period. John then turns to another senior researchers, Dr. Wong from PNNL, and finds a quite different pattern. The co-authorship edges of Dr. Wong are almost in red, which indicates that he leads most research projects during his career. Also it's notable that Dr. Wong likes to collaborate with the other senior researchers in this area. Most of his collaborators are located in the left part of the graph.

## 6 Discussions

We plan to extend our work in three directions. First, we will explore the layout algorithm with new cluster heuristics to preserve the graph clustering nature. We now partially reduce the clustering of the drawing, because we encode both the time and network information into the view space in a balanced manner. As the clustering may affect the temporal information encoding, we will develop new edge bundling technique to enhance the temporal pattern of clusters with respect to the focus node.

Second, we will try the alternative to use the spiral glyph as the temporal representation of the focus node. This will fasten the discovery of periodic temporal patterns. We will design new layout algorithms mapping time and network affinity to the polar coordinate system to cope with this new visual metaphor.

Third, we will apply our solution to multimodal network, where the central glyph represents the values of multi-dimensional data. As the data type is not necessarily ordinal, we will study the best ordering algorithm to maximize the aggregated graph aesthetics.

## 7 Conclusion

In this paper, we propose a new dynamic network visualization technique by showing the dynamic network central to a focus node each time. This approach accommodates both the temporal and network information within a traditional 2D view space, supports huge dynamic networks free of scale, and aggregates the time-evolving networks into a single static view well supporting analysis tasks. Through two case studies, we show that our approach is effective as both visual evidences and explorative tools for interactive temporal and network pattern discovery.

## References

[1] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

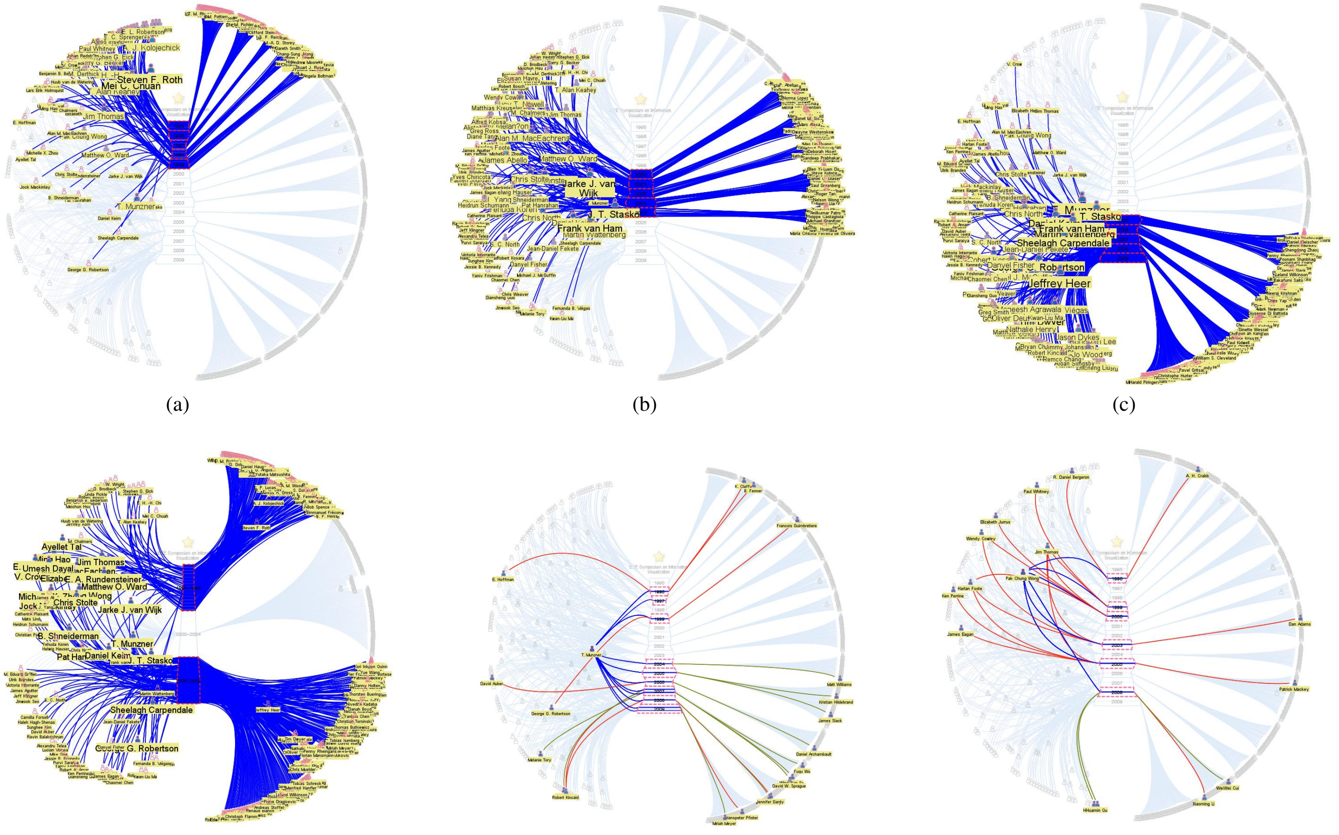
[2] B. Bederson and A. Boltman. Does animation help users build mental maps of spatial information? In *Infovis '99*, 1999.

