Performance Evaluation of Tapered, Curved, and Animated Directed-Edge Representations in Node-Link Graphs

LaQuSo* Technical Report DH/PI/JDF/JvW/415

Danny Holten†, Petra Isenberg‡, Jean-Daniel Fekete‡, and Jarke J. van Wijk†

September 30, 2010

Abstract

We present the results of a study comparing three directed-edge representations in node-link diagrams. Node-link diagrams are probably the most popular type of graph representations; nodes are depicted as dots and links as straight or curved lines connecting the nodes. An arrowhead placed at the end point of a link is traditionally used to indicate edge direction. However, Holten and Van Wijk [13] showed that this arrow representation was not optimal. Their user study compared the performance (in terms of reading time and correctness) of various directed-edge representations. A tapered representation — wide at the start and narrow at the end — showed the best performance. This paper presents the results of a follow-up study comparing the performance of the tapered representation with two other representations: one using biased curvature and the other using animation to indicate edge direction. We tested the three representations using a more realistic setting than the original article. We used random small-world graphs generated with the Barabási-Albert model, we used the Fruchterman-Reingold algorithm to lay out the graphs, and we varied graph density and link length to measure their influences. Overall, our study shows that the tapered and animated representations perform significantly better than the biased-curvature representation, with no significant differences between tapered and animated except for medium-length representations where tapered was faster. The article presents detailed results and provides practical recommendations on the use of directed-edge representations.

1 Introduction

Graph-based data, which is used to represent a collection of elements (vertices) as well as the connections (edges) between these elements, has become ubiquitous these days, not least due to the increasing popularity of social networking sites on the Internet. Apart from social network graphs (or *social networks*), where vertices represent individuals and edges represent acquaintance, the following lists some additional examples of common graph-based data sources:

- *Traffic networks*, where vertices and edges are used to represent locations and traffic routes, respectively;
- *Computer networks*, where vertices and edges are used to represent PCs and network connections, respectively;
- *Scientific citation networks*, where vertices and edges are used to represent papers and citations between papers, respectively.

Edges often have an associated weight and direction. Edge weight might be used to indicate the strength, importance, or cost of an edge. Edge direction can be used to signify the directionality of what an edge represents, e.g., which paper holds a citation and which paper is being cited in case of a citation network graph.

When visualizing a graph, an often-used and intuitive visual representation is the node-link diagram (node-link graph), in which vertices are generally represented as circular nodes and edges as straight or curved links (lines). Edge weight is commonly depicted by varying the width of a link, while edge direction is generally depicted using an arrow representation, i.e., a line with an arrowhead at Node B in case of a directed edge running from Node A to Node B.

Holten and Van Wijk noted that the use of an arrow representation can lead to visual clutter because of arrowhead overlap, which can significantly hinder effortless and correct observation of edge direction [13]. Various approaches can be used to overcome this, such as adding user interaction, using a matrix instead of a node-link representation, improving the node-link layout, or improving the directed-edge representation. However, user interaction is not suited for statically depicted, e.g., printed, graphs; matrix representations have been shown to be less intuitive than node-link graphs [11, 15]; and node-link-layout improvement is generally tackled by *node*-placement optimization, while the arrowhead-clutter problem actually stems from a sub-optimal *edge* representation.

For this reason, we contend that focusing on the use of improved directed-edge representations is warranted. This was also done by Holten and Van Wijk [13], who performed a user study (henceforth called "the initial study") in which participants performed various tasks on a collection of graphs using different directed-edge representations (including the standard arrow) to investigate which of these representations lead to the best accuracy and reading times. They concluded that a tapered representation clearly outperformed the other representations. This representation generalizes the notion of an arrowhead by gradually varying the width of a link along its length, i.e., wide at the start and narrow at the end.

This paper presents a follow-up study that was inspired by the choices, experiments, and future work suggestions that were part of the initial study [13]. Among other things, the evaluation of animated directed-edge representations is suggested. Furthermore, one of the directed-edge representations that was evaluated uses clockwise curvature to indicate edge direction, which showed poor performance. We therefore chose to evaluate the performance of 1) tapered, the best representation according to the initial study, 2) an animated rep-

*LaQuSo – Laboratory for Quality Software, Eindhoven University of Technology

†Danny Holten (d.h.r.holten@tue.nl) and Jarke J. van Wijk (vanwijk@win.tue.nl) are with Eindhoven University of Technology

‡Petra Isenberg (petra.isenberg@inria.fr) and Jean-Daniel Fekete (jean-daniel.fekete@inria.fr) are with INRIA Unité de Recherche Saclay – Île-de-France

resentation, and 3) a representation that uses *biased* clockwise curvature — high curvature at the start, low curvature at the end — as proposed by Fekete *et al.* [9]. This curvature-based representation is regularly used and might perform better than the non-biased curvature-based representation evaluated in the initial study.

We furthermore evaluated how performance in terms of reading time and correctness is affected by three levels of graph density (sparse, medium density, and dense) as well as three different link-length classes (short, medium length, and long).

Finally, we used a class of graphs that is more representative of real-world graphs such as social networks, i.e., scale-free (small-world) graphs in which the degree distribution follows a power law [25, 3]; the graph model used in the initial study to generate the graphs was a non-scale-free model [22, 23]. We used the well-known Fruchterman-Reingold algorithm [10] to generate the graph layouts.

To evaluate the actual performance of the directed-edge representations in terms of reading time and correctness, we designed and ran an experiment in which participants performed path-readability trials, i.e., they had to answer whether or not there was a directed connection between a pair of nodes. We provide initial hypotheses regarding the performance of the evaluated directed-edge representations with respect to varying graph densities and link lengths, and subsequently compare these with the outcome of a statistical analysis of participants' reading times and correctness. Based on the results of this analysis, we provide practical recommendations on the use of directed-edge representations in the context of node-link graphs.

The remainder of the paper is organized as follows. Section 2 describes previous work on directed-edge visualization, the evaluation thereof, and the use of static, animated, and curvature-based representations to show direction. Section 3 gives an overview of graph generation and layout, the evaluated representations, and the selection of node pairs during trials. The experiment design and hypotheses are presented in Section 4, followed by Section 5, which presents the statistical analysis. Practical recommendations on directed-edge usage are provided in Section 6. Finally, Section 7 presents our conclusions and suggestions for future work.

2 Related Work

The following presents an overview of various directed-edge representations that have been proposed in the past, the majority of which are straight, static representations that have been used in the context of node-link graphs to depict directed connectivity. The use of edge representations that rely on curvature and animation is treated separately in Sections 2.2 and 2.3, respectively.

