1.5D Egocentric Dynamic Network Visualization

Lei Shi, Chen Wang, Zhen Wen, Huamin Qu, Chuang Lin, and Qi Liao

Abstract—Dynamic network visualization has been a challenging research topic due to the visual and computational complexity introduced by the extra time dimension. Existing solutions are usually good for overview and presentation tasks, but not for the interactive analysis of a large dynamic network. We introduce in this paper a new approach which considers only the dynamic network central to a focus node, also known as the egocentric dynamic network. Our major contribution is a novel 1.5D visualization design which greatly reduces the visual complexity of the dynamic network without sacrificing the topological and temporal context central to the focus node. In our design, the egocentric dynamic network is presented in a single static view, supporting rich analysis through user interactions on both time and network. We propose a general framework for the 1.5D visualization approach, including the data processing pipeline, the visualization algorithm design, and customized interaction methods. Finally, we demonstrate the effectiveness of our approach on egocentric dynamic network analysis tasks, through case studies and a controlled user experiment comparing with three baseline dynamic network visualization methods.

Index Terms—Graph visualization, 1.5D visualization, dynamic network, egocentric abstraction

1 Introduction

Dynamic networks are networks that exhibit time-varying relationships as well as node and edge attributes that change over time. Important insights can be obtained through the overview, browsing and analysis of a dynamic network in the visual form. For example, in a telecom service provider, domain experts routinely check the dynamic communication network to validate the misbehavior of suspected mobile spammers [1]. In an academic scenario, a new researcher wants to study the dynamic collaboration network of a visualization fellow to discover the most influential people in her recent research activities. Though there are always analytical methods that can automatically uncover specific features, the unique strength of the visualization method is to synthesize large amount of data and reveal interesting patterns that warrant further analytical investigation. On dynamic networks, the need for novel visualizations is more critical due to the heterogeneity in the topology and temporal aspect of the network data [2], [3].

Historically, the visualization of large dynamic networks is a well-known hard problem [4]. First, new visual designs should probably be invented beyond the traditional node-link graph representation [5], [6], [7], [8] to incorporate the additional time dimension. Second, scalability issues of the visualization must be considered as the size of a dynamic network can increase significantly over time. Existing methods often introduce data reductions in the time dimension, and snap together multiple temporal views into network movies [2]; however the animation approach to display network movies is shown to be ineffective for network analysis tasks [9], [10]. Third, over the visualization design, the interaction methods to explore a dynamic network, e.g. filter and drill-down to obtain local features, are extremely valuable in the analysis process.

In this paper, unlike previous works that consider the network structure in full scale, we target a subset of dynamic network analysis tasks that take one network node as the focus and require looking at only the dynamic network central to the focus node, also known as the egocentric dynamic network. According to the taxonomy of network visualization tasks [11], this work is motivated by two types of low-level tasks frequently observed on egocentric dynamic networks: 1) checking the dynamic adjacencies between the focus node (aka the ego) and non-focus nodes (aka the alters) over time, including their strength, frequency, periodicity and directionality; 2) diagnosing the connectivity among non-focus nodes with respect to their dynamic adjacencies to the focus node, e.g. the dynamic community structure, bridges and hubs among non-focus nodes. In contrast, the method proposed here is not designed for the attribute-based, overview and browsing tasks of the entire network, though our method supports the overview and browsing of the egocentric dynamic network.

In more detail, we propose a new visualization design, namely the 1.5D dynamic network visualization (1.5D), based on the egocentric data reduction of the dynamic network (Section 3). As shown in Fig. 1, the key visual metaphors are the temporal trend glyph in the center to replace the trivial representation of the focus node, and the glyph's affiliated multiple edges carrying temporal information. All the other non-focus nodes and the edges among them remain the same as those of a simple node-link graph. The

- *L. Shi is with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. E-mail: shil@ios.ac.cn.*
- *C. Wang and Z. Wen are with IBM Research. E-mail: wangcwc@cn.ibm.com, zhenwen@us.ibm.com.*
- *H. Qu is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. E-mail: huamin@cse.ust.hk.*
- *C. Lin is with the Department of Computer Science and Technology, Tsinghua University. E-mail: chlin@tsinghua.edu.cn.*
- *Q. Liao is with the Department of Computer Science, Central Michigan University. E-mail: qi.liao@cmich.edu.*

Manuscript received 19 Sept. 2013; revised 26 Sept. 2014; accepted 9 Dec. 2014. Date of publication 17 Dec. 2014; date of current version 1 Apr. 2015. Recommended for acceptance by J.-D. Fekete.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2014.2383380

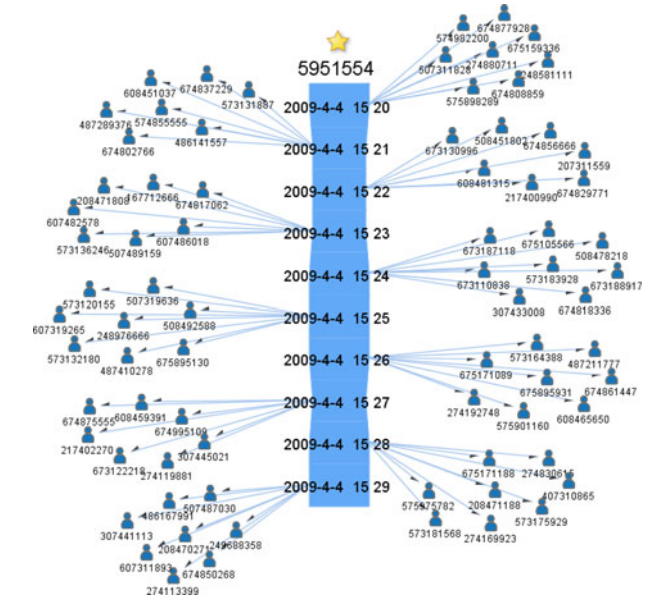


Fig. 1. The dynamic short-message communication network central to a mobile phone spammer. The spammer broadcasts messages in a continuous and constant rate, highly suspected of being the advertising behavior. Each non-focus user receives only one message from the spammer during the process, without any message sending to the spammer or among the non-focus users.

resulting visualization inherits the intuitiveness of a graph representation, while accommodating both topological and temporal information in the traditional 2D view space. Notably in this design, we encode "time" into one dimension of the view space (along the trend glyph) but do not impose a strict layout mapping on non-focus nodes. In other words, 1.5D freedom is provided for a visually aesthetic network layout, hence the name of the approach. More description of the 1.5D design is given in Section 4.

Despite the conciseness of our design, it is nontrivial to compute and accomplish a 1.5D visualization. In addition to a preliminary version of this approach [12], we further introduce three kinds of trend glyphs to be integrated into the visualization, and we discuss their selection criteria. An optimized, force-based algorithm is proposed to calculate the layout for the egocentric dynamic network, along with another radial layout model suitable for the larger event-centric dynamic network. Several customized interactions are introduced in the context of egocentric dynamic network analysis tasks. We describe two case studies in Section 5 and report one controlled user experiment in Section 6 comparing the 1.5D approach to baseline dynamic network visualization methods. The results on both objective task performance and subjective user feedback show a clear advantage of the 1.5D approach in the egocentric dynamic network analysis scenario.

2 Related Work

Traditionally, dynamic network visualization was studied on the problem of incremental node-link graph drawing, especially on specific types of graphs such as trees [13], series-parallel graphs [14] and directed acyclic graphs [15]. In DynaDAG [15], the research problem was summarized as how to maintain stability across consecutive views, i.e., preserving the user's mental map [16], [17]. Two categories of stable graph drawing algorithms had been developed. In the first category were online drawing approaches [18], [19], [20] which computed the graph layout of one time slot from the layout of the previous time slot and the delta graph change. In the second category were offline stable graph drawing algorithms, which took all the graph sequences along the timeline into consideration [21], [6], [22]. Meanwhile in SoNIA [5], two methods to display network dynamics, i.e. the static flip book and the dynamic movie, were proposed for the usage in different contexts.

While the above mentioned methods drew the dynamic network in a 2D node-link representation, more versatile visual metaphors had also been proposed in the literature. Brandes and Corman described a method to unroll the dynamic network into a 3D graph visualization [23]. Yi et al. proposed TimeMatrix [24], which visualized the temporal metrics of a dynamic network in the adjacency matrix by incorporating the TimeCell glyph. Hao et al. applied treemaps to visualize time-varying data over static hierarchies [25]. A similar hybrid approach combined the hierarchical tree layout with the timeline visualization to present the dynamic hierarchical data [26]. TimeRadarTrees [27] is another novel visual metaphor to visualize general dynamic networks. Parallel Edge Splatting [28] introduced the parallel coordinate design to the dynamic network visualization problem. Farrugia et al. [29] studied the similar problem of temporal ego network visualization. They proposed an interesting tree-ring layout in which the time was encoded into multiple concentric circles from the ego node. The alters were replicated at each active time slot and placed equidistantly on the ring. Compared to our 1.5D approach, the tree-ring layout is more compact so that each temporal ego network can be drawn as a motif to construct small multiples for the visual comparison of different ego nodes. In contrast, the 1.5D design requires more space, but is more intuitive because of the non-replicated node-link graph metaphor. Moreover, our design can better illustrate the network structure due to the 1.5D freedom on the layout. In this sense, the 1.5D approach is more suitable for the in-depth analysis of one single egocentric dynamic network.

