Analysis of a High-Dimensional Approach to Interactive Graph Drawing

Hiroshi Hosobe*
National Institute of Informatics, Japan

ABSTRACT

Graph drawing is an information visualization technology for illustrating relations between objects. Interactive graph drawing is often important since it is difficult to statically lay out complex graphs. For the interactive drawing of general undirected graphs, we have proposed the high-dimensional approach, which uses static graph layouts in high-dimensional spaces to dynamically find two-dimensional layouts according to user interaction. Although the resulting interactive graph drawing method was found to be fast, other properties of it are not yet clear. In this paper, we analyze the high-dimensional approach to further explore its properties. Specifically, we perform the following two kinds of its analysis: (1) sensitivity analysis for investigating how the high-dimensional approach places graph nodes on the two-dimensional plane; (2) empirical analysis for examining the appropriateness of underlying graph layout methods. The results show that, as an underlying graph layout method, Kruskal and Seery's method based on Torgerson's multidimensional scaling method is more appropriate for the high-dimensional approach than other methods for computing graph layouts in high-dimensional spaces.

Keywords: Interactive graph drawing, general undirected graphs, high-dimensional approach.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1 INTRODUCTION

Information visualization is often needed to illustrate relations between objects. *Graphs* are formal means for expressing such relations; they represent objects as nodes and such relations as edges. To visualize information expressed as graphs, researchers have studied *graph drawing* [2], which automatically computes appropriate positions of nodes and edges. Graph drawing methods are designed according to classes of graphs that are determined by their structures. Examples of classes are trees, directed graphs, planar graphs, and *general undirected graphs*.

General undirected graphs, whose edges have no directions, are used to express various information with network structures. Although previous methods including the force-directed approach [2] have been successful to a certain degree, drawing complex general undirected graphs with more than hundreds of nodes is still a hard problem; visualizing the structure of such a graph with a single static layout is difficult because of its high generality. An effective means for this problem is *interactive graph drawing*, which allows users to visualize graphs interactively.

For this purpose, we have proposed the *high-dimensional approach* [5, 6], which uses static graph layouts in high-dimensional spaces to dynamically find two-dimensional layouts according to user interaction. To transform such high-dimensional layouts into two-dimensional ones, it projects them onto appropriate two-dimensional planes that it determines by constraint satisfaction. The resulting interactive graph drawing method was found to be fast; it efficiently updates two-dimensional graph layouts, and processes graphs with more than one thousand nodes within a few tens of milliseconds.

*e-mail: hosobe@nii.ac.jp

However, other properties of the resulting method are not yet clear. Our experiments showed that it follows users' node dragging operations by actively moving other closely related nodes, but it has been unclear why it exhibits such behavior.

In this paper, we analyze the high-dimensional approach to further explore its properties. Specifically, we perform the following two kinds of its analysis:

1. Sensitivity analysis for investigating how a high-dimensional method places graph nodes on the two-dimensional plane;

2. Empirical analysis for examining the appropriateness of underlying methods of high-dimensional graph layout.

The results show that, as an underlying graph layout method, Kruskal and Seery's method based on Torgerson's multidimensional scaling method is more appropriate for the high-dimensional approach than Hall's method [7] and Harel and Koren's method [4], both of which are applicable to computing graph layouts in high-dimensional spaces.

The rest of this paper is organized as follows. Section 2 describes related work. Section 3 explains the high-dimensional approach that we analyze. Section 4 provides the sensitivity analysis, and Section 5 presents the empirical analysis. Finally, Section 6 mentions the conclusions and future work of this research.

2 RELATED WORK

The force-directed approach [2] is often adopted to find layouts of general undirected graphs. The approach is applicable to drawing graphs of three or higher dimensions. GEM-3D [1] uses a randomized adaptive spring-embedder algorithm to obtain three-dimensional graph layouts. In [3], a method is provided that first finds graph layouts in multidimensional (e.g., four-dimensional) spaces by using the force-directed approach and then projects the layouts onto two- or three-dimensional spaces.

Methods using eigenvectors for graph layout are attracting attention. In [9], an example is presented that finds a graph layout in the football shape by adopting eigenvectors of Laplacian matrices. The ACE algorithm [7], which is based on a similar formulation called Hall's method, computes layouts of graphs with more than 10^6 nodes within a minute by using an algebraic multigrid algorithm to speed up eigenvector calculation.

