数理解析研究所講究録
第 1644 巻 2009 年 1-11

MultiPlane Framework for Visualisation and Analysis of Large and Complex Networks

Seok-Hee Hong

School of Information Technologies, University of Sydney, Australia,
shhong@it.usyd.edu.au

Abstract

This paper presents a new framework for visualisation of large and complex networks in three dimensions. In general, the framework uses a *divide and conquer* approach. More specifically, the framework divides a graph into a set of smaller subgraphs, and then draws each subgraph in a 2D "plane" (a bounded planar region). Finally, a three dimensional drawing of the graph is constructed by arranging these planes in 3D, satisfying optimisation criteria.

The MultiPlane framework is very flexible. Algorithms that follow this framework vary in computational complexity, depending on the type of graph and the optimisation criteria that are chosen. The resulting drawing can reduce visual complexity and occlusion, and is easy to navigate. Some optimisation problems arise from the framework; we describe some simple approaches to these problems. Experimental results suggest that the new framework can be useful for visual analysis of large and complex networks such as social networks and biological networks.

Keywords: Graph drawing, Information visualisation, Three dimensions, Large and complex networks.

1 Introduction

Recent technological advances have led to the production of a lot of data, and consequently have led to many large and complex network models in many domains. Examples include:

- Social networks: These include telephone call graphs (used to trace terrorists), money movement networks (used to detect money laundering), and citation networks or collaboration networks. These networks can be very large.
- Biological networks: Protein-protein interaction (PPI) networks, metabolic pathways, gene regulatory networks and phylogenetic networks are used by biologists to analyse and engineer biochemical materials. In general, they have only a few thousand nodes; however, the relationships are very complex.
- Software engineering: Large-scale software engineering deals with very large sets of software modules and relationships between them. Analysis of such networks is essential for design, performance tuning, and refactoring legacy code.
- Webgraphs, where the nodes are web pages and relationships are hyperlinks, are somewhat similar to social networks and software graphs. They are huge: the whole web consists of billions of nodes.

Visualisation can be an effective analysis tool for such networks. Good visualisation reveals the hidden structure of the networks and amplifies human understanding, thus leading to new insights, findings and predictions. However, constructing good visualizations of such networks can be very challenging.

Recently, many methods for visualisation of large graphs have been suggested. For example, see the recent proceedings of Graph Drawing or Information Visualisation conferences. Methods include fast multi-level force directed methods, spectral graph drawing, geometric or combinatorial clustering methods, and multidimensional scaling methods. However, current visualisation methods tend to exhibit one or more of the following problems:

- *Scalability*: current methods for visualisation of large graphs can handle, at best, a few thousand nodes. Most methods do not scale well, in terms of computational efficiency (runtime).
- *Visual complexity*: humans have limited ability in cognition and perception. A drawing of millions of nodes can be cluttered, making it difficult to recognise patterns and inhibiting good insight on the data set.
- *Domain complexity*: In practice, the networks have properties that must be represented visually. For example, in a social network, some nodes are more important (higher centrality) than others. For many biological networks, there are established layout conventions for specific subnetworks. The layout algorithms must respect these domain-dependent constraints.
- *Interaction/navigation methods*: each visualisation method should be accompanied by good navigation methods for exploration of the data. This is critical for further analysis, findings, understandings or even prediction of the structure of the networks. Design of good navigation methods may dependent to a specific visualisation methods.

Affordable high quality 3D graphics in every PC has motivated a great deal of research in 3D graph drawing over the last ten to fifteen years. The proceedings of the annual Graph Drawing conferences document these developments. Three dimensional graph drawings with a variety of aesthetics and edge representations have been extensively studied (see [5, 7, 10, 11, 18, 24]). Examples include algorithms for 3D orthogonal drawing with a limited number of bends or small volume, 3D straight-line grid drawing algorithms with small volume, and 3D graph drawing algorithms that maximise symmetry.