Scalability is another key issue in visualizing dynamic networks. In the literature, only a few methods proposed to visualize the large dynamic network in full scale under the node-link representation. One exception was the small multiple display (SMD) [30, pp. 67-80] which juxtaposed networks at each time slot in the same view. Essentially a large screen is required to dilute the visual complexity, which limits their usage. In contrast, most other methods looked at the data aspect and employed some kind of data reduction to alleviate the visual complexity. Hadlak et al. gave a taxonomy of the data reduction method on dynamic network visualizations [2]. The first class of methods considered the time domain, either selecting a portion of time slots or abstracting the time into aggregated slots. Only the network of one single/aggregated time slot was drawn at each view. Multiple views were snapped together into a network movie and displayed by animations. The animation approach [31], [32], [5] offered a pleasant viewing experience for the audience, but in general its effectiveness was challenged in the recent research [33]. Experiments

comparing the animation approach with the static multiple displays [9], [10] revealed that users required more time to understand the dynamic network with the animation approach. The root cause of this slower performance was ascribed to the large degree of node movements and target separations during the animation [34]. The second class of data reduction methods started with simplifying the network structure. HiMap [35] clustered a large network hierarchically and displayed only important nodes and edges above a certain hierarchy. Van Ham and Perer proposed a method to construct a sub-graph of the large network from one or multiple nodes of interest [36]. However, very few of the structure-based data reduction methods targeted dynamic networks.

While the aforementioned methods focus on designing visualizations to interpret dynamic networks, there are fewer studies on improving their effectiveness for analytical tasks. The animation-based approaches were shown to be inadequate for analysis purposes; most other works managed to optimize the capability of static displays. One class of methods kept the analysis requirement in mind when designing visualizations. In [37], [38], temporal charts of network metrics (e.g., degree and size) were plotted together with the network graph as coordinated multiple views. Analysts can examine detailed network structures and their high-level temporal trends simultaneously. On the comparison of dynamic networks along the timeline, frameworks such as VisLink [39] can be applied. Archambault proposed a useful method to directly construct a hierarchy graph from the network difference [40]. This difference map approach was shown to be effective for many dynamic network analysis tasks [41]. Another class of methods introduced novel interactions to facilitate specific dynamic network analysis tasks. VisLink allowed manipulating (e.g., rotating) the 2D plane hosting the network at each time slot to switch among comparing visualizations. Federico et al. introduced two kinds of highlight interactions [42]: one to feature the node trajectory on network layouts over time; the other to help discover the node connectivity on all time slots of the dynamic network. In in-situ visualization [2], the user chose a base visualization at first to gain an overview of the dynamic network, and then selected one part of the network to show details in another embedded visualization. The embedding can be zoomed and filtered iteratively, and again displayed with another visualization.

3 Dynamic Network Processing

In this section, we describe the process used to transform the general dynamic network data into a format suitable for the 1.5D visualization. The raw dynamic network is represented by a time-varying graph $G = (V, E)$ spanning a time period $[0, T)$. The graph consists of a node (vertex) set V and an edge (link) set E. Each node $v \in V$ (edge $e \in E$) is associated with a time set $T(v)$ ($T(e)$), which defines the active time period of the node (edge). An example is given in the top-right part of Fig. 2. The time set can be composed of multiple time intervals for continuous dynamic networks or multiple time points for discrete dynamic networks. It is assumed that the underlying graph G is simple, i.e., no multiple edges between two different nodes and no loop edges. Both directed/undirected and weighted/unweighted graphs are allowed. For simplicity, we refer to an undirected and unweighted graph in the description below.

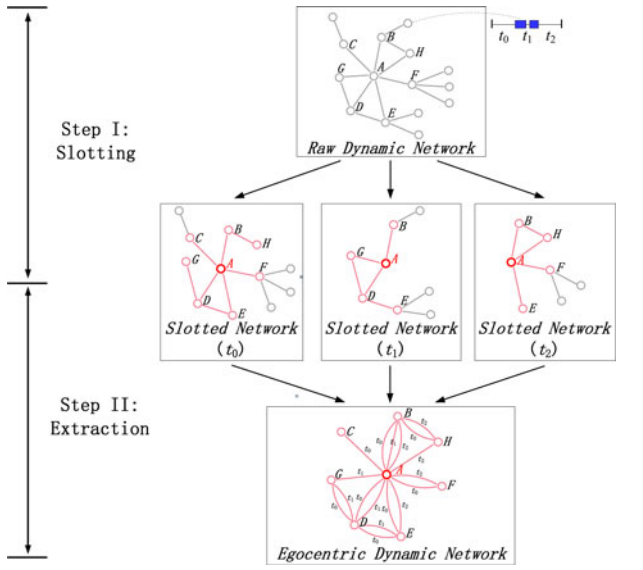


Fig. 2. An example of the egocentric dynamic network generation. The raw dynamic network is first sliced into three time slots t_0, t_1 and t_2. Egocentric graphs are then extracted and combined. The focus node A is highlighted in red; the adjacent nodes to A and their edges are drawn in pink. The edge label identifies the corresponding time slot.

3.1 Egocentric Dynamic Network

The egocentric dynamic network $D(A) = (V(A), E(A))$ central to the focus node A is defined by a discrete sub-graph of G. Formally, $D(A)$ is generated from G in two steps, as illustrated in Fig. 2.

Slotting. The first step is to discretize the dynamic network. Given an ordered time series $[t_0, t_1, \ldots, t_k]$ where $t_0 = 0$ and $t_k = T$, the slotted dynamic network graphs $G_S(t_i) = (V_S(t_i), E_S(t_i))$ are computed by

$$V_S(t_i) = \{v | v \in V \wedge T(v) \cap [t_i, t_{i+1}) \neq \emptyset\} \quad i = 0, \ldots, k-1 \tag{1}$$

$$E_S(t_i) = \{e | e \in E \wedge T(e) \cap [t_i, t_{i+1}) \neq \emptyset\} \quad i = 0, \ldots, k-1. \tag{2}$$

Normally, the slotting of the dynamic network is defined uniformly by setting the same interval on the time series. Certain granularity and network complexity control can be achieved by tuning the interval value, e.g. setting to a minute, an hour or a day.

Extraction. The second step is to extract $D(A)$ from the discrete dynamic network. By definition, the corresponding node set $V(A)$ is exactly the union of $\{A\}$ and the nodes in V adjacent to A. The edge set $E(A)$ is a little different in that each edge is replicated at each time slot in which it exists. We denote the edge in $E(A)$ by $e = (v_1, v_2, t)$ where v_1 and v_2 are two endpoints and t represents its time slot

$$V(A) = \{A\} \cup \{v | v \in V \wedge (v, A) \in E\} \tag{3}$$

$$E(A) = \{(v_1, v_2, t) | v_1 \in V(A) \wedge v_2 \in V(A) \wedge (v_1, v_2) \in E_S(t)\}. \tag{4}$$

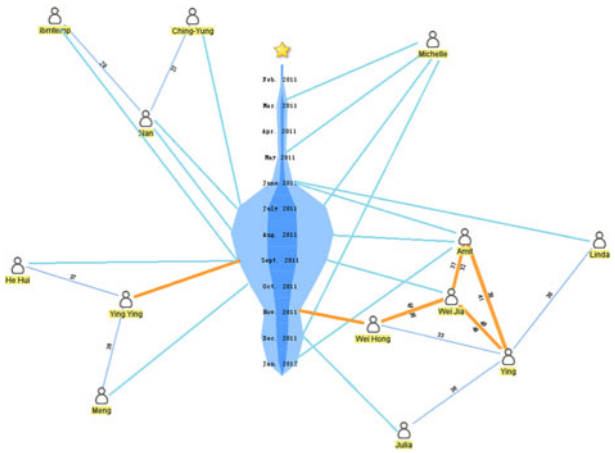


Fig. 3. 1.5D dynamic network visualization design. The data is synthesized only for the illustration purpose.

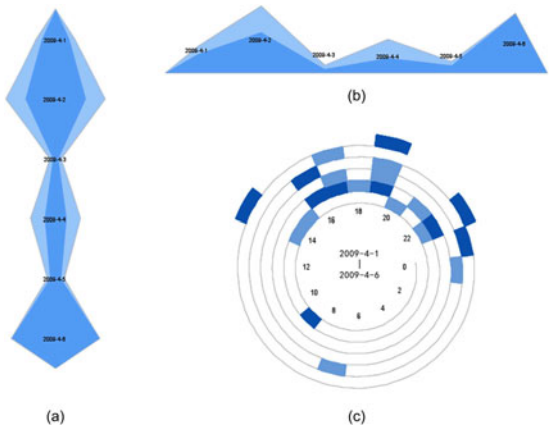


Fig. 4. Alternatives for the focus node representation: (a) The vertical double-sided trend. (b) The horizontal single-sided trend. (c) The spiral glyph. In this case, each ring in the glyph corresponds to a day and each sector (block) in a ring corresponds to an hour in a day. The filling of a block indicates there is at least one activity happening at the focus node in the corresponding hour. The brightness of the fill color encodes the number of activities.

The resulting egocentric dynamic network $D(A)$ is essentially a multigraph in that there may exist multiple edges between two endpoints, as shown in Fig. 2.

3.2 Event-Centric Dynamic Network

In an extension of the 1.5D visualization, we consider the dynamic network central to a group of events with the same type. This is done by inserting a node representing this group of events as the focused node, which is further drawn as the central trend glyph. For example, in a paper co-authorship network, the papers in the same conference can be grouped together as one event type. There will be one edge between an author and the central event (conference) if she published a paper on that. The edges between the authors remain as the co-authorship relationship.

Formally, the raw dynamic network data is processed into an event-centric graph $D(\Gamma)$ where Γ denotes the event type. Each single event in this type is represented by $Evt(\Gamma, \psi, t(\psi))$, where ψ denotes the unique event ID, $t(\psi)$ denotes the event time. A node v involved in an event $Evt(\Gamma, \psi, t(\psi))$ is denoted by $v \sim Evt(\Gamma, \psi, t(\psi))$. $D(\Gamma)$ is generated in three steps where the last two steps largely follow the process in Section 3.1. The first step is given below.