2.1 Straight, Static Directed-Edge Representations

Half-lines are used by Becker *et al.* [5] to depict directed edges; a half-line from Node A to B is a straight-line connection in which only the first half of the line is drawn. Although half-lines reduce the amount of display space used to show links, they unfortunately make it hard to determine where links end. Wong *et al.* [27] present their *GreenArrow* technique as a solution to the problem of how to balance the appearance of both a graph and its labels. The text label pertaining to a link (or one of its nodes) is drawn such that the text itself forms a tapered link between two nodes. This removes the need to explicitly visualize the edge using a line-based representation. A color-coded representation is used by Holten [12] to indicate edge direction. Direction is indicated as running from Node A to Node B by gradually changing the color from green (A) to red (B) along the length of a link, although different colors can be used as well.

The directed-edge representations that were evaluated (and partially designed) in the initial study [13] are the standard arrow, tapered, curved (non-biased, clockwise direction), green-to-red, light-to-dark, and dark-to-light. The following combinations were furthermore evaluated: tapered + curved, tapered + dark-to-light, curved + dark-to-light, and tapered + dark-to-light + curved. Apart from the recommendation of using a tapered representation in general because of its performance, is was also found that combining representations (at least the ones that were evaluated) did not result in significant performance gains. Nielsen *et al.* [18] note that they were partially inspired by these recommendations. For their *ABySS-Explorer* they use tapered, leaf-like shapes to visually depict the orientation of contigs, i.e., genome assemblies consisting of long contiguous sequences.

2.2 Curvature-Based Directed-Edge Representations

Edge representations based on curvature have been used before in the context of *Arc Diagrams* by Wattenberg [24] and *ArcTrees* by Neumann *et al.* Arcs, i.e., curved links, were used in *Arc Diagrams* to represent complex repetition patterns in string data and in *ArcTrees* to visualize hierarchical as well as non-hierarchical data relations. In case of such symmetric arcs, a curve's clockwise orientation is generally used to indicate direction. The aforementioned *GreenArrow* [27] technique by Wong *et al.* also uses curvature — albeit counter-clockwise — in addition to tapering and label-text orientation to indicate direction.

Instead of only using (counter-)clockwise orientation to indicate direction, Fekete *et al.* [9] add curvature bias to this. Their curved links are drawn as quadratic Bézier curves that vary the amount of curvature — high at the start and low at the end — to indicate direction (see Figure 2b). This is one of the representations that we evaluated since it is regularly used and we hypothesized that it might be a better alternative than the simple, non-biased clockwise-curvature-based representation evaluated in the initial study.

2.3 Animated Directed-Edge Representations

Most animated edge representations indicate edge direction based on the idea of having a dash pattern move in the direction of the link along the length of a link. An example is the effective use of animated dash-pattern textures by Wegenkittl *et al.* [26] to show the flow motion of trajectories within analytical dynamical systems.

To address node and link clutter in large and dense node-link graphs, Ware *et al.* [21] evaluated motion-based highlighting techniques to provide effective access to such graphs. One of the evaluated techniques was an edge representation

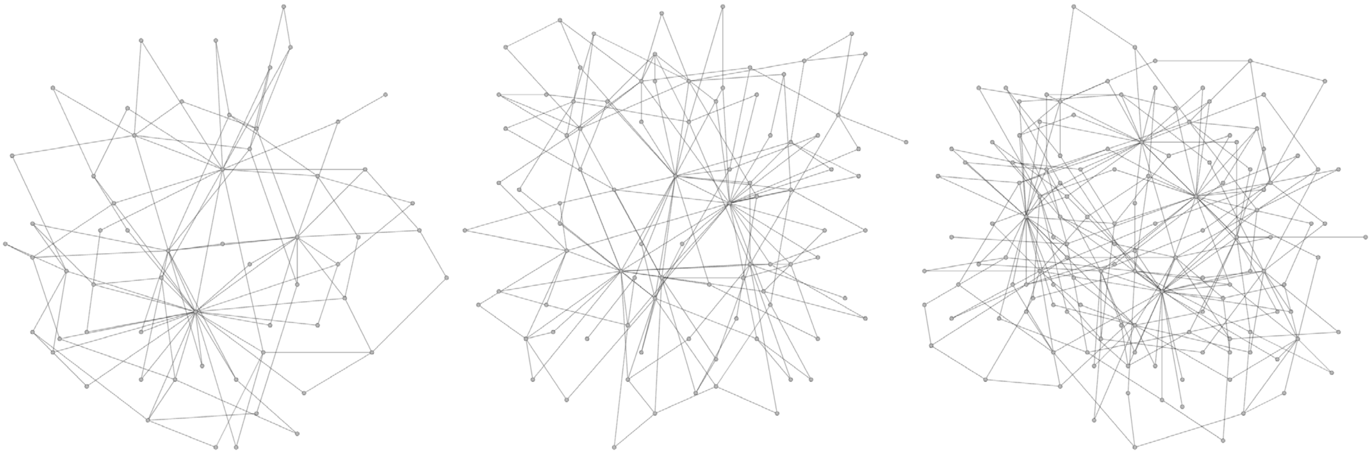

Figure 1: From left to right: a sparse (60 nodes), medium-density (90 nodes), and dense (135 nodes) graph as used during the experimental trials. The graphs were generated using the Barabási-Albert [3] scale-free model and laid out using the Fruchterman-Reingold [10] algorithm.

that uses animated sawtooth dash patterns radiating out from the source (start) node. For the path-readability and node-reachability tasks that participants performed, reading time and error rate were generally significantly lower when using animated-link highlighting than when using no highlighting. Ware *et al.* therefore argue that motion-based highlighting can be valuable in applications that require users to understand large graphs.

Bartram *et al.* [4] showed the potential of using animated causal overlays, e.g., animated links, on top of causality-depicting visualizations such as causal graphs. The idea was inspired by the fact that perceptual psychology showed that causality perception is a low-level visual event derived from certain types of motion.

Blaas *et al.* [6] present a spline-based way to smoothly visualize higher-order state transitions for the exploration of state sequences in large time-series. As noted by Blaas *et al.*, arrowheads distort the perception of a continuous spline. They therefore opted for a texture-based approach that uses animated dash patterns to visualize edge direction. They are furthermore planning to continue their investigation into the visualization of directed edges using animated textures.

Animated links have also recently been added to the well-known *Tulip* graph visualization library [2]. The dash patterns used by *Tulip* move along a straight line from start to end and are made up of repeating "greater than" ("⟩") symbols depicted using alternating colors.

2.4 Evaluation

Apart from the initial study [13] that inspired our work, only few user studies have been performed to quantify how directed-edge representations perform in the context of node-link graphs. One of these is the user study performed by Wong *et al.* for their *GreenArrow* approach. However, their focus was on how to balance the appearance of a graph and its labels, not on determining how well their approach works as a technique for depicting directed edges in comparison with other directed-edge representations.

An additional motivation for our follow-up evaluation of the performance of directed-edge representations is the remark of Kosara *et al.* [16], who note that visualization as currently practiced is mostly a craft and evaluation is often only performed informally. Because of this, user studies in visualization should be encouraged. This is further supported by North [19], who also states that there are too many informal usability studies that only indicate whether or not a small number of users *like* a certain visualization technique. Visualization evaluation should seek to quantifiably determine how well visualizations perform, and one way of doing this is to perform controlled user studies.