Laboratory experiments have shown that 3D graph visualisations can be up to three times more readable than 2D [25]. However, the availability of the 3rd dimension has made little impact on graph visualisation industry; currently no major graph visualization provider uses 3D. Even though these 3D algorithms of the past 10 years are theoretically significant, none of them have been adopted by the commercial graph drawing software providers. Thus achieving good 3D visualisation remains a challenging problem.

A number of researchers have recently pointed out that full use of 3D layout may not be helpful [2, 4, 6, 21, 26]. Ware [26] advocates a "2.5D design attitude", using 3D depth selectively and paying special attention to 2D layout. He indicates that this may provide the best match with the limited 3D capabilities of the human visual system.

In this paper, we propose a new flexible framework for drawing graphs in three (more exactly "2.5D") dimensions, consistent with the guidelines of Ware. The new MultiPlane framework uses a divide and conquer approach. More specifically, we divide a graph into a set of subgraphs, and then draws each subgraph in a plane (bounded planar region) using well-established 2D drawing algorithms. Finally, a 2.5D drawing of the whole graph is constructed by combining the 2D drawings, satisfying chosen optimisation criteria. Specific algorithms are instantiations of the framework. These require solutions to optimisation problems.

Our framework generalises some of existing methods. For example, PolyPlane methods draw *trees* in 2.5D [21] using a 2D plane for each subtree. Another method is to use 2.5D to visualise a set of related networks in *parallel planes* [2, 4, 6, 26].

MultiPlane methods can be effective in reducing visual complexity and occlusion, and easing navigation. For example, the drawing in Figure 1(a) clearly shows the problem of occlusion and a great deal of 3D clutter. It is very difficult to see the inside of the drawing in order to see more details of the tree, say the center of the tree. However, the drawing in Figure 1(b), a tree drawn with the PolyPlane method [21], clearly shows the inside of the tree structure, thus making it easier to identify the center of the tree. Further, while rotating the drawing as in Figure 1(c), some of the planes can be displayed as lines; this both reduce visual complexity and occlusion and allows the user to concentrate on their plane of interest.

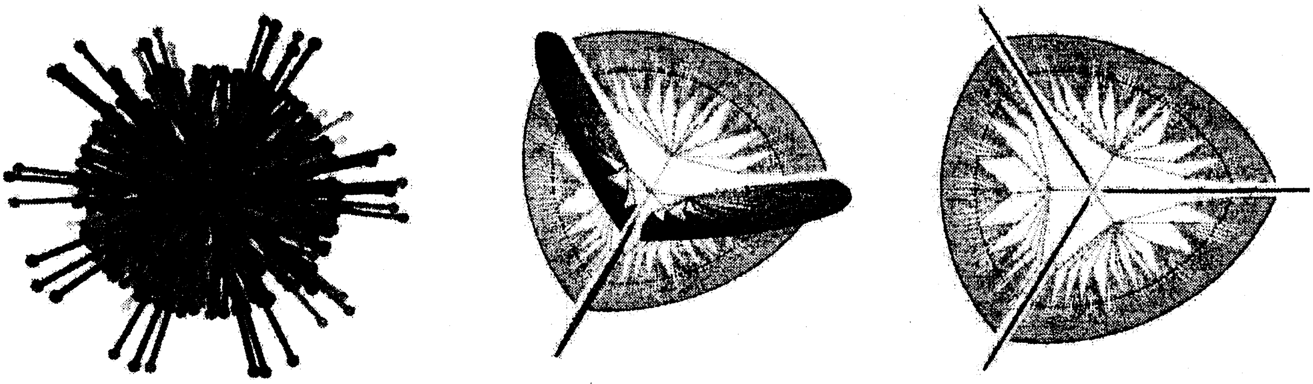

Figure 1: *(a) Example of occlusion: a tree with 483 nodes; (b) a tree with 8613 nodes drawn with PolyPlane method using 6 subplanes; (c) navigating a 2.5D drawing.*

Based on the MultiPlane framework, we have developed a series of algorithms for various types of graphs including general graphs, directed graphs and clustered graphs, and various types of network models such as scale-free networks, dynamic networks, temporal networks, overlapping networks and multi-relational networks. Preliminary experimental results suggest that the MultiPlane framework can be useful for visual analysis and insight into large and complex networks arising in social network and biological network domains. For details, see [2] for scale-free social networks and biological networks, [19, 20] for directed (or hierarchical) graphs, [17] for clustered graphs, [13] for temporal email networks, [8] for the visual comparison of network centralities, and [14, 15] for overlapping biological networks. These methods are implemented in GEOMI, a visual analysis tool for large and complex networks [1].