Insertion. On the input time-varying graph G, Γ is added to the node set V as the focus node, which spans the entire time period of G. For the edge set, edges are added from every non-focus node in G to the focus node Γ. Each such edge indicates that an event in the type of Γ involving a non-focus node has happened. The graph insertion step is defined by

$$V = V \cup \{\Gamma\}, T(\Gamma) = [0, T) \tag{5}$$

$$\begin{aligned} E &= E \cup \{(v, \Gamma) | v \in V \wedge v \neq \Gamma\}, \\ T(e = (v, \Gamma)) &= \{t(\psi) | v \sim Evt(\Gamma, \psi, t(\psi))\}. \end{aligned} \tag{6}$$

4 1.5D Visualization

4.1 Design

An example of the proposed 1.5D visualization is given in Fig. 3. The main idea is to introduce a temporal glyph to represent the trend of the focus node. As a result, the multiple edges between each non-focus node and the focus node can be decoupled by design. The time information of each of these edges is encoded by the location of the edge's endpoint on the trend glyph, exactly at the brim of the corresponding time slot. We call these edges *time-dependent*. On the other hand, between the non-focus nodes, the multiple edges are combined into a single edge in the view, which is called *time-independent*. Basically the 1.5D design follows the traditional network visualization paradigm with nodes and straight-line edges, so that the visual network theme can be easily identified by a user.

In Fig. 3, the graph illustrates a dynamic email network of the focus node (person). The focus node is drawn in a vertical glyph, showing the trend of email communications (send + receive) of this person through a whole year. The width of the trend at each monthly-slotted sub-glyph encodes the number of emails in a particular month. The non-focus nodes, which represent the contact persons having email communications with the focus person, are placed on either side of the central trend glyph. For example, the non-focus node "Michelle" in the top-right part of Fig. 3 connects to the focus node in four separate months. The central trend glyph uses a stacked drawing to visualize the ratio of send/receive in the personal email communications. The inner stack in dark blue indicates the number of emails sent by the focus person in each month. Correspondingly, the outer stack of the trend glyph in light blue indicates the number of receives. In this graph, the majority of the email communications of the focus person are inward.

On the edge coloring, unidirectional communications are drawn in blue, while bidirectional communications are drawn in orange. Edge thickness indicates the number of emails. Upon mouse-hovering, the selected node (e.g., Van Ham in Fig. 8a) is drawn in a red outline, and all the neighboring nodes are drawn in pink outlines. The sending edges of the selected node are drawn in green, and the receiving edges are drawn in red. The corresponding time slots on the trend glyph turn red for the receiving stack and green for the sending stack.

Trend glyphs and selection criterion. We have implemented three kinds of trend glyphs to represent the timeline of the focus node, as listed in Fig. 4. Other visual encodings are

also possible, e.g., the recursive pattern [43]. Our first design is a double-sided trend glyph placed vertically (Fig. 4a). This choice applies a symmetric design so that a non-focus node can be placed on either side of the central glyph. The layout space is utilized better, which allows us to accommodate more non-focus nodes in the view. Also, the temporal network patterns of the non-focus node are better illustrated in this design. It is especially helpful when the number of time-independent edges among non-focus nodes is low. However, an obvious drawback, the placement of non-focus nodes on either side of the trend glyph can introduce unnecessary ambiguity in the data encoding. In some cases, the crossings of the time-independent edge over the trend glyph lead to a poor graph readability. On the other hand, the single-sided trend glyph (Fig. 4b) employs a horizontal design and lays out non-focus nodes only above the glyph. This avoids edge crossings of the time-independent edge with the trend glyph, however provides less flexibility to maximize the overall graph readability. This design works better in the scenario where the network is small in size but complex in structure. In a third choice, a spiral glyph [44] can be applied in the case where periodic patterns in the focus node's network activity are significant, as shown in Fig. 4c, where the focus node's timeline is drawn in a spiral line. Each ring in the glyph can represent a month, a week, or an hour, and in this graph, a day. Each sector (block) in the ring corresponds to a finer granularity, e.g., in this graph, an hour in a day. Each filled block indicates there are network-related activities in this hour. The color brightness of the filling encodes the number of such activities, the darker the blue, the larger the number. Labels in the center of the spiral glyph show the active period of the focus node measured in days, and also mark the hour of each block on the ring. Time-dependent edges are connected to the brim of the outermost block in this design. Note that since the spiral design occupies more space than the other two glyphs when a longer time period is considered, it generally works better for small egocentric graphs with periodic patterns.

4.2 Graph Layout

In this part, we describe the layout algorithm for the 1.5D dynamic network visualization. Without loss of generality, the trend glyph adopts the vertical double-sided design throughout the algorithm description. The layouts with the other two trend glyphs have little difference from the standard process. In a default setting, the vertical trend glyph is reasonably placed at the center of the view space, partially mapping the Y axis to the time dimension in the dynamic network. The ultimate goal for the layout algorithm is to place all the non-focus nodes in appropriate locations so that both their temporal affinities to the focus node and the topological characteristics of the dynamic network can be revealed. We introduce two layout algorithms to serve the smaller egocentric dynamic network and the larger event-centric dynamic network respectively.

4.2.1 Force-Directed Layout Model

The egocentric dynamic network is generally small in size. For example, though the friends/followers of an online SNS user can reach a thousand or more, the number of users she interacts with, is often much smaller (e.g., below a hundred). We apply the classical force-directed layout model [45] on the egocentric dynamic network, which can compute an aesthetic layout for small graphs in real time.

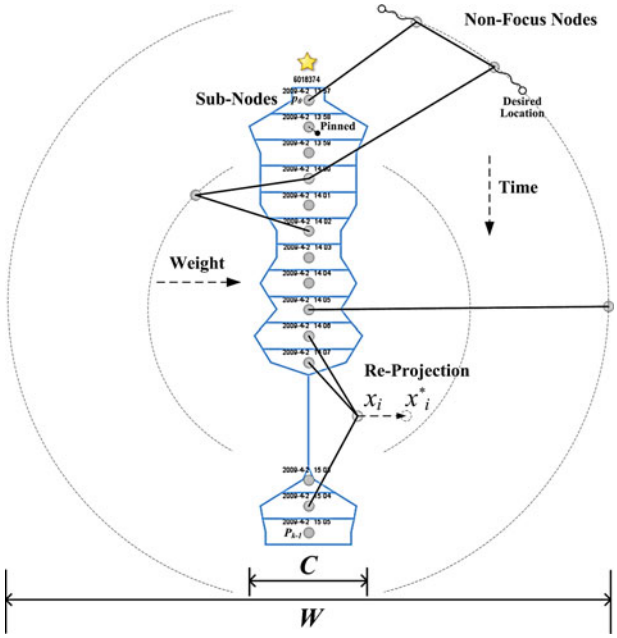


Fig. 5. An illustration of the proposed stable force-directed layout model with the sub-node split and re-projection processes.

There are three major challenges in directly applying the force-directed model. First, the classical force-directed algorithms assume an infinitely small size of the node, while in the 1.5D visualization, the shape of the trend glyph in the center is nontrivial, which can lead to a severe node overlapping problem. Second, the 1.5D graph is essentially a multi-graph, due to the multiple edges between each non-focus node and the focus node. Given the nontrivial shape of the focus node, the standard force-directed algorithm can not take the multiple edges into account in the layout process. Third, as the time is mapped to the Y-axis of the trend glyph (vertical setting), there is a desire for non-focus nodes to follow this visual mapping. In this paper, we propose a customized force-directed model for the 1.5D graph layout. It works in three steps:

Split. We first virtually split the focus node A by the predefined time slot into several sub-nodes $\{p_0, p_1, \ldots, p_{k-1}\}$, as shown in Fig. 5. Then each time-dependent edge between the non-focus node and the focus node is decoupled into several time-independent edges between the non-focus node and the sub-nodes, making the resulting graph a simple graph. Another gain is that each sub-node has a much smaller size, favoring the force-directed layout assumption. The new graph after the split is denoted as $L = (V_L, E_L)$.

Stable layout. Over the simple graph after the split, we apply a stable layout algorithm to compute the node placement. In the literature, most force-directed algorithms [45], [46] define energy functions over the graph and solve the energy minimization problem to compute the final optimized layout, which maximizes the layout aesthetic. To introduce the temporal information to the dynamic graph layout, we extend from the classical Kamada-Kawai (KK) layout model [46]. Our energy function consists of two

terms. The first term implements the KK layout's energy function and the second term works as a stable function to encode the temporal constraint of the non-focus nodes.

Formally, the energy function is written as

$$F = (1-\alpha)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\omega_{ij}(\|X_i - X_j\| - d_{ij})^2 + \frac{(n-1)\alpha}{2}\sum_{i=1}^{n}\mu_i\|X_i - X_i'\|^2, \tag{7}$$

where X_i denotes the position of the ith node in graph L, d_{ij} defines the optimal distance between the ith node and the jth node, ω_{ij} and μ_i are the parameters controlling the weight of each node (pair), X_i' denotes the desired position of the ith node according to its temporal information, and α controls the degree of stability.

In the vertical setting, the position of the sub-nodes split from the focus node are fixed at the center of their sub-glyphs. We remove irrelevant terms from the energy function by setting parameters as (8), which helps to alleviate the negative effect of fixed-nodes on the layout aesthetics. ω_{ij} and μ_i are set according to the classical model [47]. By default, α is set to 0.5 to strike a balance between the temporal and topology graph aesthetics. Users can adjust α online to favor a different layout strategy. For example, setting $\alpha = 1$ fixes the non-focus nodes at their desired positions by the temporal affinity, while setting $\alpha = 0$ only considers their topology aesthetics.

$$\omega_{ij} = \begin{cases} 0 & i\text{th and } j\text{th nodes are both sub-nodes of } A \\ d_{ij}^{-2} & \text{otherwise,} \end{cases}$$
$$\mu_i = \begin{cases} 0 & i\text{th node is sub-node of} A \\ \|X_i' - \overline{X'}\|^{-2} & \text{otherwise,} \end{cases} \tag{8}$$

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

The desired position of each non-focus node (X_i') is set on the circumference of a circle centered at the trend glyph. For the unweighted graph L, the angular position of a non-focus node is computed from the average time slot of all the incident edges connecting to the focus node A. The radius is inversely proportional to the total number of these edges to A. Formally, for a non-focus node v_i in graph L, $X_i' = (x_i, y_i)$ is computed by

$$\begin{aligned} x_i &= \overline{x} + syn(v_i)\rho\cos\theta, \\ y_i &= \overline{y} + \rho\sin\theta, \\ \rho &= \frac{\rho_0}{\|\{j|(v_i, p_j) \in E_L\}\|}, \\ \theta &= \theta_0 + \frac{(\theta_{k-1} - \theta_0)(s_i - t_0)}{t_{k-1} - t_0}, \\ s_i &= \overline{t_j}, \forall j, (v_i, p_j) \in E_L, \end{aligned} \tag{9}$$

where $(\overline{x}, \overline{y})$ is the center of the trend glyph, $syn(v_i)$ is the signal function indicating whether the non-focus node is placed on the left (-1) or on the right (1) of the trend glyph, ρ_0 denotes the maximal node distance from the center, and θ_0 and θ_{k-1} denote two bounding angular positions from the center, by default set to $\frac{\pi}{2}$ and $-\frac{\pi}{2}$.

