

A User Study on Visualizing Directed Edges in Graphs

Danny Holten

Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, The
Netherlands
d.h.r.holten@tue.nl

Jarke J. van Wijk

Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, The
Netherlands
vanwijk@win.tue.nl

ABSTRACT

Graphs are often visualized using node-link representations: vertices are depicted as dots, edges are depicted as (poly)lines connecting two vertices. A directed edge running from vertex *A* to *B* is generally visualized using an arrow representation: a (poly)line with a triangular arrowhead at vertex *B*. Although this representation is intuitive, it is not guaranteed that a user is able to determine edge direction as quickly and unambiguously as possible; alternative representations that exhibit less occlusion and visual clutter might be better suited. To investigate this, we developed five additional directed-edge representations using combinations of shape and color. We performed a user study in which subjects performed different tasks on a collection of graphs using these representations and combinations thereof to investigate which representation is best in terms of speed and accuracy. We present our initial hypotheses, the outcome of the user studies, and recommendations regarding directed-edge visualization.

Author Keywords

Directed Edges, Lines, Curves, Graphs, User Studies, Information Visualization

ACM Classification Keywords

I.3.3 Computer Graphics: Picture/Image Generation - Line and Curve Generation; H.5.2 Information Interfaces and Presentation: User Interfaces - Screen Design

1 INTRODUCTION

When visualizing graphs using a node-link representation, directed edges are often shown using simple straight lines or polylines with triangular arrowheads at one or both ends. Although such a standard arrow representation is intuitive to most people, it can lead to problems in case of high-degree graph vertices, i.e., vertices that have many incoming and/or outgoing edges. For instance, it might be hard to determine to which edge a specific arrowhead belongs because of occlusion problems and visual clutter in case of a high number of arrowheads that are drawn around a high-degree vertex.

Furthermore, since the arrowheads often have approximately the same size and aspect ratio as the dots (circles) that depict graph vertices, the graph as a whole might be perceived as cluttered with detail to the point of being distracting.

Certain types of edge clutter can be remedied by means of user interaction, e.g., using the *EdgeLens* approach by Wong et al. [13], which interactively curves graph edges away from a person's focus of attention without changing the node positions. However, we contend that the static depiction of a graph should be visually optimized as well as possible before resorting to user interaction techniques to solve a problem that is, in essence, more related to visualization than to user interaction.

It is possible to develop alternative depictions of directed edges that do not suffer from the aforementioned problems. Representations in which hue or luminance vary along the length of an edge or representations in which width varies along the length of an edge are examples of such alternative representations based on color and shape. Furthermore, it is possible to combine representations, e.g., a combination of varying hue, intensity, and width along the length of an edge. We were also interested in examining if such multi-cue representations, i.e., those based on multiple visual cues, perform better than single-cue representations in terms of speed and accuracy when performing certain tasks.

To determine the performance of the different representations, we first performed an experiment involving six single-cue directed-edge representations. In short, subjects performed two tasks in which they had to answer whether or not there was 1) a single-step directed connection and 2) a two-step directed connection from one vertex to another in a randomly generated directed graph. Response times and accuracy were measured and analyzed using one-way ANOVA (Analysis of Variance) and Tukey's HSD (Honestly Significantly Different) post-hoc test to determine whether or not there were significant differences between the six single-cue directed-edge representations.

We performed a follow-up experiment involving four multi-cue directed-edge representations. These were developed using all possible combinations of three single-cue directed-edge representations that emerged from the first user experiment as viable candidates. Subjects performed the same tasks mentioned above using randomly generated directed graphs. Response times and accuracy were again analyzed using one-way ANOVA and Tukey's HSD post-hoc test.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2009, April 4 - 9, 2009, Boston, Massachusetts, USA.

Copyright 2009 ACM 978-1-60558-246-7/09/04...\$5.00.