In [4], a method is given that finds layouts of graphs with 10^5 nodes within a few seconds by first computing graph layouts of relatively high dimensions such as 50 and then by projecting them onto two-dimensional planes according to principal component analysis.

3 HIGH-DIMENSIONAL METHOD

This section explains the basic high-dimensional method for interactive graph drawing [5].

3.1 Multidimensional Graph Layout

The basic high-dimensional method uses Torgerson's method [8, 10] as its fundamental basis. Given distances between any pairs of objects, it finds a layout of them that satisfies the distances.

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/APVIS.2007.329281.

Torgerson's method is described below. Assume that we have distances d_{ij} between any pairs i and j of n objects, and also that they satisfy the distance axioms. First, define a_{ij} as follows:

$$a_{ij} = \frac{1}{2}\left(\frac{1}{n}\sum_{k=1}^{n} d_{ik}^2 + \frac{1}{n}\sum_{k=1}^{n} d_{kj}^2 - \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n} d_{kl}^2 - d_{ij}^2\right).$$

Next, define an $n \times n$ real symmetric matrix $A = (a_{ij})$. Then A is diagonalizable as $X^T AX = \Lambda$ for an orthogonal matrix X, where Λ is a diagonal matrix. With the eigenvalues λ_k of A and the eigenvectors $\boldsymbol{x}_k$ corresponding to λ_k, such X and Λ are obtained as follows: Λ has λ_k as its (k,k) elements, and $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)$.

Let $P = X\Lambda^{1/2}$, where $\Lambda^{1/2}$ is the diagonal matrix with $\sqrt{\lambda_k}$ as its (k,k) elements. For an ideal set of d_{ij}, each eigenvalue λ_k is nonnegative. Torgerson's method regards each i-th row $(p_{i1}, p_{i2}, \dots, p_{in})$ of P as the coordinates of the location of the object i in the n-dimensional real Euclidean space. It should be noted that actual data usually result in occurrences of negative eigenvalues. Ordinary applications of Torgerson's method use only the coordinates corresponding to the first and second largest eigenvalues.

Kruskal and Seery proposed a method that uses Torgerson's method to lay out connected general undirected graphs [8] (which we call the TKS method). It is realized as follows: given a graph, first compute the graph-theoretic distances (or the lengths of the shortest paths) between any pairs of its nodes; next, perform Torgerson's method by using the graph-theoretic distances, to obtain a layout of the nodes on a two-dimensional plane. Although they assumed two dimensions, the method is easily extensible to multi-dimensional graph layouts.

3.2 Interactive Graph Drawing

The basic high-dimensional method [5] computes two-dimensional graph layouts by projecting graph layouts in high-dimensional spaces onto two-dimensional planes. It handles connected general undirected graphs, and represents edges as straight lines.

Adopting the TKS method described in the previous subsection, the basic high-dimensional method computes graph layouts in high-dimensional spaces. It uses all the coordinates corresponding to positive eigenvalues. Generally, since the TKS method exploits graph-theoretic distances in Torgerson's method, it obtains many positive eigenvalues, which means that the dimensionalities of the resulting graph layouts are high. Assume that the eigenvalues λ_1, λ_2, ... are sorted in descending order, and also $d \geq 2$, where d is the number of positive eigenvalues; that is, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d > 0$. Then the position of each node i in the high-dimensional space is $\boldsymbol{p}_i = (p_{i1}, p_{i2}, \dots, p_{id})$.

The method projects such a d-dimensional graph layout onto a two-dimensional plane (called the projection plane) as follows. Consider the projection plane as the plane spanned by two orthonormal d-dimensional vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Using these vectors, the two-dimensional coordinates of node i are obtained as $(\boldsymbol{p}_i \cdot \boldsymbol{e}_1, \boldsymbol{p}_i \cdot \boldsymbol{e}_2)$.

