ISSN 1342-2812

Research Reports on Mathematical and Computing Sciences

A Projection-Based Method for Interactive Visual Exploration of Complex Graphs in A Three-Dimensional Space

Masanori Takami, Hiroshi Hosobe, Ken Wakita

March 2012, C–278

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

A Projection-Based Method for Interactive Visual Exploration of Complex Graphs in A Three-Dimensional Space

Masanori Takami*
Department of Information Science
Tokyo Institute of Technology

Hiroshi Hosobe†
National Institute of Informatics

Ken Wakita‡
Tokyo Institute of Technology/
Japan Science and Technology Agency

Abstract

Complex networks can be hard for us human beings to conceptually grasp. We propose a new graph visualization method, which projects the static high-dimensional layout of a complex graph onto a two-, three-, or higher-dimensional space and allows viewers to explore the graph by interactively altering the way the layout is projected. The viewer could visually "untangle" what initially appeared to be a tight knot — a simply impossible operation in conventional methods, with which you could just either zoom in or rotate. This is made possible by interpreting the user's action as a rotation command of the original graph in its original space and then re-projecting it onto the target space. This work extends Hosobe's previous work, whose target space could only be two dimensional, not three dimensional. The theoretical foundation of our technique and the lessons acquired from our early prototype are presented.

1 Introduction

Complex networks can be hard for us human beings to conceptually grasp. There exist visualization techniques to address this issue. They commonly project the original graph onto a two-dimensional or three-dimensional space, and use affine transformation to offer zooming and rotation facilities in that target space. In these techniques, the skeleton structure of the resulting graph layout is static and cannot be modified by the viewer.

We propose an alternative method, which projects the static high-dimensional layout of a complex graph onto a two-, three-, or higher-dimensional space and allows viewers to explore the graph by interactively altering the way the layout is

*e-mail: takami.m.aa@m.titech.ac.jp
†e-mail: hosobe@nii.ac.jp
‡e-mail: wakita@is.titech.ac.jp

projected. When the manner of projection is changed, sometimes you get unique visual effects. For example, the viewer could "untangle" what initially appeared to be a tight knot. It is simply impossible in conventional techniques, with which you can just either zoom in or rotate.

Our proposal is based on Hosobe's previous work [1]. His work was unique in that the viewer could get a different view of the projected graph by rotating the projection plane, which in turn was triggered by him dragging a graph node on the screen. High-dimensional rotation gives much more flexibility in the choice of easy-to-understand viewpoints than standard two-dimensional rotation and zooming.

However, Hosobe's work was limited in that the target space can only be two-dimensional. We will extend his work and remove this limitation in this paper. For this, instead of rotating the projection plane, we will rotate the whole "projection space". Our method occasionally yields visual untangling effects similar to those by Hosobe's previous method.

2 Foundation

This section briefly explains how a high-dimensional layout of the graph is projected onto a lower-dimensional space in our method and how the user interacts with it.

2.1 High-Dimensional Graph Layout and Its Projection onto Lower-Dimensional Space

Given an undirected graph, we first build a *distance matrix* that corresponds to graph-theoretic distances for all the pairs of the nodes in the graph. Then a *classical multidimensional scaling* called *Torgerson scaling* is applied to the distance matrix to assign *static high-dimensional locations* to the graph nodes [2, 3]. During this process, the *error matrix* (E) of the distance matrix is eigendecomposed as $X^T E X = diag(\lambda_1, \lambda_2, \ldots)$, where λ_i's are E's eigenvalues, which are sorted in a descending order, and correspond to eigenvectors, $(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots) = X$.

The proposed method projects such computed high-dimensional graph layouts onto two-, three-, or higher-dimensional spaces, while in [1] high-dimensional graph layouts are projected onto two-dimensional spaces only. This extended ability allows for the interactive technique to be used in three- and higher-dimensional graph visualization.

A high-dimensional graph layout is projected onto a lower-dimensional space by a $(d_H \times d)$*-projection matrix*, $P = (\boldsymbol{e_1}, \boldsymbol{e_2}, \ldots)$, where $\boldsymbol{e_i} = \boldsymbol{f_i}/\|\boldsymbol{f_i}\|$, and d_H and d are the dimensions of the high- and lower-dimensional spaces, respectively. The projection matrix is obtained from the normalization of orthogonal bases $\boldsymbol{f_i}$, which are arranged by picking every d element from E's positive eigenvalues.

$$\boldsymbol{f_i} = (\overbrace{0, \ldots, 0}^{i-1}, \sqrt{\lambda_i}, \overbrace{0, \ldots, 0}^{d-1}, \sqrt{\lambda_{d+i}}, \overbrace{0, \ldots, 0}^{d-1}, \ldots)^T$$

Let $\boldsymbol{p}$ be a high-dimensional location of a graph node. We can obtain its lower-dimensional position, $\boldsymbol{p_d}$, by multiplication of $\boldsymbol{p}$ and P, i.e., $\boldsymbol{p_d} = \boldsymbol{p}P$.

