



HAL
open science

Edge Visual Encodings in Matrix-Based Diagrams

Joris Sansen, Romain Bourqui, Bruno Pinaud, Helen Purchase

► **To cite this version:**

Joris Sansen, Romain Bourqui, Bruno Pinaud, Helen Purchase. Edge Visual Encodings in Matrix-Based Diagrams. Information Visualisation (iV), 2015 19th International Conference on , Jul 2015, barcelone, Spain. pp.62 - 67, 10.1109/iV.2015.22 . hal-01189166

HAL Id: hal-01189166

<https://hal.science/hal-01189166>

Submitted on 1 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Edge Visual Encodings in Matrix-Based Diagrams

Joris Sansen, Romain Bourqui, Bruno Pinaud, Helen Purchase
Bordeaux University, LaBRI, France and University of Glasgow, UK
{jsansen, romain.bourqui, bruno.pinaud}@labri.fr, Helen.Purchase@glasgow.ac.uk

Abstract

The most common depictions of graphs are node-link diagrams (NLDs) and matrix-based diagrams (MBDs). Making valid comparisons between these two visualisation techniques is difficult because they are each subject to a variety of representation parameters with respect to graph layout (NLD) and node ordering (MBD), meaning that any given choice of layout and order (even if they fulfil some aesthetic criteria) may influence experimental results. To overcome this problem, we propose a MBD-based technique which hybridises the entity visual encoding of a MBD with the edge visual encoding of a NLD. Using a typical MBD, we propose three edge visual encoding evolutions to ultimately render edges like in a NLD while preserving nodes depiction and order. Such encoding evolutions allow us to perform an experimental evaluation of user performances for a path finding task without the above limitations. We show that for a path finding task, our edge visual encoding evolutions tend to improve the user experience when analysing and interacting with a MBD.

1 Introduction

In modern contexts, visual analytics of social-like relationships has become increasingly more popular. In biology, epidemiology, sociology and marketing for example, the need for effective visualisation tools has increased, and finding connections between entities and/or groups is therefore a common task. Popular approaches for visually exploring such relational data include the use of node-link diagrams (NLDs) and matrix-based diagrams (MBDs). In a NLD, each entity is represented as one shape, whereas a relationship between two entities is represented as a polyline (*link*) between the corresponding shapes. MBDs are square matrices in which entities are displayed twice, on the abscissa and on the ordinate, and a relationship between two entities is represented by a shape in the corresponding row and column.

Improving NLDs and MBDs often corresponds to optimising aesthetic criteria. Some criteria are common to both diagrams, such as symmetry, while others are specific to one representation technique. For example, reducing the number of edge crossings is one of the most impor-

tant criteria for NLDs [17]. For MBDs, one of the main aesthetic criteria is to represent close-proximity entities (in terms of graph distance) as close as possible in the representation [15]. Both techniques have been improved using different layouts or ordering algorithms. The graph drawing community focuses on the design of efficient graph layout algorithms. Among these algorithms, one can find force-directed, linear algebra, constraint-based, topological decomposition of the graph or space-filling techniques (e.g., [14, 10, 4, 1, 2]). For MBDs, node ordering optimisation relates to the minimum linear arrangement problem, which is well-known to be *NP*-Hard. Therefore the community mainly focuses on the design of heuristic methods. The main approaches are spectral sequencing, hill-climbing heuristics, a combination of both or partial orders that arise from clustering technique (e.g., [11, 19, 16]). Mueller and Ma [15] compared the effects of different node ordering algorithms on the readability of a MBD for connectivity models. Instead of focusing on the improvement of a particular technique, other methods attempt to take advantage of both representation techniques and to introduce hybrid visualisation techniques (e.g., [8, 7, 20]).

