3D TECHNIQUES FOR VISUALIZING USER ACTIVITIES ON MICROBLOGS

Masahiko Itoh

Institute of Industrial Science, The University of Tokyo, Japan
Email: imash@tkl.iis.u-tokyo.ac.jp

ABSTRACT

Microblogging platforms such as *Twitter* have recently attracted attention and gained popularity. By using them, people can easily write their opinions and comments on events in the real world in real time. Content on the timeline of *Twitter* immediately reflects real and virtual activities. The purpose of this paper is to provide 3D components and methods of visualization that enable us to visualize user activities and the roles they play in their networks, i.e., when and how much they *tweet*, how and when information spreads in their networks, or what kinds of topics they *tweet* in each community. This paper presents 3D interactive components for visualizing relations between users, their messages, and changes in the number of messages over time. Moreover, we explain methods of comparing variations caused by different timing and by multiple groups of users.

1. INTRODUCTION

Microblogging platforms have recently attracted attention and gained in popularity. *Twitter* [1] is one of the biggest microblogging platforms, which has over 75 million user accounts [2], and its users have spread all over the world (Java et al. (11)). By using *Twitter*, people can easily write their opinions and comments on events in the real world in real time. Content on the timeline of *Twitter* immediately reflects both real and virtual activities.

Twitter enables users to write and read text messages called *tweets* that are up to 140 characters long, are displayed on user home pages, and are also shown on their *followers*' home pages. They can read the *tweets* of their *followings* and also *reply* to such *tweets*. Moreover, users can select *tweets* by others and retransmit them, called *retweets*. We can obtain these data through the *twitter* public API in XML format.

Tweeting and *retweeting* sometimes cause information to spread such as that on restaurants or new products by word-of-mouth (WOM) communication. Therefore, *Twitter* has attracted attention as a tool for WOM. Gladwell's ”The Tipping Point” discusses three types of key people for WOM such as connectors, mavens, and salesmen (Gladwell (6)). Information on *Twitter* is occasionally spread by these kinds of key people. It is important to analyze the connection between users and message flows on *Twitter* to consider methodologies for advertising and propaganda using WOM on the Internet. For example, we may effectively bring information to the attention of others by finding WOM influencers on the Internet, and then by being *followed* and *retweeted* by them.

The purpose of this paper is to provide 3D components and methods of visualization that enable us to visualize user activities and the roles they play in their networks, when and how much they *tweet*, how and when information spreads in their networks, or what kinds of topics they *tweet* in each community.

Visualizing the time-series of data allows us to answer seven important questions (i) what kinds of elements appear at specific times, (ii) when do such elements appear and disappear, (iii) how long do they exist on a timeline, (iv) how rapidly do they change, (v) how often do they appear, (vi) what kind of order do data elements appear in, and (vii) which elements appear together? (Müller and Schumann (15)). Being able to visualize the time series of data content on *Twitter* enables us to display changes in thought and activities that occur in the real and/or virtual world, and allows us to analyze social phenomena.

This paper proposes five 3D-interactive components for the visualization of relations between users, their messages, and changes in the numbers of messages over time: (i) *TimeSlices* for visualizing user networks with the time-series of data, (ii) *TimeSkewers* for visualizing the content of each user such as *tweets* on a timeline, (iii) *TimeFluxes* for visualizing changes in the amount of information such as the number of *tweets* for each user at each timing, (iv) *MessageTrails* for visualizing message flows caused by *replying* and *retweeting*, and (v) *TimeCloud* for visualizing the global tendency of user activities on the timeline. These elements are introduced in Section 3.

Moreover, we provide methods of comparing multiple variations caused by different timing and multiple-user groups. To compare different timing, our sys-

[1] http://twitter.com/

[2] http://themetricsystem.rjmetrics.com/2010/01/26/new-data-on-twitters-users-and-engagement/

tem provides functions for adding multiple *TimeSlices* with different time stamps, and it provides *parallel views* and *overlay views* in a 3D environment. They enable us to seamlessly change these viewing modes without losing user cognition. To compare different communities, our system provides functions for adding multiple layers representing different groups of users, and also provides different viewing modes such as an *aggregate view*, a *pile view*, or a *split view* to compare different layers that sometimes have overlapping nodes, and that change differently along a timeline. These methods are introduced in Section 4.