3 Graphs and Directed-Edge Representations

This section provides details on the choices that were made with regard to the generative graph model, the graph layout, the directed-edge representations, and the way in which node pairs were chosen during trials to ensure controlled testing of different link lengths.

3.1 Graph Model

To ensure the availability of enough different graphs with similar statistical properties for all of the participants and for all of the combinations of graph density, edge representation, and link length, we chose to generate random graphs using a graph generation model.

The graph model used in the initial study was proposed by Ware *et al.* [22, 23]. It generates graphs as follows: "*For each graph node, form a directed edge to one or two other nodes, randomly selected, so that a single connection occurs* $p\%$ *of the time and two connections occur* $(100 - p\%)$ *of the time*," where p controls the density. This model is more suited for real-world graphs than the model by Erdős *et al.* [8], who define a random graph as $G(n, p)$ in which each of the $\binom{n}{2}$ possible edges occurs with probability p. However, the node-degree distribution of the Ware-*et-al.*-graphs does not follow a power law (is not scale-free), i.e., the fraction $P(k)$ of nodes having k connections to other nodes is not modeled by $P(k) \sim k^{-\gamma}$, typically with $2 < \gamma < 3$. Many empirically observed networks, however, *do* appear to be scale-free, including social, citation, and protein networks.

We therefore chose a graph generation model that produces scale-free (small-world) graphs, i.e., graphs with a power law distribution, such as the Barabási-Albert (BA) [3] or the Watts-Strogatz [25] model. We used the Network Workbench [20], a network analysis, modeling, and visualization toolkit, to auto-

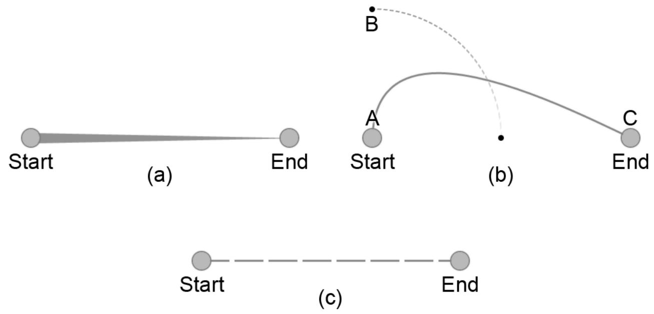

Figure 2: The directed-edge representations that were evaluated: (a) *tapered* from wide (start) to narrow (end), (b) *biased* clockwise curvature from high (start) to low (end) curvature, and (c) *animated*, in which a pattern moves from start to end along a straight line.

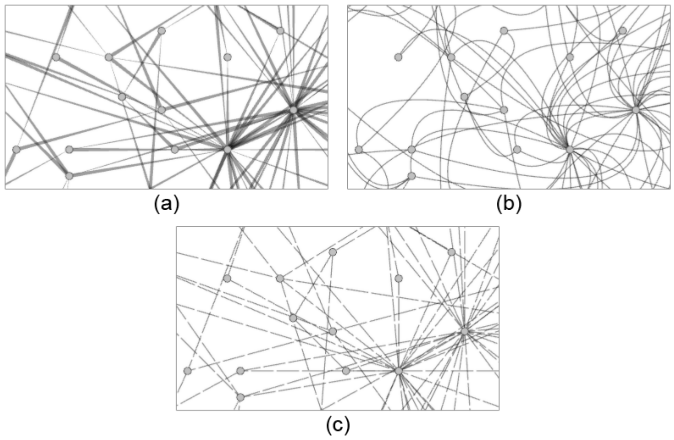

Figure 3: Identical regions of a medium-density graph shown using a (a) tapered, (b) biased, and (c) animated representation.

matically generate a set of random graphs with varying densities according to the BA model. After initial pilot runs that colloquially assessed the difficulty experienced by people when deciding if there is a connection between a pair of nodes, we settled for sparse, medium-density, and dense graphs, with 60, 90 and 135 nodes, respectively ($60 \times 1.5 = 90 \times 1.5 = 135$). A couple of representative example graphs are shown in Figure 1.

3.2 Graph Layout

There are many node-link-graph layout models available, the majority of which make use of force-directed node placement algorithms (or adaptations thereof), such as Eades' spring-embedder model [7], the Kamada-Kawai [14] model, or the Fruchterman-Reingold [10] model.

Due to its real-world popularity, ease of implementation, and overall good results for graphs up to a couple hundred nodes, we chose the Fruchterman-Reingold (FR) algorithm to lay out our graphs. All layouts were generated by a *JUNG*-based [17] FR implementation as provided by the *GUESS* graph exploration system [1] from within the Network Workbench (see Figure 1).

3.3 Directed-Edge Representations

Many feasible directed-edge representations remain to be comparatively evaluated. We chose to evaluate the performance of three representations that seemed particularly promising based on previous work. Specifically we evaluated 1) tapered, the best representation according to the initial study, 2) an animated representation, and 3) a representation that uses *biased* clockwise curvature as proposed by Fekete *et al.* [9]. This biased-curvature-based representation might be a better alternative than the non-biased representation evaluated in the initial study. Based on information directly provided by Holten, Van Wijk, and Fekete, we modeled all representations to adhere to their original design and to be similar to the representations used in the initial study.

We used a 23" LCD display with a 16:10 width:height ratio and a native resolution of 1680×1050 pixels to represent all of our graphs at 4× anti-aliasing on a white background. All visualizations were generated within our OpenGL-based evaluation tool, which was written in Borland Delphi. Given the display diagonal and the resolution, our display provided approximately 86 DPI.

The tapered representation was similar to the one used in the initial study: 4.31px (0.05") wide at the start, 0.43px (0.005") wide at the end, and drawn in black at an opacity of 35% (see Figure 2a).

Based on information directly provided by Fekete, we modeled the biased-curvature representation as a quadratic Bézier curve defined by the points A, B, and C as shown in Figure 2b; Point B is generated by 90° counterclockwise rotation of the midpoint of A and C around A. The curve is drawn at 1.29px (0.015") wide in black at an opacity of 40%. We used a higher opacity than tapered to compensate for the fact that — due to its thinner design — it covers less display area.

Finally, the animated representation used a moving dash pattern with a cycle length of 30px (0.35"), 90% of which, i.e., 27px (0.31"), was occupied by a line, and 10% of which, i.e., 3px (0.04"), was occupied by a gap (see Figure 2c). The pattern moved along a straight line from start to end at a speed of 15px (0.17") per second. Each animated link used a different, randomly assigned phase to prevent distracting "bursts" of activity caused by multiple incoming/outgoing links around a node. The animated links were also drawn in black at an opacity of 40% using a line width of 1.29px (0.015").