The projection described in [1] can be regarded as a special case of our extended formulation, where $d = 2$.

2.2 Interaction Framework

We have seen that the projection matrix, P, gives a projection of the high-dimensional space to the lower-dimensional one. Hosobe's interaction technique is based on an insight that a slightly modified projection matrix, P', can reposition the user's viewpoint *in the high-dimensional space*. Standard graph manipulation techniques based on affine geometry allow the user to scale, rotate, and transform the graph layout but those operations are restricted in two- or three-dimensional spaces and they fail to make use of the rest of the dimensions that are available in the high-dimensional graph layout. In contrast, the proposed system interprets user's drag operation as a command for rotation in the high-dimensional space. Intuitively, this gives visual effects similar to rotating a three dimensional object and viewing the other side of the object, which could not be seen in the initial layout. Note, however, that this rotation is performed in the high-dimensional space and the rotation axis is automatically determined from the user's simple drag operation.

In the proposed interaction scheme, the only operation required for the user is to drag a graph node. Suppose that a user grabs a graph node placed at $\boldsymbol{q_d} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and tries to reposition it to another position, $\boldsymbol{q'_d} = (x'_1, \ldots, x'_d) \in \mathbb{R}^d$. We consider that both of these positions are projected from an identical point, $\boldsymbol{q} \in \mathbb{R}^{d_H}$, in the high dimensional space, or more formally, $\boldsymbol{q_d} = \boldsymbol{q}P$ and $\boldsymbol{q'_d} = \boldsymbol{q}P'$. We also assume that such a change in the projection occurred by a rotation performed in the high-dimensional space.

In the following, we will explain how a rotation that translates a projection, P, to another, P', can be found. In other words, since a projection matrix consists of normal orthogonal bases, $\boldsymbol{e}_i$'s, the problem are to find another normal orthogonal bases that are obtained by a rotation of the former bases and also project $\boldsymbol{q}$ onto $\boldsymbol{q'_d}$.

In the following formulae, the graph presentation that the user is observing is represented by a $(d_H \times d)$-projection matrix $P = (\boldsymbol{e_1}, \boldsymbol{e_2}, \ldots)$, which consists of normal orthogonal bases, $\boldsymbol{e_i}$'s. When the user's drag operation is performed, the system finds a high-dimensional rotation that translates P to another projection matrix $P' = (\boldsymbol{e'_1}, \boldsymbol{e'_2}, \ldots)$, which also consists of normal orthogonal bases, $\boldsymbol{e'_i}$'s. The following is a mathematical account of this explanation.

$$\boldsymbol{e_0} = \boldsymbol{q} - \sum_{i=1}^{d} x_i \boldsymbol{e_i} \tag{1}$$

$$\boldsymbol{e'_i} = \sum_{j=0}^{d} a_{ij} \boldsymbol{e_j} \quad (1 \le i \le d) \tag{2}$$

$$\boldsymbol{r_i} = \sum_{j=1}^{d} b_{ij} \boldsymbol{e_j} \quad (1 \le i \le d-1) \tag{3}$$

$$\boldsymbol{q} \cdot \boldsymbol{e'_i} = x'_i \quad (1 \le i \le d) \tag{4}$$

$$\boldsymbol{e'_i} \cdot \boldsymbol{e'_j} = \delta(i,j) \quad (1 \le i, j \le d) \tag{5}$$

$$\boldsymbol{r_i} \cdot \boldsymbol{r_j} = \delta(i,j) \quad (1 \le i, j \le d-1) \tag{6}$$

$$\boldsymbol{r_i} \cdot \boldsymbol{e'_j} = \boldsymbol{r_i} \cdot \boldsymbol{e_j} \quad (1 \le i \le d-1, 1 \le j \le d) \tag{7}$$

Equations 1 and 2 characterize the space where rotation is performed, which is formed from the original bases, $\boldsymbol{e_i}$'s $(1 \le i \le d)$, as well as the node being dragged, $\boldsymbol{e_0}$, to increase the degree of freedom to enable rotation. The added vector, $\boldsymbol{e_0}$, is arranged such that it is normal and orthogonal to all the bases. The system tries to find a rotation axis, $\boldsymbol{r_i}$, on the plane spanned by the original bases (Equation 3). Equation 4 states that the projection matrix formed from bases $\boldsymbol{e_i'}$'s projects the graph node being dragged to the target of the drag operation. This requirement guarantees the projection gives consistent viewpoints with respect to user's designation. Equations 5 and 6 state that the modified projection bases and the rotation axis are normal orthogonal bases.[1] Finally, Equation 7 requires that the modification of the projection matrix be a rotation.