This paper is organized as follows: the framework is described in Section 2. Section 3 presents specific instances of the framework and experimental results including visualisation of social networks and biological networks. Section 4 concludes.

2 The MultiPlane Framework

In this section, we describe our new framework for drawing graphs in 2.5D. In particular, we use graph theoretic approaches and network analysis methods to reduce the scalability and complexity of the large and complex network.

The framework uses *planes*. In general, the planes are bounded planar regions in 3D. An outline of the framework, which will call the `MultiPlane` method is below:

`MultiPlane Framework`

1. Choose a partitioning of a graph G into a set of subgraphs $\{G_i : 1 \leq i \leq k\}$.
2. For each i, $1 \leq i \leq k$, draw G_i in a plane P_i using a 2D drawing algorithm.
3. Arrange each plane P_i in 3D satisfying chosen criteria.
4. Connect inter-plane edges between the planes.

A simple example of a drawing drawn with the framework is illustrated in Figure 2.

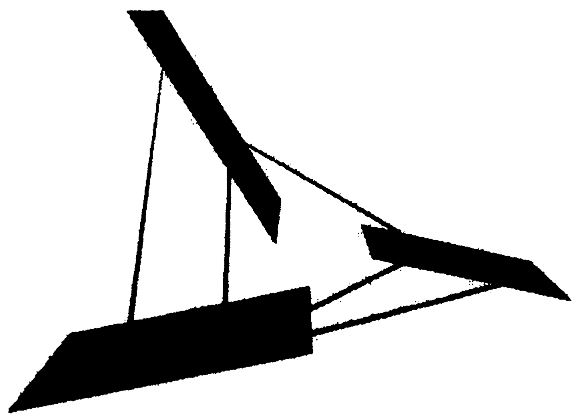

Figure 2: *A 2.5D drawing drawn with the MultiPlane framework.*

The framework is very flexible, as there are many steps at which an arbitrary choice can be made. Furthermore, there are combinatorial optimisation problems involved in each step. Thus, each MultiPlane algorithm should attempt to optimise some criteria chosen at each step.

For example, Step 1 involves the well studied problem of finding a good partitioning of a graph. In some cases, the partitioning is given by the application domain; for example, the entities in a software system may be clustered into modules *a priori*. Otherwise, the problem of finding a good partitioning can be a classical optimisation problem; for example, finding minimum cuts or a balanced partitioning. In most cases, such problems are NP-hard. However, fast heuristics and approximation algorithms are available [3, 12, 22]. Note that the number of planes used should be small; otherwise, the visualisation loses the 2.5D attitude.

For Step 2, one can choose a preferred 2D graph drawing algorithm based on the application domain [9, 23]. For example, *intra-plane* edge crossings can be minimised using one of well established 2D graph drawing algorithms [9, 23]. However, sometimes modifications are necessary due to the newly introduced optimisation criteria chosen at Step 4 for *inter-plane* edges.

Step 3 involves some new criteria for arranging planes; these can vary from one instance of the framework to another. As design guidelines for MultiPlane framework, the following general criteria apply:

- plane-plane crossing: no two planes cross each other.
- plane-edge crossing: no edge cross each plane more than once, and no edge cross more than one plane.
- plane angular resolution: the angles between the planes should be large.

For example, we can define a new problem of *plane-edge* crossing minimisation, to minimise the number of inter-plane edges which cross more than one planes.

At Step 4, we need to solve a new problem of minimising the total inter-plane edge length, which maintaining the crossing-free properties of the inter-plane edges.