For the initial two-dimensional layout, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are initialized by letting $\boldsymbol{e}_1 = \boldsymbol{f}_1/\|\boldsymbol{f}_1\|$ and $\boldsymbol{e}_2 = \boldsymbol{f}_2/\|\boldsymbol{f}_2\|$, where $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are the d-dimensional vectors defined as $\boldsymbol{f}_1 = \left(\lambda_1^\delta, 0, \lambda_3^\delta, 0, \dots\right)$ and $\boldsymbol{f}_2 = \left(0, \lambda_2^\delta, 0, \lambda_4^\delta, \dots\right)$. Here δ is a parameter, typically set to $1/2$, to adjust how the coordinates affect the two-dimensional layout. Note that $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are orthonormal.

The method enables users to interactively update two-dimensional graph layouts by dragging a single node at a time. It is realized by moving projection planes. Since it is not necessary to modify graph layouts in high-dimensional spaces, the method provides high efficiency in updating two-dimensional layouts.

The basic idea of the method is that it rotates the projection plane in the three-dimensional space spanned by the current vectors for the projection plane and the vector positioning the dragged node. To compute this, it performs constraint satisfaction by imposing the constrains that should be satisfied by the vectors spanning the projection plane.

It is described in detail below. First, constants that work as input are defined. Let $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ be the current vectors spanning the projection plane, which are orthonormal. Let i be the index of the dragged node, and (x_i, y_i) and (x_i', y_i') be its current and new two-dimensional coordinates respectively. Then we have $(x_i, y_i) = (\boldsymbol{p}_i \cdot \boldsymbol{e}_1, \boldsymbol{p}_i \cdot \boldsymbol{e}_2)$ by definition. Also, assume $\|(x_i, y_i)\| < \|\boldsymbol{p}_i\|$ and $\|(x_i', y_i')\| < \|\boldsymbol{p}_i\|$ (this assumption is not restrictive; see [5]). Also, let $\boldsymbol{e}_3$ be the vector obtained by the orthonormalization of $\boldsymbol{p}_i$ using $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, which is determined as $\boldsymbol{e}_3 = \boldsymbol{f}_3/\|\boldsymbol{f}_3\|$, where $\boldsymbol{f}_3$ is the d-dimensional vector defined as $\boldsymbol{f}_3 = \boldsymbol{p}_i - x_i \boldsymbol{e}_1 - y_i \boldsymbol{e}_2$. Then we have $\|\boldsymbol{f}_3\| > 0$ since $\|\boldsymbol{f}_3\| = \sqrt{\|\boldsymbol{p}_i\| - x_i^2 - y_i^2}$ and $\|(x_i, y_i)\| < \|\boldsymbol{p}_i\|$.

Next, let $\boldsymbol{e}_1'$ and $\boldsymbol{e}_2'$ be the new vectors spanning the projection plane. These vectors are considered to be in the three-dimensional space spanned by $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$. Then they can be expressed with six variables α_1, α_2, α_3, β_1, β_2, and β_3 as $\boldsymbol{e}_1' = \alpha_1 \boldsymbol{e}_1 + \alpha_2 \boldsymbol{e}_2 + \alpha_3 \boldsymbol{e}_3$ and $\boldsymbol{e}_2' = \beta_1 \boldsymbol{e}_1 + \beta_2 \boldsymbol{e}_2 + \beta_3 \boldsymbol{e}_3$. Also, let $\boldsymbol{r}$ be the vector indicating the rotation axis of the projection plane. Then it can be represented with two variables γ_1 and γ_2 as $\boldsymbol{r} = \gamma_1 \boldsymbol{e}_1 + \gamma_2 \boldsymbol{e}_2$.

Now, using these constants and variables, the following eight constraints are imposed: $\|\boldsymbol{e}_1'\| = 1$, $\|\boldsymbol{e}_2'\| = 1$, $\boldsymbol{e}_1' \cdot \boldsymbol{e}_2' = 0$, $\|\boldsymbol{r}\| = 1$, $\boldsymbol{r} \cdot \boldsymbol{e}_1' = \boldsymbol{r} \cdot \boldsymbol{e}_1$, $\boldsymbol{r} \cdot \boldsymbol{e}_2' = \boldsymbol{r} \cdot \boldsymbol{e}_2$, $\boldsymbol{p}_i \cdot \boldsymbol{e}_1' = x_i'$, and $\boldsymbol{p}_i \cdot \boldsymbol{e}_2' = y_i'$. The first three constraints mean that $\boldsymbol{e}_1'$ and $\boldsymbol{e}_2'$ are orthonormal. The next three constraints indicate that $\boldsymbol{r}$ is a unit vector, and that $\boldsymbol{e}_1'$ and $\boldsymbol{e}_2'$ are the rotations of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ around $\boldsymbol{r}$. The last two constraints imply that the coordinates (x_i', y_i') are obtained by projecting $\boldsymbol{p}_i$ onto the new projection plane. These eight constraints can be efficiently solved by a basic numerical method for simultaneous nonlinear equations such as Newton's method.