Although our texture-based animation model is capable of generating more complex animated links using gradient-based patterns with smoothly varying colors and opacity, we chose the aforementioned settings to prevent the animated representation from claiming visual attributes such as (varying) color, opacity, and line width that can be used to encode other attributes in addition to directionality. The exact settings for our candidate representation were determined using pilot runs that assessed the length, density, and speed of the dash patterns that the majority of the participants felt comfortable with. Figure 3 shows an identical region of a medium-density graph using each of the three directed-edge representations.

3.4 Link-Length Selection

For each path-readability trial (see Section 4.4 for a detailed task description), a pair of start and end nodes A and B was selected between which a directed link L might be present. Since difficulties caused by visual clutter and overlap occur more often in the central region of a force-directed graph layout, we ensured the selection of node pairs from the central part of the graph layout. This was done by calculating a bounding circle for each graph layout and adding a normalized, cut-off 2D Gaussian on top of this with a value of 1

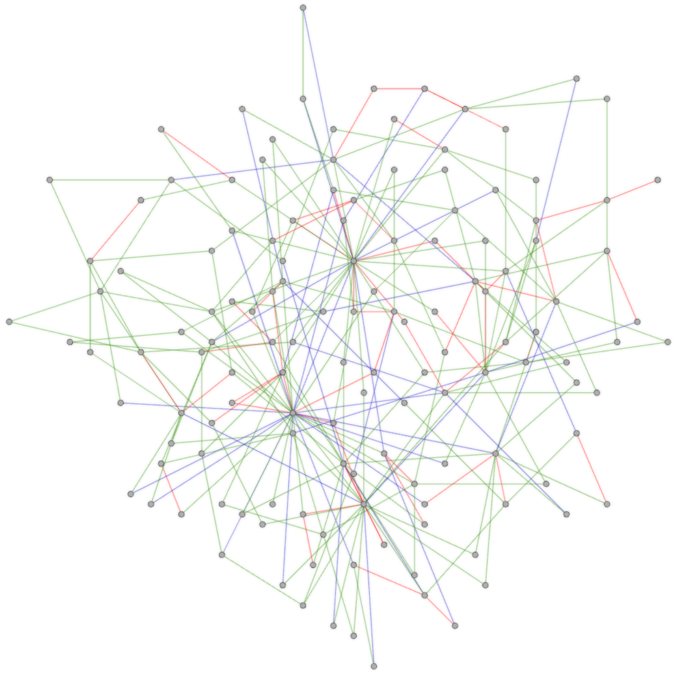

Figure 4: A medium-density, undirected graph in which representative links for each link-length class are highlighted using red (short), green (medium length), and blue (long).

at the center and a value of 0 (at 3σ) at the circular graph boundary. This value, denoted as $P_{center}(A, B) \in [0, 1]$, represents the probability of a randomly chosen node pair consisting of nodes A and B getting accepted based on the position of the node-pair midpoint within the bounding circle. More specifically, we implemented this by randomly picking a node pair with node positions A_{pos} and B_{pos} until a node pair was accepted. The probability of acceptance $P_{center}(A, B)$ depended on $d = \frac{A_{pos}+B_{pos}}{2} - C$, where C is the center of the graph bounding circle.

To select a short, medium-length, or long link for a trial (see Figure 4 for examples), the minimum- and maximum-length links L_{min} and L_{max} were first determined. Based on the range $D = L_{max} - L_{min}$, the links were sorted from short to long and divided into three bins, i.e., $B_{short} = [L_{min}, L_{min} + \frac{1}{3}D\rangle$, $B_{medium} = [L_{min} + \frac{1}{3}D, L_{min} + \frac{2}{3}D\rangle$, and $B_{long} = [L_{min} + \frac{2}{3}D, L_{max}]$. To determine the probability of a randomly chosen link from Bin B (a certain length class) getting accepted, a 1D normalized, cut-off Gaussian was centered on the median-length link in B, having a value of 1 at the center and 0 (at 3σ) at the bin boundaries. The probability of a randomly chosen link in Bin B getting accepted is $P_B(L) \in [0, 1]$.

Taking the position of a link with respect to the graph center into account, the combined probability of a randomly chosen link in Bin B getting accepted now becomes $P_{center}(L_A, L_B) \cdot P_B(L) = P(L) \in [0, 1]$, where L_A and L_B are the start and end nodes of Link L, respectively.

4 Experiment – Comparing Directed-Edge Representations

As previously stated, the goal of our experiment was to extend previous work on the readability of directed edges. We chose to evaluate tapered, the best directed-edge representation based on the results of an initial study [13], and added two new ones: animation and biased curvature (henceforth called "biased"). In addition, we tested three graph densities and three link lengths.

4.1 Hypotheses

Based on the results of previous work, we hypothesized that overall, tapered links would outperform the other two in this experiment as well (*H1*). However, we hypothesized that interaction effects would be present for the different graph densities and link lengths (*H2*). In terms of interaction effects we specifically had the following hypotheses:

H3: For sparse graphs, tapered will outperform the other two representations due to minimal overlap and its strong indication of direction;

H4: For dense graphs, tapered will perform worse than the other two representations due to its increased use of "ink" (display area);

H5: For long links, biased will perform significantly worse than the other two representations due to increased edge overlap.

We were unsure how the animated links would perform. Animation has a strong perceptual focus and may lead to increased visual clutter of the graph. On the other hand, the strong perceptual cues provided by animation may lead to an increased ability to follow links. We were further unsure how tapered and animated links would compare for long edges.

4.2 Design

We used a repeated-measures design with the following within-subjects independent variables: *edge representation* (tapered, biased, animated), *graph density* (sparse, medium density, dense), and *link length* (short, medium length, long). Each participant performed 10 repetitions of trials, each repetition containing 27 trials. The order of combinations of graph density, link length, and edge representation was randomized per repetition of trials. Before each repetition of trials, participants were allowed to rest and could continue to the next repetition whenever they were ready. Experimental sessions lasted about 30 minutes including training. In summary, the design included:

3	edge representations	×
3	graph densities	×
3	link lengths	=
27	trials per repetition	×
10	repetitions per participant	=
270	trials per participant	×
27	participants	=
7,290	**trials in total**	

The dependent variables that we measured are reading time and correctness on a per-trial basis. In addition, we analyzed qualitative feedback from the post-study questionnaire.