Our approach provides 3D interactive environments and visualization techniques using multiple 2D planes, lines, and plots in 3D space, which enable users to dynamically and simultaneously visualize user networks, changes in their attribute values, messages and message flows, and the global distribution of their activities. Our techniques enable us to observe overviews and their details from global and local aspects by seamlessly switching them. Moreover, combinations of our visualization techniques allow us to display the relationships between interactions among users and statistical changes in trends in 3D space at the same time.

2. RELATED WORK

Four main methods have been proposed to enable the evolution of information structures to become visualized; however, these have their own pros and cons. The first is (a) using *animation* to dynamically display changes in structures (Toyoda and Kitsuregawa (23), Nakazono et al. (16)). Although this enables users to dynamically observe changes in structures, it reduces user recognition, because they lose the context in previous situations. Users occasionally miss where changes have occurred and when they have changed throughout the entire space. The second is (b) mapping a *timeline* to one of the axes in a 3D environment (H.Chi et al. (9)). This enables users to observe global differences between multiple graphs. However, it is difficult to check local differences in detail. The third is (c) using multiple *tiled views* to display multiple Web graphs (Toyoda and Kitsuregawa (23), H.Chi and K.Card (8)). Users can compare the differences between Web graphs in parallel, enabling them to comprehend global differences. However, it is difficult to intuitively understand time intervals between Web graphs. Users occasionally cannot determine how long it has taken for changes to have occurred. The fourth is (d) *overlaying* graphs for different time periods on one view (Nakazono et al. (16), Brandes and Corman (1)). This is advantageous for comparing graphs in detail; however, is difficult to display global changes in structures.

LifeLines (Plaisant et al. (20)) and *TimeMachine* (Rekimoto (21)) enable users to display recorded events in personal histories on timelines. *VisuaLinda* [12], and *TimeTunnel* (Notsu et al. (18)) use the combination of 3D space and a timeline to visualize the time series of events or attribute values. They effectively use 3D spaces to simultaneously represent two kinds of relations including time relations to avoid disrupting user cognition caused by them having to reconstruct their mental models.

Minard's famous chart for showing the terrible fate of Napoleon's army in Russia is a classical example of mapping changes in values over time (Tufte (24)). *ThemaRiver* (Havre et al. (7)) and *Wormplots* (Matthews and Roze (14)) provide methods of visualizing changes in values of multiple attributes in 2D or 3D spaces. Dwyer and Eades (2) visualizes a flow graph, and simultaneously demonstrated time-dependent changes in the values of elements on a flow graph using 3D space.

There has been research on visualizing both time-dependent changes in elements in a graph and message flows in it (Koike et al. (13), Dwyer and Eades (2)).

Moving Phenomenon (Kim et al. (12)) visualize levels of activities or the distribution of moving objects using scatter plots in 3D environments. They enable us to display time-dependent changes in distribution from a global viewpoint.

Many systems for different data domains use a 2.5-D representation to visualize multiple situations in a 3D environment. The 2.5-D representation is used for three kinds of visualizations that involve: (i) visualizing different content (Fung et al. (5), Erten et al. (3)), (ii) visualizing time sequential changes (H.Chi et al. (9), Brandes and Corman (1), Nocke et al. (17)), and (iii) using different visual representations and/or models (Shen et al. (22)). Our framework supports the functions for (i) and (ii); however, the function for (iii) is still not supported, and we intend to explore this in future work.

Techniques for coordinated multiple visualizations such as linked views of a histogram or a *ThemaRiver* and multiple 3D graphs also enable us to observe time-sequential changes in graphs. However, it is difficult to freely add arbitrary numbers of graphs. Moreover, multiple tiled views require adequately sized display areas.

