# Visualizing Information Evolution

Yu-Ting Wong* Kai-Yin Cheng* Sheng-Jie Luo* Bing-Yu Chen
National Taiwan University
*{callia,keynes,forestking}@cmlab.csie.ntu.edu.tw robin@ntu.edu.tw

## ABSTRACT

In this paper, we propose a novel graph representation, *Railway Visualization*, to reveal the information evolution with temporal relationship over time. Through *Railway Visualization*, it cannot only reveal the changes of an entity's intra-state over time, but also can effectively disclose the intrinsic characteristics of the complicated relationship evolution, such as branching, crossing, splitting, converging and parallelism. Using the railway route map in people's daily life as the metaphor to visualize the information evolution can leverage people's existing knowledge and mental models to quickly interpret the graph and gain the insight. We then conclude the drawing aesthetic rules by observing the railway route maps and formulate them as the energy terms. By solving them through a two-steps optimization method, the railway-route-map-style layout can be rendered automatically. Furthermore, an interactive level-of-detail mechanism is also provided. Finally, several types of data, including movies, photo albums, news, and co-author network are visualized to prove the virtues of the *Railway Visualization*.

## 1. INTRODUCTION

Relationship visualization has received lots of attention in information visualization community, and it is not only widely adopted in Social Network Analysis (SNA) and software development for practical usages, but also adopted in new research areas like Casual Information Visualization. Traditional information visualization techniques usually use a node-link diagram (graph) to represent the relationship between some entities [11, 3], where nodes are encoded as the entities, and links are encoded as the relationship between them. Hence, the graph can be used to represent a snapshot of the relationship network as a static graph, but omits the evolution features of the relation data in the real world [15]. However, data and its relationship usually vary over time. A snapshot of a static graph cannot provide sufficient messages, and cannot reveal the information about the trend, which is critical while analyzing the temporal changes of the data and its relationship. Hence, there are many visualization techniques arising in order to represent the evolution characteristics of the data recently.

To visualize the relationship changes, Toyoda and Kitsuregawa [26] used a series of snapshots to reveal the evolution about link relationship among the web pages. Collberg *et al.* [4] used an animation to visualize the changes of a software system. Ogawa and Ma [19] even used an organic information visualization technique to represent the qualitative view for software development history and evolution like a music video. Though the prior works are dedicated to reveal the relationship evolution with different proposed visualization methods, they still fail to show the evolution overview over time in a single intuitive represented view. Moreover, they are limited to answer the following questions:

- What is an overview of the evolution trend over a period of time?
- What major events were taken placed during the period? How did such events affect the participated entities?
- Did the entities change their intra states over time? Did any relationship be changed while the intra states were changed?
- For the interested entities, what were the interactions happened between them during a period of time?

The above questions are important especially when users read the graph and want to know the evolution of a network relationship. Therefore, how to effectively visualize the information evolution is a challenge.

In this paper, we propose a novel graph representation, *Railway Visualization*, to reveal the information evolution among data over time. As the name shown, our work is inspired by the railway route map in our daily life. We use the railway as the metaphor to encode the characteristics of the information evolution. Unlike the traditional encoding of graph or network visualization, we encode each railway track (linked-edges) as an individual data entity and a railway station (node) as an event. Through this way, nodes can be encoded with states as well as the relationship, such that the information about relationship evolution and intra-state evolution can be visualized at the same time. Moreover, the *Railway Visualization* can effectively visualize the temporal relationship changes among the entities, such as branching, merging and parallelism [5, 14]. On the one hand, the evolution of the relationship can be easily observed while they intersect at certain stations (nodes) during a period of time. On the other hand, the temporal intra-state changes of a specific data entity can also be traced by traversing


*Computer Graphics Workshop* 2010 Hsinchu, Taiwan

a sequence of stations (nodes) in a track (edges). Therefore, benefiting from the mental model through the visual metaphor, the relationship evolution among the interested entities and temporal changes of an entity's intra-state can be understood at one glance.

Hence, we first leverage people's existing knowledge about the railway route maps as the metaphor to design the way to visualize the information evolution, and then define the aesthetic rules for the railway-route-map-style visualization. Then, a two-steps optimization method is used to minimize a set of energy terms modeled by the railway aesthetic rules for drawing the railway layout automatically. Furthermore, we also provide an interactive level-of-detail mechanism for users. Moreover, to prove the virtues of the *Railway Visualization*, several types of data including movies, photo albums, news and co-author network are visualized through that way.

# 2. RELATED WORK

Rather than portraying the conclusion of the relationship between some entities over a period of time as traditional graph or network visualization [11, 3], our work aims to reveal the information evolution which includes both of the relationship evolution and intra-state evolution. The former one focuses on the characteristics of temporal relationship between some entities, including crossing, branching, merging and parallelism. The latter one focuses on the changes of an entity's intra states and highlights the experienced major events. In the following, we will review the prior literatures of the above two topics.