4.3 Participants and Procedure

Twenty-seven participants (15 male, 12 female) were recruited from two research institutions. They ranged in age from 22

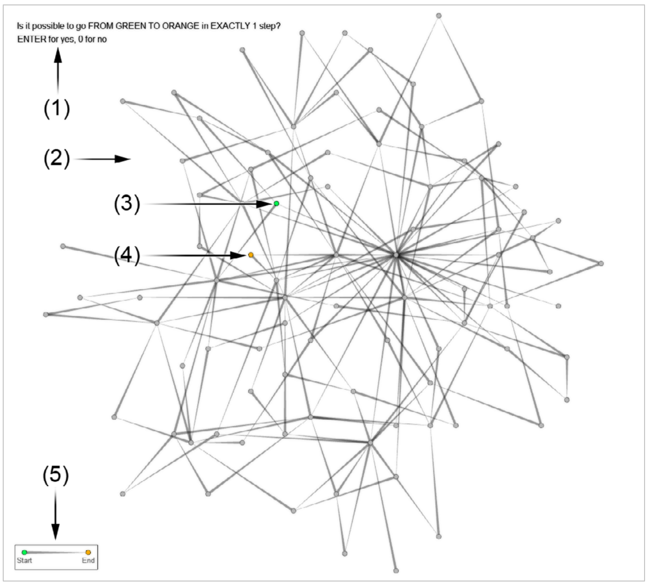

Figure 5: Example view of a trial as presented to participants. Task instructions resided in the top-left corner (1), the graph in the screen center (2), and a legend on how to read the directed-edge representation in the lower-left corner (5). Test nodes were highlighted in green for the start node (3) and orange for the end node (4).

to 58 years (median 30 years). All participants had normal or corrected-to-normal vision and were tested for color-blindness. Seven participants reported looking at node-link graphs at least weekly, 13 participants reported monthly or yearly exposure, and six reported to be unfamiliar with node-link-graph representations. Eleven participants were students and 16 were non-students with varying technical as well as non-technical occupations, e.g., nurse, administrator, or researcher. Participants were not paid for their involvement in the study.

Participants were seated in front of a Dell LCD display at a viewing distance of approximately 50cm; subsection 3.3 includes further details on the display settings. Participants were first given a short introduction to node-link graphs and the different types of edge representations used in the study. The experimenter then described the task, how to step through the trials, and how to record an answer. Participants answered by pressing the "0" (no directed connection present) and "Enter" (directed connection present) keys on the keyboard and they pressed the "Space" key to start a new repetition of trials. Participants first conducted 50 practice trials to ensure that they were familiar with how to read the edge representations and with the experimental setup and procedure. After the practice trials, participants were asked if they had any further questions. If not, they continued to the experiment. After the experiment, participants filled out a post-session questionnaire to elicit qualitative feedback and demographic information.

4.4 Tasks

We used a path-readability task in which participants had to answer the following question throughout the study: *"Is it possible to go from the green node to the orange node in exactly one step?"* This task was chosen as it tests local readability of node connections and gives an indication of how well the chosen edge representation allows participants to infer directionality. Participants were instructed to be both as accurate and fast as possible. The correctness of their answer was shown to participants only during practice trials (immediately after a trial) and not during the actual experiment.

Each trial was shown to participants full-screen in three stages. During the first stage participants saw an empty white screen for 400ms. During the second stage we showed two randomly selected nodes A (green) and B (orange) to the participants for 600ms. This stage was introduced to ensure that participants did not spend time finding the two nodes in the graph. Finally, in stage three the complete graph (with A and B in the same positions) was displayed and timing for the trial was started.

A legend for the currently used edge representation was displayed in the lower-left corner of the screen (see Figure 5). Participants were presented with a new graph for each trial and Nodes A and B were chosen such that there was a 50% chance of a connection being present; in 50% of the cases in which a connection was present, the connection also had the correct direction, i.e., there was a 25% overall chance of a connection being present and having the correct direction (from A to B). The chance of a bidirectional connection being present was not explicitly controlled for. The randomly selected graphs were generated and laid out as discussed in Sections 3.1 and 3.2 and link lengths were selected as discussed in Section 3.4.

5 Results

The completion times collected during the experiment were first log-transformed to comply with the normality assumption of the data analysis. Then time was analyzed using a repeated-measures ANOVA. Timing data was recorded and analyzed at millisecond scale; average task times are reported here at second scale. The error data was analyzed using the non-parametric Friedman analysis of variance by ranks and Wilcoxon signed rank tests as the error data did not conform to the normality assumption.

5.1 Overall Effect of Edge Representation

There was a significant effect of edge representation on overall task completion time ($F(2, 52) = 75.315; p < .001$) with mean times diminishing from 1.81s ($SD = 1.48$) for tapered, to 1.92s ($SD = 1.2$) for animated, and 2.72s ($SD = 2.25$) for biased (see Figure 6). Post-hoc pairwise comparisons only showed a significant difference between biased and the two other representation with $p < .001$ in both cases.

Overall, participants made errors in only 6% of the trials.

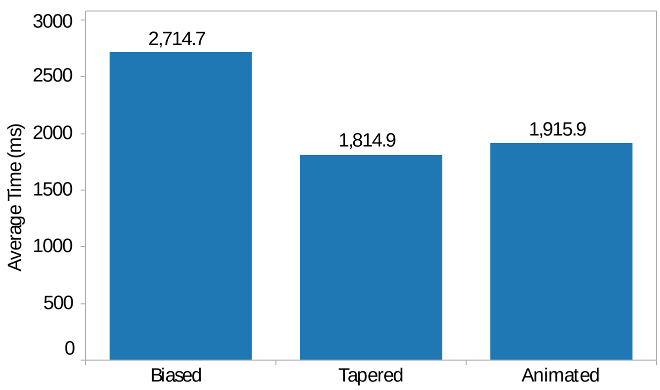

Figure 6: Average completion time per representation.

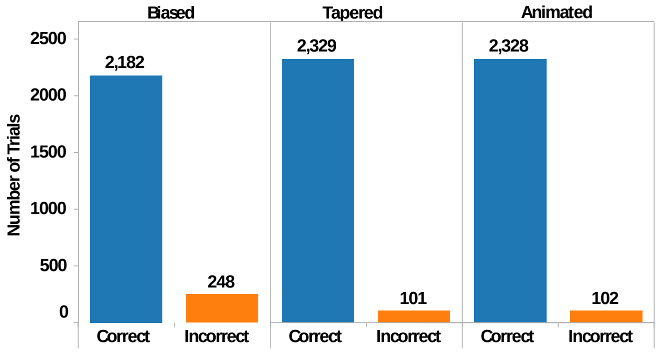

Figure 7: Number of (in)correct trials per representation.

However, a Friedman analysis showed a significant effect of edge representation on the correctness of the trials ($\chi^2(2) = 29.16; p < .001$) with errors increasing from 101 for tapered, to 102 for animated, and 248 for biased links out of 2,430 trials per edge representation (see Figure 7). Wilcoxon signed rank tests again only showed a significant effect between biased and the two other representations, with $Z = -4.12$ for comparison to animated, $Z = -4.05$ for comparison to tapered, and $p < .001$ in both cases. We, thus, confirmed our first hypothesis (H1), that tapered would perform well overall, but found animated to be a competitive alternative.

