

An Interactive Visualization Framework for Time-series of Web graphs in a 3D Environment

Masahiko Itoh, Masashi Toyoda, and Masaru Kitsuregawa
Institute of Industrial Science, The University of Tokyo, Japan
{imash,toyoda,kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract

We propose an interactive framework for the 3D visualization of the time-series of Web graphs. The purpose of our framework is to enable users to examine the evolution of Web graphs by comparing multiple graphs that have different timings and topics. To achieve this, we provide TimeSlices that are 2D planes to visualize Web graphs in a 3D environment. Users can interactively add new TimeSlices along the timeline, and they can manipulate them to animate Web graphs. Visualized Web graphs on TimeSlices are snapshots of different timings. Moreover, we provide mechanisms to generate and manipulate overlay views and parallel views in the 3D environment to enable Web graphs to be compared. Users can seamlessly change these viewing modes. Our system can also visualize two Web graphs created from different keywords to display the evolutionary correlation between them. This system enables us to observe structural and temporal changes in multiple Web graphs such as shifts in influential blogs or power shifts in the diffusion of blogs between topics.

1. Introduction

The Web currently reflects real movements in society. Since blog platforms have become popular, people can easily write their opinions and comments on events in the real world. We have collected and archived Japanese Web content including blogs for over a decade [1] and we can extract the time-series of Web graphs from this Web archive. This enables us to analyze structural and temporal changes in content on the Web that reflect both real and virtual activities.

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 timelines, (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? [2]. Being able to visualize the time-series of Web graphs enables us to display changes in thought that occur in the real and/or virtual world.

Four main methods have been proposed to enable the evolution of the time-series of Web graphs to be visualized. The first is (a) using *animation* to dynamically display changes in structures [1, 6]. 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 of the Web. The second is (b) mapping a *timeline* to one of the axes in a 3D environment [3]. This enables users to observe global differences between multiple Web graphs. However, it is difficult to check local differences in detail. The third is (c) using multiple *tiled views* to display multiple Web graphs [4, 5]. 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 sometimes cannot determine how long it has taken for changes to have happened. The fourth is (d) *overlaying* Web graphs for different time periods on one view [6, 19]. This is advantageous for comparing Web graphs in detail; however, is difficult to display global changes that have occurred in structures.

This paper proposes an interactive framework for the 3D visualization of the time-series of Web graphs. The purpose of our framework is to enable users to examine the evolution in Web graphs by comparing multiple graphs that have different timings and topics. Our method integrates the four methods of *animation*, *timelines*, *tiled views*, and *overlay views*, which enable the evolution of the time-series of Web graphs using a 3D visualization environment to be visualized. It enables us to seamlessly change the four kinds of viewing modes.

To achieve this, we provide TimeSlices, which are 2D planes that enable Web graphs to be visualized, and these enable users to utilize multiple views in a 3D environment. Users can interactively add new TimeSlices along a timeline, which is one axis in the 3D environment, and they can manipulate them to animate Web graphs. Visualized Web graphs on TimeSlices are snapshots of different timings or are created from different keywords. These mechanisms enable users to observe when and how these graphs have changed and