## 2.1 Relationship Evolution

In order to portray the dynamically changed information, there are a wide variety of techniques have been developed in Social Network Analysis (SNA) [7], Web Document Visualization and Software Visualization [6]. Viégas *et al.* proposed *Themail* [30], which utilizes the conversational keywords inside the email archives as the visualization material, and uses a typographic visualization method to represent the interactive and conversational relationship between the individuals. *Semantic Timelines* [12] draws a sequence of events by stacking timelines vertically, and uses different shapes and colors of arrow lines to convey the semantic relationship. However, due to the timeline constraints of vertical alignment, it leverages space encoding with less freedom than graph or network layout. Besides, the above timeline-base approaches usually limit to visualize the evolution of relationship between two entities only.

Pinzger *et al.* [20] modified the traditional Kiviat diagram as *RelVis* to visualize the temporal relationship changes in one static graph with multiple metrics. It still cannot visualize the changes of complicated temporal relationship, like crossing, branching, merging and parallelism. *WebRelieo* [26] uses a series of snapshots to reveal the evolution about the relationship among some linked web pages, such as appearance or disappearance of a web page and the relation changes. Every snapshot represents a time slice of the graph in the dynamic change processes. Though it uses different colors to highlight the changes of temporal relation among the web links, it still needs lots of efforts to compare the different snapshots to match the changed relationship between the web pages.

Instead of using one or several snapshots to visualize the information evolution at a time, animation is another approach. Kumar and Garland [13] proposed an interaction visualization technique in order to explore a dynamic time-varying graph. Collberg *et al.* proposed *GRVOL* [4], which uses an animation with colors to visualize the evolution of a software system. Though colors are used to encode the authors, it can only reveal a few relationship between them. Inspired by organic information visualization techniques, *code_swarm* [19] uses an animation video to represent software development history and evolution. The animation can convey the involvement of different authors through time, but cannot reveal the changes of their relationship. Viégas *et al.* [29] used an animation to portray the rhythm of relationship. Though the animation-based approaches can visualize the information evolution over time, it is time consumption and is limited to provide a clear overview at one glance.

## 2.2 State Evolution

Intra-state evolution focuses on the changes of an entity's intra-state during a period of time. It has been well studied by the visualization techniques for time-varying data, such as timeline, stacked graph, sector graph, circle graph, multivariate visualization, spiral graph, etc. [1, 24, 16] Among the different visualization methods, the timeline-based approaches are the most general way to represent the changes of an intra-state during a period of time. *Google living stories*[1] uses a straight line to sketch the development of a specific news topic. Plaisant *et al.* proposed *Lifeline* [21] to visualize personal history records. It is intuitive for timeline-based visualization techniques to represent the state evolution for each entity in a linear chronological order, but it is limited to reveal more information about the branching characteristic of the time-varying data [5].

The stacked graph, such as *ThemeRiver* [10], *NameVoyager* [31] and *The Ebb and Flow of Movies: Box Office Receipts 1986-2008*[2] is another common visualization technique to reveal the evolution over time. Using the river as the metaphor, stacked graph encodes each theme as a stream and leverages the thickness of each stream to show the quantitative changes during a period of time [2]. Through such a visual design, stacked graph cannot only reveal the quantitative changes for each individual stream, but also can provide a global view of the evolution among different streams to compare their relative strength. However, it still cannot reveal the branching characteristic of the time-varying data. Besides, it also lacks of the landmarks to highlight the significant events which cause changes.

Hence, though the prior works can effectively deal with the evolution of relationship or the evolution of intra-state separately, they cannot provide a single overview with those two aspects at the same time.

# 3. DESIGN

This work is motivated by the lack of revealing the information evolution with temporal relationship. Hence, our design of the visual encoding has to capture the following features: (1) revealing the characteristics of the temporal relation between any two data entities, such as crossing, split-

[1]http://livingstories.googlelabs.com/

[2]http://www.nytimes.com/interactive/2008/02/23/movies/20080223_REVENUE_GRAPHIC.html

ting, merging and parallel relationships; (2) representing the temporal changes and major events, which an entity has experienced; (3) highlighting the important events among the temporal data; (4) providing a clear global overview at one glance. Considering all of the mentioned aspects, we designed a new visual encoding, *Railway Visualization*, by leveraging the railway metaphor in order to help users gain more insights about the evolution of the temporal relationship.

## 3.1 Using Railway Route Map as the Metaphor

Inspired by the railway route map, we found that its natural characteristics is similar with the features of information evolution with temporal relation. Firstly, a track traverses a sequence of sites as an individual entity had experienced a series of events over time. Secondly, several tracks cross at an exchange site as several entities had experienced a same event. Thirdly, there is a parallel relationship among different tracks as different entities had different experiences at the same period. For example, each individual in a social network can be encoded as a track, and an experienced event can be encoded as a railway site. By traversing several sites along a track sequentially, a user can easily understand what a person had experienced during a period of time. The intra-state evolution of an individual is encoded in a track with a series of sites. On the other hand, when two tracks cross at the same site, the crossing implies that the two people had participated in the same social event. When two tracks pass through several same sites sequentially, the merging implies that the two people had experienced several events together. When two tracks merge and then separate from a site, the splitting implies the two people gone forward to participate different activities individually. Therefore, the relationship evolution among the individuals can be effectively encoded in the railway route map form.