Some researchers have also attempted to determine when to use NLD or MBD [5, 9]. Ghoniem *et al.* [5] compared a force-directed layout algorithm for NLDs and an alphabetical ordering for MBDs. When considering large and dense graphs, user evaluations showed that MBDs perform better than NLDs for almost all tested tasks, *i.e.*, estimate the number of nodes and links, find the given nodes and find the most connected nodes and node neighbours. However, in case of small and/or sparse graphs, NLDs statistically outperform MBDs. Keller *et al.* [9] confirmed these results by comparing these two visualisation techniques with the same experimental settings. These studies provide interesting findings for improving user performance. However, the differences between MBDs and NLDs in terms of user performance are difficult to determine. The manner in which entities and relations are visually encoded in MBDs and NLDs introduce a limitation in such studies because no algorithm for positioning the entities/relations is common to both visualisation techniques. The node position and order actually induce variations in

the visual quality of both NLDs and MBDs (*e.g.*, edge-crossing number or average/maximum edge length); thus, variations are induced in the experimental results.

In this study, we propose three edge visual encodings that smoothly transform edges from a MBD visual encoding to a NLD-like visual encoding while preserving node depictions and ordering. The second contribution is an experimental evaluation which compares MBD and NLD edge visual encoding efficiencies for a path-finding task without the bias of node positioning/ordering. As far as we know, this is the first experimental evaluation that makes it possible to compare NLDs and MBDs without the bias inherent in choosing a NLD layout or a MBD node ordering.

The remainder of this paper is structured as follows. In Section 2, we first introduce the three evolutions of edge visual encoding in MBDs. Then in Section 3, we present the design and setup of the experimental evaluation to compare the efficiencies of these encodings for a path-finding task. We provide and discuss the results in Sections 4 and 5. Finally, we present conclusions and discuss possible future work in Section 6.

2 Edge Visual Encodings

In a MBD, a relation between two entities is represented by a shape at the intersection of the row and column of the two entities. Henry *et al.* [8] consider that these rows and columns are part of the entity visual encoding, while an edge is only represented by a shape at their intersection. In this study, we make the opposite assumption and consider the row and the column as part of the edge visual encoding, *i.e.*, an edge is encoded as two rectangles (one for each row and column) and a shape at their intersection (see Fig. 1(a)). This assumption is supported by the fact that rectangles ease the identification of edge extremities, particularly when considering medium and large matrices.

In this study, the key idea is to reduce the space necessary to encode edges as much as possible and therefore to reduce the visual complexity. We propose three variations of edge visual encoding to smoothly transform traditional MBD visual encoding into *link* encoding through different simplification processes (see Fig. 1).

The first variation consists of pulling out unused rows and columns and shortening the two rectangle representing a connection to their essential components (see Fig. 1(b)). Using this visual encoding, the rectangles do not span the entire row/column but rather stop at their intersection. In a typical MBD, the grid formed by the intersections of all rectangles contains empty cells (except for complete graphs) which display *no-relation* information. This process reduces the number of such cells and therefore reduce visual complexity. The next variation consists of reducing the two rectangles into a single polyline with an orthogonal bend at their intersection (see Fig. 1(c)). To distinguish

bends from crossings, the shape of the edges are conserved and their sizes are reduced to help distinguish segments between contiguous shapes. Such encoding reveals some visual similarities to *DAGView* [12] which aims to combine advantages of both techniques to produce a NLD. Rather than duplicating nodes, Kornaropoulos and Tollis lay the nodes out within the matrix using two orders (one for each axis) and edges are routed in an orthogonal manner. The last variation consists of replacing both the polyline and the shape with a single curved polyline (Fig. 1(d)). Similar to the previous variation, the curve is needed to prevent ambiguities between bends and crossings. This edge encoding has been introduced by Henry *et al.* [8] as a potential transitional states to transform a MBD into a NLD.

Notably, the four variations are visually similar because entity visual encoding is preserved at each step of the transformation. However, we can certainly consider the two latter encodings as NLDs because the edges are encoded as links between the entities they connect. This statement is the starting point for the bias-free comparison between MBD and NLD edge visual encodings presented in this study. In the remainder of the paper, we refer to the original matrix as Encoding 1 and the variations respectively as Encoding 2, Encoding 3 and Encoding 4.

3 Experimental Evaluation

We design an experimental evaluation based on 3 intermediate research questions: (i) Which of the 4 encodings eases task completion (completion time and error rate)? ; (ii) Which of the 4 encodings makes it possible to find shorter paths? and (iii) Which of the 4 encodings induces fewer errors (number of selections to build a path)?