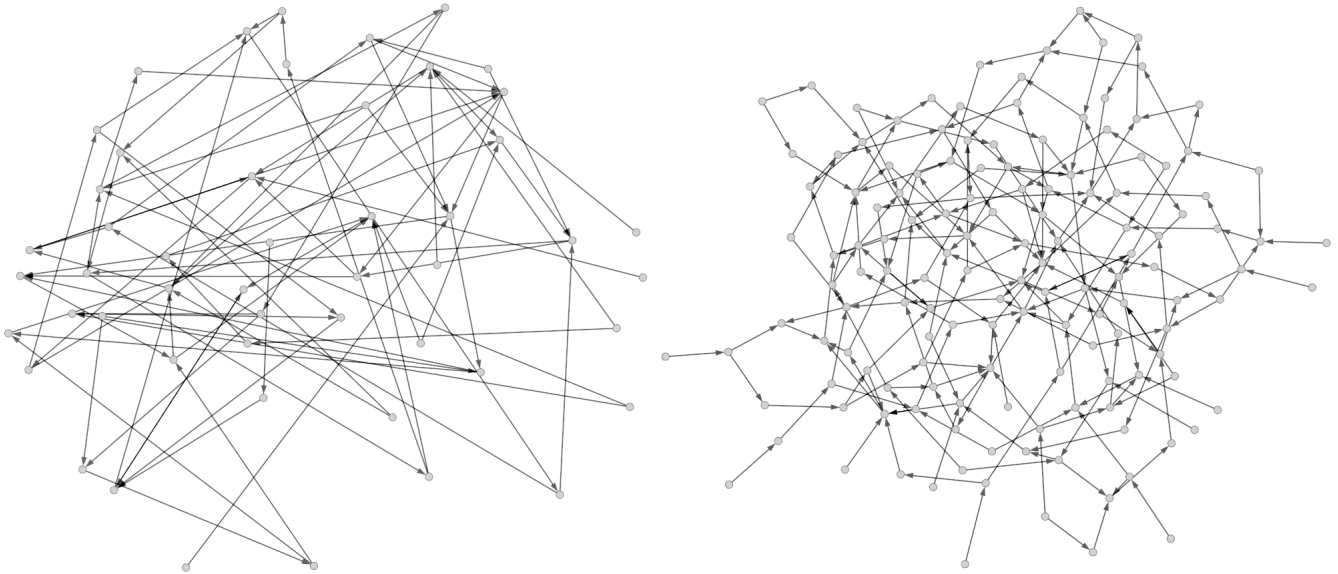


Figure 1. (left) A 50-vertex graph generated with $p = 50\%$ using a random layout; (right) a 140-vertex graph generated with $p = 50\%$ using a force-directed layout provided by AT&T's *Graphviz* NEATO algorithm.

2 RELATED WORK

Graph visualization as a whole is a well-studied field that generally focuses on methods for the generation of graph layouts. However, techniques that specifically deal with the visualization of directed edges within a node-link-based graph layout are less well-studied. Examples of such techniques are the following.

Becker et al. [1] use half-lines to visualize directed edges. A half-line from vertex A to vertex B is a straight-line connection between these vertices in which only the first half of the line is drawn, starting at vertex A . Fekete et al. [5] use curved links on top of treemaps to indicate edge direction. The edges (links) are drawn as a quadrilateral Bézier curve where the curvature indicates the direction of the link. Wong et al. [14] 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 an edge (or one of its vertices) is visualized in such a way that the text itself forms a curved, tapered edge between two vertices, removing the necessity to explicitly visualize the edge using a line-based representation. A color-coded representation is used by Holten [4] to indicate edge direction. Direction is indicated as running from vertex A to vertex B by gradually changing the color from green (A) to red (B) along the length of the edge.

To our knowledge, almost no user studies have been performed to quantify how the techniques mentioned above actually perform. An exception is the user experiment performed by Wong et al. for their *GreenArrow* approach. However, their focus was on how to balance the appearance of both a graph and its labels, not on determining how well their approach works as a general technique for depicting directed edges in comparison to other directed-edge representations.

Kosara et al. [7] note that visualization as currently practiced is mostly a craft and evaluation is generally performed informally. The use of user studies in visualization should therefore be encouraged, something which is also noted by North [9], who states that there are too many (informal) usability studies that only indicate whether or not users liked a certain visualization technique. Visualization evaluation should therefore seek to quantifiably determine how well visualizations perform, and one way of doing this for detailed design choices is to perform controlled user experiments. We therefore performed a controlled user study to quantify how the different directed-edge representations perform with respect to each other in terms of speed and accuracy.