Compared with the traditional map metaphor, both of them leverage people's spatio-cognitive ability to help them navigate through information space representing in cartographic form [25]. However, the general map metaphor is usually used to reveal the quantitative or qualitative information as a static map form in a time slice like choropleth map, or as trajectory visualization by visualizing the activities' trajectories over time. Most general map metaphors are limited to reveal the branch characteristic of temporal data and the trajectories of different entities at the same time. Hence, the general map metaphor is hard to effectively represent the spitting, merging, and parallel relations among the trajectories of different entities in a single view.

Besides, a railway route map also provides an easily understanding view because of the grid layout characteristic [32]. Consequently, by leveraging the natural characteristics of the railway route map, we can visualize the information evolution of different entities with temporal relation.

## 3.2 Visual Encoding

"*Visual encoding is the mapping of information to display elements* [17]." In order to amplify users' cognition of information evolution, we designed the visual encoding that maps the features of information evolution into users' mental model of dairy railway route maps. There are three basic graphical components on the railway route map: track, site, and background.

**Track.** Traditional relationship visualization commonly encodes an individual entity, like a person, as a node, and represents the relationship between two entities as a linked edge between the corresponding two nodes. Unfortunately, it is not suitable for representing the temporal relationship among a sequence of nodes. Hence, we inverted the graph encoding and represented an individual data entity as a track in the railway route map. To differentiate different tracks, we built our color scheme to maximize the difference of hue between sampled neighboring colors.

**Site.** To represent the changes of the intra-state and relation evolution between the entities at the same time, events are encoded as the sites. Therefore, when a track passes by a site, it implies that an entity had experienced an event. Thereby, the changes of a specific entity's states can be traced by traversing the sites along a specific track. Besides, the semantic meanings of an event can be encoded with the site's preattentive features, like color, size and shape [27], and the design of the preattentive features depends on the properties of the applications. For example, the size of a site can be proportional to the duration of an event. To effectively utilize human's perception, the nodes are leveled as circle with three sizes, that users can easily discriminate the semantic meanings at one glance [18]. To represent an exchange site, where several tracks cross at, we used a concentric circle. Consequently, the meaning of the intra states, major events and co-experiences can be embedded into the site nodes at the same time in the graph. Besides, because *Railway Visualization* encodes a node as an event and a track as an individual entity, an **edge**, which is a segment belonging to a track, becomes a representation of temporal relationship between two linked events. Such edge encoding can reveal the sequential order of a series of relevant events.

**Background.** Traditionally, the background of the railway route map is usually encoded with the cartographic information in GIS. Inspired by the contour map in Cartography, we used the contour lines to mark the metadata or a specific attribute. For example, we can mark the global time interval by using contour lines. Through this way, people can get insight of the additional information at one glace through the background color. Hence, the background information can be used as a filter tool.

## 3.3 Aesthetic Rules

Using the railway route map metaphor in visualization does not only have the mentioned utility benefits, but also have the benefit of attractive layout [25]. Therefore, considering both the layout features of the railway route maps[3][4][5] in daily life and the significant aesthetic rules of graph drawing [22], we identified the aesthetic rules for the railway layout as follows:

- Straight lines: Most tracks in the railway route map are drawn as straight lines. Straight lines cannot only allow people easily discriminating different tracks, but also improve the readability through the grid layout [32]. Therefore, the line bending for each track should be as less as possible.

- Preferred angles: After crossing a site, the tracks are often drawn with the angles with multiples of 45 de-

[3]http://en.wikipedia.org/wiki/File:TubeMapZ1_TFL.png
[4]http://www.tokyometro.jp/global/en/service/pdf/routemap_en.pdf
[5]http://www.treehugger.com/galleries/2009/07/worlds-most-impressive-subway-maps.php?page=8

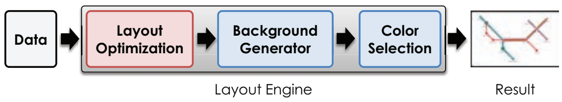


**Figure 1: The system architecture of *Railway Visualization***

grees. Therefore, the orientation of a track edge should be horizontal, vertical or diagonal.

- Avoiding overlapping: Overlapping causes visual ambiguity, and can make the graph hard to be interpreted. Hence, the overlapping among the graphical components should be reduced.
- Keeping in the center: To keep the visual balance of the railway map layout, the center of gravity of the route map should be aligned with the center of the display.

# 4. IMPLEMENTATION

The overview of our system is shown in Fig. 1. The input of our system is the raw data embedded the evolution with temporal relation among the data entities. At first, our system automatically generates the layout while considering the aesthetic rules mentioned in Sec. 3. The layout problem is formulated as an optimization problem which minimizes a set of energy terms modeling the aesthetic rules. Secondly, the metadata contour map is generated by a diffusion algorithm based on a specific data attribute. Then the suitable colors are generated by the concept inspired by Harrower *et al.*'s work [9] for representing the different tracks. In the following, we will focus on the layout algorithm as well as the optimization scheme.