3.1 The task

The task consists of selecting edges one by one to find a path between two highlighted nodes (called *start* nodes) within a time limit. A path can be built from either of the *start* nodes at the same time. If the path is built from both *start* nodes, then the aim is to join each sub-path to ultimately connect the two *start* nodes. Three rules are introduced to maintain a reasonable difficulty level: (i) it is possible to select an edge only if it extends one of the sub-path extremities; (ii) self-loops are forbidden, and (iii) if two sub-paths are created, then one path can join the other only by its extremity. To complete the task, the user has to select a path and to validate it within the time limit.

3.2 Preliminary hypotheses

Our hypothesis is that it is possible to enhance user performances for a path-finding task in a MBD by reducing visual clutter and simplifying visual reading rules using a hybrid edge visual encoding. Our expectation is that Encoding 3 would demonstrate a real difference in user performance, as it seems to be easier to follow a single line than to keep one's eyes between two lines. Furthermore, a

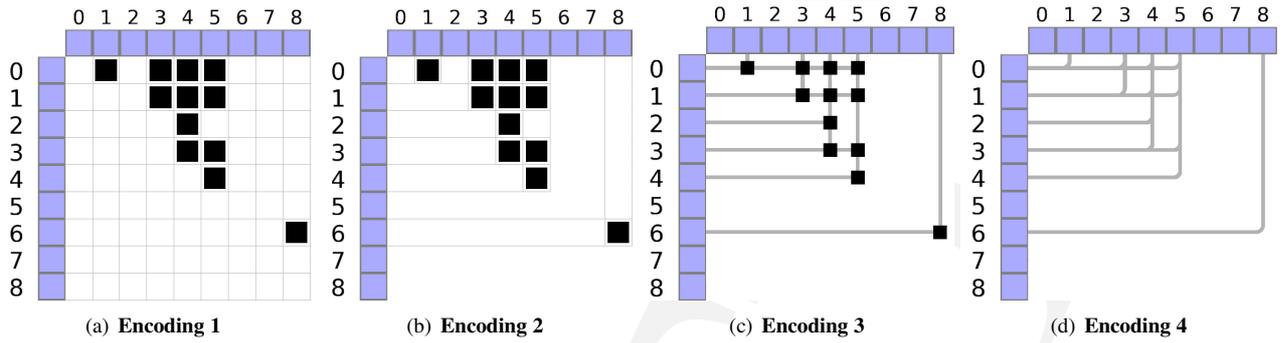


Figure 1: Evaluated edge visual encodings. (a) An edge is encoded as two rectangles and a black square. (b) Rectangles do not span the entire row/column. (c) The edge is encoded with a polyline and a black square. (d) Both the polyline and shape are replaced by a single curved polyline.

path-finding task necessarily relates to neighbour identification. Our hypothesis is that such an identification would be easier with Encoding 3. Similar results could be expected for Encoding 4. However, the distinction between bends and crossings seems easier with shapes than with curves, and it should induce distinction issues. Thus, we expect that this encoding would make matrix reading more difficult and induce more errors.

3.3 Datasets

We extracted experimental data from the graph of co-occurrence relation of actors of the *Infovis Contest 2007* dataset [13]. It has been achieved using a breadth first search (BFS) algorithm from a random starting node. The extraction algorithm has two parameters n and k , where n is the desired number of nodes and k is the number of visited neighbours at each step of the BFS. The input parameter k can also be interpreted as the community size in the resulting network. The BFS stops once the desired graph size has been reached, and the extracted graph is induced in the entire network by the visited nodes. In the experiment, we used two graph sizes (50 and 100 nodes) and two community sizes (5 and 10) which led to 4 categories of graphs. The limit of 100 nodes was set to maintain an adequate level of readability. Two graphs for each combination of n and k were generated. For each of these graphs, we randomly extracted two *start* nodes with a shortest path length of 4.