Based on the MultiPlane framework, we have developed a series of algorithms for various types of graphs including trees, planar graphs, general graphs, hierarchical graphs, directed graphs and clustered graphs, and various types of network models such as small-world networks, scale-free networks, dynamic networks, evolution networks, temporal networks, overlapping networks and multi-relational networks. The time complexity of a MultiPlane algorithm depends on the time complexity of the method chosen at each step. In practice, we usually choose fast heuristics in each step for scalability issue.

Preliminary experimental results suggest that the MultiPlane framework can be useful for visual analysis and insight into large and complex networks arising in social network and biological network domains. For details, see [2] for scale-free social networks and biological networks, [19, 20] for directed (or hierarchical) graphs, [17] for clustered graphs, [13] for temporal email networks, [8] for the visual comparison of network centralities, and [14, 15] for visualisation of overlapping biological networks. These methods are implemented in GEOMI, a visual analysis tool for large and complex networks [1].

In the next section, we present specific instances of the framework together with experimental results including visualisation of social networks and biological networks.

3 MultiPlane Algorithms and Experimental Results

In this section, we first present a MultiPlane algorithm for drawing general graphs in 2.5 dimensions, and then present experimental results produced based on the MultiPlane framework. Let G be a general undirected graph.

We first divide the graph G into a set of $k \geq 2$ smaller subgraphs. More specifically, at Step 1 of `Framework MultiPlane`, we can use $k-way$ balanced partitioning with minimum cut to divide the graph into a set of k subgraphs. This is an NP-hard problem; however there are many good heuristics and approximation algorithms available. See [3, 12, 22] for details.

At Step 3 and Step 4, we have a new problem of arranging each plane in 3D to minimise the number of plane-edge crossings and the total inter-plane edge length.

To solve this problem effectively and efficiently, we need to consider the structure of the partitioning. More specifically, we can define a supergraph defined by the relation between the set of induced subgraphs from the partitioning. Suppose that we have a set of induced subgraph $G_1, G_2, \ldots, G_k$ where $G_i = (V_i, E_i)$. We can define a supergraph G_S such that each G_i is a node v_i in G_S and if there is an edge between a node in G_i and a node in G_j then there is an edge between v_i and v_j in G_S. Note that we can assign weights to each node and edge in G_S depending on the size of the subgraph and the number of edges between the subgraphs.

We can design a series of algorithms based on the structure of G_S: path, cycle, tree, planar graph and general graph. Suppose that G_S is a tree. We first draw G_S using a weighted 3D tree drawing algorithm to determine the arrangement of the planes. Then we replace each node in the drawing with a 2D drawing of each G_i drawn in a 2D plane. See Figure 3 for an example [17]. The drawing has no plane-plane crossings and no plane-edge crossings.

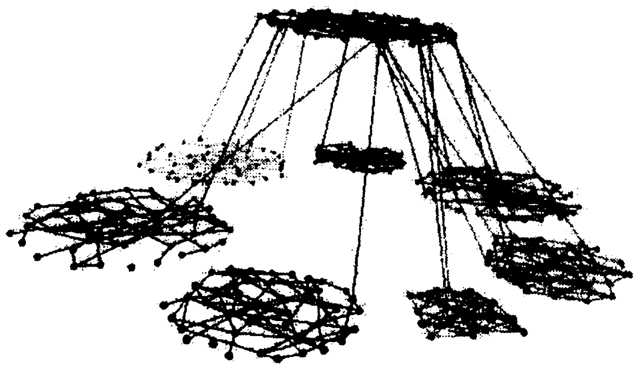

Figure 3: *2.5D drawing of a graph with hierarchical tree clustering structure.*

Figure 4 shows a 2.5D drawing of eight related metabolic pathways from KEGG database, produced by the method for general clustered graphs [17]. The most important pathway is emphasized as the top plane, while inter-plane edges reveal complex relationships between related pathways. The drawing has no plane-plane crossings while minimising the total inter-plane edge lengths and plane-edge crossings.