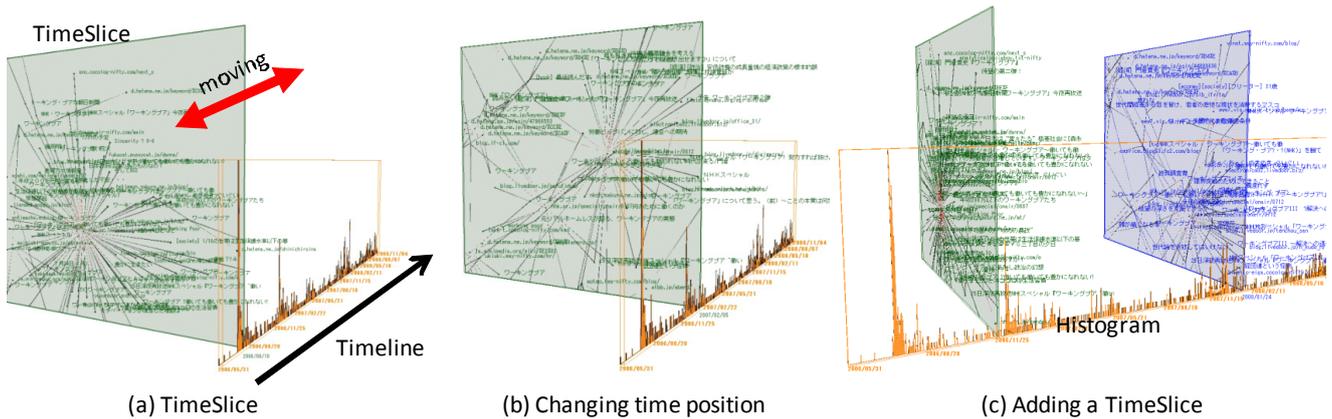


Figure 1 TimeSlices

how long it has taken for changes to have occurred. Moreover, we provide methods of generating overlay views and parallel views using TimeSlices in the same 3D environment. Overlay views and parallel views enable users to observe where changes in Web graphs have occurred and how they have been caused by allowing them to be compared in detail. By using TimeSlices to generate and manipulate multiple views in 3D space, users can seamlessly switch types of views on demand, and they can observe them from global and local aspects.

Our system can also visualize two Web graphs created from different keywords to show the evolutionary correlation between them. Two Web graphs occasionally have overlapping nodes, and change differently along a timeline. Users can compare them using an *aggregate view*, a *pile view*, or a *split view*.

Our framework enables us to observe structural and temporal changes in multiple Web graphs such as shifts in influential blogs over time, those in common blogs and uncommon blogs on different topics, and power shifts in the diffusion of blogs between topics.

2. TimeSlices

A TimeSlice [23] is a plane for visualizing a Web graph arranged on a timeline, which is one axis in 3D space. It represents a snapshot of a Web graph with a specified timing (Figure 1 (a)).

Web graphs are generated from a blog archive, which is constructed from 2006 and includes over 1.4 million blogs and over 300 million posts. Users can extract a Web graph that uses specified keywords from the blog archive. Each node in a Web graph represents a blog feed or a Web page, and each edge represents a link between feeds or Web pages. In the example given in Figure 1, each TimeSlice visualizes a Web graph on the keyword “working poor”.

All TimeSlices can be dragged. We can seamlessly change the positions of TimeSlices along a timeline. Such manipulations generate an animated time sequence of structural changes in the Web graph (Figures 1(a-b)).

By using these, we can easily find source blogs and influential blogs that have diffused topics in the blog space. We can also add new TimeSlices to the timeline by clicking on it (Figure 1(c)) to compare structural differences between Web graphs with different timings in detail. The added TimeSlices are positioned at the place the time is clicked. TimeSlices can be colored differently to easily distinguish them. The system visualizes a histogram on the timeline representing the number of nodes generated at each timing. Users can intuitively recognize the timing when the selected keywords become popular on the Web (Figure 1).

In the example in Figure 11, we can see that many people wrote their opinions on the blog after the TV program called the “working poor” had been broadcast. Most of their blogs were linked to the official page of the TV program. The second peak appeared after the sequel “working poor II” had been broadcast. We can find the center shifts in focus from the official pages of “working poor” to “working poor II”. We can also see that many clusters have spread and the term “working poor” has become popular after these two peaks. We can also see the variety in topics in these clusters. These are derived from or related to the term “working poor”.