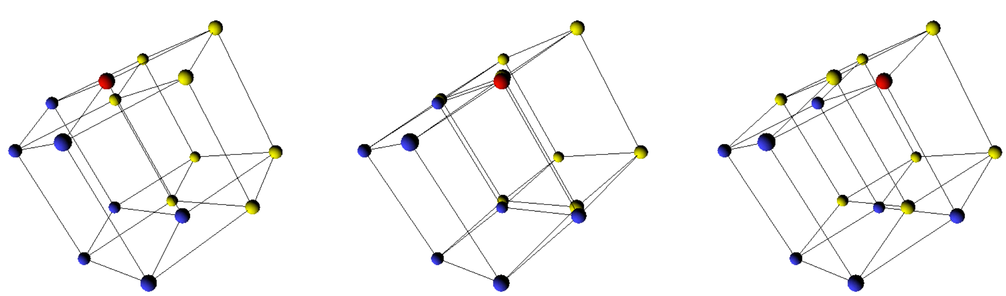
This series of equations contain $d(d+1)$ variables that correspond to a_{ij}, and $(d-1)d$ variables to b_{ij} and have equal number of equations. A standard numerical algorithm can therefore solve the equations.

3 Experiment

We have implemented a prototype system that presents graphs on the three-dimensional space and features the proposed interaction capability. Its GUI is implemented in Java with Java 3D. The equation solver is implemented in C using GNU Science Library that offers APIs for linear algebra and a solver for nonlinear equations.

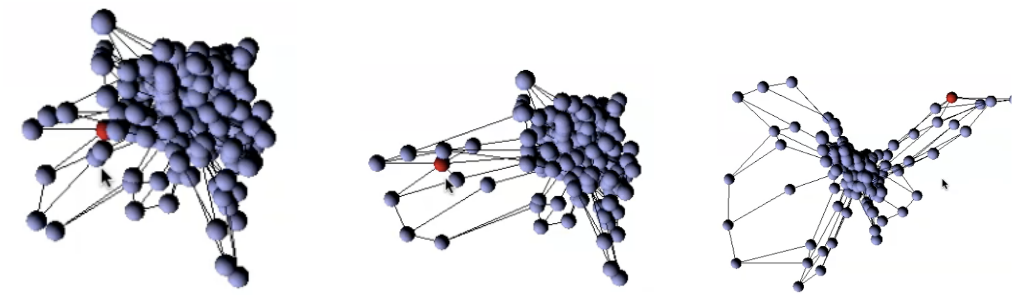
The prototype presents a three-dimensional visualization of the graph, whose high-dimensional layout is computed as explained in subsection 2.1. The user interacts with the graph layout by picking a graph node and repositions it in the three-dimensional space. The rotation in the high-dimensional space that projects the graph node being picked to the location where the user repositioned it, is computed as explained in subsection 2.2.

The next three images demonstrates interaction with a three-dimensional presentation of a four-dimensional hypercube structure. The red point represents the node being dragged to the right by the user. Other nodes are painted either yellow or blue to illustrate the sub-cube structures. Note that the leftmost plane of the blue sub-cube and the backmost plane of the yellow one remain almost stationary. On the other hand, the backmost plane of the blue sub-cube and the foremost plane of the yellow one exchange their positions. Unlike the standard two- and three-dimensional rotation techniques, our high-dimensional rotation technique allows us to change our viewpoint by a drag operation in such a way that the resulting changes will remain local and not distract our attention by global changes.



[1] δ denotes Kronecker's delta.

The following are a few snapshots from our interaction with an example from AT&T graph dataset `ug_380`, which contains 1,104 nodes and 3,231 edges. The leftmost image is the initial layout, which is obtained from a projection of multidimensional scaling, to three dimensional space. From this dense cluster a point is picked and dragged to the left. This action effectively untangles the nearby substructures and offers a better view of them. Similar dragging operations are repeated for tow more points to further untangle the structure and we obtain the rightmost image.



4 Concluding Remarks

The article has proposed a method for interactive three-dimensional graph visualization. The idea has been tested using an early prototype. The framework can theoretically be applied to four- and higher-dimensional visualization; such higher-dimensional visualization remains as future work.

Acknowledgements The authors would like to thank Dr. Makoto Yamashita for giving us insight into solving non-linear equations. The gratitude goes to members of our research group and the members of the graph CREST visualization team. This work is being supported in part by a grant from JST CREST.

References

[1] H. Hosobe. A high-dimensional approach to interactive graph visualization. In *SAC '04: Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 1253–1257, New York, NY, USA, 2004. ACM.

[2] J. B. Kruskal and J. B. Seery. Designing network diagrams. In *Proceedings of the First General Conference on Social Graphics*, pages 22–50, Washington, D.C., USA, July 1980.

[3] F. W. Young. Multidimensional scaling. In *Encyclopedia of Statisical Sciences*, volume 5, pages 649–658. Wiley, 1985.