3.3 Example

Next, an example of performing the high-dimensional approach is presented. The graph used here is the one obtained by modifying the graph called `erdos1graph`, which was produced in the Erdös Number project.[1] This project is collecting data on coauthorships of papers starting from Paul Erdös, a mathematician who passed away in 1996. `erdos1graph` is the graph whose nodes are the coauthors of the papers written by Erdös (which exclude Erdös himself), and whose edges connect the nodes whose corresponding persons have collaboratively written one or more papers (which may be irrelevant to Erdös). Therefore, this graph exhibits a human social network in the real world. This example uses the graph that excludes, from the original `erdos1graph`, the 46 nodes and 4 edges that do not belong to the maximum connected component (which we call the modified `erdos1graph`). The modified `erdos1graph` consists of 463 nodes and 1,547 edges.

Figure 1(a) illustrates the initial two-dimensional layout of the modified `erdos1graph` obtained by the above method. Next, Figure 1(b) depicts the graph obtained by dragging to the right side a node that was initially located in the center of the layout and that is connected by many edges (the circle indicates the dragged node).

Figure 1(c) shows the graph that is zoomed in around the dragged node and to which the name labels of the persons corresponding to the nodes are added. The dragged node corresponds to Frank Harary, a researcher on Graph Theory, who has the second most coauthorships among the coauthors of Erdös (that is, its node has the second most edges in `erdos1graph`.) In the lower part of Figure 1(c), there is a node corresponding to William T. Tutte, who proposed a classical graph layout method [2]. Tutte has only 4 coauthorships with Erdös' coauthors, and Harary is one of them. Therefore, Tutte's node was placed close to Harary.

[1] `http://www.oakland.edu/enp/`

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/APVIS.2007.329281.

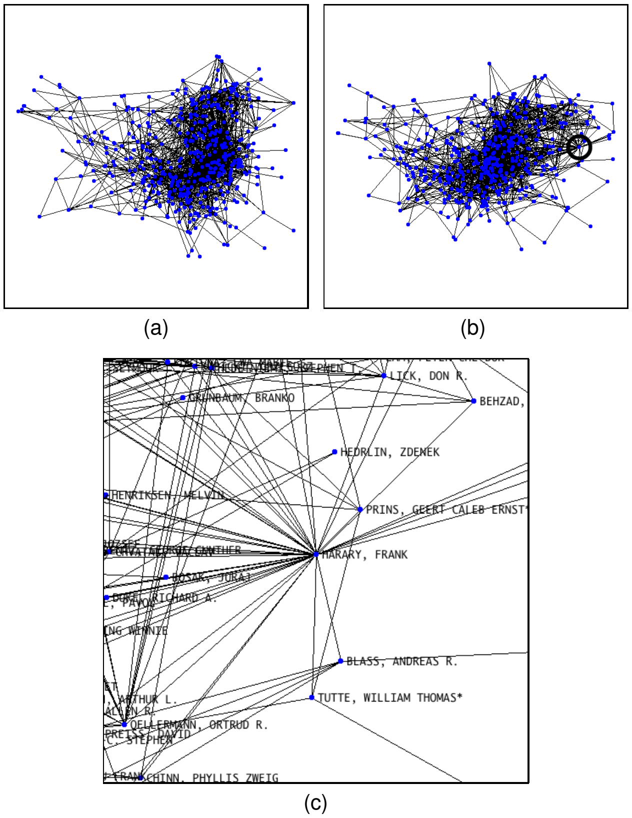

Figure 1: An application to a social network.