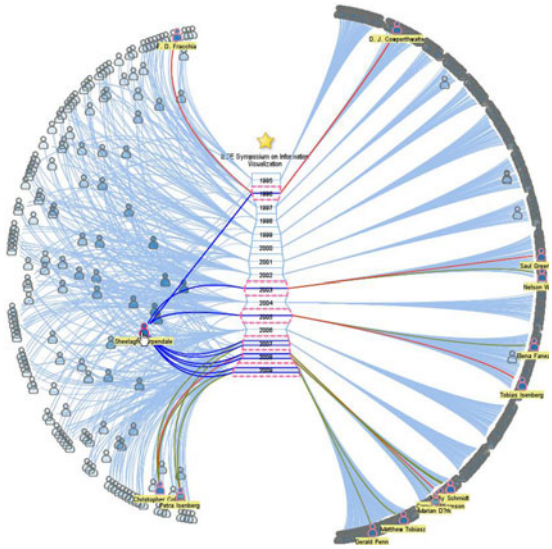


Fig. 6. 1.5D visualization of the InfoVis co-authorship network 1995 ~ 2009. The left part of the graph shows the authors who publish InfoVis papers in multiple years; the right part shows the authors who present in only one year (maybe multiple papers). Both time-dependent and time-independent edges are drawn. Event-centric edge bundling is applied. The node representing "Carpendale" is highlighted.

We apply a modified version of the stress majorization solver [47] to compute the optimization result of the above energy function. To decide on which side the non-focus node is placed, we implement a uniform graph bisection algorithm to partition the non-focus node set. The case with weighted graphs is handled similarly, except that edge weights are added to the computation in (9).

Re-projection. After the layout is computed, it is possible that some non-focus nodes lie in the contour of the central trend glyph. We introduce a linear re-projection on the X coordinate of non-focus nodes to alleviate this effect. Formally, their new X coordinates are calculated as below

$$x_i^* = \begin{cases} W - \frac{W-C}{W}(W - x_i) & W/2 \le x_i \le W, \\ \frac{W-C}{W}x_i & 0 \le x_i < W/2, \end{cases} \tag{10}$$

where x_i denotes the X coordinate before the re-projection, W is the width of the layout space, C is the maximal width of the trend glyph. Fig. 5 illustrates this process.

4.2.2 Radial Layout Model

For the event-centric dynamic networks, the event and the resulting network can involve thousands of entities, e.g., authors in a conference series. In this size, the force-directed layout will be quite slow. Although there are approximation-based multi-level layout algorithms for large graphs [48], the final drawing is often too cluttered to be understood, especially for the 1.5D visualization having many edge crossings on the central trend glyph. The event-centric edge bundling is proposed to alleviate this effect, as shown in Fig. 6 and described in Section 4.3. By edge bundling, time-independent edges which connect two non-focus nodes are not drawn in straight lines, and topological adjacencies among non-focus nodes are weakened to favor their temporal affinities to the focus node. Motivated by this observation, we propose a radial layout model which places

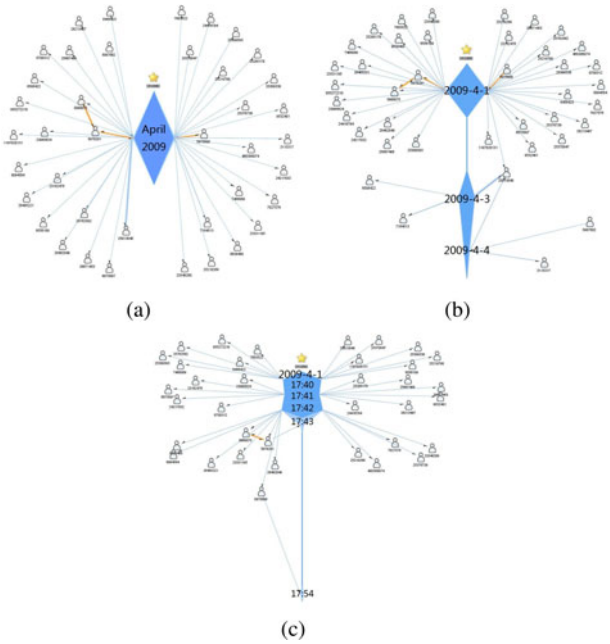


Fig. 7. 1.5D network visualization in different time granularities: (a) slotted by month, the network contains one-month's data; (b) showing the separate days in a month; (c) drilling down to a few minutes of April 1st, 2009.

the non-focus nodes rapidly for large event-centric dynamic networks. The layout result is shown in Fig. 6.

The radial model determines the graph layout in a polar coordinate system. The position of the non-focus nodes are computed only from their temporal affinity to the focus node. The center of the polar coordinate is set to the center of the trend glyph. The radius of each non-focus node is inversely proportional to the number of edges connecting to the focus node. The computation of a non-focus node's angular position involves three steps:

Partition. All the non-focus nodes are divided into two subsets and placed in the left and right side of the central trend glyph respectively. The default partition method separates the nodes having only one edge connecting to the focus node from the other nodes having multiple edges to the focus node. Other partition methods can also be applied for customized comparison purposes.

Sort. The average time affinity of each non-focus node to the focus node, denoted as s_i, is calculated by (9). Then for each subset generated in the first step, their non-focus nodes are sorted according to this average time affinity. The node rank is assigned starting from zero.

Assign. The non-focus node v_i with rank r_i in subset S is assigned the angular position θ_i by

$$\theta_i = \theta_0 + \frac{r_i(\theta_{k-1} - \theta_0)}{\|S\| - 1}. \quad (11)$$

4.3 User Interaction for Analysis

We design a few customized interactions for the analysis of egocentric dynamic networks by the 1.5D visualization:

Timeline navigation. In our design, the dynamic network is processed and visualized by pre-defined time slots. Switching to a new slotting granularity will lead to a quite different view of the same network. Inspired by the geometric zoom-in/zoom-out operation, we introduce the timeline navigation interaction which allows a user to select an interesting timeline period, zoom-in to show the network with a finer time granularity for the detailed analysis, and/or zoom-out to a higher-level view for the overview purpose.

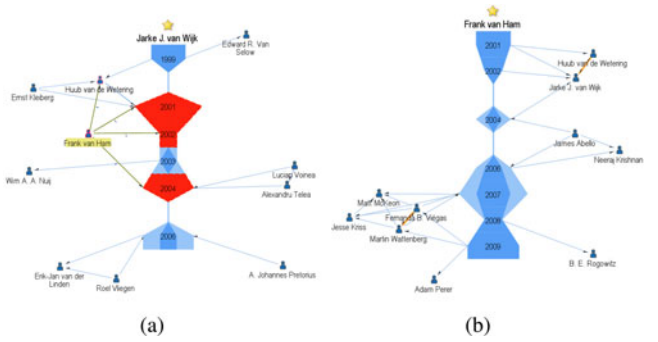


Fig. 8. Egocentric network navigation in the InfoVis co-authorship network: (a) the network central to Van Wijk, the node for Van Ham is hovered; (b) switch to the network central to Van Ham by double-clicking the node.

Fig. 7 gives an example of this interaction. In Fig. 7a, the network is slotted by month; however no temporal trend is visible as the time span is only one-month. Then the user zooms to the day granularity (Fig. 7b), and it can be quickly discovered that the behavior of the focus node is divided into two periods: April 1st and April 3rd $\sim$ 4th. As he proceeds to select the day of April 1st and zooms to the minute granularity (Fig. 7c), the pattern of a constant-rate burst in three minutes is located.

Egocentric network navigation. A major trade-off of the 1.5D design is to show only the egocentric dynamic network, rather than the entire network. Moreover, the temporal patterns associated with the time-independent edges among the non-focus nodes can not be revealed. We mitigate these limitations by allowing the user to navigate across many egocentric networks through a simple interaction. Upon a double-click of one non-focus node, the dynamic network view will switch to a new network central to the clicked node, as illustrated in Fig. 8.