3 GRAPH GENERATION

For our user experiments we needed a large collection of graphs with uniform characteristics that could be randomly generated on demand. The concept of a random graph is defined by Erdős et al. [3] as $G(n, p)$, in which each of the $\binom{n}{2}$ possible edges occurs independently with probability p . This model has been challenged by more focused classes of real-world graphs; it is therefore wrong to simply draw random graphs using this model when designing an experiment to assess the usability of a visualization technique [8].

Instead, we use the model by Ware et al. [10, 11] for our user experiments. Their graphs were generated using the following algorithm: “For each node in the graph, form a directed link to either 1 or 2 other nodes, randomly selected, with a generator biased so that a single directed connection occurs $p\%$ of the time and two directed connections occur $(100 - p\%)$ of the time.” The value of p can be adjusted to generate graphs with varying edge densities. We settled on a value of $p = 50\%$, which generates graphs that can have an adequate number of vertices without causing a large amount of visual clutter caused by an excessively high edge density.

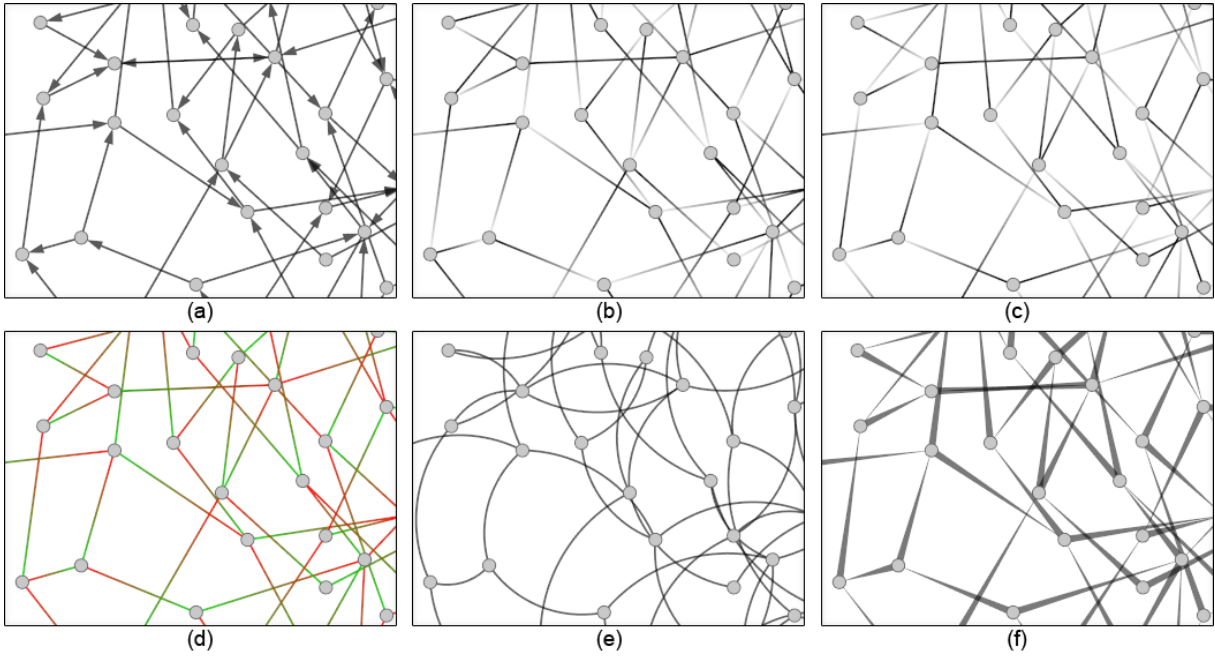


Figure 2. The six single-cue directed-edge representations used in the first user experiment. (a) “arrow”; (b) “light-to-dark”; (c) “dark-to-light”; (d) “green-to-red”; (e) “curved”; (f) “tapered”.

Since we also wanted to test the effect of the layout of a graph on the directed-edge representations, we chose to use two classes of layouts: a random layout and a force-directed layout. A force-directed layout algorithm treats a graph as a mechanical system with attracting spring forces between edges and repelling electrostatic forces between nodes. This whole system is iteratively simulated, which is repeated until the node positions stabilize.