3. VISUALIZATION OF USER ACTIVITIES

This section presents 3D techniques for visualizing user networks, *tweets* including *replies* and *retweets*, flows of *replies* and *retweets*, and changes in the strength and distribution of user activities. We utilize the *IntelligentBox* system (Okada and Tanaka (19)) as the platform to implement these. This is a component-based visual software development system for interactive 3D graphics applications.

3.1. TimeSlices

A *TimeSlice* (Itoh et al. (10)) is a plane for visualizing a user graph arranged on a timeline, which is one axis in 3D space. It represents a snapshot of the states of user networks with specific timing (Figure 1).

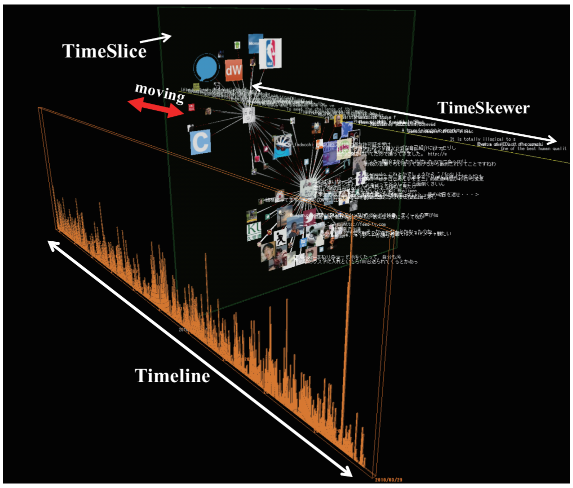

Figure 1: *TimeSlice* and *TimeSkewer*

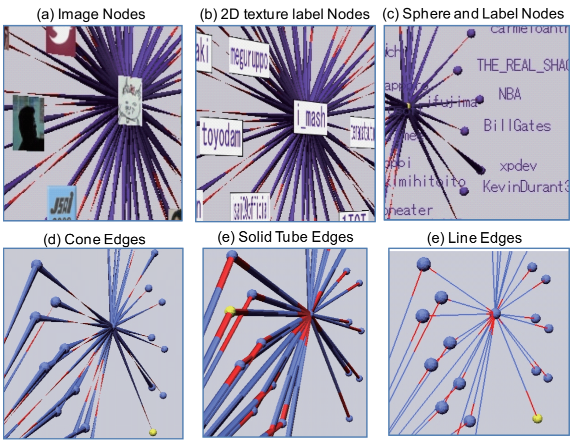

Figure 2: Representations of nodes and edges

A graph on the *TimeSlice* consists of *followers* and the *followings* of a specified user. To display these, we first input a *twitter* user name or id, and select whether we only obtain *followers* or *followings*, or both of these. We can also obtain friends of friends by inputting the depth level, or by selecting the visualized nodes and expanding them.

To visualize connections between users, we have adopted automatic and dynamic graphlayout algorithms to visualize graphs based on a kind of force-directed model (Fruchterman and Reingold (4)). We can select and move nodes and edges to interactively check details on relationship such as the degrees of separation between nodes. Moreover, we can zoom and pan a canvas to interactively change the focusing point in very large graph spaces.

TimeSlices can adopt various kinds of representations for nodes and edges. We provide images, 2D texture labels, and spheres and labels for the nodes described in Figures 2 (a-c), as well as cones, solid tubes, and lines for the edges described in Figures 2 (d-f). The directions of edges to distinguish between followers and followings are represented by using colors and/or acute shapes.

This also enables us to filter visible nodes according to the number of in- and/or out-links, or their ratios so that we only focus on hub and/or authority nodes. Such filtering functions enable us to find key people, and spam or bot user accounts.