Figure 5(a) shows a 2.5D drawing of a BFS tree of the School of IT, University of Sydney webgraph, with 4485 nodes, using 12 planes. Figure 5(b) shows 2.5D visualisation of Erdos network of mathematician collaboration. We first compute a BFS tree rooted at Erdos, and construct a 2.5D drawing with 30 planes. One can easily identify relationships between Erdos and people with Erdos number 1 and Erdos number 2. Both use a variation of PolyPlane algorithms [21].

Figure 6 shows a 2.5D drawing of a temporal email network [13]. It clearly shows a propagation of email virus based on the time stamp drawn in each plane, enabling monitoring and easy detection of email servers infected by computer virus.

Figure 7 shows a 2.5D drawing of three overlapping biological networks: metabolic pathways, protein protein interaction networks, and gene regulatory networks [15]. The drawing enables integrated analysis and supports complex high level analysis by relating

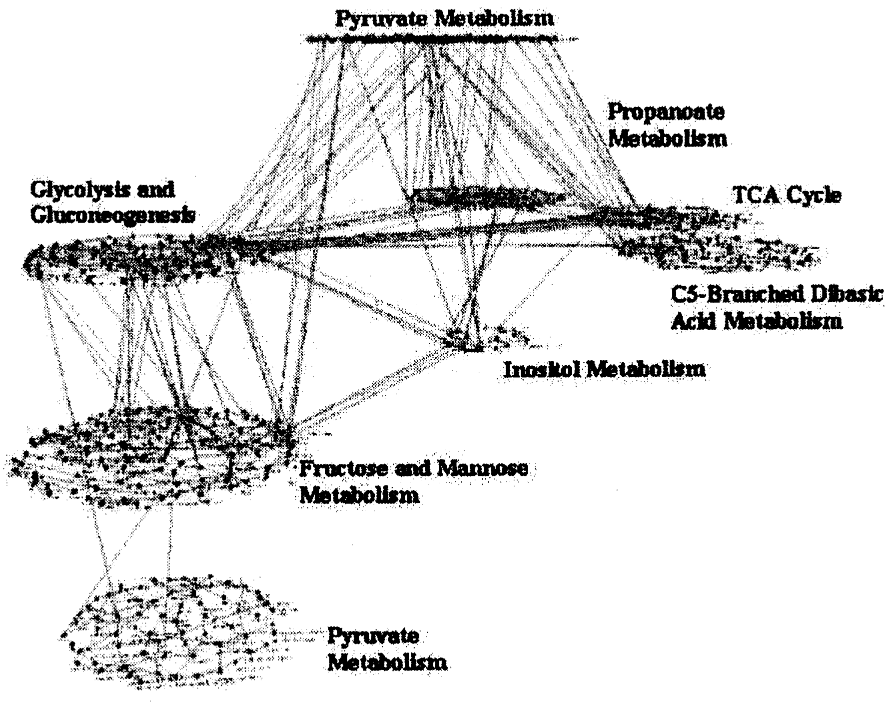

Figure 4: *Visualisation of eight metabolic pathways in 2.5D.*

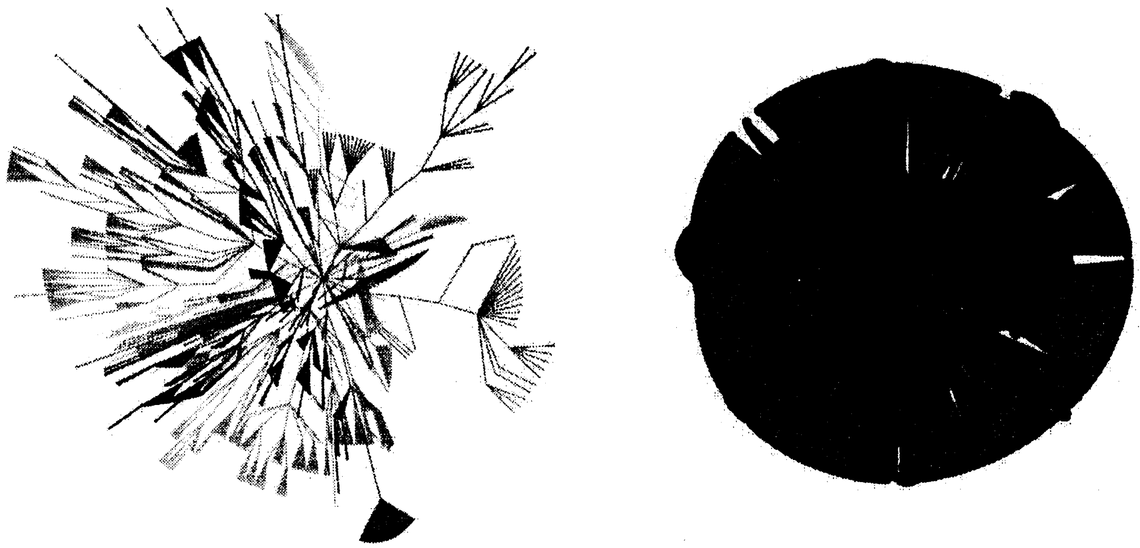

Figure 5: *(a) BFS tree of School of IT website with 4485 nodes drawn with 12 planes; (b) Erdos Number visualisation using 30 planes.*

Figure 6: *2.5D drawing of temporal email networks.*