We consider that Web graphs have three kinds of relations. (i) *Intra-graph* relations mean the relationships between nodes in one TimeSlice. Users can observe links between nodes, the distance between nodes, clusters of nodes, and hub and authority nodes. (ii) *Inter-time* relations mean the relationships between Web graphs with different timings. Users can observe the appearance and disappearance of nodes, changes in clusters, and changes in hub and authority nodes. (iii) *Inter-facet* relations mean the relationships between facets. We consider Web graphs extracted from different keywords to be facets that project the entire blog space from the aspect of selected keywords. Users can observe common nodes and/or common clusters in different facets, and which clusters have increased in size earlier than the others.

Our framework provides an interactive visualization environment enabling users to explore time sequential Web graphs using three kinds of relations. We utilized

the IntelligentBox system [14] as the platform to implement this. This is a component-based visual software development system for interactive 3D graphics applications.

3. Visualizing intra-graph relations

Intra-graph relations are visualized on a graph that consists of the nodes and edges of a selected time in a TimeSlice. We adopt automatic and dynamic graph layout algorithms to visualize graphs based on a kind of force-directed model [7, 11] that considers a graph to be a physical system. Attractive forces in the force-directed model are exerted on all pairs of connected nodes, and repulsive forces are exerted on all pairs of nodes.

We first construct a combination of all graphs and calculate its layout to avoid different layouts on the same nodes belonging to different TimeSlices and to avoid drastic movements by nodes in the animation. Each TimeSlice then only visualizes nodes and edges related to a specified time by controlling when they appear and disappear.

Users can select and move nodes and edges to interactively check details on the relationship between nodes. Moreover, they can zoom and pan the canvas to interactively change the focusing point in very large Web graph spaces. The system also enables URLs to be opened from the pop-up menu of the selected nodes.

Users can adopt various kinds of representations for nodes and edges. We provide 3D shapes such as spheres and cubes, labels, and 2D texture labels for the nodes described in Figures 2 (a-c), as well as lines, solid tubes, and cones for the edges described in Figures 2 (d-f). Users can interactively change the representations of nodes and edges according to their situation in exploration. TimeSlices can change the color, size, shapes, and transparency of nodes and edges according to the attributes values of nodes. The directions of edges are represented by using colors and/or acute shapes.

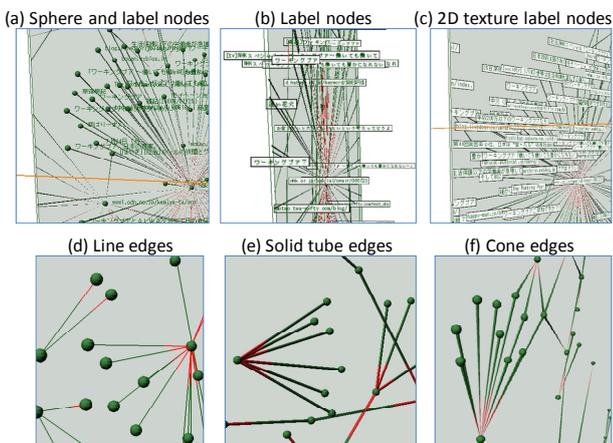


Figure 2 Representations of nodes and edges

We can use three types of techniques in our 3D environment for drawing labels: the (i) bitmap label used in Figure 2 (a), the (ii) texture map label used in Figure 2 (b), and the (iii) 3D polygon font label. The bitmap label enables us to avoid background texts from being hidden. However, users sometimes miss the position of depth in texts because they use the same bitmaps despite their depth positions. Texture map labels and 3D polygon font labels are easy to read, and can change the size of the drawing depending on depth positions. However, they occasionally hide background objects.

Labels sometimes cause problems with occlusion in 3D environments. Many techniques have been proposed [15] to solve the problem with occluded labels. We use the four techniques described by Fekete and Plaisant [15], i.e., *no-label*, *rapid label-all*, *label-what-you-can (lwy)* as a static type, and a *cursor sensitive balloon label (csbl)* as a dynamic type. Users can select the type of labeling mode. The *lwy* mode requires a function to filter nodes so that labels can be displayed. For example, a filter controls ranges to display whether or not labels will be shown according to the number of in- and out-links. Users can only see the detail on nodes that have numerous or few links. They can change the distance level to display labels from selected nodes with the *lwy* and *csbl* modes. Figure 3 (a) only shows the label of a clicked node, and Figure 3 (b) shows the labels of nodes, whose distances from the clicked node are two steps or less.