4 Sensitivity Analysis

In this section, we explore the high-dimensional approach by performing sensitivity analysis. This analysis investigates how the high-dimensional approach places graph nodes on the two-dimensional plane. We specifically analyze the basic method, which was presented in the previous section, as an instance of the high-dimensional approach.

Consider again the same high-dimensional graph layout as in Subsection 3.2, and also assume that it is projected onto the two-dimensional plane, and that the user drags the node i in the same way. For each node j, let (x_j, y_j) and (x'_j, y'_j) be its positions on the initial and updated projection planes respectively.

Now we consider how the coordinates of each node are changed by the dragging. For the x-coordinate, we can derive the following:

$$x'_j = x_j\alpha_1 + y_j\alpha_2 + (\|\boldsymbol{p}_j\| \cos\theta_j)\alpha_3, \tag{1}$$

where θ_j indicates the angle between $\boldsymbol{p}_j$ and $\boldsymbol{f}_3$ (and hence $\boldsymbol{p}_j \cdot \boldsymbol{f}_3 = \|\boldsymbol{p}_j\|\|\boldsymbol{f}_3\| \cos\theta_j$ holds). Note that we can obtain a similar equation for the y-coordinate.

Let us further examine (1). We can classify the constants and variables in (1) into four categories:

1. Constant $\boldsymbol{p}_j$ that is defined by the high-dimensional graph layout;
2. Constants x_j and y_j that are determined when the initial projection plane is given;
3. Constant θ_j that is settled when the user decides to drag node i;
4. Variables α_1, α_2, and α_3 that are repeatedly updated while the user is dragging the node i.

In (1), the term $(\|\boldsymbol{p}_j\| \cos\theta_j)\alpha_3$ is the most important, since θ_j depends on both the nodes i and j (by contrast, $\boldsymbol{p}_j$, x_j, and y_j do not depend on the node i, and α_1, α_2, and α_3 affect all the nodes in the same way). We can expect that the magnitude of α_3 tends to become large when the user drags the node i far from the initial position. This is because the contribution of the term $\|\boldsymbol{f}_3\|\alpha_3$ is significant in satisfying $x'_i = x_i\alpha_1 + y_i\alpha_2 + \|\boldsymbol{f}_3\|\alpha_3$ (note that α_1, α_2, and α_3 range between -1 and 1 since $\alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 1$).

Let us turn our attention to θ_j, which is the angle between $\boldsymbol{p}_j$ and $\boldsymbol{f}_3$. Since $\boldsymbol{f}_3$ is obtained by the orthonormalization of $\boldsymbol{p}_i$ using $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, we can regard $\boldsymbol{f}_3$ as having a similar direction to $\boldsymbol{p}_i$. Therefore, we can approximate $\cos\theta_j$ by using $\cos\theta_{ij}$, where θ_{ij} denotes the angle between $\boldsymbol{p}_j$ and $\boldsymbol{p}_i$.

We conclude that in general, node j is positively sensitive to the dragging of the node i if $\|\boldsymbol{p}_j\| \cos\theta_{ij}$ is large.

5 Empirical Analysis

In this section, we empirically analyze actual graph layout methods to see if they are appropriate for the high-dimensional approach.

5.1 The TKS Method

First, we perform an experiment to analyze the TKS method. In this experiment, we use the modified `erdos1graph` as the test data. We consider the case of dragging Harary's node (which is regarded as node i), and see the $\|\boldsymbol{p}_j\|$, $\cos\theta_{ij}$, and $\|\boldsymbol{p}_j\| \cos\theta_{ij}$ of each node j.

Figures 2(a1)–(a3) show the results of this experiment. The graph-theoretic distances are calculated from Harary's node (and therefore, the distance for Harary's node is zero). In this figure, the diamond marks indicate the results of Tutte's node.

From these results, we can observe the following:

- Figure 2(a1) shows that Harary's node is close to the origin in the high-dimensional space. Perhaps this is because it has many edges.
- Figure 2(a2) indicates that nodes j with short graph-theoretic distances to Harary's node result in large $\cos\theta_{ij}$.
- Figure 2(a3) suggests that nodes j with short graph-theoretic distances to Harary's node result in large $\|\boldsymbol{p}_j\| \cos\theta_{ij}$. More importantly, Tutte's node has larger $\|\boldsymbol{p}_j\| \cos\theta_{ij}$. This means that the high-dimensional approach works well for indicating the strong relationship between Tutte and Harary.