three heterogeneous networks. It achieves both drawing aesthetics for each individual network, and an optimization criteria for minimising the total inter-plane edge lengths used for highlighting the overlapping between networks.

Figure 8 shows 2.5D drawings of directed graphs, arising in software engineering domain (from the ROME graph drawing data set), produced by the 2.5D hierarchical layout method for directed graphs in [19, 20]. It uses two partitioning algorithms with different optimisation criteria such as balanced min-cut and minimising the total inter-plane edge length.

4 Conclusion

A new flexible framework for drawing graphs in 2.5D using planes is presented with application to visual analysis of large and complex networks. Our current work involves theory of 2.5D graph drawing, inspired by the MultiPlane framework with various graph models, combinatorial optimisation criteria and application domains.

References

[1] A. Ahmed, T. Dwyer, M. Forster, X. Fu, J. Ho, S. Hong, D. Koschützki, C. Murray, N. Nikolov, A. Tarassov, R. Taib and K. Xu, GEOMI: Geometry for Maximum Insight, Proceedings of Graph Drawing 2005, Lecture Notes in Computer Science 3843, pp. 468-479, 2006.

[2] A. Ahmed, T. Dwyer, S. Hong, C. Murray, L. Song and Y. Wu, Visualisation and Analysis of Large and Complex Scale-free Networks, Proceedings of EuroVis 2005 (EUROGRAPHICS-IEEE VGTC Symposium on Visualization), pp. 1-8, IEEE, 2005.