The system also enables users to filter visible nodes according to the number of in- and/or out-links to only focus on hub and/or authority nodes.



(a) Labeling only selected node (b) Labeling nodes within 2 steps

Figure 3 Labeling area

4. Visualizing 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).

Koike [10] used a 3D space to simultaneously visualize two relations such as physical placement and time relation to avoid disturbing user cognition caused by reconstruction of their mental model. Our framework simultaneously uses *intra-graph* relations and *inter-time* relations. It enables users to explore the diffusion of Web

graphs while considering time relations by using animation and multiple TimeSlices, as outlined in Figure 4 (a). Our framework also provides overlay views and parallel views to enable TimeSlices to be compared in detail in 3D space (Figures 4 (b, c)).

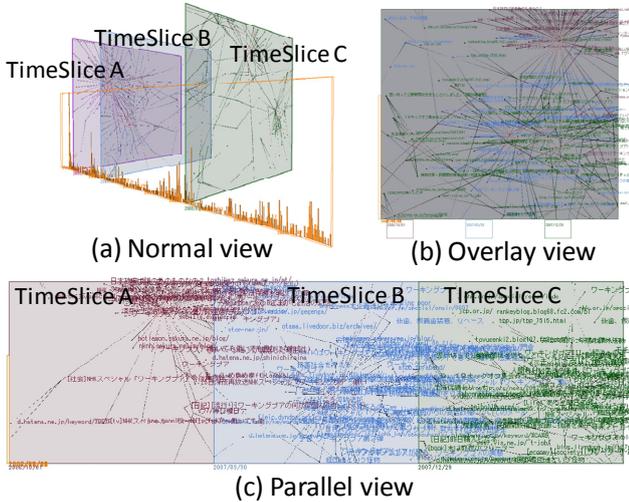


Figure 4 Comparison of Multiple TimeSlices

Many visualization systems have adopted multiple view environments [13, 17] to enable information to be explored and avoid rich data being missed caused by using a single state model [16]. Three patterns can be considered for generating and comparing multiple variations: (i) *overlaying* variations in one view, (ii) displaying other variations in separate views *in parallel*, and (iii) *sequentially* replacing the current visualization with another one in one view.

Our framework provides mechanisms for using these techniques in a 3D environment. It enables users to interactively explore information through different perspectives using multiple views. In the example in Figure 4, we have visualized Web graphs related to the keyword “working poor”, and can observe how word-of-mouth information has spread via blogs and how influential nodes have changed in the blog space.

The system with our graph-layout algorithm described in Section 3 constructs one graph including nodes and edges for whole time stamps, then calculates their layouts, and only displays the necessary nodes and edges for a selected timing. The positions of nodes on different TimeSlices can be completely synchronized with one another with this technique even if users drag a node as seen in Figure 5. Panning and zooming operations are also propagated to other TimeSlices.

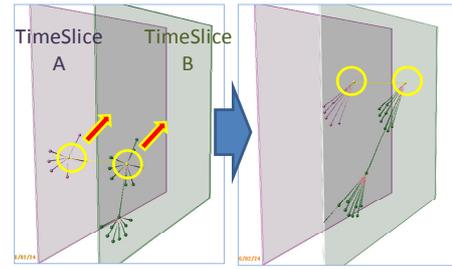


Figure 5 Synchronized graph layout

Figure 6 shows links between the same nodes belonging to different TimeSlices. They enable users to quickly recognize common nodes on other TimeSlices. Moreover, they allow users to intuitively monitor the timing when nodes appear and/or disappear.