5.2 Effects of Graph Density

Tests showed a significant effect of graph density on task completion time ($F(2, 52) = 54.56; p < .001$) with average times of 1.97s ($SD = 1.24$) for sparse graphs, 2.13s ($SD = 1.39$) for medium-density graphs, and 2.34s ($SD = 1.59$) for dense graphs. Post-hoc pairwise comparisons showed a significant effect between all graph densities with $p < .001$ each. Error analysis showed a significant effect between the three densities ($\chi^2(2) = 10.65; p < .005$). Post-hoc pairwise comparisons showed a significant difference for medium-density and dense $p < .002$ as well as sparse and dense graphs $p < .001$. Although the difference between sparse and medium-density graphs was not significant, the other differences indicate that there is a monotonic increase in error. This analysis confirms that we chose an appropriate increase in density for the graphs.

Next, we looked at the effect of graph density with respect to the three different edge representations. The analysis of task completion time showed a significant effect for $edge\ representation \times graph\ density$ ($F(4, 104) = 3.535; p < .002$). Post-hoc comparisons showed a significant difference between biased and tapered as well as biased and animated representations for all three graph densities ($p < .001$ in each case). No significant difference was observed between tapered and animated links for any of the graph densities. Figure 8 gives an overview of the average trial times observed for each edge representation and graph density.

During the analysis of trial errors we observed the same pattern. An overall significant difference for $edge\ representation \times graph\ density$ ($\chi^2(8) = 46.76; p < .001$) emerged. Further cross-comparisons per type of graph density showed significant differences between the biased representation and the two other ones. Similar to the analysis of trial time, no significant difference was observed between tapered and animated links. Table 1 gives an overview of the respective significance values.

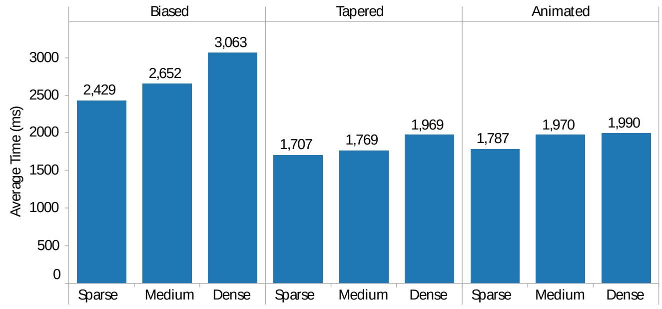

Figure 8: Average trial time per representation by graph density.

	Sparse		Medium Density		Dense	
	A-B	T-B	A-B	T-B	A-B	T-B
Z	-3.0	-2.61	-3.06	-3.05	-3.67	-3.36
p	$< .003$	$< .009$	$< .002$	$< .002$	$< .001$	$< .001$

Table 1: Significant differences for cross comparisons according to trial error for animated (A), biased (B), and tapered (T) links in sparse, medium-density, and dense graphs.

Thus, we partially confirmed our third hypothesis (H3). Tapered performed well in sparse graphs but there was no significant difference in trial time or error to animated links. We did not, however, confirm H4. Against our initial hypothesis, tapered links performed well for dense graphs as well, and again we did not show a difference to animated links. Biased links performed the worst according to both time and error.

5.3 Effects of Link Length

Tests showed a significant effect of link length on task completion time ($F(2, 52) = 105.26; p < .001$) with average task completion times of 1.95s ($SD = 1.37$) for short links, 2.09s ($SD = 1.4$) for medium-length links, and 2.4s ($SD = 1.43$) for long links. Post-hoc pairwise comparisons showed a significant effect between each of the three lengths with $p < .001$ in each case. No significant effect was observed between link lengths in terms of correctness. Participants were 94.2% correct with short links, 94.4% correct for medium-length links, and 92.9% correct with long links. Although the differences in error rate were not significant for the different link lengths, the timing information shows that participants spent significantly longer for trials with longer links indicating an increase in difficulty.

Next we analyzed interaction effects for $edge\ representation \times link\ length$. The analysis of task time showed a significant difference between the three lengths and representation types ($F(4, 104) = 4.48; p < .002$). Post-hoc comparisons showed significant differences between the biased representation and the two other ones with $p < .001$ in each case. For short and long links the data exhibited no significant difference between tapered and animated links. Only for medium-length links, a significant difference ($p < .022$) occurred for those two representations with tapered being significantly faster at 1.69s on average than animated at 1.86s on average. Figure 9 gives an overview of the average trial time for each edge representation separated by the three link lengths.

The Friedman analysis of variance showed a significant effect of error rate for $edge\ representation \times link\ length$ ($\chi^2(8) = 57.46; p < .001$). Cross-comparisons by link

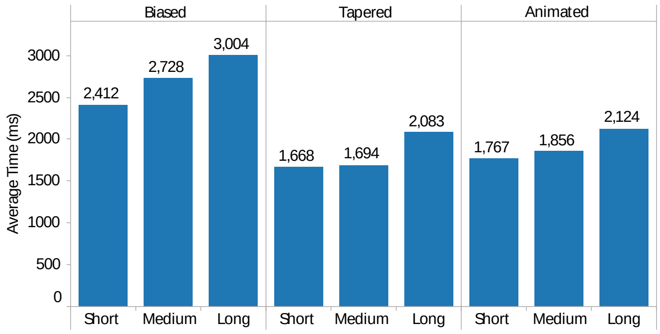

Figure 9: Average trial time per representation by link length.

	Medium Length		Long	
	A-B	T-B	A-B	T-B
Z	-3.97	-3.59	-4.08	-3.89
p	$< .001$	$< .001$	$< .001$	$< .001$

Table 2: Significant differences for cross comparisons according to trial error for animated (A), biased (B), and tapered (T) links with medium-length and long links.

lengths showed no significant difference in errors among the three representations for short links. For medium-length and long links, the data exhibited a significant effect between the animated and biased as well as biased and tapered representations, but not between tapered and animated. Table 2 gives an overview of the relevant significance scores for these tests.

In summary, we saw interaction effects for edge representation for both density and length, confirming our second hypothesis (H2). We could confirm H5 as biased links performed significantly worse than the others for long links. However, it should be noted that it in fact performed significantly worse than the other two techniques for all link lengths.

5.4 Qualitative Feedback

The post-session questionnaire elicited feedback from participants on their subjective preferences and ratings for the three edge representations.