3.4 Edge orientation and node ordering

The unoriented graphs can be represented in two ways using a MBD, either with a double display of each edge that results in symmetric matrices, or with a single-edge display, that results in *half*-matrices. Pilot experiments revealed that a double-edge display induces clutter and reduces matrix readability; half-matrices were therefore chosen.

To compute a node ordering, we combine a graph clustering algorithm [3] and visit orders during the graph extraction process (*e.g.*, communities are ordered in the same order as they appeared during the BFS).

3.5 Protocol

We followed the recommendations of Purchase [18] and designed a 4 practice trials and 32 experimental trials evaluation. From a participant's point of view, the practice trials are indistinguishable from the experimental trials. Practice trials are performed first to ensure that users are well settled into the trials and are at their peak performance. Different experimental trials order were possible, by blocks of encodings or on the contrary, avoiding an encoding to appear twice in a row. We chose to use the latter order but both methods present their own advantages and drawbacks. We ask users to perform the task in a limited amount of time. During the experiment, the remaining time is always displayed. We included this time limit to encourage users to complete the tasks as quickly as possible. Furthermore, we expect that it will avoid any transformation of the path-finding task into a shortest path-finding task. Preliminary tests reveal that 3 minutes to complete a task is a good compromise as it gives users enough time to complete the major part of the tasks while keeping the total time of the evaluation at an acceptable level. A maximum break of 20 seconds between tasks is offered and a *no-limit* break is available after 20 tasks. Once all tasks are completed, the user has to fill in a questionnaire about her/his favourite encoding and the pros and cons of each encoding.

3.6 The evaluation user interface

The evaluation GUI consists of a single view of an adjacency matrix, using one of the 4 edge visual encodings within a standard web browser in full-screen mode (see Fig. 2). A reminder of the experiment question is displayed on top of the evaluation interface; and below, is an interactive view of the matrix and two buttons. The first button

(bottom left) centres the view on the matrix and the other (bottom right) validates the task. To perform the evaluation, the interface provides two interaction tools: a classical zoom-and-pan and a selection interaction tool that adheres to the rules described in Section 3.1.

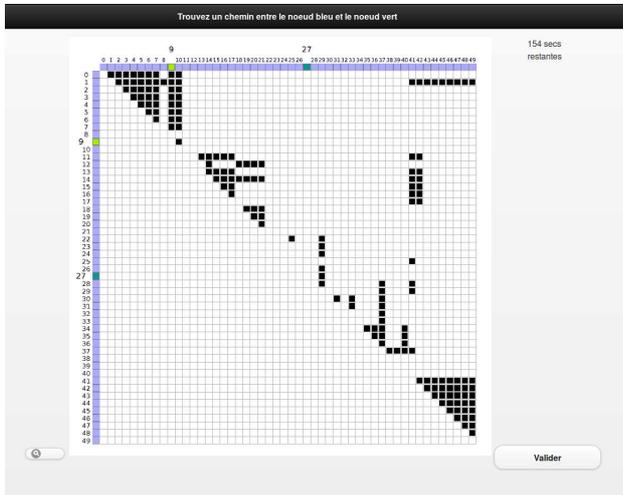


Figure 2: Screenshot of the evaluation GUI.

3.7 The participants

The participants consist of undergraduate students, PhD students and researchers in various fields of computer science. All subjects are familiar with graphs, NLDs and MBDs but do not manipulate these tools on a daily basis. A total of 33 persons volunteered for the pilot tests and evaluations (6 and 27 respectively).

4 Results

We observe four different parameters during this evaluation: error rate, response time, path length and total number of selected edges. Because of high error rates, we filtered out 7 of the 27 volunteers, 4 (resp. 3) of whom had a nearly 50% rate of incomplete task (resp. 33%).

4.1 Quantitative results

Because the data are not normally distributed, we used non-parametric statistical tests. For the error rate, we used a Friedman test, as all data from the 20 participants are used. For response time, path length and number of selected edges, only correct answers have to be considered and we therefore use a Kruskal-Wallis test. A standard significance level $\alpha = 0.05$ is used to test whether significant differences occur.