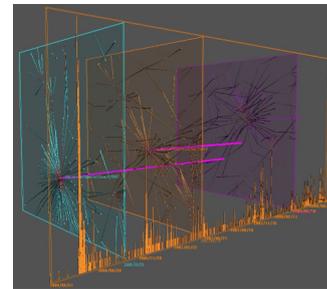


Figure 6 Links between the same nodes on different TimeSlices

4.1. Overlay View

An *overlay view* is represented by changing eye positions, and by changing projection modes in a 3D environment. If users change their eye positions to the right of Figure 4 (a), they can obtain results such as those in Figure 7 (a). We normally use perspective projection in a 3D environment. The same nodes in different TimeSlices are then displayed in different positions because of perspective as seen in Figure 7 (a). To solve such problems, we organize an orthogonal projection mode, where the same nodes in different TimeSlices completely overlap positions with one another, as can be seen in Figure 7 (b).

We also provide a function to change the transparency of TimeSlices to avoid background objects from being hidden.

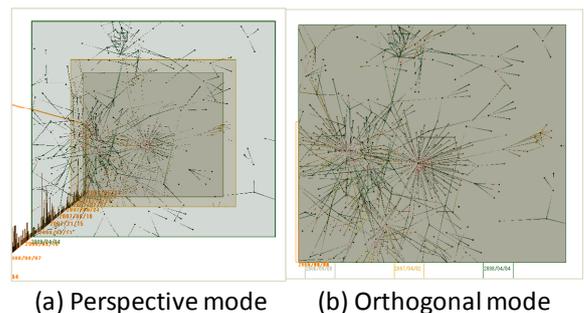


Figure 7 Overlay view

Our system provides a function to control the visibility and invisibility of TimeSlices. If users change the visible state in the *overlay view*, they can obtain the same effect as with undo/redo operations (Figure 8), which are normally used for *sequentially* comparing multiple variations.



Figure 8 Sequentially changing visible TimeSlices for comparison

4.2. Parallel View

Our framework enables users to seamlessly change a normal view to a *parallel view*. To achieve this, the system can automatically slide TimeSlices (Figure 9). After that, users can obtain a *parallel view* (Figure 4 (c)) by changing eye positions and projection modes in the same way as in the *overlay view*.

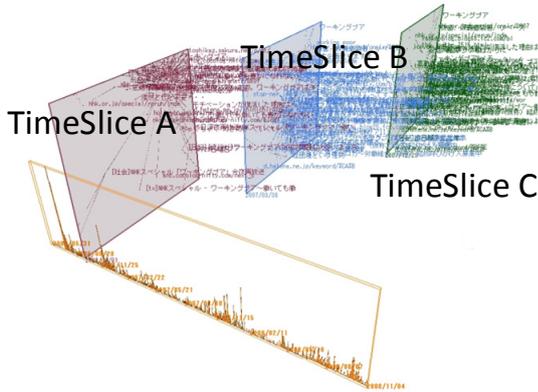


Figure 9 Sliding TimeSlices

5. Visualizing inter-facets relations

Inter-facets relations are visualized by using multiple TimeSlices on different facets that are Web graphs generated from different keywords.

Our framework provides three types of views to compare multiple facets, which are similar to the ideas introduced by Fung [8] and Erten [9], i.e., (a) *aggregate*, (b) *pile*, and (c) *split views*. An *aggregate view* visualizes two facets in one TimeSlice (Figure 10 (a)), where nodes and edges in different facets are in different colors. Common nodes are in light green. A *pile view* visualizes two facets in different stacked TimeSlices (Figure 10 (b)), where common nodes in different facets have the same color, and the same nodes have red links, as seen in Figure 10 (b). A *split view* visualizes two