The random layout was created by randomly placing the graph vertices in a circular container, whereas the force-directed layout was generated by using the NEATO program that is part of AT&T’s *Graphviz* [2] library. Since the random layout generates edges that are longer on average, the visual edge density is higher in these graphs.

To balance the visual edge density between both classes of graphs, we settled on generating the graphs that were laid out by NEATO’s force-directed algorithm using 140 vertices, while the graphs that use a random layout were generated using 50 vertices (see Figure 1).

4 FIRST USER EXPERIMENT

This section describes the first user experiment involving six single-cue directed-edge representations.

4.1 Single-Cue Directed-Edge Representations

We chose the intuitive and well-known black, straight-line arrow on a white background as the base case for the single-cue representations (“arrow”, see Figure 2a). The first alternative was created by observing that one of the most straightforward ways of changing a black straight-line connection between two vertices on a white background into a

directed connection (without adding elements and/or changing the curvature and/or shape of the line) is to change the color of the line. Due to the high sensitivity of the human visual system to changes in intensity, an obvious candidate is a line that varies from opaque/dark (near black) to near-transparent/light (near white) or vice-versa (“light-to-dark” and “dark-to-light”, see Figures 2b and 2c). The aforementioned green-to-red representation used by Holten [4] also belongs to the class of color-varying lines and was therefore added as well (“green-to-red”, see Figure 2d). The curved links by Fekete et al. [5] use curvature and a clockwise orientation to show the direction of an edge. Furthermore, the idea of using clockwise curvature to depict direction has also been used in other types of visualizations than graphs drawn as node-link diagrams [12, 6]. We therefore added a clockwise-curvature representation (“curved”, see Figure 2e). The final representation was created by simply generalizing the notion of the intuitive arrowhead by gradually varying the width of the line – wide at the start and narrow at the end – along its length (“tapered”, see Figure 2f).

In the remainder of this paper, we use the following abbreviations to refer to the single-cue directed-edge representations described above: “arrow” = “A”, “light-to-dark” = “LD”, “dark-to-light” = “DL”, “green-to-red” = “GR”, “curved” = “C”, and “tapered” = “T”.

To keep the visual density (“ink coverage”) as constant as possible for each of the representations, we slightly varied the amount of alpha-blending for each of the six representations as well as possible without visually compromising the representation. Alpha-blending pertains to the combination of two or more colors to allow for transparency effects. We

furthermore adjusted the red and green colors in the “green-to-red”-representation to be of equal luminance when converted to grayscale.

All of the visualizations were shown on a 22” Dell UltraSharp 2208WFP screen at a resolution of 1680x1050 with 8x anti-aliasing (using OpenGL). This setup was used throughout the user experiments.

4.2 Hypotheses

Our initial hypotheses with regard to the representations mentioned above were as follows.

We expected the “arrow”-representation to perform fairly well since this representation is quite intuitive and well-known to most users. However, due to the use of a triangular arrowhead at the end of each edge it might sometimes be hard to determine to which edge a specific arrowhead belongs because of occlusion problems and visual clutter in case of a high number of arrowheads that are drawn around a high-degree vertex. This might consequently diminish performance.

We were furthermore unsure whether the “light-to-dark”- or the “dark-to-light”-representation would be more intuitive to users. This is because a black line that varies in intensity (or transparency) along its length could intuitively be viewed as either a comet leaving a trail or as a needle pointing in a specific direction (see Figure 3). The comet-interpretation would be correct for the “light-to-dark”-representation while the needle-interpretation would be correct for the “dark-to-light”-representation. It was unclear to us in advance which of these interpretations – comet or needle – would be more intuitive to users.

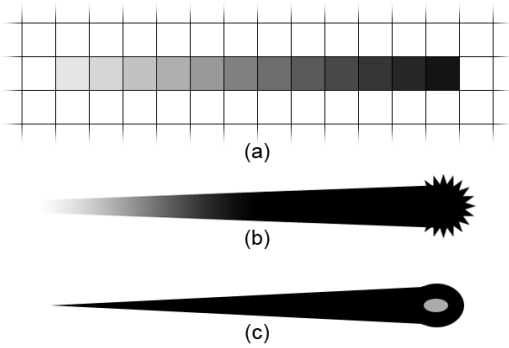


Figure 3. Possible ambiguous interpretation of intensity/transparency. The pixel-based line in (a) can be interpreted as (b) a comet leaving a trail, in which case the line is viewed as running from left to right, while it might also be interpreted as (c) a needle pointing in a specific direction, in which case the line is viewed as running from right to left.