The results (Fig. 3) show that no significant difference exists between the 4 visual encodings for any parameter considered (error rate, response time, path length and number of selections). Some trends are however noticeable. Encoding 3 seems to be the most efficient in terms of the

error rate. In contrast, Encoding 1 has the poorest results. If we consider the mean response time and path length, Encoding 1 has the shortest path length but the highest response time. Finally, Encoding 4 has the smallest number of selections. These trends are consistent with the preliminary hypotheses, despite the lack of significant differences.

Analysing the results by graph category, as a combination of n and k , we obtain statistically significant differences for all observed parameters (see Fig. 3) as the corresponding p-values are lower than the significance level $\alpha = 0.05$. Such analysis confirms our expectation that among the dataset, some categories induce harder (resp. easier) tasks. The graph categories can thus be split into 3 levels of difficulty: *low* (n100k10), *medium* (n50k5 and n50k10) and *high* (n100k5).

We further investigate and split our data to analyse the performance of each encoding per graph category (Fig. 3). We apply a Bonferoni correction with a new significant level $\alpha = 0.025$. Again, no significant differences are seen but trends exist. For the *high-difficulty* graph category (n100k5), linked encodings (Encoding 3 and Encoding 4) seem to work best. For *medium-* and *high-difficulty* graph categories, MBD encoding (Encoding 1) has the most errors. Concerning the *low-difficulty* graph category, there is almost no difference between the 4 encodings.

4.2 Qualitative results

We asked all participants which visual encoding they preferred. As expected, Encoding 3 is preferred by the majority (47.6%), followed by Encoding 2 (21.4%), Encoding 1 (7.1%) and Encoding 4 (4.8%). The remaining participants (19.1%) did not express a preference. These values are consistent with our preliminary hypotheses and the trends revealed by the quantitative results.

An analysis of the participant comments revealed that Encoding 1 generally appears more intuitive and easier to use for small graphs. The majority of participants found that Encoding 2 was better than Encoding 1 because it improves the readability of edges and communities. Encoding 3 appeared to be the best for visualising edges, although it reduces the identification of communities. Users also felt that the interaction area was smaller compared to that of Encodings 1 and 2. Encoding 4 was the least appreciated. Users described it as unsuitable for large graphs and un-intuitive with a reduced interaction area (for edge selection). They also mentioned that it makes difficult the distinction between bends and crossings and the identification of communities. These results confirm the hypothesis that edge simplification and clutter reduction would increase matrix readability and that Encoding 4 would not be easy to use because of the lack of contrast and the crossing/bend distinction issue.