[3] C. Binucci, U. Brandes, G. D. Battista, W. Didimo, M. Gaertler, P. Palladino, M. Patrignani, A. Symvonis, and K. A. Zweig. Drawing trees in a streaming model. In *Proceedings of the Symposium on Graph Drawing*, pages 292–303, 2009.

[4] K. Boitmanis, U. Brandes, and C. Pich. Visualizing internet evolution on the autonomous systems level. In *International Symposium on Graph Drawing*, pages 365–376, 2007.

[5] U. Brandes and S. R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. In *Infovis '02*, 2002.

[6] C. Collberg, S. 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*, 2003.



(a) (b) (c)

(d) (e) (f)

Figure 9: Infovis co-authorship network: (a) Year 1995∼1999 highlighted; (b) Year 2000∼2004 highlighted; (c) Year 2005∼2009 highlighted; (d) Two periods highlighted, slotted by 5-year; (e) "Prof. Munzner" ego network; (f) "Dr. Wong" ego network

[7] S. Diehl, C. Görg, and A. Kerren. Preserving the mental map using foresighted layout. In *IEEE TCVG Symposium on Visualization (VisSym'01)*, 2001.

[8] T. Dwyer, Y. Koren, and K. Marriott. IPSep-CoLa: an incremental procedure for separation constraint layout of graphs. In *Infovis '06*, 2006.

[9] P. Eades. A heuristic for graph drawing. In *Congressus Numerantium*, volume 42, pages 149–160, 1984.

[10] P. Eades, W. Lai, K. Misue, and K. Sugiyama. Preserving the mental map of a diagram. In *COMPUGRAPHICS '91*, pages 34–43, 1991.

[11] J.-D. Fekete, G. Grinstein, and C. Plaisant. IEEE InfoVis 2004 Contest, the history of InfoVis, http://www.cs.umd.edu/hcil/iv04contest.

[12] C. Friedrich and P. Eades. Graph drawing in motion. *Journal of Graph Algorithms and Applications*, 6(3):353–370, 2002.

[13] Y. Frishman and A. Tal. Dynamic drawing of clustered graphs. In *Infovis '04*, 2004.

[14] Y. Frishman and A. Tal. Online dynamic graph drawing. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):727–740, 2008.

[15] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software, Practice & Experience*, 21(11):1129–1164, 1991.

[16] E. R. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In *International Symposium on Graph Drawing*, 2004.

[17] C. Görg, P. Birke, M. Pohl, and S. Diehl. Dynamic graph drawing of sequences of orthogonal and hierarchical graphs. In *International Symposium on Graph Drawing*, pages 228–238, 2004.

[18] J. Heer and G. G. Robertson. Animated transitions in statistical data graphics. In *Infovis '07*, 2007.

[19] I. Herman, G. Melancon, 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.

[20] M. L. Huang, P. Eades, and J. H. Wang. On-line animated visualization of huge graphs using a modified spring algorithm. *Journal of Visual Languages & Computing*, 9(6):623–645, Dec. 1998.

[21] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, 1989.

[22] Y. Koren, L. Carmel, and D. Harel. ACE: A fast multiscale eigenvector computation for drawing huge graphs. In *Infovis '02*, 2002.

[23] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. In *Infovis '06*, 2006.

[24] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183–210, June 1995.

[25] J. Moody, D. McFarland, and S. Bender-deMoll. Dynamic network visualization. *American Journal of Sociology*, 110(4):1206–1241, 2005.

[26] S. C. North. Incremental layout in DynaDAG. In *International Symposium on Graph Drawing*, pages 409–418, 1995.

[27] S. C. North and G. Woodhull. Online hierarchical graph drawing. In *International Symposium on Graph Drawing*, pages 77–81, 2002.

[28] L. Nowell, E. Hetzler, and T. Tanasse. Change blindness in information visualization: a case study. In *Infovis '01*, 2001.

[29] A. Papakostas and I. G. Tollis. Interactive orthogonal graph drawing. *IEEE Transactions on Computers*, 47(11):1297–1309, 1998.

[30] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

[31] C. Wang, Y. Zhang, X. Chen, Z. Liu, L. Shi, G. Chen, F. Qiu, C. Ying, and W. Lu. Spamwatcher: A behavior-based SMS anti-spam system. *IBM Journal of Research & Development*, in press.

[32] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *Infovis '01*, 2001.