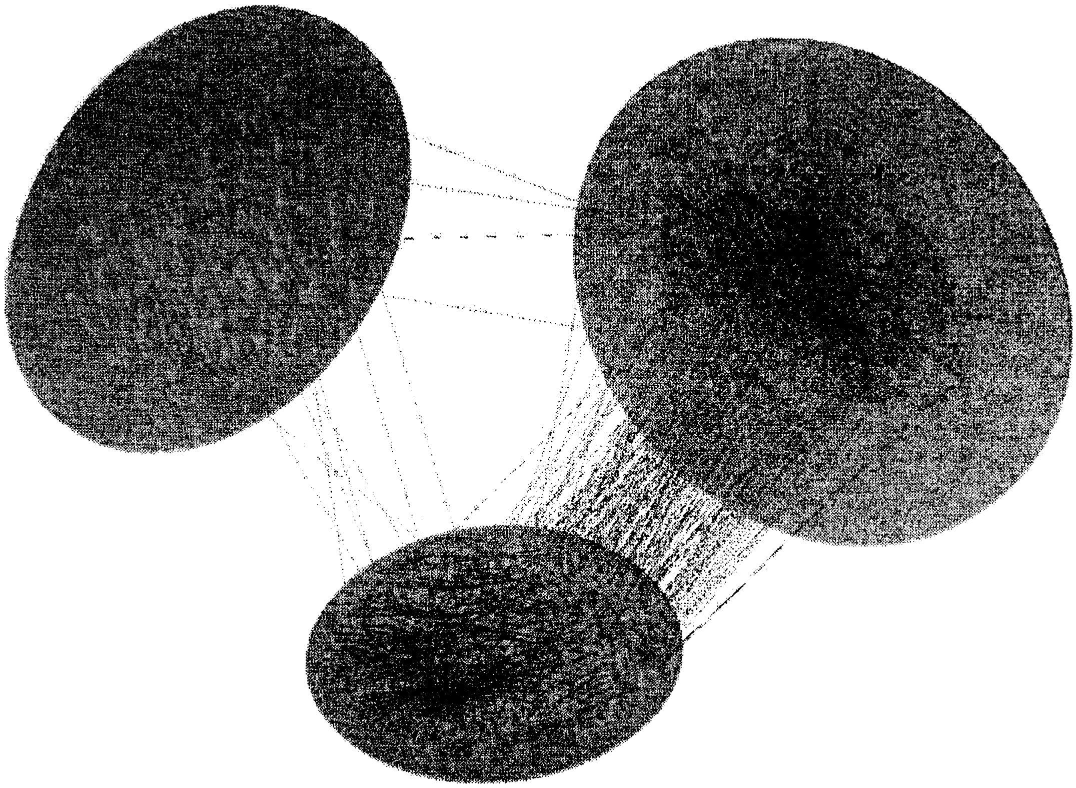

Figure 7: *Visualisation of three overlapping biological networks in 2.5D.*

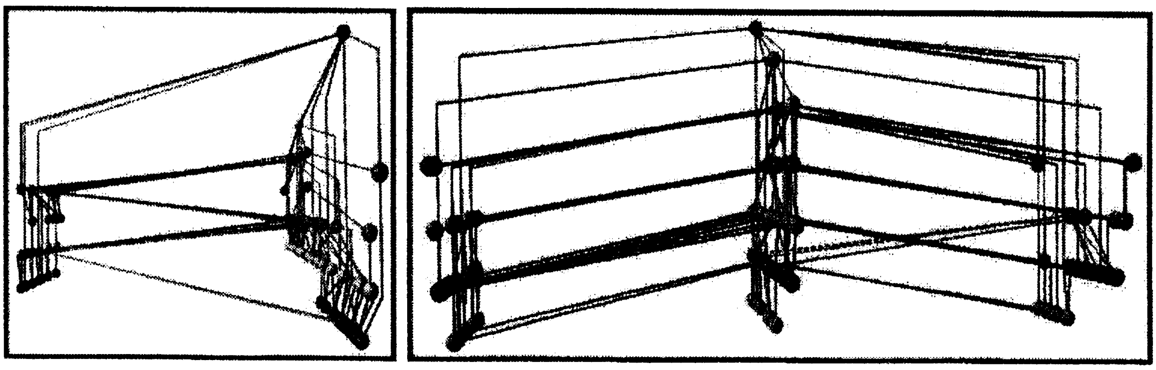

Figure 8: *2.5D drawings of hierarchical graphs with (a) 2 planes and (b) 3 planes.*

[3] C. J. Alpert and A. B. Kahng, Recent Developments in Netlist Partitioning: A Survey, *Integration: the VLSI Journal*, 19(1-2), pp. 1-81, 1995.

[4] M. Baur, U. Brandes, M. Gaertler and D. Wagner, Drawing the AS Graph in 2.5 Dimensions, Proceedings of Graph Drawing 2004, Lecture Notes in Computer Science 3383, pp. 43-48, 2005.

[5] T. Biedl, T. Thiele and D. R. Wood, Three-Dimensional Orthogonal Graph Drawing with Optimal Volume, Proceedings of Graph Drawing 2000, Lecture Notes in Computer Science 1984, pp. 284-295, 2001.

[6] U. Brandes, T. Dwyer and F. Schreiber, Visualizing Related Metabolic Pathways in Two and a Half Dimensions, Proceedings of Graph Drawing 2003, pp. 111-122, 2004.