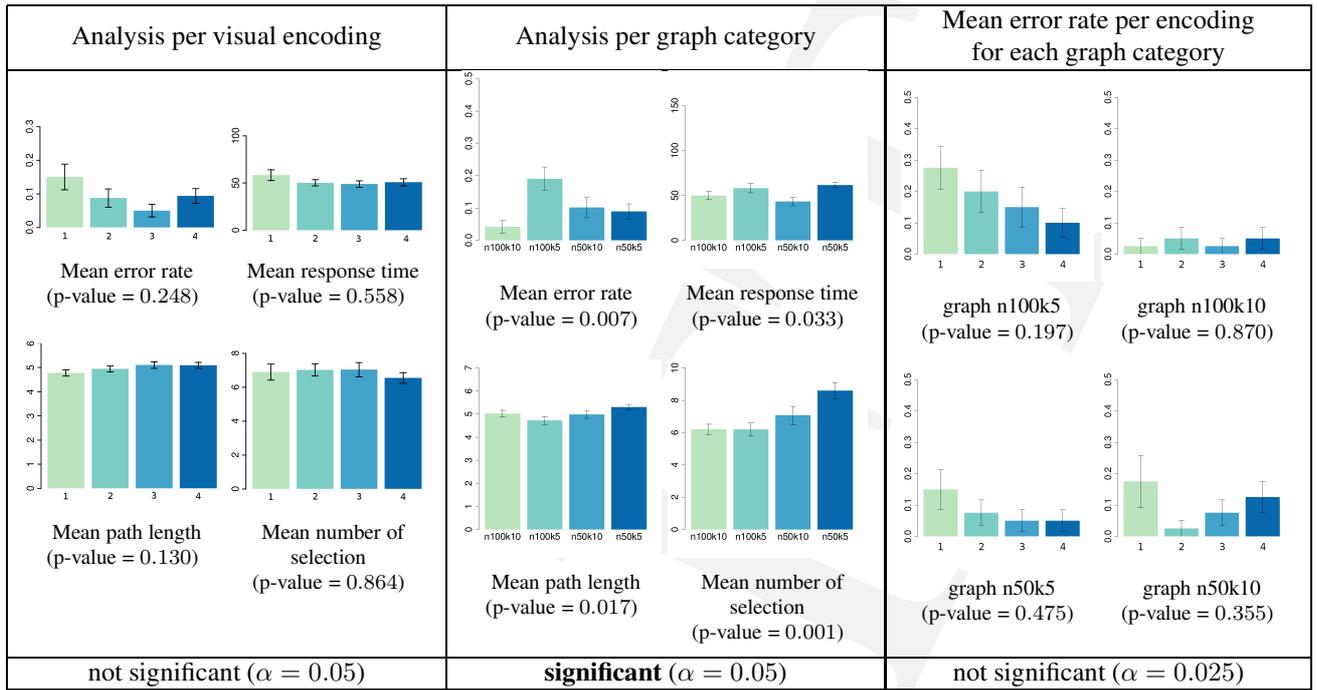


Figure 3: Statistical analyses performed in this experimental evaluation.

5 Discussion

An analysis of the quantitative data indicates no significant user performance differences between the four encodings even when taking graph categories into account. We suggest that this is due to a floor effect where the tasks were too easy for the given time period, and so any variation in the ease of understanding of the four encodings is not revealed in the data. This is evidenced by the mean error rates, which all lie in a narrow band between 0.07 and 0.16. These results are surprising, but some trends are noticeable. Simplifying visual encodings seems to reduce the error rate for a path-finding task. The visual encodings according to the mean error rates (Fig. 3) have the following order (highest to lowest): Encoding 1, Encoding 4, Encoding 2 and Encoding 3. These results tend to favour Encodings 3 and 2 rather than Encodings 1 and 4. Concerning the participant preferences, most of them ranked Encoding 3 as highest while Encoding 4 was least preferred. Users slightly prefer Encoding 2 over Encoding 1. This finding is in accordance with the preliminary hypotheses expressed in Section 3.2.

Encodings 3 and 4 can be observed as standard NLDs with node duplication's. Using curved lines complicates the distinction between edge bends and edge crossings in Encoding 4. We believe that an extreme simplification of the visual complexity makes the readability more difficult. Because the other recorded parameters do not vary, rea-

sonably simplifying the visual complexity ease matrix usage without affecting user performances. We therefore assume that the results of prior graph representation readability studies were not only related to MBD and NLD edge visual encodings but rather to node positioning/ordering and/or node duplication. The latter has already been used to untangle the network [6] and showed a strong impact on user performances.

These results must be put into perspective because we only evaluated user performances for a path-finding task in social networks. Other classical graph analysis tasks (*e.g.*, node degree, community size and node neighbourhood) should also be evaluated to understand how edge visual encoding can alter response times or error rates. Analyses using weighted connections might also confirm our trends, as weight can be mapped to both links and shapes on the reduced encoding (Encoding 3), whereas only shapes or links are available for traditional MBDs or NLDs.

6 Conclusion and future work

We have proposed a 3-step smooth transformation for building a node-link diagram from a matrix-based diagram by only modifying edge visual encoding. We also presented the results of an experimental evaluation that compares each step of the process on a path-finding task. This experiment attempted to remove the bias induced by node positioning/ordering when comparing NLDs and MBDs. In addition to these results, qualitative results revealed an

important user preference for the simplified encoding (Encoding 3). Although a lack of encoding-dependent variations existed in user performances during the experiment, the encoding type had a significant impact on the user experience in completing the tasks.

For future works, we plan to evaluate user performances in other tasks, such as neighbourhood identification or graph connectivity, to determine whether one of the proposed encodings may better facilitate particular tasks. Additionally, the benefits of such encodings in the case of multiscale visualisation would be interesting to evaluate. An evaluation that aims to compare DAGView with Encoding 3 should be a worthy thing to do.