## 4.1 Railway Graph

Traditional graph definition denotes a graph as: $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ and $\mathbf{E}$ describe the node and edge sets of the graph. The edge is usually used to encode the linking relation between two nodes. However, it cannot represent the relation among a series of nodes, such as the track relation in our *Railway Visualization*. Therefore, we define a railway graph as a node-link diagram $\mathbf{R} = \{\mathbf{V}, \mathbf{E}, \mathbf{T}\}$, where $\mathbf{V} = \{v_1, v_2, ..., v_n\}$ denotes the positions of the sites (nodes) in $\mathbb{R}^2$, $\mathbf{T} = \{t_1, t_2, ..., t_m\}$ denotes the tracks, and $\mathbf{E} = \left\{e^k_{i,j} | i, j \in \mathbf{V}, k \in \mathbf{T}\right\}$ represents the directional edges on a track $k$ for connecting two sites from $i$ to $j$.

## 4.2 Formulation

Given a railway layout problem, we formulate it as an optimization problem which minimizes a set of energy terms penalized for violating the aesthetic rules to determine the positions of the sites in $\mathbf{V}$, which is formulated as:

$$E(\mathbf{V}) = \sum_i \omega_i E_i(\mathbf{V}), \qquad (1)$$

where $E_i$ is the penalized cost defined as the following terms, and $\omega_i$ is its corresponding weight.

**Line Bending.** Because bending a track will reduce the readability, the line bending should be penalized. The penalty for line bending is defined as:

$$E_{Bending} = \sum_{t \in \mathbf{T}} \sum_{v^t \in \mathbf{V}} (1 - \cos(\angle v^t)), \qquad (2)$$

where $\angle v^t$ is the included angle of site $v$ along the track $t$, which can be calculated by the inner product of its two adjacent directional edges as: $\arccos(e^t_{v-1,v} \cdot e^t_{v,v+1})$.

**Edge Orientation.** In order to improve the readability, the orientation for each edge should be horizontal, vertical or diagonal. Therefore, the edges $e \in \mathbf{E}$ are compared with a unit horizontal vector $u$ which is parallel with the x-axis, and the orientation energy is defined as:

$$E_{Orientation} = \sum_{e \in \mathbf{E}} \min_{\theta \in \Theta} (\arccos(u \cdot e) - \theta)/\alpha, \qquad (3)$$

where $\Theta = \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi\}$ is used for penalizing the arbitrary unaligned edges, and $\alpha = \frac{\pi}{8}$ is used for normalization.

**Overlapping.** Because overlapping causes visual ambiguity, the overlapping among the edges and sites should be prevented. The energy penalized for overlapping is defined as:

$$E_{Overlapping} = \sum_{i,j \in \mathbf{V}} (1 - \frac{d_{i,j}}{D_{i,j}}) + \sum_{i \in \mathbf{V}, j \in \mathbf{E}} (1 - \frac{d_{i,j}}{D_{i,j}}) + \sum_{i,j \in \mathbf{E}} (1 - \frac{d_{i,j}}{D_{i,j}}), \qquad (4)$$

where $d_{i,j}$ is the shortest distance between arbitrary two components $i$ and $j$, which can be either an edge or a site, and $D_{i,j}$ is the minimal required margin of the two components $i$ and $j$ for avoiding the overlapping and is proportion to the size of the site(s).

**Alignment.** In order to keeping the visual balance, we place center of gravity of whole graph near the centric area in the display region as much as possible. Suppose $d$ measures the distance between the center of gravity and the display center, and $D$ is the half diagonal length of the display region. The alignment penalty is defined as:

$$E_{Alignment} = \begin{cases} 0 & \text{if } d < \varepsilon_{Alignment} \\ \frac{d - \varepsilon_{Alignment}}{D} & \text{otherwise} \end{cases}, \qquad (5)$$

where $\varepsilon_{Alignment}$ is a small constant.

## 4.3 Optimization.

Given a layout problem with a railway graph $\mathbf{R} = \{\mathbf{V}, \mathbf{E}, \mathbf{T}\}$, we aim to find the suitable positions for all sites by an optimization approach in order to generate the railway layout. Here, a two-steps optimization method is used. We first perform a simple data abstraction for the railway graph, and then generate the approximated layout result by a discrete optimization method. Then, the layout result is refined in the continuous domain.