[7] R. Cohen, P. Eades, T. Lin and F. Ruskey, Three Dimensional Graph Drawing, *Algorithmica*, 17 (2), pp. 199-208, 1996.

[8] T. Dwyer, S. Hong, D. Koschützki, F. Schreiber and K. Xu, Visual Comparison of Network Centralities, Proceedings of APVIS (Asia-Pacific Symposium on Information Visualisation) 2006, pp. 189-197, CRPIT 60, ACS, 2006.

[9] G. Di Battista, P. Eades, R. Tamassia and I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice-Hall, 1998.

[10] V. Dujmović, P. Morin and D. R. Wood, Path-Width and Three-Dimensional Straight-Line Grid Drawing of Graphs, Proceedings of Graph Drawing 2002, Lecture Notes in Computer Science 2528, pp. 42-53, 2003.

[11] P. Eades, A. Symvonis and S. Whitesides, Three Dimensional Orthogonal Graph Drawing Algorithms, *Discrete Applied Mathematics*, 103, pp. 55-87, 2000.

[12] G. Even, J. Naor, S. Rao and B. Schieber, Fast Approximate Graph Partitioning Algorithms, *SIAM Journal on Computing*, Vol. 28, No. 6, pp. 2187-2214, 1999.

[13] X. Fu, S. Hong, N. S. Nikolov, X. Shen, Y. Wu and K. Xu, Visualization and Analysis of Email Networks, Proceedings of APVIS (Asia-Pacific Symposium on Visualisation) 2007, IEEE, pp. 1-8, 2007.

[14] D. Fung, S. Hong, D. Koschützki, F. Schreiber and K. Xu, 2.5D Visualisation of Overlapping Biological Networks, *Journal of Integrative Bioinformatics*, 5(1), 2008.

[15] D. Fung, S. Hong, D. Koschützki, F. Schreiber and K. Xu, Visual Analysis of Overlapping Biological Networks, *submitted.*

[16] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness*, Freeman, 1979.

[17] J. Ho and S. Hong, Drawings Clustered Graphs in Three Dimensions, Proceedings of Graph Drawing 2005, Lecture Notes in Computer Science 3843, pp. 492-502, 2006.

[18] S. Hong, Drawing Graphs Symmetrically in Three Dimensions, Proceedings of Graph Drawing 2001, Lecture Notes in Computer Science 2265, pp. 189-204, 2002.

[19] S. Hong and N. Nikolov, Layered Drawings of Directed Graphs in Three Dimensions, Proceedings of APVIS (Asia Pacific Symposium on Information Visualisation) 2005, CPRIT 45, pp. 69-74, 2005.

[20] S. Hong, N. Nikolov, and A. Tarassov, A 2.5D Hierarchical Drawing of Directed Graphs, *Journal of Graph Algorithms and Applications*, Vol. 11, no. 2, pp. 371-396, 2007.

[21] S. Hong and T. Murtagh, Visualization of Large and Complex Networks Using PolyPlane, Proceedings of Graph Drawing 2004, Lecture Notes in Computer Science 3383, pp. 471-482, 2005.

[22] H. Nagamochi and T. Ibaraki, *Algorithmic Aspects of Graph Connectivities*, Encyclopedia of Mathematics and Its Applications, Cambridge University Press, 2008.

[23] M. Kaufmann and D. Wagner, (eds.), *Drawing Graphs: Methods and Models*, Lecture Notes in Computer Science Tutorial 2025, Springer Verlag, 2001.

[24] J. Pach, T. Thiele and G. Toth, Three-dimensional Grid Drawings of Graphs, Proceedings of Graph Drawing 1997, Lecture Notes in Computer Science 1353, pp. 47-51, 1998.

[25] C. Ware and G. Franck, Viewing a Graph in a Virtual Reality Display is Three Times as Good as a 2-D Diagram, IEEE Conference on Visual Languages, pp. 182-183, 1994.

[26] C. Ware, Designing with a 2 1/2D Attitude, Information Design Journal 10, 3, pp. 171-182, 2001.

[27] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.