Acknowledgement

This work has been carried out as part of SpeedData project supported by the French Investissement d'Avenir Program (Big Data - Cloud Computing topic) managed by Direction Gnrale des Entreprises (DGE) and BPIFrance.

References

- [1] D. Archambault, T. Munzner, and D. Auber. Topo-layout: Multilevel graph layout by topological features. *Trans. on Visualization and Computer Graphics*, 13(2):305–317, 2007.
- [2] D. Auber, C. Huet, A. Lambert, B. Renoust, A. Sal-laberry, and A. Saulnier. Gospermap: Using a gosper curve for laying out hierarchical data. *Trans. on Visualization and Computer Graphics*, 19(11):1820–1832, 2013.
- [3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
- [4] T. Dwyer, K. Marriott, and M. Wybrow. Topology preserving constrained graph layout. In *Graph Drawing*, LNCS, pages 230–241. Springer, 2009.
- [5] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4(2):114–135, 2005.
- [6] N. Henry, A. Bezerianos, and J. Fekete. Improving the Readability of Clustered Social Networks using Node Duplication. *Trans. on Visualization and Computer Graphics*, 14(6):1317–1324, 2008.
- [7] Nathalie Henry and Jean-Daniel Fekete. MatLink: Enhanced Matrix Visualization for Analyzing Social Networks. In *Human-Computer Interaction - INTER-ACT 2007*, volume 4663, pages 288–302, 2007.
- [8] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. NodeTriX: A Hybrid Visualization of Social Networks. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.
- [9] R. Keller, C. M. Eckert, and P. J. Clarkson. Matrices or Node-link Diagrams: Which Visual Representation is Better for Visualising Connectivity Models? *Information Visualization*, 5(1):62–76, 2006.
- [10] Yehuda Koren, Liran Carmel, and David Harel. Ace: A fast multiscale eigenvectors computation for drawing huge graphs. In *IEEE Symp. on Information Visualization (INFOVIS)*, pages 137–144. IEEE, 2002.
- [11] Yehuda Koren and David Harel. A multi-Scale Algorithm for the Minimum Linear Arrangement Problem. In *Graph-Theoretic Concepts in Computer Science*, pages 296–309. Springer, 2002.
- [12] EvgeniosM. Kornaropoulos and IoannisG. Tollis. DAGView: An Approach for Visualizing Large Graphs. In *Graph Drawing*, volume 7704 of LNCS, pages 499–510. Springer, 2013.
- [13] R. Kosara, T. J. Jankun-Kelly, and E. Chlan. IEEE InfoVis Contest: InfoVis goes to the movies, 2007.
- [14] Ulrich Lauther. Multipole-based force approximation revisited—a simple but fast implementation using a dynamized enclosing-circle-enhanced kd-tree. In *Graph drawing*, LNCS, pages 20–29. Springer, 2007.
- [15] C. Mueller, B. Martin, and A. Lumsdaine. A Comparison of Vertex Ordering Algorithms for Large Graph Visualization. In *6th Int. Asia-Pacific Symp. on Visualization APVIS'07*, pages 141–148, 2007.
- [16] Jordi Petit. Experiments on the Minimum Linear Arrangement Problem. *J. of Experimental Algorithmics (JEA)*, 8:2–3, 2003.
- [17] Helen Purchase. Which aesthetic has the greatest effect on human understanding? In *Graph Drawing*, LNCS, pages 248–261. Springer, 1997.
- [18] Helen C. Purchase. *Experimental Human-Computer Interaction - A Practical Guide with Visual Examples*. Cambridge University Press, 2012.
- [19] Eduardo Rodriguez-Tello, Jin-Kao Hao, and Jose Torres-Jimenez. An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. *Computers & Operations Research*, 35(10):3331–3346, 2008.

- [20] Zeqian Shen and Kwan-Liu Ma. Path Visualization for Adjacency Matrices. In *EuroVis07: Joint Eurographics - IEEE VGTC Symp. on Visualization*, pages 83–90, 2007.

DRAFT