5.2 Comparison with Other Methods

Next, we compare the TKS method with Hall's method [7] and Harel and Koren's method [4] (which we call the HK method). Hall's method and the HK method are known to work well for very large graphs with mesh structures [4, 7]. Since these methods can compute high-dimensional graph layouts, it is interesting to see if they can be applicable to our high-dimensional approach.

Figure 3(a) depicts the initial two-dimensional graph layout obtained by applying Hall's method to the high-dimensional approach. Figures 2(b1)–(b3) show the results of performing the same experiment on Hall's method as in the previous subsection. The results suggest that Hall's method is completely inappropriate to the high-dimensional approach.

Figure 3(b) illustrates the initial two-dimensional graph layout obtained by applying the HK method to the high-dimensional approach. Figures 2(c1)–(c3) give the results of conducting the same experiment on the HK method as in the previous subsection. Note that Tutte's node exhibits small $\|\boldsymbol{p}_j\| \cos\theta_{ij}$. This means that the HK method does not perform well for the high-dimensional approach.

From these results, we can conclude that the TKS method is better than both Hall's method and the HK method when used in the high-dimensional approach.

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/APVIS.2007.329281.

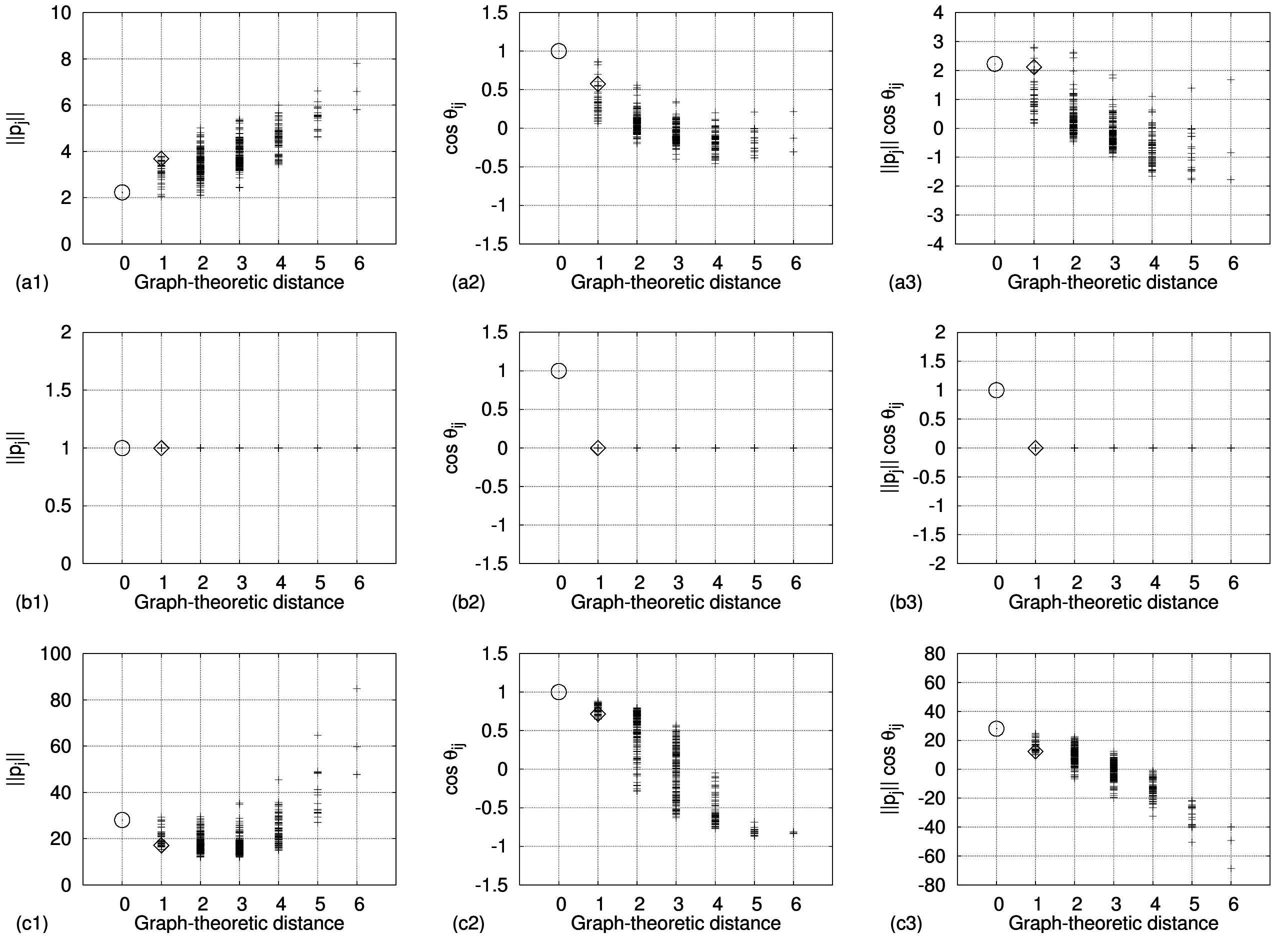

Figure 2: Results of (a1)–(a3) the TKS, (b1)–(b3) Hall's, and (c1)–(c3) the HK methods.

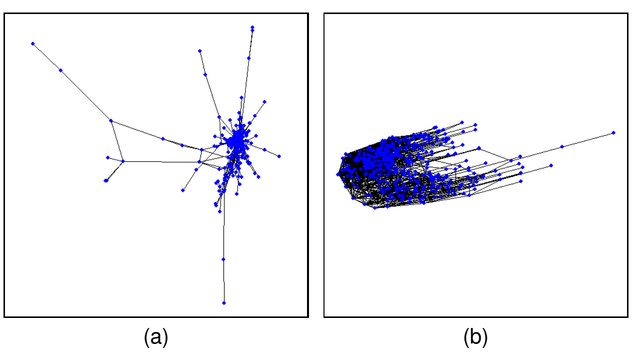

Figure 3: Graph layouts based on (a) Hall's and (b) the HK methods.

6 Conclusions and Future Work

In this paper, we analyzed the high-dimensional approach to interactive drawing of general undirected graphs, by performing sensitivity and empirical analysis. The results showed that the TKS method is more appropriate for the high-dimensional approach than Hall's and the HK methods.