facets in different TimeSlices side-by-side (Figure 10 (c)). Users can optionally show edges for the same nodes in different facets in the *aggregate* and *split views*. In the *pile* and *split views*, the positions of 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 [8, 9, 12]. We provide three types of methods for layouts: (a) *merge*, (b) *pivot*, and (c) *independent layouts*. We treat the same nodes in different facets as one node in the *merge layout* mode, and create a union of nodes in different facets. We then calculate their layouts (Figure 10(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 facets. 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 of two facets (Figure 10 (b)). The shared nodes are treated like pins in the results of this method, and others are spread around these pins. The *independent layout* mode independently calculates the layouts of all graphs for facets (Figure 10(c)). Therefore, the visualized results obtained with this method appear compact. It is good for independently exploring the evolution of all facets. However, if we use this method in the *aggregate view* or *pile view*, nodes and edges belonging to different facets 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 facets. To avoid this situation, the system can add links between the same nodes as seen in Figure 10 (b).

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

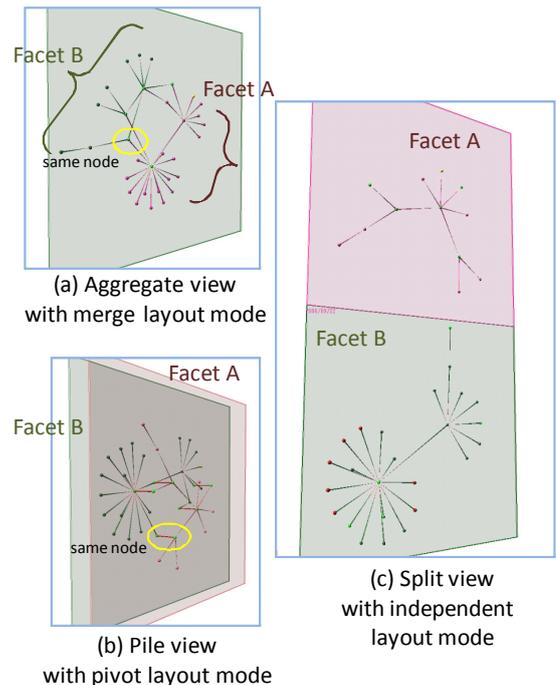


Figure 10 Visualizing inter-facet relations

6. Related work

There have been many systems and/or research projects for visualizing the evolution of information structures [1, 3, 4, 5, 6, 19]. These have used animation, a timeline on 3D, tiled views, and overlaying views to represent time-sequential changes in the structure of information. All these techniques have their pros and cons as discussed in Section 1. However, there have been no systems that have integrated these techniques to solve problems with limitations.

Many systems for different data domains use a 2.5D representation [3, 8, 9, 18, 19, 20, 21] to visualize multiple situations in a 3D environment. The 2.5D representation is used for three kinds of visualizations that involve: (i) visualizing different content [8, 9], (ii) visualizing time sequential changes [3, 19, 20, 21], and (iii) using different visual representations and/or models [18, 21]. 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.

Some systems use a combination of 3D space and a timeline [3, 10, 22]. They effectively use the 3D space to simultaneously represent two kinds of relations including the time relation.

Conclusion

This paper proposed an interactive framework for visualizing the time-series of Web graphs using 3D space. Our framework provides multiple planes representing Web graphs, called TimeSlices, on a timeline in a 3D environment. Our system enables users to use animation, tiled views, and overlay views in 3D space to explore the evolution of time sequences. Moreover, it provides a function for comparing the evolution of two Web graphs related to different keywords, called facets. We have presented the details on these functions of the proposed framework through examples implemented on the system.

Our methods can be applied to various kinds of time-series in networks such as co-citation networks. We currently plan to provide functions to enable other visual representations and layout algorithms to be used, such as multidimensional scaling (MDS) and self-organizing maps (SOMs), for visualizing the evolution of topics in blog spaces.