**Approximation in discrete domain.** It is hard to find the global optimization solution in the continuous domain. In order to find a solution to approximate the global optimization solution, we adopt the branch and bound (BB) algorithm. The whole display region is first segmented into several uniform grids to downscale the search space. Our goal is to find a solution, which places all sites in the railway graph to the grid nodes with minimal cost, through the partial graph reconstruction. Hence, we define a processing order for all sites in the railway graph based on the sites' degrees. The site's degree is defined as the traditional node degree in the graph representation. Considering a site with more degree would affect the cost more, the processing order begins with the site with maximal degree in the

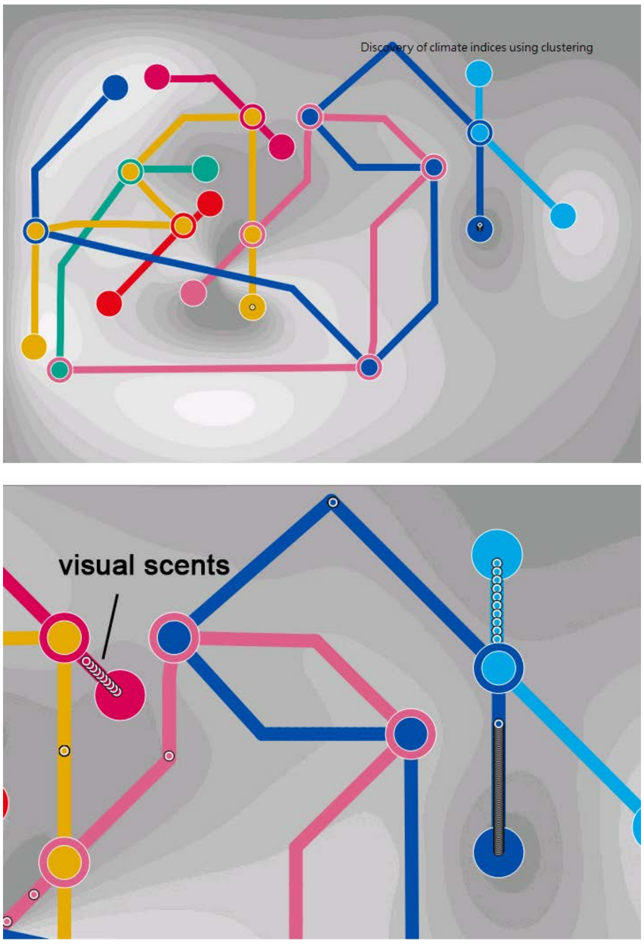


**Figure 2: While zooming into the graph, the detailed will be shown according to the calculated DOI scores, and on the contrary, while zooming out the picture, the whole graph will be shown and the elements of lower DOI scores will be hidden.**

railway graph. A BB tree is constructed for approximating the minimal lower bound in the partial graph reconstruction. In each step of the partial graph reconstruction, we find a branch of the minimal lower bound and then extend this branch by adding a new site at different position. Finally, after traversing possible solution for all sites, we can find a solution, which approximates the minimal cost, in the discrete domain.

**Refinement in continuous domain.** Based on the approximated layout in the previous step, our system further refines the local layout by the optimization in the continuous domain. In this step, we adopt the Davidon Fletcher-Powell method, which is a variation of the gradient decent approach. In each optimization iteration, the Davidon Fletcher-Powell method seeks the local minimum based along with the gradient direction. Through such way, we can refine the layout by seeking the local minimization.

## 5. INTERACTION

In order to help people explore the relationship among data, based on the seven basic operations [23], we also designed some interaction mechanisms for *Railway Visualization*.

**Design for scale.** Considering the *Power Law* characteristics of the relation data, to effectively use the limited display, a suitable data abstraction method is necessary for the input raw data. Rather than using the clustering-approach by gathering and merging the related nodes, a graph simplification method is proposed based on the importance score for each data component. While simplifying the graph, the graph topology should be maintained; otherwise, the semantic meaning of the relationship will be changed while the topology is altered.

Firstly, *A Prior Importance* (API) score for each component is calculated, by the given metadata or the basic properties derived from the graph. Therefore, a site's API score can be calculated by the weighted sum of the metadata and node degree in the graph. A track's API score can be represented as the sum of the sites' API scores along the track. For example, an actor or actress is represented as a track and a movie is represented as a site in *Railway Visualization*. Then, a movie's API score is measured by the evaluation score from the audience (metadata) as well as the number of actors and actresses participated in (node degree). Hence, the API score of an actor or actress depends on the movies that they participated. Then after simplifying, those sites with lower API scores are abstracted as the small visual scents, and the tracks with lower API scores are desaturated to maximize the usage of display in a clear view, but without altering the semantic topology of the original graph. If there are too many components on the screen, the elements with lower API scores will be hidden.

**Level of detail.** In order to provide a whole picture as well as the detail information based on a user's interest, the concept of Degree of Interest (DOI) [8] is used. Like van Ham and Perer [28] applied the DOI concept to exploring a large graph, we modified the original DOI function in order to navigate the graph embedded time-varying data and relation evolution.

By considering an observation that the longer duration between two time-varying data has, the less relation between them is. The concept of attenuation is modeled in the DOI function as:

$$DOI(x|y) = API(x) * e^{\frac{D(x,y)}{\alpha}}, \quad (6)$$

where $API(x)$ is the data term which measures the mentioned prior importance score for component $x$, $D(x,y)$ is the distance function in the temporal domain for the component $x$ and the user's focused one $y$, $\alpha$ is the normalized factor, and $e^{\frac{D(x,y)}{\alpha}}$ is the attenuation function modeled by the duration between $x$ and $y$. Therefore, we can effectively provide the most relevant and important information by considering both of temporal relationship and correlations depending on a user's focus.