TimeSlices can be dragged and we can seamlessly change their positions along a timeline. Such manipulations generate an animated time sequence of changes in the graph and content on it (Figure 1). A *Timeslice* displays the tweets of all users with selected timing. The number of actions such as *#tweet*, *#reply*, *#replied*, *#retweet*, or *#retweeted* around the selected timing is mapped to the size, color, or transparency of nodes and edges. Such functions enable us to display which users are active or have strong connections with others in a particular timing.

We can interactively add new *TimeSlices* along a timeline, and generate multiple views for visualizing graphs with different timing in the 3D environment to compare time-dependant differences as described in Figure 4 (a). To avoid different layouts on the same nodes belonging to different *TimeSlices* and to avoid drastic movements by nodes in the animation, the positions of nodes on different *TimeSlices* can be completely synchronized with one another even if users drag a node. Panning and zooming manipulations are also propagated to other *TimeSlices*.

The system visualizes a histogram on the timeline representing the number of *tweets*, *replies*, and/or *retweets* at each timing (Figure 1). We can dynamically change the scope of the visualized range of time along the timeline by zooming or panning it.

3.2. TimeSkewers

A *TimeSkewer* is a line for visualizing events such as *tweets*, *replies*, and *retweets* on a selected user, as shown in Figure 1. Moreover, the *TimeSkewer* can only display *tweets* with specified keywords, or *replies* or *retweets* for selected persons. By using these functions, we can observe when people talked about certain topics and who they were.

3.3. TimeFluxes

A *TimeFlux* is a line of spheres or solid tubes for visualizing the changes in the number of activities such as *tweets*, *replies*, and *retweets* about selected users within a given period of time, e.g. one day or three hours, as shown in Figure 3 (a). We can map *#tweet*, *#reply*, *#replied*, *#retweet*, and *#retweeted* to the radii, colors, and transparencies of spheres or tubes. We can add two or more *TimeFluxes* to one person to display multiple attribute values, or different keywords. In the case of using *#reply*, *#replied*, *#retweet*, and *#retweeted*, we can select target users to communicate with. By using these functions, we can observe when and how much a person has communicated with particular people on specific topics.

3.4. MessageTrails

MessageTrails form a flow graph for visualizing message flows caused by *replying* and *retweeting*, as shown in Figure 3 (b). We can select target users and source users to *reply* and *retweet* with, and can select particular messages to find the path the messages have spread along. We can also specify keywords to filter *MessageTrails*. They can have different colors depending on the types of messages such as *reply* or *retweet*, target or source users, or selected messages. The directions of the arrows are represented by using the same method as that in Figures 2 (d-f). Optionally, a *MessageTrail* can display messages of *replies*, *retweets*, targets of *replies*, sources of *retweets*, and their time stamps. By using these functions, we can observe how specified topics have spread, and how long they have continued.

3.5. TimeCloud

A *TimeCloud* is a 3D scatter plot mapped on user networks and a timeline for visualizing global tendency on user activities such as *tweets*, *replies*, and *retweets*, as shown in Figure 3 (c). Each plot can change color and transparency according to the kinds of messages, or numbers of messages within a given period of time, e.g., one day or three hours. We can use filters for keywords. By using these functions, we can observe the sizes and positions of user groups that have had interest in the keywords, and when the topic has attracted attention.

4. METHODS OF COMPARISON

Comparing situations with different timing and different user groups is an important task for observing changes in social phenomena in detail. We have taken into consideration two kinds of relations, such as inter-time relations and inter-community relations, and provide methods for comparing them.

4.1. Inter-time relations

Inter-time relations are visualized by multiple *TimeSlices* in different positions on the timeline. Our system allows users to add and compare multiple *TimeSlices*. It enables us to explore changes in graphs by using animation, and by comparing multiple *TimeSlices* as described in Figure 4 (a). Our system also provides *overlay views* and *parallel views* to compare *TimeSlices* in detail in 3D space (Figures 4 (b and c)). These kinds of views enable us to interactively explore information through different perspectives.