References

- [1] M. Kitsuregawa, T. Tamura, M. Toyoda, and N. Kaji. Socio-Sense: A System for Analysing the Societal Behavior from Long Term Web Archive. In APWeb, LNCS, Vol. 4976, pp. 1–8, 2008.
- [2] Müller, W. and Schumann, H, Visualization methods for time-dependent data - an overview, In Proceedings of the 35th Conference on Winter Simulation, pp. 737–745, 2003.
- [3] E. H. Chi et al. Visualizing the evolution of Web ecologies. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 400–407, 1998.
- [4] 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, pp. 18–25, 1999.
- [5] Toyoda, M. and Kitsuregawa, M. 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, pp. 151–160, 2005.
- [6] Nakazono, N., Misue, K., and Tanaka, J., NeL2: network drawing tool for handling layered structured network diagram. In Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation, pp. 109–115, 2006.
- [7] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software Practice and Experience*, 21 (11), pp. 1129–1164, 1991.
- [8] Fung, D. C., Hong, S., Koschutski, D., Schreiber, F., and Xu, K., Visual Analysis of Overlapping Biological Networks. In Proceedings of the 2009 13th International Conference on Information Visualisation, pp. 337–342, 2009.
- [9] Erten, C., Kobourov, S. G., Le, V., and Navabi, A., Simultaneous Graph Drawing: Layout Algorithms and Visualization Schemes, The 11th Symposium on Graph Drawing, pp. 437–449, 2003.
- [10] H. Koike. The Role of Another Spatial Dimension in Software Visualization, *ACM Trans. on Information Systems*, Vol. 11, No. 3, pp. 266–286, 1993.
- [11] Alexander Shapiro. Touchgraph. <http://www.touchgraph.com/>.
- [12] S. Diehl and C. Görg. Graphs, They are Changing. In The 10th Symposium on Graph Drawing, pp. 23–30, 2002.
- [13] J. C. Roberts. Multiple-View and Multiform Visualization. In *Visual Data Exploration and Analysis VII*, Proceedings of SPIE, Vol. 3960, pp. 176–185, 2000.
- [14] Y. Okada and Y. Tanaka. IntelligentBox: A Constructive Visual Software Development System for Interactive 3D Graphic Applications. In Proceedings of Computer Animation '95, pp. 114–125, 1994.
- [15] Fekete, J. and Plaisant, C., Eccentric labeling: Dynamic neighborhood labeling for data visualization. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 512–519, 1999.
- [16] M. Terry and E. D. Mynatt. Recognizing Creative Needs in User Interface Design. In Proceedings of the 4th Conference on Creativity and Cognition, pp. 38–44, 2002.
- [17] M. Itoh, J. Fujima, M. Ohigashi, and Y. Tanaka, Spreadsheet-based Framework for Interactive 3D Visualization of Web Resources, In Proceedings of the 11th International Conference on Information Visualization, pp. 65–73, 2007.
- [18] Shen, Z., Ogawa, M., Teoh, S. T., and Ma, K., BiblioViz: a system for visualizing bibliography information. In Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation, pp. 93–102, 2006.
- [19] Brandes, U. and Corman, S. R., Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization* 2, 1, pp. 40–50, 2003.

- [20] Nocke, T., Flechsig, M., and Böhm, U. Visual exploration and evaluation of climate-related simulation data. In Proceedings of the 39th Conference on Winter Simulation, pp. 703–711, 2007.
- [21] von Pilgrim, J. and Duske, K. GEF3D: a framework for two-, two-and-a-half-, and three-dimensional graphical editors. In Proceedings of the 4th ACM Symposium on Software Visualization, pp. 95–104, 2008.
- [22] Notsu, H., Okada, Y., Akaishi, M., and Nijjima, K. 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, pp. 167–172, 2005.
- [23] M. Itoh, M. Toyoda, and M. Kitsuregawa, TimeSlices: An Interactive 3D Visualization System for Time-series of Web graphs, 17th Workshop on Interactive Systems and Software, pp. 141–142, 2009.