Hence, with the updated DOI score and different zoom-in/out levels, the revealed information is varied according to the user's interest. Through this way, we do not only provide a context + focus mechanism for graph navigation, but also use intuitive visual scents to direct users' attention to hot spot or to unexplored region.

## 6. APPLICATIONS

In this section, we show some of the *Railway Visualization*

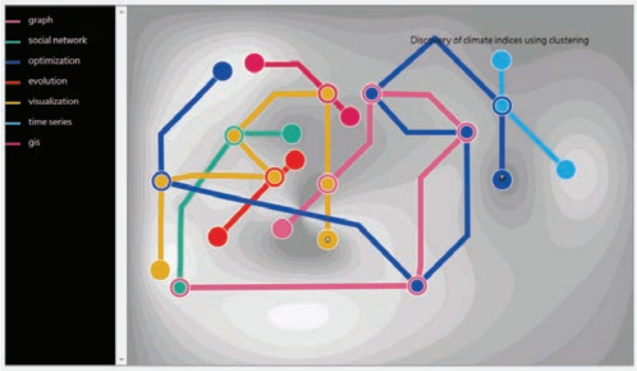


**Figure 3: The result of visualizing the relation among selected keywords of paper data.**

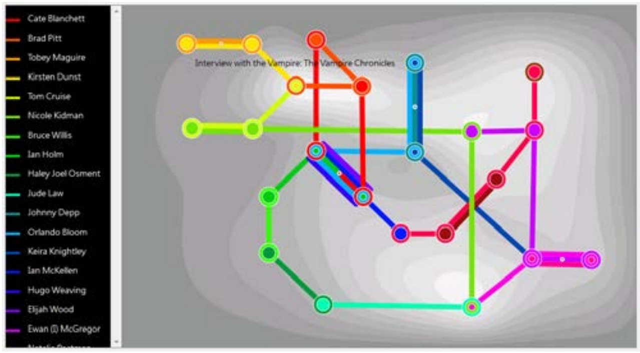


**Figure 4: The result of visualizing the co-star relation among the actors or actresses.**

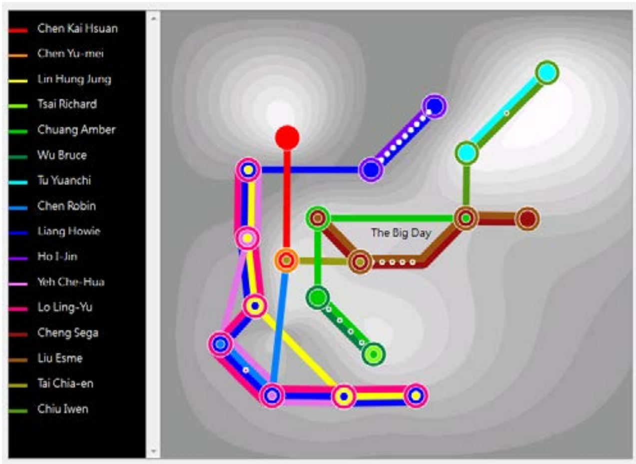


**Figure 5: The result of visualizing people's relation through web albums.**

results with four kinds of application data, which are papers, movies, photo albums and news.

**Paper.** We parsed the paper data from *ACM portal*[6] and then portrayed the evolution of research topics by our system as shown in Fig. 3. Through this way, users can survey a specific research topic by traversing along a track. The papers are selected by using the following keywords: graph, Social Network, optimization, evolution, visualization, time series, and GIS.

- Track: A keyword is encoded as a track.
- Site: A paper is represented as a site. Since we visualize the evolution of the research topics, an edge links the relevant successive work in the specific research topic.
- Background: The publish time of the papers.
- Data scale: We parsed 1084 papers from *ACM portal*. After data abstraction, we represented 37 papers as the sites and 1047 papers as the visual scents (abstract level: 1047(96.3 %)).

**Movie.** We parsed the movie data from the *IMDB*[7]. Using our *Railway Visualization*, we can represent the co-star relation among the actors or actresses over time as shown in Fig. 4. The components are represented as follows.

- Track: An actor or actress is encoded as a track.
- Site: A movie is represented as a site. An edge links the sequential works of an actor or actress in chronological order.
- Background: The contour map encodes the show time of the movies.

**Facebook.** We adopted *Facebook* public API to parse personal photo albums in *Facebook*[8]. Base on the tags of People's name in *Facebook*, we visualized the social activities of a person whom we chose arbitrarily, and the result is shown in Fig. 5. Therefore, given a user, we used *Railway Visualization* to represent the interaction among his or her social network over time. The components are represented as follows.