An *overlay view* is represented by changing eye positions, and by changing projection modes in a 3D environment. We normally use perspective projection in 3D environments. The same nodes in different *TimeSlices* are then displayed in different positions because of perspective. To solve such problems, we prepared an orthogonal projection mode, where the same nodes in different *Timeslices* completely overlapped positions with one another as can be seen in Figure 4 (b). We also provided a function to change the transparencies of *TimeSlices* to avoid background ones from being hidden.

Our framework enables us to seamlessly change a normal view to a *parallel view*. To achieve this, the system can automatically slide *TimeSlices*. After that, users can obtain a *parallel view* by changing eye positions and projection modes, in the same way as that in the *overlay view* in 3D space, as can be seen in Figure 4 (c).

4.2. Inter-community relations

Inter-community relations are visualized by using multiple *TimeSlices* on different communities that are graphs generated from different groups of users.

Our framework provides three types of views to compare these, which are similar to the ideas introduced by Fung (Fung et al. (5)) and Erten (Erten et al. (3)), i.e., (a) *aggregate*, (b) *pile*, and (c) *split views*. An *aggregate view* visualizes two communities in one *TimeSlice* (Figure 5 (a)), where edges in different communities are in different colors. A *pile view* visualizes two communities in different stacked *TimeSlices* (Figure 5 (b)), where common nodes in differ-

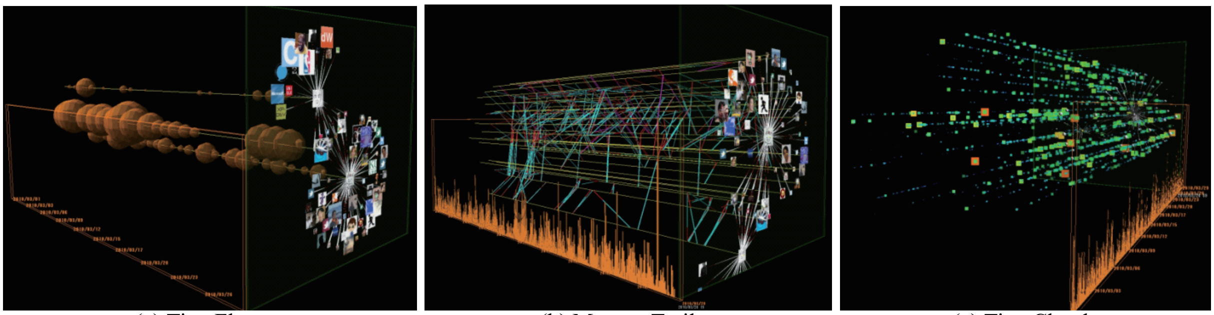
(a) TimeFluxes (b) MessageTrails (c) TimeCloud

Figure 3: *TimeFluxes*, *MessageTrails*, and *TimeCloud*

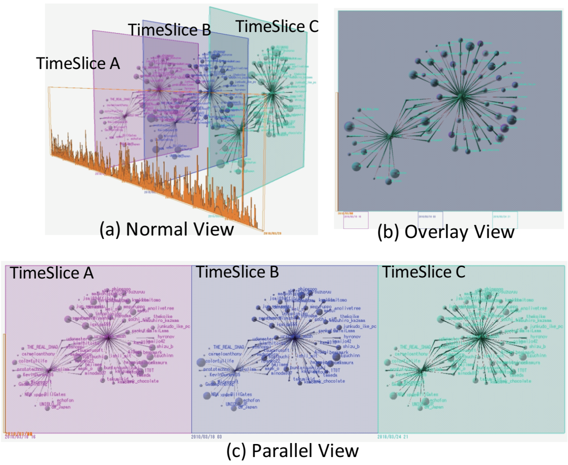

Figure 4: Comparison of multiple *TimeSlices*

ent communities have red edges, as seen in Figure 5 (b). A *split view* visualizes two communities in different *TimeSlices* side-by-side (Figure 5 (c)). In *pile* and *split views*, the positions of two *TimeSlices* along the timeline are synchronized with one another. Users can also add sets of *TimeSlices* to the timeline.