5.4.1 General Preference

Two participants chose the biased representation as their overall favorite, eleven chose the animated type, and the remaining fourteen participants preferred the tapered representation. Figure 10 gives an overview of participants' preferences. The reasons participants gave for choosing animated or tapered as opposed to the biased representation were fairly similar. Participants found them to be "intuitive" (two responses each for tapered (2-T) and animated (2-A)), preferred the direct connection in comparison to the biased representation (3-T; 5-A), named them "easier to follow" (4-T, 1-A), faster (1-T, 1-A), and thought that those two representations produced minimal clutter (2-T, 3-A). Not all participants produced an answer to why they preferred a specific representation and some named several reasons, therefore the answer numbers to this question do not represent the total number of participants. Some participants further explained why they preferred a specific representation. Participants saw an advantage of the tapered links in that the varying width allowed them to more easily infer whether a node intersecting a link was a start or end node. Participants also reported that biased links were difficult to follow

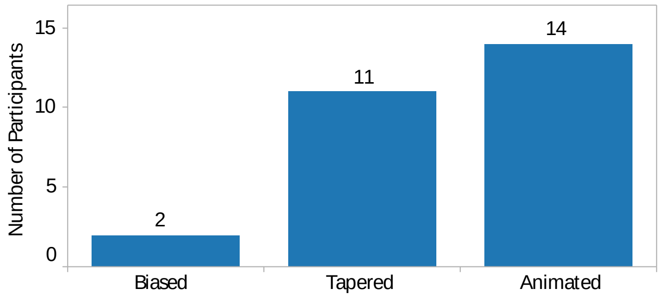

Figure 10: Participant-preferred representation type.

in dense situations as the angle of the curve was hard to guess and had to be searched for. Participants further stated that whether a connection was present or not was easy to see for both tapered and animated but not for biased links. One participant was also concerned that animated links would imply variations or flow over time (such as the movement of people between cities), which may not always properly represent the semantics of a link in a graph.

5.4.2 Preference by Graph Density and Link Length

Participants were further asked to rank the techniques for each graph density they encountered. Overall, biased was the first choice for each graph density only five times or less. Animated and tapered links were chosen equally often for dense graphs. For medium-density graphs, animated links were the number one choice of most participants: they were chosen fourteen times compared to ten times for tapered links. Participants showed a clear preference for tapered links for sparse graphs, however: they were ranked first sixteen times compared to 7 times for animated links. Figure 11 gives an overview of which edge representation participants rated first for each graph density.

Furthermore, participants were asked to rank the edge representations for the two extreme link lengths: short and long links. As the experiment randomly presented participants with

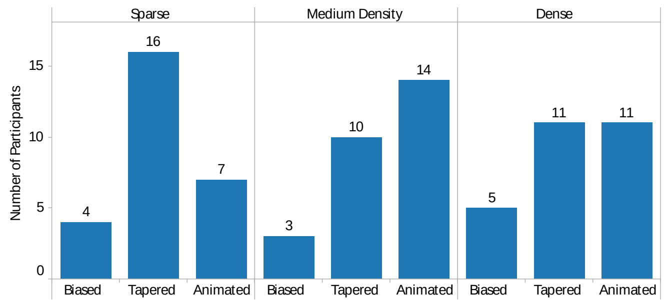

Figure 11: Participants' first choice of technique by graph density.

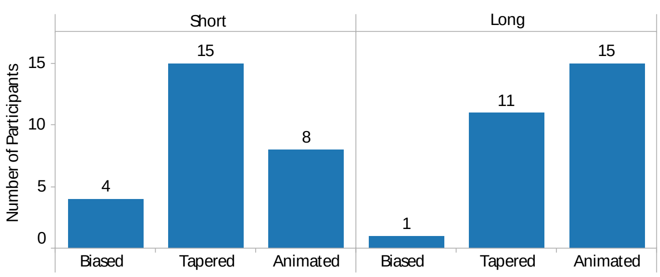

Figure 12: Participants' first choice of technique by link length.

	The technique allowed me to be…			
	…correct		…fast	
	Mean	Median	Mean	Median
Biased	3.96	4 (no opinion)	2.93	2 (disagree)
Tapered	5.67	6 (agree)	5.52	6 (agree)
Animated	5.3	5 (somewhat agree)	5.41	6 (agree)

Table 3: Results of the Likert-scale questions: "The technique allowed me to be (fast/correct)."

different link lengths, medium-length links were excluded from this question. For short links, the tapered representation was the most preferred, while animated links were the most preferred technique for long links (Figure 12). Together with the analysis of graph density, participants tended to prefer tapered links for sparse graphs and short, animated links for longer links and medium-density graphs; they were equally effective with both in the quantitative evaluation.

5.4.3 Self-Rated Speed and Correctness per Representation

To capture possible differences in actual and perceived efficiency and effectiveness with a given edge representation, participants were asked to rank their perceived correctness and speed on a 7-point Likert scale. The scale ranged from 1: strongly disagree to 7: strongly agree. Table 3 summarizes the answers to these questions. Reflecting the results of the quantitative measurements, participants were least sure about their performance with the biased representation. The answers to perceived speed and correctness for both animated and tapered were again significantly different to the biased representation ($p < .001$ for speed and $p < .007$ for correctness) but not significantly different between animated and tapered.

5.4.4 Aesthetics

Participants were further asked to rate the three representations on their ability to produce "nice-looking graphs." Participants rated the biased representation with a median of 4 (no opinion, mean = 4.15), tapered with a median of 5 (somewhat agree, mean = 5.22), and animated also with a median of 5 (mean = 5.15). While the difference in rank of the biased link to the two other representations was smaller for this question compared to previous ones, there was still a significant difference to the two other techniques with $p < .022; Z = -2.29$ compared to tapered and $p < .034; Z = -2.12$ compared to animated. No significant difference between the ratings for tapered and animated links was observed.

6 Discussion

The quantitative results are consistent with the self-rated perception of the participants. From the experiment, the biased representation is always less effective than the two others and is also the least preferred. Therefore, it should probably not be used if any of the two others is an option.

The animated representation is effective, even competing with the tapered representation on almost all the conditions except for medium-length links. The reason why this condition is different from the others needs further study; so far, we have no explanation. Since tapered links were more effective than all the others tested in the initial study [13], we expect animated links to also be more effective.

We can now choose the link representation of a node-link diagram in a more rational way than before. There are still tradeoffs to keep in mind. One tradeoff concerns the visual attributes that can be used along with the representation. For example, tapered links require transparency and a varying thickness to be readable. If changing the link thickness or transparency according to an edge attribute is important, tapered links cannot be used or could become less effective. Animated links, on the other hand, can support varying thickness and variation of transparency but cannot be used on paper or any static medium. Further research is also required to test different types of animation patterns for screen use. We proposed one that tested very well in comparison to tapered but many different patterns are possible and may not support path readability in the same way (they may be better or worse).

As another tradeoff, animated links are more complex to program and require more graphics performance to work smoothly on large graphs. Finally, edge routing or bundling cannot be used with biased links because the shape of the curve is constrained by the routing itself.

To summarize: tapered links were always good and should be preferred by default, unless link thickness or transparency should vary. Animated links can be used in that case. Finally, tapered and animated links can likely be combined or used to differentiate categorical attributes of links with up to three values: tapered, animated and tapered-animated.