- Track: A person in the social network is encoded as a track.
- Site: A photo album, which belongs to a person in the social network, is represented as a site. An edge links two related photo albums.
- Background: The albums' upload time is encoded in the the contour map.
- Data scale: We parsed 59 photo albums from *Facebook*. After data abstraction, we represented 33 photo albums as the sites and 26 photo albums as the visual scents (abstract level: 26(44.1 %)).

**News.** People usually are interested in certain themes and wonder to know the relevant news, and how the topic is going on. Therefore, we parsed the techniques news on *Engadget*[9] to demo the usage, and the visuliazation result is shown in Fig. 6.

[6]http://portal.acm.org/
[7]http://www.imdb.com/
[8]http://www.facebook.com/
[9]http://www.engadget.com/

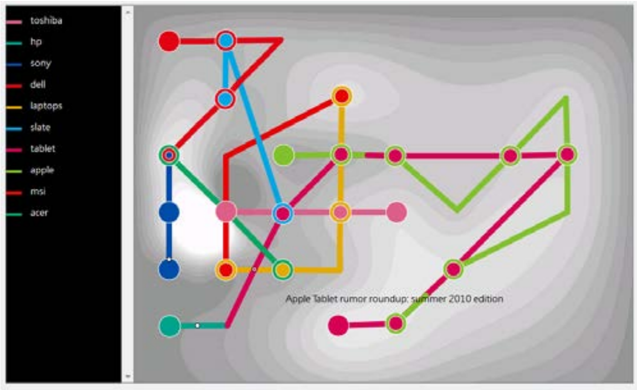


**Figure 6: The result of visualizing news by using the selected tags.**

- Track: A keyword is encoded as a track.
- Site: A news article is represented as a site. An edge represents relevant successive relation for a specific topic.
- Background: The background contour map is encoded with the post time of the articles.

## 7. DISCUSSION

Compared with general graph visualizations, the strength and limitation of the *Railway Visualization* are addressed at various aspects as shown in Table 1. Generally, the graph visualization uses a node-link diagram, which encodes an entity as a node and an edge as the relationship between two entities. We compared our approach with two kinds of graph visualization: static graph and dynamic graph. The former one portrays the whole graph in a static time slice, and the later one represents the dynamic graphic evolution by animation.

**Relation.** Yet general graph visualization can represent the relationship between two nodes, it cannot portray the relation among a series of nodes. By inverting the general graph encoding, *Railway Visualization* represents an individual entity as a track, and represents an experienced event as a site in a railway route map. Through this way, *Railway Visualization* can represent the relation among several sequential nodes through by a passed link.

**Evolution.** Because the visual encoding of *Railway Visualization* has the advantages of network as well as timeline, the relationship of any two entities can be represented as graph visualization can do. Moreover, *Railway Visualization* can render the evolution of relationship in one snapshot to provide a global view. Furthermore, tracing a series of edges along a track is like tracing the events on the timeline. People can easily know the serial relation of the relevant events as well as an entity's intra-state evolution by the passed serial events.

**Clustering.** Due to the *Power Law* of the relation data, the characteristics of clustering and celebrity are commonly seen in the general graph visualization. Because the semantic distance is encoded into the edge length, the general graph visualization can cluster the nodes with similar attributes in adjacent spatial regions. Instead, *Railway Visualization* uses the node degrees to represent the concept of clustering. When a node has higher degrees, it means that there are many tracks intersected at the node. Then, while tracks are encoded as the entities, and nodes are encoded as the events, we can interpret the intersection as the entities are clustered at that event.

Compared with the traditional graph visualizations, the inverted graph encoding of the *Railway Visualization* cannot only visualize the relationship between the entities, but also can visualize the complicated evolution of the relationship, like branching, crossing, splitting, converging and parallelism. Generally, *Railway Visualization* provides more information than traditional graph visualizations.

| | Static Graph | Dynamic Graph | Railway Visualization |
|---|---|---|---|
| Relation of two Nodes | ○ | ○ | ○ |
| Serial Relation | × | ○ | ◎ |
| Relationship Evolution | × | △ | ◎ |
| Intra-state Evolution | × | × | ◎ |
| Clustering | ○ | ○ | ○ |

**Table 1: The Comparison of Graph and Railway Visualizations.**

## 8. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel visualization technique, *Railway Visualization*, to reveal the information evolution with temporal relationship. By leveraging people's mental models of the railway route map in our daily life, our method cannot only portray the changes of an entity's intra-state over time, but also can effectively represent the characteristics of the complicated relationship evolution, such as branching, crossing, splitting, converging and parallelism. Besides, the atheistic rules for showing the railway layout are defined and formulated as an energy minimization problem, which is solved by a two-steps optimization scheme. In addition, we also provide an interactive level-of-detail mechanism while considering the importance score of the data. Furthermore, we demonstrate the visualization results with several kinds of data such as papers, movies, photo albums and news.

All of applications' data demonstrated here focus on reveal the features of *Railway Visualization*, hence they are processed by some simple abstraction mechanisms. However, the result of *Railway Visualization* can be better if we preprocess the raw data with a powerful information retrieval method. Beyond the data scale concern, we are also interesting in how to drawing a railway layout more effectively with the global minimal cost. Besides, we are interesting in exploring other applications by *Railway Visualization* in real world. Nevertheless, our *Railway Visualization* provides a clear and effective representation with dedicated interaction mechanism for exploring the information evolution.