Several methods of drawing two or three overlapping graphs have been introduced (Fung et al. (5), Erten et al. (3)). We provide three types of methods for layouts: (a) *merge*, (b) *pivot*, and (c) *independent* layouts. We treat the same nodes in different communities as one node in the *merge layout* mode, and create a union of nodes in different communities. We then calculate their layouts (Figure 5 (a)). As two graphs in this mode are treated like one graph, nodes in the results never overlap. This is advantageous for exploring the relationships between nodes in two groups. However, it needs a large space to visualize the results. We treat the same nodes as one node in the *pivot layout* mode, and independently calculate the layouts of nodes for two communities (Figure 5 (b)). The shared nodes are treated like pins in the results of this method, and the others are spread around these pins. The *independent layout* mode independently calculates the layouts of all graphs for communities (Figure 5 (c)). Therefore, the visualized results with this method appear compact. This is advantageous for independently exploring changes in each community. However, if we use this method in *aggregate view* or *pile view*, nodes and edges belonging to different communities can easily overlap. As the same nodes in different graphs are treated independently in this layout mode, it is difficult to identify which nodes are shared in both communities. To avoid this situation, the system can add red edges between the same nodes in the same way as seen in Figure 5 (b).

Our framework visualizes inter-community relations using a combination of the *view* and *layout* types.

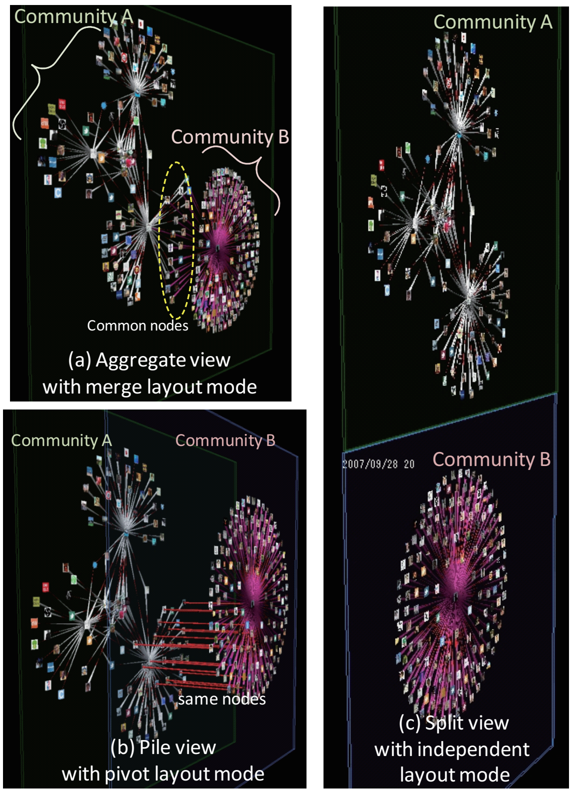

Figure 5: Visualization of multiple communities.

5. CONCLUSIONS

We have proposed 3D visualization techniques. They enables us to interactively explore user activities including their relationships, communications with others, transitions in the volume of sent messages, and message flows on *Twitter*. These combinations enabled us to observe user relationships, communications, and statistical changes in trends in 3D space.

In this work, we only focused on a visualization method of structural analysis. However, a combination of structural analysis and content-based analysis methods is required for analyzing social connections in detail. We intend to provide methods of analyzing and visualizing the content of messages.

6. REFERENCES

[1] U. Brandes and S. R. Corman. Visual Unrolling of Network Evolution and the Analysis of Dynamic Discourse. *Information Visualization*, 2(1):40–50, 2003. ISSN 1473-8716.

[2] T. Dwyer and P. Eades. Visualising a Fund Manager Flow Graph with Columns and Worms. In *Proceedings of the Sixth International Conference on Information Visualisation*, pages 147–152, 2002.