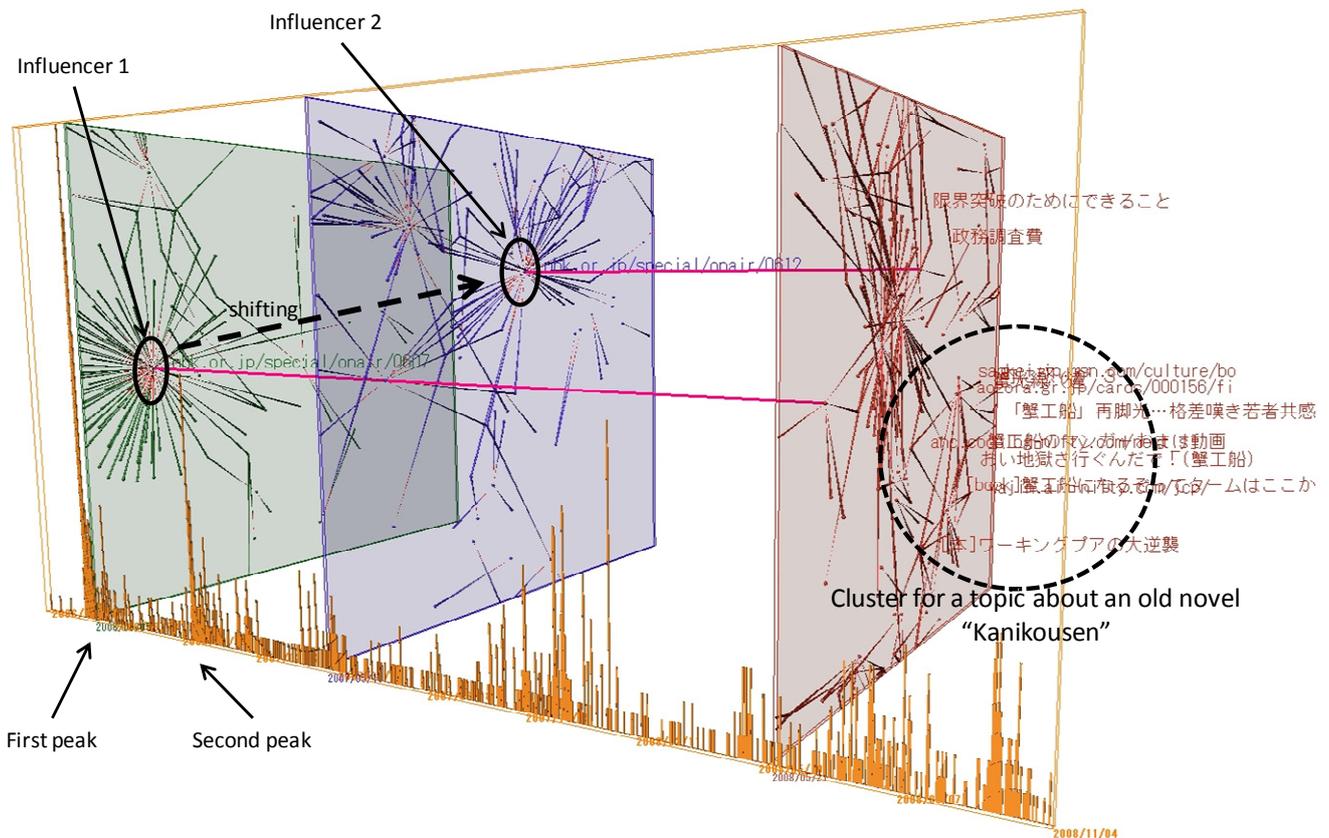


Figure 11 Example: We can see that many people wrote their opinions on the blog after the TV program called the "working poor" had been broadcast. Most of their blogs were linked to the official page of the TV program. The second peak appeared after the sequel "working poor II" had been broadcast. We can find the center shifts in focus from the official pages of "working poor" to "working poor II". We can see that many clusters have spread and the term "working poor" has become popular after these two peaks. We can also see the variety in topics in these clusters. These are derived from or related to the term "working poor".