Event-centric edge bundling. In the 1.5D design, the time-independent edges will sometimes pass through the central trend glyph, which can introduce significant visual clutters. On the event-centric dynamic networks, each time-independent edge is associated with a few events happening at particular time slots. We can deliberately bundle all the edges on the same event together by letting them go through the center of the trend glyph at the event's time slot. When a user hovers one non-focus node for its connections, its incident edges bundled at the same time slot of the trend glyph are decoupled into different events to reflect the details. This is called the event-centric edge bundling. In an example of the InfoVis co-authorship network, Fig. 6 shows the result after the bundling. The overall visual clutter is alleviated. Carpendale's connection patterns are highlighted in detail. She published nine papers with 13 co-authors during the history of the InfoVis conference. Note that by the event-centric bundling, the time-dependent and time-independent edges will overlap with each other. Our design differentiates

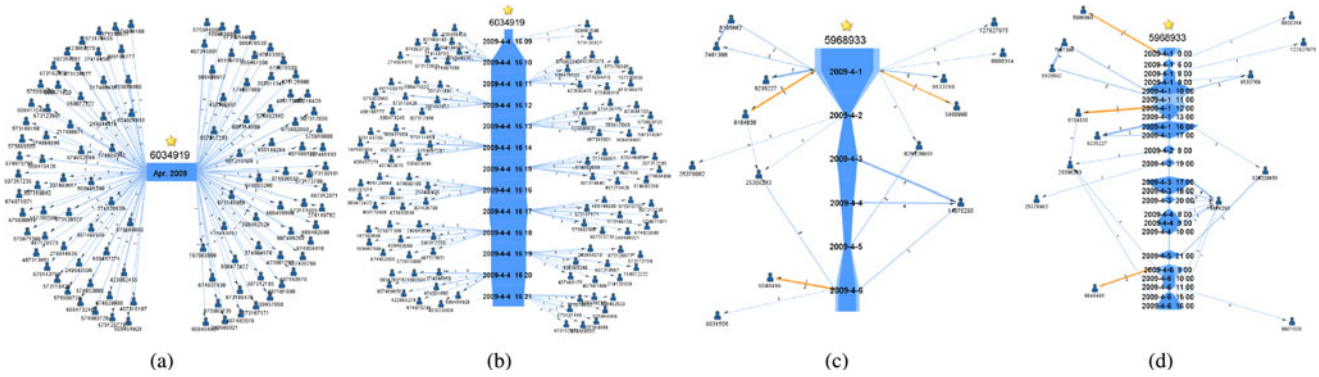


Fig. 9. 1.5D dynamic network visualization in a telecommunication network scenario: (a) a typical spammer behavior slotted by month; (b) spammer slotted by minute; (c) a typical non-spammer behavior slotted by day; (d) Non-spammer slotted by hour.

them by the edge coloring. As shown in Fig. 6, upon mouse hovering a node, the first-half segment of its time-independent edge, which is also a full time-dependent edge, is drawn in deep blue; while the second half segment of this time-independent edge is drawn in green and red, according to the priority of the target node over the hovered node on the event.

5 Case Studies

We present two case studies covering the two targeted task scenarios for the 1.5D visualization: 1) the dynamic adjacency between the focus node and non-focus nodes; 2) the connectivity of non-focus nodes with respect to their dynamic adjacencies to the focus node.

5.1 Telecommunication Network

In the first case, we visualize the telecommunication network collected by a service provider. As shown in Fig. 9, each node in the network, as well as the central trend glyph, represents one mobile phone user. The directed edges among them indicate short message communications (Fig. 9c). The resulting network is essentially a dynamic social network in the time period of the data set. Our previous work has developed a learning-based system [1] which detects mobile users who spam, based on the temporal and topological features of the social network. In the real usage, it is important for the service provider to evaluate the accuracy of the system. In case the system has wrong classifications, the provider needs to find root causes. Even if the system is shown to be accurate in most time, there is a need to prepare a summary of the spamming behavior, preferably in the visual form.

We invited Adam, an analyst from the telecom service provider, to use our visual tool to check the learning-based spammer/non-spammer classification results. He started by selecting one spammer in the list and accessing its egocentric dynamic network. As in Fig. 9a, the network slotted by month showed up a star-like pattern where the suspected spammer sent out only one message to quite a few users without receiving any messages from them. Meanwhile, there was no communication among neighbors of the spammer, a situation which indicated an extremely abnormal social network. These observations corresponded to the features applied in the spammer classifier: high outbound degree but low inbound degree, low average outbound edge weight, high sending/receiving ratio, and low clustering coefficient. Further, Adam drilled down to more details by changing the slotting granularity to minute, as shown in Fig. 9b. The temporal patterns in the spammer's behavior were located. The spammer tended to send messages out in a constant rate within a short time span. In this case, nine messages were sent per minute for 12 minutes. There was no user who communicates with the spammer in more than one time slots. This corresponded to the temporal feature applied in the classifier: the long-term bursty and short-term smooth sending rates.

In his second trial, one non-spammer classified by the system was selected, as shown in Fig. 9c with the egocentric network slotted by day. The orange edge indicated bidirectional communications, and the edge thickness displayed the number of messages on the edge (also drawn as the edge label). In this view, opposite patterns to the spammer's network were discovered: between the non-spammer and non-focus users, there were both inbound/outbound and bidirectional edges; the number of messages exchanged was larger than one in many cases; communications were found among non-focus users; there were several users who talked to the non-spammer in multiple time slots, and the sending/receiving trend of the non-spammer had no significant temporal pattern. Drilling-down to the hour granularity, as shown in Fig. 9d, more details were revealed. Although there were few high-level patterns to discover in this scale, more clues can be found in personal communications. For example, some had a double-handshake like contact with the central user within an hour and some others received a lot of messages continuously without replying. This is highly useful for scenarios such as crime network analyses.

5.2 Co-Authorship Network in the Visualization Community

In this part, we present another case study on the analysis of paper co-authorship dynamic networks in the visualization community. The data set is extracted from the ArnetMiner database [49]. It contains all the 9,557 papers of nine major visualization conferences and journals, including SciVis, InfoVis, VAST, EuroVis, PacificVis, TVCG, CGF, IV journal and CG&A, from 1982 to Jan. 2013. The co-authorship network is generated by adding one directed edge between any two authors of the same paper, from the lower-ranked to the higher-ranked author. This sums up to a network of 11,016

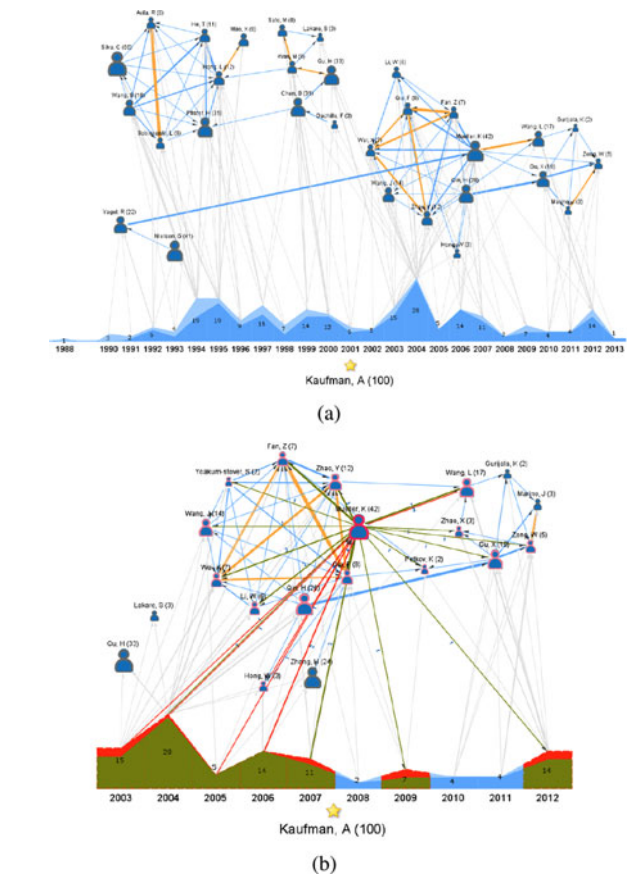


Fig. 10. 1.5D visualization of Arie E. Kaufman and his co-authors in the visualization community: (a) the egocentric dynamic network with his top 30 co-authors; (b) the top influencer in the recent 10 years.

author nodes and 40,839 co-authorship edges. Some tags are attached to the co-authorship edges according to the topic of the corresponding paper (e.g. "network visualization"). This is done by matching the paper title, index terms and abstract with relevant keywords and manually double-checking all the matched papers for the final classification. In visualization, we apply the horizontal single-sided trend glyph design. In most cases, we mute the time-dependent edges in grey (except Fig. 11 with a small network), so that the network structure of non-focus nodes can be better perceived.

Jane, a junior visualization researcher, helped us in evaluating the 1.5D visualization tool. As a newcomer to the visualization community, Jane first selected Arie E. Kaufman, the prestigious fellow on scientific visualization, to study his collaboration history in this field. The initial view of Kaufman's egocentric dynamic network was a bit cluttered because of his 104 co-authors in history. Jane decided to apply the node filter in our tool to leave only his top 30 co-authors who published at least three papers together with Kaufman. In Fig. 10a, Jane found that Kaufman's top co-authors were naturally divided into two disconnected components (i.e. network community) over time. The community on the left connected to Kaufman mainly before 2003, as indicated by the horizontal position of these non-focus nodes and time-dependent edges (mouse hover to access a better view). The community on the right worked with Kaufman mainly after 2003. She then drilled down to the recent ten years after 2003, which was displayed in Fig. 10b. She found that the most influential author in

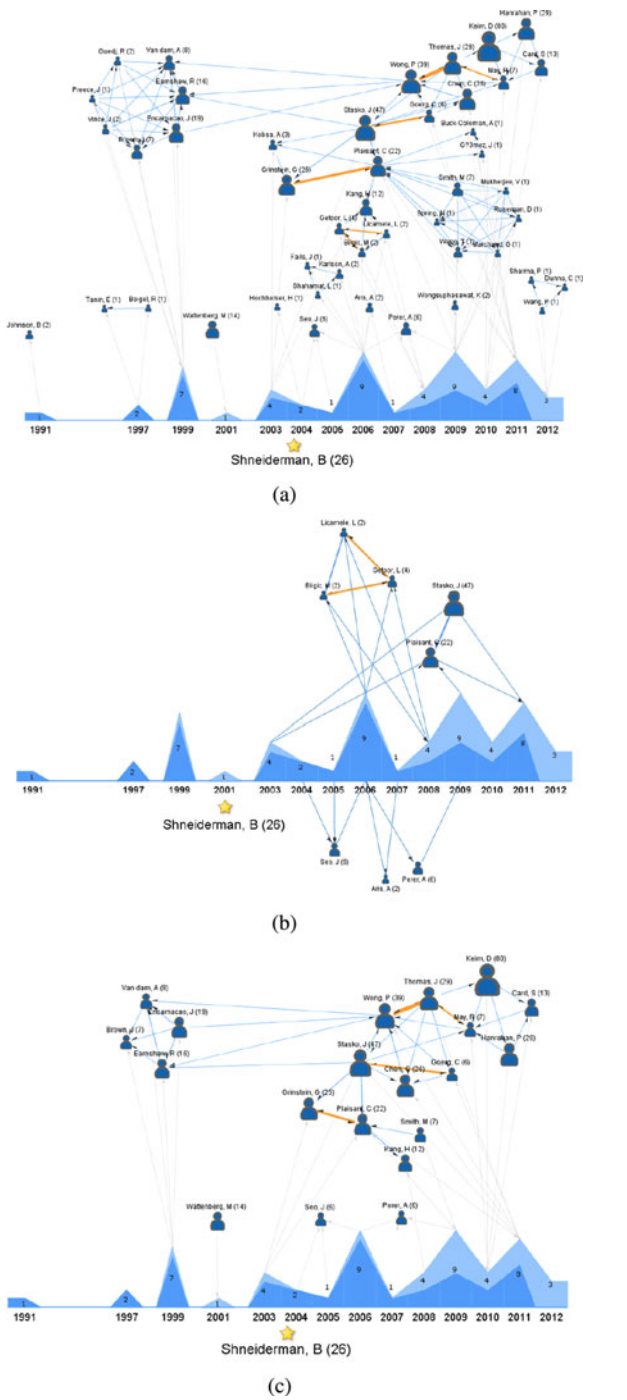


Fig. 11. 1.5D visualization of Ben Shneiderman and his co-authors in the visualization community: (a) a full egocentric dynamic network; (b) after filtering out one-time co-authors; (c) leave only the top 20 productive co-authors.