[3] C. Erten, S. G.Kobourov, V. Le, and A. Navabi. Simultaneous Graph Drawing: Layout Algorithms and Visualization Schemes. In *The 11th Symposium on Graph Drawing*, pages 437–449, 2003.

[4] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software Practice and Experience*, 21(11):1129–1164, 1991. ISSN 0038-0644.

[5] D. C. Y. Fung, S.-H. Hong, D. Koschutzki, F. Schreiber, and K. Xu. Visual Analysis of Overlapping Biological Networks. In *Proceedings of the 2009 13th International Conference Information Visualisation*, pages 337–342, 2009.

[6] M. Gladwell. *The Tipping Point: How Little Things Can Make a Big Difference*. Little Brown, Boston, MA, USA, 2000.

[7] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002. ISSN 1077-2626.

[8] E. H.Chi and S. K.Card. Sensemaking of Evolving Web Sites Using Visualization Spreadsheets. In *Proceedings of the 1999 IEEE Symposium on Information Visualization*, pages 18–25, 1999.

[9] E. H.Chi, J. Pitkow, J. Mackinlay, P. Pirolli, R. Gossweiler, and S. K.Card. Visualizing the evolution of Web ecologies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 400–407, 1998.

[10] M. Itoh, M. Toyoda, and M. Kitsuregawa. An Interactive Framework for Visualizing Time-series of Web Graphs in a 3D Environment. In *Proceedings of the 14th International Conference on Information Visualization*, 2010.

[11] A. Java, X. Song, T. Finin, and B. Tseng. Why We Twitter: Understanding Microblogging Usage and Communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 56–65, 2007.

[12] K.-S. Kim, K. Zettsu, Y. Kidawara, and Y. Kiyoki. Moving Phenomenon: Aggregation and Analysis of Geotime-Tagged Contents on the Web. In *Proceedings of the 9th International Symposium on Web and Wireless Geographical Information Systems*, pages 7–24, 2009.

[13] H. Koike, T. Takada, and T. Masui. VisuaLinda: A Framework for Visualizing Parallel Linda Programs. In *Proceedings of the 1997 IEEE Symposium on Visual Languages*, pages 174–182, 1997.

[14] G. Matthews and M. Roze. Worm Plots. *IEEE Computer Graphics and Applications*, 17(6):17–20, 1997. ISSN 0272-1716.

[15] W. Müller and H. Schumann. Visualization Methods for Time-dependent Data - An Overview. In *Proceedings of the 35th conference on Winter simulation*, pages 737–745, 2003.

[16] N. Nakazono, K. Misue, and J. Tanaka. NeL2: Network Drawing Tool for Handling Layered Structured Network Diagram. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation*, pages 109–115, 2006.

[17] T. Nocke, M. Flechsig, and U. Böhm. Visual Exploration and Evaluation of Climate-Related Simulation Data. In *Proceedings of the 39th conference on Winter simulation*, pages 703–711, 2007.

[18] H. Notsu, Y. Okada, M. Akaishi, and K. Niijima. Time-Tunnel: Visual Analysis Tool for Time-Series Numerical Data and Its Extension toward Parallel Coordinates. In *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*, pages 167–172, 2005.

[19] Y. Okada and Y. Tanaka. IntelligentBox: A Constructive Visual Software Development System for Interactive 3D Graphic Applications. In *Proceedings of Computer Animation '95*, pages 114–125, 1995.

[20] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: Visualizing Personal Histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 221–227, 1996.

[21] J. Rekimoto. Time-Machine Computing: a Time-centric Approach for the Information Environment. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 45–54, 1999.

[22] Z. Shen, M. Ogawa, S. T. Teoh, and K.-L. Ma. BiblioViz: a System for Visualizing Bibliography Information. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation*, pages 93–102, 2006.

[23] M. Toyoda and M. Kitsuregawa. A System for Visualizing and Analyzing the Evolution of the Web with a Time Series of Graphs. In *Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia*, pages 151–160, 2005.

[24] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1983.