## 9. REFERENCES

[1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on*

*Visualization and Computer Graphics*, 14(1):47–60, 2008.

[2] L. Byron and M. Wattenberg. Stacked graphs – geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, 2008.

[3] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., 1999.

[4] C. Collberg, S. Kobourov, J. Nagra, J. Pitts, and K. Wampler. A system for graph-based visualization of the evolution of software. In *ACM Software Visualization 2003 Conference Proceedings*, pages 77–86, 2003.

[5] M. J. Egenhofer and R. G. Golledge, editors. *Spatial and Temporal Reasoning in Geographic Information Systems*. Oxford University Press, 1998.

[6] S. G. Eick, T. L. Graves, A. F. Karr, A. Mockus, and P. Schuster. Visualizing software changes. *IEEE Transactions on Software Engineering*, 28(4):396–412, 2002.

[7] L. C. Freeman. Visualizing social networks. *Journal of Social Structure*, 1(1), 2000.

[8] G. W. Furnas. Generalized fisheye views. *ACM SIGCHI Bulletin*, 17(4):16–23, 1986.

[9] M. Harrower and C. A. Brewer. Colorbrewer.org: An online tool for selecting colour schemes for maps. *Cartographic Journal*, 40(1):27–37, 2003.

[10] S. Havre, B. Hetzler, and L. Nowell. Themeriver: Visualizing theme changes over time. In *IEEE Information Vizualization 2000 Conference Proceedings*, pages 115–123, 2000.

[11] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[12] M. Jensen. Visualizing complex semantic timelines. Technical Report NBTR2003-001, NewsBlip, 2003.

[13] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):805–812, 2006.

[14] V. Kumar, R. Furuta, and R. B. Allen. Metadata visualization for digital libraries: interactive timeline editing and review. In *ACM Digital Libraries 1998 Conference Proceedings*, pages 126–133, 1998.

[15] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *World Wide Web 2008 Conference Proceedings*, pages 685–694, 2008.

[16] W. Müller and H. Schumann. visualization methods for time-dependent data - an overview. In *Proceedings of 2003 Conference on Winter Simulation*, pages 737–745, 2003.

[17] T. M. Munzner. *Interactive visualization of large graphs and networks*. PhD thesis, Stanford University, 2000.

[18] L. Nowell, R. Schulman, and D. Hix. Graphical encoding for information visualization: An empirical study. In *IEEE Information Visualization 2002 Conference Proceedings*, pages 43–50, 2002.

[19] M. Ogawa and K.-L. Ma. code swarm: A design study in organic software visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1097–1104, 2009.

[20] M. Pinzger, H. Gall, M. Fischer, and M. Lanza. Visualizing multiple evolution metrics. In *ACM Software Visualization 2005 Conference Proceedings*, pages 67–75, 2005.

[21] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: visualizing personal histories. In *ACM CHI 1996 Confernece Proceedings*, pages 221–227, 1996.

[22] H. C. Purchase. Which aesthetic has the greatest effect on human understanding? In *Graph Drawing 1997 Conference Proceedings*, pages 248–261, 1997.

[23] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Visual Languages 1996 Conference Proceedings*, pages 336–343, 1996.

[24] S. F. Silva and T. Catarci. Visualization of linear time-oriented data: A survey. In *Proceedings of 2000 International Conference on Web Information Systems Engineering*, pages 310–319, 2000.

[25] A. Skupin. From metaphor to method: Cartographic perspectives on information visualization. In *IEEE Information Visualization 2000 Conference Proceedings*, pages 91–97, 2000.

[26] M. Toyoda and M. Kitsuregawa. A system for visualizing and analyzing the evolution of the web with a time series of graphs. In *ACM Hypertext and Hypermedia 2005 Conference Proceedings*, pages 151–160, 2005.

[27] A. Treisman. Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31(2):156–177, 1985.

[28] F. van Ham and A. Perer. "search, show context, expand on demand": Supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):953–960, 2009.

[29] F. B. Viégas, D. Boyd, D. H. Nguyen, J. Potter, and J. Donath. Digital artifacts for remembering and storytelling: Posthistory and social network fragments. In *Proceedings of 2004 Annual Hawaii International Conference on System Sciences - Track 4*, page 40109a, 2004.

[30] F. B. Viégas, S. Golder, and J. Donath. Visualizing email content: portraying relationships from conversational histories. In *ACM CHI 2006 Conference Proceedings*, pages 979–988, 2006.

[31] M. Wattenberg. Baby names, visualization, and social data analysis. In *IEEE Information Vizualization 2005 Conference Proceedings*, pages 1–7, 2005.

[32] A. Zanella, M. S. T. Carpendale, and M. Rounding. On the effects of viewing cues in comprehending distortions. In *Proceedings of 2002 Nordic Conference on Human-Computer Interaction*, pages 119–128, 2002.