Kaufman's recent egocentric dynamic network was Klaus Mueller, a professor on visualization at the same department. Notably, Mueller's co-authorship with Kaufman distributed broadly over time and he virtually collaborated with most of Kaufman's top co-authors in this time period.

In the next trial, Jane conducted the same analysis on Ben Shneiderman, the well-known InfoVis fellow. As an overview, the tool displayed a full dynamic network visualization egocentric to Shneiderman (Fig. 11a). He published 26 visualization papers and had 46 co-authors during 22 years. From the graph, Jane quickly found that Shneiderman

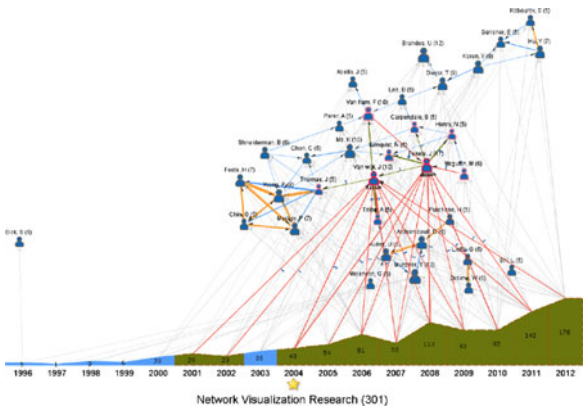


Fig. 12. 1.5D co-authorship network visualization central to the "network visualization" topic having 301 papers. The top authors with at least five network visualization papers are shown. Two key influencers in this topic are identified and highlight in the graph.

became more active in the field from 2003. Two thirds of his co-authors were connected to the same component (community), and the other one third were isolated from the main community, who might be doing independent research with Shneiderman. Jane further studied the connection patterns between Shneiderman and his co-authors using our tool. She filtered out the one-time co-authors who wrote only one paper with Shneiderman. The result is shown in Fig. 11b. It is clear that only a few people co-authored at least two papers with Shneiderman. Using another filter, Jane could create an egocentric dynamic network of Shneiderman and his top 20 productive co-authors (Fig. 11c), according to their number of papers in the visualization community. From Figs. 11a and 11c, Jane found that Plaisant, Stasko and Wong stood at the center of Shneiderman's egocentric dynamic network and connected a few local communities together.

Different from the above cases that look at the dynamic network central to one node (a mobile phone user or an paper author), we also apply the 1.5D visualization to event-centric dynamic networks. As Jane was very interested to the network visualization research, she selected this topic as the central event, which included 301 papers classified in our data pre-processing stage. In Fig. 12, the network was organized with respect to this research topic, drawn as the single-sided horizontal trend glyph in the bottom. She learned that this field was growing steadily, with most of papers published after 2000. The non-focus nodes, which represented the authors ever published on this topic, connected to the focus node (the topic) at each paper publication year. The edges among these authors still indicated the co-authorship relationship. Because there were 651 authors who ever published network visualization papers, Jane applied a filter to show only the authors with at least five such papers, which left 35 authors in Fig. 12. Most of these productive authors were connected into one single component, showing the close tie in this research field. At the center of this egocentric dynamic network, Jane found a few influential authors, notably Jack Van Wijk and Jean-Daniel Fekete (highlighted in Fig. 12), who connected several local communities together and also published frequently in the recent decade.

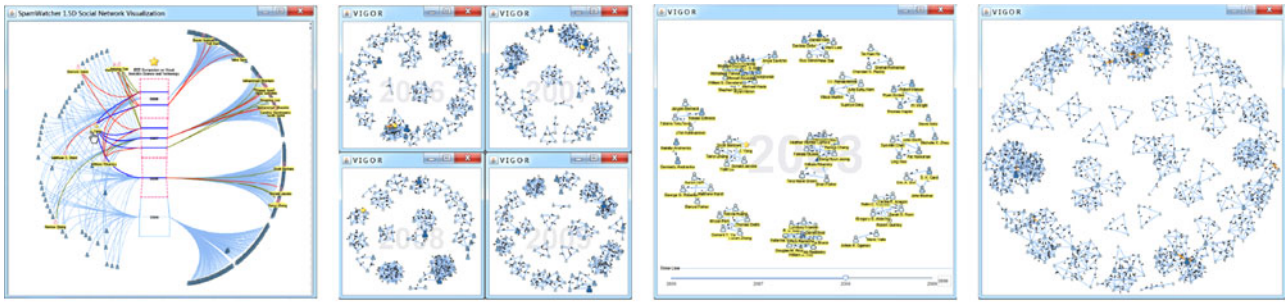
6 User Evaluation

We conducted a controlled user experiment to evaluate the performance of the 1.5D approach (1.5D Vis) in the context of the egocentric dynamic network analysis scenario. Our approach was compared with two baseline dynamic network visualization methods: small multiple display, dynamic network movie (Movie); as well as the static visualization aggregating the dynamic network over time (Static). Each method was implemented in a separate tool with a similar visual design, as shown in Fig. 13. In the Movie approach, the user was required to control the timeline to navigate dynamic networks. Auto-play is disabled because it is hard to select a fair animation speed for comparison. In all the tools, it is not allowed to switch the focus node or apply any filters.

Participant and apparatus. Twelve participants were recruited for the experiment. Eight were novices in the network visualization, three had experience, and another one was an expert. All the experiments were carried out in the same laptop workstation with a 17" widescreen LCD and a high performance graphics card. A 800×800 window size was set for all the visualization tools, except for SMD which used a smaller 400×400 window size for each timeslot ($\leq 2 \times 3$ timeslots) or 200×200 window size ($\leq 4 \times 6$ timeslots).

Experiment design. Participants were asked to complete several tasks with each visualization method, and then answered corresponding questions. We measured their accuracy and performance time in completing each task. Participants also responded to a quantitative questionnaire regarding their experience in using each visualization. The experiment followed a within-subject design: each user completed one trail per task ("T1 $\sim$ T4 or T5 $\sim$ T8" + "Q1 $\sim$ Q2") $\times$ visualization method ("1.5D Vis", "SMD", "Movie", "Static"). To obtain independence among results from the same user, we introduced four data sets so that each participant worked on tasks of each visualization method with a different data set. We applied a Latin square design that counterbalanced both learning and ordering effects. On each participant's turn, a training session was held before using each visualization. The session included readings of a half-page material on a paper describing the visualization, a short oral instruction from the organizer, and a trial of the tool with an irrelevant sample data to understand the basic visual encodings and interactions. The participant was told to complete each task in best-effort and wrote down their answers on paper.

Data and task. Four data sets were used in the experiment. The first two were egocentric short message communication networks from the first case study: one was the network central to a suspected spammer (12 timeslots by minute and 109 nodes in total); the other was the network central to a typical non-spammer (five timeslots by day and 16 nodes in total). Four egocentric dynamic network analysis tasks were designed on the first two data sets, as listed below. T1 and T2 were used to examine the performance involving topological features of the egocentric dynamic network. T3 and T4 were used to evaluate the tasks further combining temporal features of the network. On each task, six



(a) 1.5D visualization (b) Small multiple display (c) Dynamic network movie (d) Static aggregated visualization

Fig. 13. The interface of 1.5D Vis and three alternative dynamic network visualization methods. The data set is the co-authorship network of VAST conference from 2006 to 2009. The star icon indicates "J. Yang" who is a relevant subject in the task questions.

candidate answers were provided including one "can not answer" option.

T1: Estimate the number of unique non-focus users who ever SEND short messages to the focus user.

T2: Estimate the number of unique connections among non-focus users.

T3: Among all the time slots, find the time slot when the focus user connects to (sends to or receives from) a maximal number of non-focus users.

T4: Estimate the number of non-focus users who connect to the focus user in more than one time slots.

The other two data sets were co-authorship dynamic networks in the visualization community. One was extracted from the InfoVis conference publications from 1995 to 2009 (15 timeslots by year and 674 nodes in total). The other was extracted from the VAST publications from 2006 to 2009 (four timeslots by year and 298 nodes in total, see Fig. 13). Four similar tasks were designed.

T5: Find the researcher that publishes the most InfoVis (VAST) papers.

T6: Find the researcher that co-authors the most InfoVis (VAST) papers with Frank Van Ham (J. Yang).

T7: Find the researcher that co-authors the most InfoVis (VAST) papers with Frank Van Ham (J. Yang) in the years 2005 $\sim$ 2009 (2007).

T8: Find the year in which the InfoVis (VAST) conference has the most (least) unique paper authors.

Two subjective questions were asked to rate each visualization, immediately after a participant completed all the four tasks. Answers were selected from a 1 $\sim$ 7 Likert scale.

Q1: How much does this visualization help you in completing the tasks and finding the correct answers?