A future direction of our research is to search for methods other than the TKS that are appropriate to the high-dimensional approach. Our plan also includes extending our prototype graph drawing system by further enhancing its display and user interaction functions.

References

[1] I. Bruß and A. Frick. Fast interactive 3-D graph visualization. In *Graph Drawing—GD'95*, volume 1027 of *LNCS*, pages 99–110. Springer, 1996.

[2] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[3] P. Gajer, M. T. Goodrich, and S. G. Kobourov. A multi-dimensional approach to force-directed layouts of large graphs. In *Graph Drawing—GD2000*, volume 1984 of *LNCS*, pages 211–221. Springer, 2000.

[4] D. Harel and Y. Koren. Graph drawing by high-dimensional embedding. *J. Graph Algorithms and Applications*, 8(2):195–214, 2004.

[5] H. Hosobe. A high-dimensional approach to interactive graph visualization. In *Proc. ACM SAC*, pages 1253–1257, 2004.

[6] H. Hosobe. An extended high-dimensional method for interactive graph drawing. In *Proc. APVIS*, volume 45 of *CRPIT*, pages 11–16. Aust. Comput. Soc., 2005.

[7] Y. Koren, L. Carmel, and D. Harel. Drawing huge graphs by algebraic multigrid optimization. *Multiscale Modeling and Simulation*, 1(4):645–673, 2003.

[8] J. B. Kruskal and J. B. Seery. Designing network diagrams. In *Proc. 1st General Conf. on Social Graphics*, pages 22–50. U.S. Dept. Census, 1980.

[9] T. Pisanski and J. Shawe-Taylor. Characterizing graph drawing with eigenvectors. *J. Chem. Inf. Comput. Sci.*, 40(3):567–571, 2000.

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

This is the author's version. The final authenticated version is available online at https://doi.org/10.1109/APVIS.2007.329281.