7 Conclusion

In this article, we have reported on a controlled experiment comparing the readability of three directed-edge representations for node-link diagrams: tapered, biased, and animated. The study used graphs randomly generated using the Barabàsi-Albert model and laid out using the Fruchterman-Reingold algorithm. The factors were the three edge representations, the density of the graphs (sparse, medium density, and dense) and the link lengths (short, medium length, and long). We tested one low-level connectivity task: showing two nodes, asking if the first node was connected to the second. We collected the time to complete and the number of errors, as well as a user questionnaire eliciting subjective feedback from participants.

The study showed that tapered and animated links were always faster to read and more accurate than the biased-link representation. Tapered was significantly faster than animated for only one condition: medium-length links. The questionnaire reported consistent conclusions regarding user preferences and self-assessment.

From this study and the initial one, we can conclude that the best directed-edge representation for the readability task is tapered, followed by animated. We also provide advice and recommendations in the discussion section for choosing a representation according to various constraints and tradeoffs.

This study revealed that animated links were effective at depicting directed edges, matching the best technique in most conditions. The participants provided interesting reasons explaining why tapered links are still better in some cases: when a link crosses a node but does not end, the thickness of the tapered link allows one to resolve the ambiguity while animated links remain ambiguous.

7.1 Future Work

The design space of animated links is large and we have only tried a very simples design so far. Given the user feedback, we need to improve the design of animated links to disambiguate between crossings and end points. We should also try to find the optimal parameters in terms of dash sizes and speed, or vary the shape of the moving part using glyphs or other patterns.

Now that we have evaluation results from several types of link representations, we can also study how to combine them either as visual attributes for categorical attributes or to visualize overlapping edges that have identical start and end points. Another possibility is to combine different representations in dense or sparse graph regions or to differentiate short and long links.

We have only tested edge representations for one task. There are, or course, several other tasks involved in graph visualization and the results of the representations we have tested are likely to be different for some of them. For example, tapered and biased links provide an easy assessment of the overall link length (the thickness variation of the tapered link, the derivative of the curve of the biased link). Higher-level tasks could also benefit from different edge representations but have not yet been formally tested.

8 Acknowledgements

We would like to thank all of the participants in our study for their time. This project is partly funded by the Netherlands Organization for Scientific Research (NWO) View programme under research grant no. 638.100.502 (*Expression of Interest Project*).

References

[1] E. Adar. GUESS: A Language and Interface for Graph Exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*, pages 791–800, 2006.

[2] D. Auber. Tulip: A Huge Graph Visualisation Framework. In P. Mutzel and M. Jünger, editors, *Graph Drawing Software*, pages 105–126. Springer-Verlag, 2003.

[3] A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286:509–512, 1999.

[4] L. Bartram and M. Yao. Animating Causal Overlays. *Computer Graphics Forum (Proceedings of EUROVIS)*, 27(3):751–758, 2008.

[5] R. A. Becker, S. G. Eick, and A. R. Wilks. Visualizing Network Data. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 1:16–28, 1995.

[6] J. Blaas, C. Botha, E. Grundy, M. Jones, R. Laramee, and F. Post. Smooth Graphs for Visual Exploration of Higher-Order State Transitions. *IEEE Transactions on Visualization and Computer Graphics (TVCG; Proceedings of INFOVIS'09)*, 15(6):969–976, 2009.

[7] P. A. Eades. A Heuristic for Graph Drawing. In *Congressus Numerantium*, volume 42, pages 149–160, 1984.

[8] P. Erdős and A. Rényi. On Random Graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[9] J.-D. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant. Overlaying Graph Links on Treemaps. In *Poster Compendium of the IEEE Symposium on Information Visualization (INFOVIS),*, pages 82–83, 2003.

[10] T. M. J. Fruchterman and E. M. Reingold. Graph Drawing by Force-Directed Placement. *Software: Practice and Experience (SPE)*, 21(11):1129–1164, 1991.

[11] M. Ghoniem, J.-D. Fekete, and P. Castagliola. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 17–24. IEEE, 2004.

[12] D. Holten. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics (TVCG; Proceedings of INFOVIS'06)*, 12(5):741–748, 2006.

[13] D. Holten and J. J. van Wijk. A User Study on Visualizing Directed Edges in Graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09)*, pages 2299–2308. ACM, 2009.

[14] T. Kamada and S. Kawai. An Algorithm for Drawing General Undirected Graphs. *Information Processing Letters*, 31(1):7–15, 1989.

[15] 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.

[16] R. Kosara, C. G. Healey, V. Interrante, D. H. Laidlaw, and C. Ware. Thoughts on User Studies: Why, How, and When. *IEEE Computer Graphics & Applications (CG&A)*, 23(4):20–25, 2003.

[17] J. Madadhain, D. Fisher, P. Smyth, S. White, and Y. B. Boey. Analysis and Visualization of Network Data Using JUNG. *Journal of Statistical Software*, 10:1–35, 2005.

[18] C. B. Nielsen, S. D. Jackman, I. Birol, and S. J. M. Jones. ABySS-Explorer: Visualizing Genome Sequence Assemblies. *IEEE Transactions on Visualization and Computer Graphics (TVCG; Proceedings of INFOVIS'09)*, 15(6):881–888, 2009.

[19] C. North. Toward Measuring Visualization Insight. *IEEE Computer Graphics & Applications (CG&A)*, 26(3):6–9, 2006.

[20] NWB Team. Network Workbench Tool (Network Workbench: A Large-Scale Network Analysis, Modeling and Visualization Toolkit for Biomedical, Social Science and Physics Research). Indiana University, Northeastern University, and University of Michigan, 2006. `http://nwb.slis.indiana.edu/`.

[21] C. Ware and R. Bobrow. Motion to Support Rapid Interactive Queries on Node-Link Diagrams. *ACM Transactions on Applied Perception*, 1(1):3–18, 2004.

[22] C. Ware and R. Bobrow. Supporting Visual Queries on Medium-Sized Node-Link Diagrams. *Information Visualization*, 4(1):49–58, 2008.

[23] C. Ware and P. Mitchell. Visualizing Graphs in Three Dimensions. *ACM Transactions on Applied Perception (TAP)*, 5(1):1–15, 2008.

[24] M. Wattenberg. Arc Diagrams: Visualizing Structure in Strings. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS)*, pages 110–116. IEEE, 2002.

[25] D. J. Watts and S. H. Strogatz. Collective Dynamics of Small-World Networks. *Nature*, 393(6684):440–442, 1998.

[26] R. Wegenkittl, M. E. Gröller, and W. Purgathofer. A Guided Tour to Wonderland: Visualizing the Slow-Fast Dynamics of an Analytical Dynamical System. Technical Report TR-186-2-96-11, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 1996.

[27] P. C. Wong, P. Mackey, K. Perrine, J. Eagan, H. Foote, and J. Thomas. Dynamic Visualization of Graphs with Extended Labels. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS)*, pages 73–80, 2005.