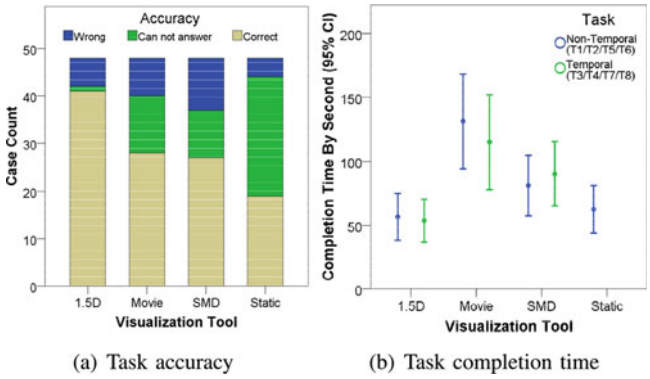
Q2: How much do you like the experience using this visualization?

In the first two smaller data sets, the force-directed layout model was applied (Section 4.2.1); in the other two data sets, the radial layout model was applied (Section 4.2.2).

Result and analysis. We collected 288 data entries in total, each corresponding to one task question completed by a user. Statistical analysis was conducted on the effect of alternative visualization methods over the measure of task accuracy, completion time and subjective rating. The choice of data set and task were considered as contributing factors. The significant level was set at 0.05 throughout the analysis. We also compared the performance difference between non-temporal and temporal tasks. On non-temporal tasks (T1/T2/T5/T6), users can complete the study without accessing the dynamics of the egocentric connection pattern over time. In contrast, on temporal tasks (T3/T4/T7/T8), users must take connection dynamics into consideration. Because all the users could not answer temporal tasks with the static visualization by design (Fig. 14a), we avoided comparing the Static approach on temporal tasks and subjective ratings.

Task accuracy. We translated task answers into binary accuracy variables, either true or false, by comparing to ground-truth answers. "Can not answer" choice is classified into false. We conducted binary logistic regressions to capture the Boolean value of the accuracy. The choice of visualization, data set and task were used as independent variables, and the binary accuracy variable was used as the dependent variable. Results show that the contribution of the visualization method to the task accuracy variation is statistically significant ($p < .005$). Compared to the 1.5D Vis, the Movie approach decreases the likelihood (odds) of correctly answering each task to 17.1 percents of the 1.5D Vis (95% CI = [5.6, 52], $p < .005$), controlling for differences in data set and task. Similarly, the SMD approach decreases this likelihood (odds) to 15.3 percents of the 1.5D Vis (95% CI = [5, 46.7], $p < .005$). The goodness of fit of this logistic regression model is 0.361 (Nagelkerke R Square). The raw task accuracy distribution in Fig. 14a indicates the same result: the 1.5D Vis approach receives the lowest overall error rate (7/48) than both the Movie approach (20/48) and the SMD approach (21/48). In the split view, the 1.5D Vis again receives the lowest error rate on non-temporal tasks (4/24), close to the Static approach (5/24) and much better than Movie (13/24) and SMD approaches (10/24). On temporal tasks, the result is similar: the 1.5D Vis has a much lower error rate (3/24) than Movie (7/24) and SMD approaches (11/24).

Task completion time. We applied the analysis of variance (ANOVA) test to study the impact of visualization, data set and task choice on the task completion time. Because of our Latin square study design, we can not use the repeated-measure ANOVA test to partition out the variability of individual participants. Instead, we applied a three-way ANOVA model, in which the numerical task completion time was used as the dependent variable, the visualization, data set, and task choice were used as three independent variables. Only main effect on each factor was modeled, high-order interactions among three factors were not captured. We validated both the normality ($p > .1$ in Shapiro-Wilk



(a) Task accuracy (b) Task completion time

Fig. 14. User experiment results.

test) and homogeneity of variance ($p > .05$ in Levene's test) assumptions on the dependent variable before conducting the ANOVA test. Results show that, with three-way ANOVA, there are significant main effects of the visualization method ($F(2, 131) = 16.3, p < .001$) and the task choice ($F(6, 131) = 5.45, p < .001$) on the task completion time. There is no significant main effect of the data set choice on the task completion time. A Tukey's post-hoc comparison among different visualization groups indicates that the 1.5D Vis group (M = 55.0, 95% CI = [38.4, 71.7]) leads to significantly shorter task completion times than the Movie group (M = 123.3, 95% CI = [106.5, 140.2]), $p < .001$, and the SMD group (M = 85.6, 95% CI = [69.0, 102.3]), $p < .05$. The SMD group also has significantly shorter task completion times than the Movie group, $p < .01$, which is coherent with previous study results on general dynamic networks [33]. The raw task completion time shown in Fig. 14b indicates the same comparative result on both non-temporal and temporal tasks. On the non-temporal tasks only, the difference between the 1.5D Vis group (M = 56.4, 95% CI = [38.1, 74.7]) and the Static group (M = 62.4, 95% CI = [43.8, 81.0]) is not significant.

Here we should note that, during the experiment we did not distinguish the time to read the question and write-down the answer from the task completion time. Therefore, the task completion time measure may not be exactly representative to account for the technique differences, though in general participant's difference in reading and writing speed does not vary much when they are told to work in best-effort on short, simple tasks.

Subjective feedback. We analyzed participant's subjective ratings by the Kruskal-Wallis test, which does not require a normality assumption of the observed data. The dependent variable was set to the 7-scale Likert rating from Q1/Q2, the independent variable was set to the visualization method and the data set separately (Kruskal-Wallis test allows only one independent variable in each time). Results show that there are statistically significant differences among visualization groups ($H(2) = 11.0, p < 0.005$) on the subjective rating of Q1. The mean rank value is 26.5 for 1.5D Vis, 14.1 for Movie, and 14.9 for SMD (the rank value has a range of 1 to 36 from 36 feedbacks on three visualization groups). On the rating of Q2, there are also significant differences among visualization groups ($H(2) = 8.48, p < 0.05$). The mean rank is 25.5 for 1.5D Vis, 14.6 for Movie, and 15.3 for SMD. Follow-up Mann-Whitney tests were conducted to evaluate the pairwise difference among visualization groups. Results show that the subjective rating of the 1.5D Vis is significantly higher than the rating of the Movie approach, on both Q1 ($U = 23.0, p < .005$) and Q2 ($U = 27.5, p < .01$). Similarly, the subjective rating of the 1.5D Vis is significantly higher than the rating of the SMD approach, on both Q1 ($U = 24.5, p < .01$) and Q2 ($U = 32.0, p < .05$).

Discussion. From the above analysis, we can summarize that on both non-temporal and temporal tasks, the 1.5D approach gains an advantage over two baseline dynamic network visualization methods (the self-controlled dynamic network movie and the small multiple display) by higher task accuracies, shorter task completion times, and better subjective ratings from participants. On non-temporal tasks only, the performance of the 1.5D approach is close to that of the static visualization aggregating the dynamic network over time. We caution that our result should be taken on the egocentric dynamic network analysis scenario only, and we haven't compared it with various special-purpose network visualization tools.

7 Conclusion

In this paper, we propose a general framework, namely the 1.5D visualization, for displaying and analyzing egocentric dynamic networks. Through formal case and user studies, we show that the 1.5D approach can effectively guide a user in the analysis process of egocentric dynamic networks, notably by optimizing low-level tasks such as analyzing egocentric dynamic adjacencies and egocentric network structures. The success of our approach can be attributed to three key innovations: the egocentric dynamic network abstraction that reduces the network complexity for a better human perception; the 1.5D visual metaphor with a variety of trend glyphs that reveal both interesting temporal patterns and topological egocentric network features; and various interaction methods that allow temporal and network navigation beyond the basic single view representation.

Acknowledgments

This work is supported by China National 973 project 2014CB340301 and NSFC grant 61379088. The authors would like to thank Xiaoxiao Lian, Xiaohua Sun and Frank Van Ham for their suggestions in the visual design and user study, and anonymous reviewers for their valuable suggestions in improving the paper.

References

[1] C. Wang, Y. Zhang, X. Chen, Z. Liu, L. Shi, G. Chen, F. Qiu, C. Ying, and W. Lu, "A behavior-based sms antispam system," *IBM J. Res. Develop.*, vol. 54, no. 6, pp. 1–16, 2010.

[2] S. Hadlak, H.-J. Schulz, and H. Schumann, "In situ exploration of large dynamic networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2334–2343, Dec. 2011.

[3] L. Shi, Q. Liao, Y. He, R. Li, A. Striegel, and Z. Su, "SAVE: Sensor anomaly visualization engine," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, 2011, pp. 201–210.

[4] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner, "Visual analysis of large graphs," in *Proc. EuroGraph.—State of the Art Report*, pp. 37–60, 2010.

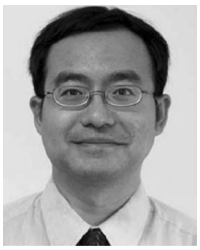
[5] J. Moody, D. McFarland, and S. Bender-deMoll, "Dynamic network visualization," *Amer. J. Sociol.*, vol. 110, no. 4, pp. 1206–1241, 2005.

[6] S. Diehl and C. Görg, "Graphs, they are changing: Dynamic graph drawing for a sequence of graphs," in *Proc. Int. Symp. Graph Drawing*, 2002, pp. 23–31.
[7] S. C. North and G. Woodhull, "Online hierarchical graph drawing," in *Proc. Int. Symp. Graph Drawing*, 2001, pp. 77–81.
[8] C. Görg, P. Birke, M. Pohl, and S. Diehl, "Dynamic graph drawing of sequences of orthogonal and hierarchical graphs," in *Proc. Int. Symp. Graph Drawing*, 2004, pp. 228–238.
[9] D. Archambault, H. C. Purchase, and B. Pinaud, "Animation, small multiples, and the effect of mental map preservation in dynamic graphs," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 4, pp. 539–552, Apr. 2011.
[10] M. Farrugia and A. Quigley, "Effective temporal graph layout: A comparative study of animation versus static display methods," *Inf. Vis.*, vol. 10, no. 1, pp. 47–64, 2011.
[11] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry, "Task taxonomy for graph visualization," in *Proc. AVI Workshop Beyond Time Errors: Novel Eval. Methods Inf. Vis.*, 2006, pp. 82–86.
[12] L. Shi, C. Wang, and Z. Wen, "Dynamic network visualization in 1.5D," in *Proc. IEEE Pacific Vis. Symp.*, 2011, pp. 179–186.
[13] S. Moen, "Drawing dynamic trees," *IEEE Softw.*, vol. 7, no. 4, pp. 21–28, Jul. 1990.
[14] R. F. Cohen, G. D. Battista, R. Tamassia, and I. G. Tollis, "Dynamic graph drawings: Trees, series-parallel digraphs, and planar st-digraphs," *SIAM J. Comput.*, vol. 24, no. 5, pp. 970–1001, 1995.
[15] S. C. North, "Incremental layout in DynaDAG," in *Proc. Int. Symp. Graph Drawing*, 1995, pp. 409–418.
[16] P. Eades, W. Lai, K. Misue, and K. Sugiyama, "Preserving the mental map of a diagram," in *Proc. Compugraphics*, 1991, pp. 34–43.
[17] K. Misue, P. Eades, W. Lai, and K. Sugiyama, "Layout adjustment and the mental map," *J. Vis. Lang. Comput.*, vol. 6, no. 2, pp. 183–210, 1995.
[18] K. Boitmanis, U. Brandes, and C. Pich, "Visualizing internet evolution on the autonomous systems level," in *Proc. Int. Symp. Graph Drawing*, 2007, pp. 365–376.
[19] Y. Frishman and A. Tal, "Online dynamic graph drawing," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 4, pp. 727–740, Jul. 2008.
[20] Y. Frishman and A. Tal, "Dynamic drawing of clustered graphs," in *Proc. IEEE Symp. Inf. Vis.*, 2004, pp. 191–198.
[21] S. Diehl, C. Görg, and A. Kerren, "Preserving the mental map using foresighted layout," in *Proc. IEEE TCVG Symp. Vis.*, 2001, pp. 175–184.
[22] G. Kumar and M. Garland, "Visual exploration of complex time-varying graphs," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 805–812, Sep./Oct. 2006.
[23] U. Brandes and S. R. Corman, "Visual unrolling of network evolution and the analysis of dynamic discourse," in *Proc. IEEE Symp. Inf. Vis.*, 2002, pp. 145–151.
[24] N. E. Ji Soo Yi and S. Lee, "Timematrix: Analyzing temporal social networks using interactive matrix-based visualizations," *Int. J. Human-Comput. Interaction*, vol. 26, no. 11–12, pp. 1031–1051, 2010.
[25] M. C. Hao, U. Dayal, D. A. Keim, and T. Schreck, "Importance-driven visualization layouts for large time series data," in *Proc. IEEE Symp. Inf. Vis.*, 2005, pp. 203–210.
[26] M. Burch, F. Beck, and S. Diehl, "Timeline trees: Visualizing sequences of transactions in information hierarchies," in *Proc. Working Conf. Adv. Vis. Interfaces*, 2008, pp. 75–82.
[27] M. Burch and S. Diehl, "TimeRadarTrees: Visualizing dynamic compound digraphs," *Comput. Graph. Forum*, vol. 27, no. 3, pp. 823–830, 2008.
[28] M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf, "Parallel edge splatting for scalable dynamic graph visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2344–2353, Dec. 2011.
[29] M. Farrugia, N. Hurley, and A. Quigley, "Exploring temporal ego networks using small multiples and tree-ring layouts," in *Proc. 4th Int. Conf. Adv. Comput.-Human Interactions*, 2011, pp. 79–88.
[30] E. Tufte, *Envisioning Information*. Cheshire, CT, USA: Graphics Press, 1990.
[31] C. Friedrich and P. Eades, "The marey graph animation tool demo," in *Proc. Int. Symp. Graph Drawing*, 2000, pp. 396–406.
[32] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst, "Animated exploration of dynamic graphs with radial layout," in *Proc. IEEE Symp. Inf. Vis.*, 2001, pp. 43–50.
[33] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko, "Effectiveness of animation in trend visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1325–1332, Nov./Dec. 2008.
[34] N. E. S. Ghani and J. S. Yi, "Perception of animated node-link diagrams for dynamic graphs," *Comput. Graph. Forum*, vol. 31, no. 3, pp. 1205–1214, 2012.
[35] L. Shi, N. Cao, S. Liu, W. Qian, L. Tan, G. Wang, J. Sun, and C.-Y. Lin, "HiMap: Adaptive visualization of large-scale online social networks," in *Proc. IEEE Pacific Vis. Symp.*, 2009, pp. 41–48.
[36] F. van Ham and A. Perer, "'Search, show context, expand on demand': Supporting large graph exploration with degree-of-interest," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 953–960, Nov./Dec. 2009.
[37] M. Pohl, F. Reitz, and P. Birke, "As time goes by: Integrated visualization and analysis of dynamic networks," in *Proc. Working Conf. Adv. Vis. Interfaces*, 2008, pp. 372–375.
[38] A. Perer and B. Shneiderman, "Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2008, pp. 265–274.
[39] C. Collins and S. Carpendale, "VisLink: Revealing relationships amongst visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1192–1199, Nov./Dec. 2007.
[40] D. Archambault, "Structural differences between two graphs through hierarchies," in *Proc. Graphics Interface*, 2009, pp. 87–94.
[41] D. Archambault, H. C. Purchase, and B. Pinaud, "Difference map readability for dynamic graphs," in *Proc. Int. Symp. Graph Drawing*, 2010, pp. 50–61.
[42] P. Federico, W. Aigner, S. Miksch, F. Windhager, and L. Zenk, "A visual analytics approach to dynamic social networks," in *Proc. Int. Conf. Knowl. Manag. Knowl. Technol.*, 2011, pp. 47:1–47:8.
[43] D. A. Keim, H.-P. Kriegel, and M. Ankerst, "Recursive pattern: A technique for visualizing very large amounts of data," in *Proc. IEEE Conf. Vis.*, 1995, pp. 279–286.
[44] M. Weber, M. Alexa, and W. Müller, "Visualizing time-series on spirals," in *Proc. IEEE Symp. Inf. Vis.*, 2001, pp. 7–13.
[45] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Upper Saddle River, NJ, USA: Prentice-Hall, 1998.
[46] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, 1989.
[47] E. R. Gansner, Y. Koren, and S. North, "Graph drawing by stress majorization," in *Proc. Int. Symp. Graph Draw.*, 2004, pp. 239–250.
[48] Y. Hu, "Efficient and high quality force-directed graph drawing," *Mathematica J.*, vol. 10, no. 1, pp. 37–71, 2005.
[49] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Min.*, 2008, pp. 990–998.

Lei Shi received the BS and MS degrees in 2003 and 2006, and the PhD degree in 2008 from the Department of Computer Science and Technology, Tsinghua University. He is an associate research professor in the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. Previously, he was a research staff member and a research manager at IBM Research, China, working on information visualization and visual analytics. His research interests include information visualization, visual analytics, network science, and networked systems. He has published more than 40 papers in refereed conferences and journals. He received the IBM Research Accomplishment Award on "Visual Analytics" and the VAST Challenge Award twice in 2010 and 2012.

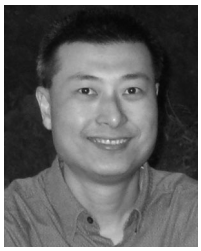
Chen Wang received the BS and MS degrees in 2003 and 2006, respectively, from the Department of Computer Science, Fudan University. He is currently a research staff member and a research manager at the Information Management Department, IBM Research, China. His research interests include database technology, stream computing, and big data systems. He has published 20+ papers in refereed conferences and journals. He holds 15 issued and pending patents in the US and China.

Zhen Wen received the PhD degree from the University of Illinois at Urbana-Champaign in computer science in 2004. He is currently a research staff member at IBM T.J. Watson Research Center. He is a co-PI of the Social Network Analytics Research in IBM T.J. Watson Research Center. He has broad interests in data mining, signal processing, and human computer interaction with applications on social network analysis and multimedia analysis. He was an area chair of social media at the ACM conference on Multimedia 2012. His work received 2011 Association of Information System ICIS Best Theme Paper Award, the best paper award at the ACM Conference on Intelligent User Interfaces (IUI) 2005, and an IBM Research division award in 2005.



Huamin Qu received the BS degree in mathematics from Xi'an Jiaotong University, China, and the MS and PhD (2004) degrees in computer science from the Stony Brook University. He is an associate professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His main research interests include visualization and computer graphics. He has co-authored more than 60 refereed papers. He is in the steering committee of the IEEE Pacific Visualization Conferences, and served as the program co-chair for IEEE PacificVis 2011 and 2012 and the conference co-chair for VINCI 2011 and VINCI 2012. He is a guest editor for the *IEEE Transactions on Visualization and Computer Graphics*, *IEEE Computer Graphics and Applications*, and *ACM Transactions on Intelligent Systems*. He received Honorable Mention for the Best Paper Award at IEEE Visualization 2009 and is a winner of the 2009 IBM Faculty Award.

Chuang Lin received the PhD degree in computer science from the Tsinghua University in 1994. He is an Honorary visiting professor, University of Bradford, United Kingdom. He is a professor at the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His current research interests include computer networks, performance evaluation, network security analysis, and Petri net theory and its applications. He has published more than 400 papers in research journals and IEEE conference proceedings in these areas and has published four books.



Qi Liao received the BS degree in 2005 with Departmental Distinction in computer science from Hartwick College, with a minor concentration in mathematics, and the MS and PhD degrees in computer science and engineering (CSE) from the University of Notre Dame in 2008 and 2011, respectively. He is an assistant professor in the Department of Computer Science, Central Michigan University. His research interests include computer security, anomaly detection, visual analytics, and graph mining. His research has been recognized with a Best Paper Award from USENIX 22nd Large Installation System Administration Conference (LISA '08), the Second Prize Winner of the NSIC '09 Competition, and the IEEE VAST Challenge Award 2012.