Regarding “green-to-red” we speculated that it might benefit from an intuitive mnemonic device because red and green might readily be associated with “stop” and “go”, respectively. However, we also took into account that variation in hue might be less salient than variation in intensity from a perceptual point of view. Furthermore, the red-green color axis is notorious as far as color blindness is concerned,

something which should definitely be taken into account when using such a representation. Based on these observations, we expected the “green-to-red”-representation to have average performance; probably better than “curved”, roughly the same as “light-to-dark” and “dark-to-light”, and probably worse than “arrow” and “tapered”.

We expected “curved” to have the worst performance, mainly because we noticed that we got confused quite easily while informally working with our testing program. Determining the direction of an edge by forcing ourselves to “read” the edge in a clockwise fashion seemed to require quite some mental effort.

Finally, the “tapered”-representation appeared to have the benefit of the “arrow”-representation (edge direction is intuitively clear) while possibly having none of the aforementioned occlusion and visual-clutter problems of this representation. We therefore expected “tapered” to perform best, but possibly only slightly better than the “arrow”-representation.

4.3 Experiment Setup

Subjects performed two tasks in our first experiment. In task 1 they had to determine whether or not there was a single-step directed connection from vertex *A* to *B* (see Figures 4a and 4b). In task 2 they had to determine whether or not there was a two-step directed connection from vertex *A* to *B*, i.e., if vertex *B* could be reached from vertex *A* by visiting exactly one in-between vertex (see Figures 4c and 4d).

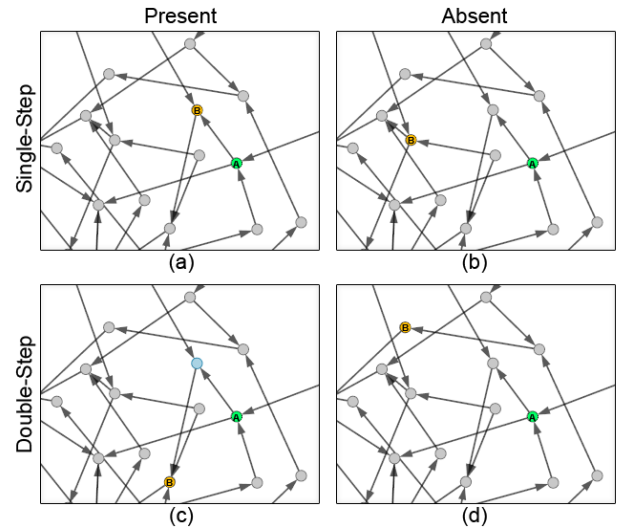


Figure 4. Tasks during first user experiment. (a) single-step connection from vertex *A* to *B* is present; (b) single-step connection from vertex *A* to *B* is absent; (c) double-step connection from vertex *A* to *B* is present (in-between vertex is highlighted in blue; this is not the case during the experiment); (d) double-step connection from vertex *A* to *B* is absent.

Subjects first performed task 1 on each of the six single-cue directed-edge representations, followed by task 2, which was also performed on each of the six representations. For both tasks, the order in which the representations were presented was randomized for each subject. Each representation was

first tested on a force-directed layout and subsequently on a random layout. Since initial usage of the testing program indicated that task 2 took longer than task 1, and that tasks performed on a random layout took longer than tasks performed on a force-directed layout, the number of trials per directed-edge representation was adjusted as follows:

Trials per Representation	Single-Step	Double-Step
Force-Directed Layout	20	10
Random Layout	12	6

This amounted to 48 trials per representation \times 6 representations = 288 trials per test instance. Completing a test instance took subjects on average around 20 minutes. A typical random test instance as described above might look as follows:

Task 1 – Single-Step			Task 2 – Double-Step		
Repr.	Layout	Trials	Repr.	Layout	Trials
LD	Force-D.	20	T	Force-D.	12
	Random	10		Random	6
T	Force-D.	20	DL	Force-D.	12
	Random	10		Random	6
GR	Force-D.	20	A	Force-D.	12
	Random	10		Random	6
DL	Force-D.	20	C	Force-D.	12
	Random	10		Random	6
A	Force-D.	20	GR	Force-D.	12
	Random	10		Random	6
C	Force-D.	20	LD	Force-D.	12
	Random	10		Random	6
Total Trials		180	Total Trials		108

Thirty test subjects with normal or corrected-to-normal vision participated in the first user experiment. Twenty of them were (under)graduate students at our university; ten were part of the scientific staff. They were not paid for participating in the experiment. In order to familiarize themselves with the tasks and program controls, subjects were given a couple of practice trials before commencing with the actual test instance. We continued with this until we felt certain that each subject fully understood the task and program controls.

Individual trials were presented to a subject as follows. The question presented to a subject for each trial remained constant within tasks. For task 1, the question was “*Is it possible to go FROM A TO B in EXACTLY 1 step? Press ‘+’ for ‘yes’, ‘-’ for ‘no’*”; for task 2, the question was “*Is it possible to go FROM A TO B in EXACTLY 2 steps? Press ‘+’ for ‘yes’, ‘-’ for ‘no’*”. Each trial was presented to a subject in three stages. Stage 1 displayed an empty white screen to the subject for 400ms. Stage 2 subsequently displayed two randomly chosen vertices *A* (green) and *B* (orange) to the subject for 600ms. This stage was introduced to ensure that subjects did not have to spend time on finding the two vertices. Finally, stage 3 displayed the complete graph on the screen as well and timing for the trial was started. Furthermore, a legend describing how to interpret the current edge representation was displayed in the lower-left corner during each trial. Figure 5 shows the screen contents as presented to

a subject during a trial. For each trial, the response time and correctness were registered. The testing program automatically halted between edge representations to give subjects the ability to have a short pause; subjects could resume the test by pressing the space bar.

Test subjects were presented with a new graph at each trial and vertices *A* and *B* were chosen such that there was a 50% chance that a directed connection was indeed present. Furthermore, in case of the force-directed layouts, vertices *A* and *B* were chosen such that vertices near the central region of the graph were preferred. This was done to ensure the selection of vertex pairs from the more crowded parts of the graph, since problems related to visual clutter and occlusion occur more often in the central regions of force-directed graph layouts.

All graphs were randomly generated, visually checked, and stored before the experiment. This was done to ensure that the graphs that were presented to the test subjects did not contain errors as the result of a degenerate layout produced by the NEATO layout algorithm (this happened in rare cases). It furthermore allowed us to present the same collection of graphs to each test subject.

4.4 Statistical Analysis and Results

The response times and error measurements collected during the first user experiment were analyzed using one-way ANOVA and Tukey’s HSD post-hoc test. Before discussing the outcome of these tests, we first take a look at the overall results depicted in Figure 6. This figure shows both the average response time as well as the response error per directed-edge representation; results are grouped by layout and task.

If we take a look at the influence of task and layout on response time and error, it can be seen that task 2 was clearly more difficult for subjects than task 1. Average response time and error for task 1 are 2362ms and 3.22%, respectively, while these values are 7866ms and 17.82%, respectively, for task 2. Switching from task 1 to task 2 increases response time and error by a factor of 3.32 and 5.59, respectively, while switching from a force-directed to a random layout only increases response time and error by a factor of 1.18 and 1.19, respectively. This shows that the influence of the task was much higher than the influence of the layout. A random layout appeared to be only slightly harder to interpret here than a force-directed layout. However, we had adjusted the number of nodes, so this is not directly comparable.

We can furthermore observe that response error appears to follow response time quite well. We can quantify this by calculating the correlation coefficient between response time and error as the average of the correlation coefficient for the single-step tasks and the correlation coefficient for the double-step tasks, which is 0.72. When calculated for all four tasks at once, the correlation coefficient is even higher: 0.96. However, this high value is mainly due to the widely differing values of response time and error between the single- and double-step tasks. It is also apparent that

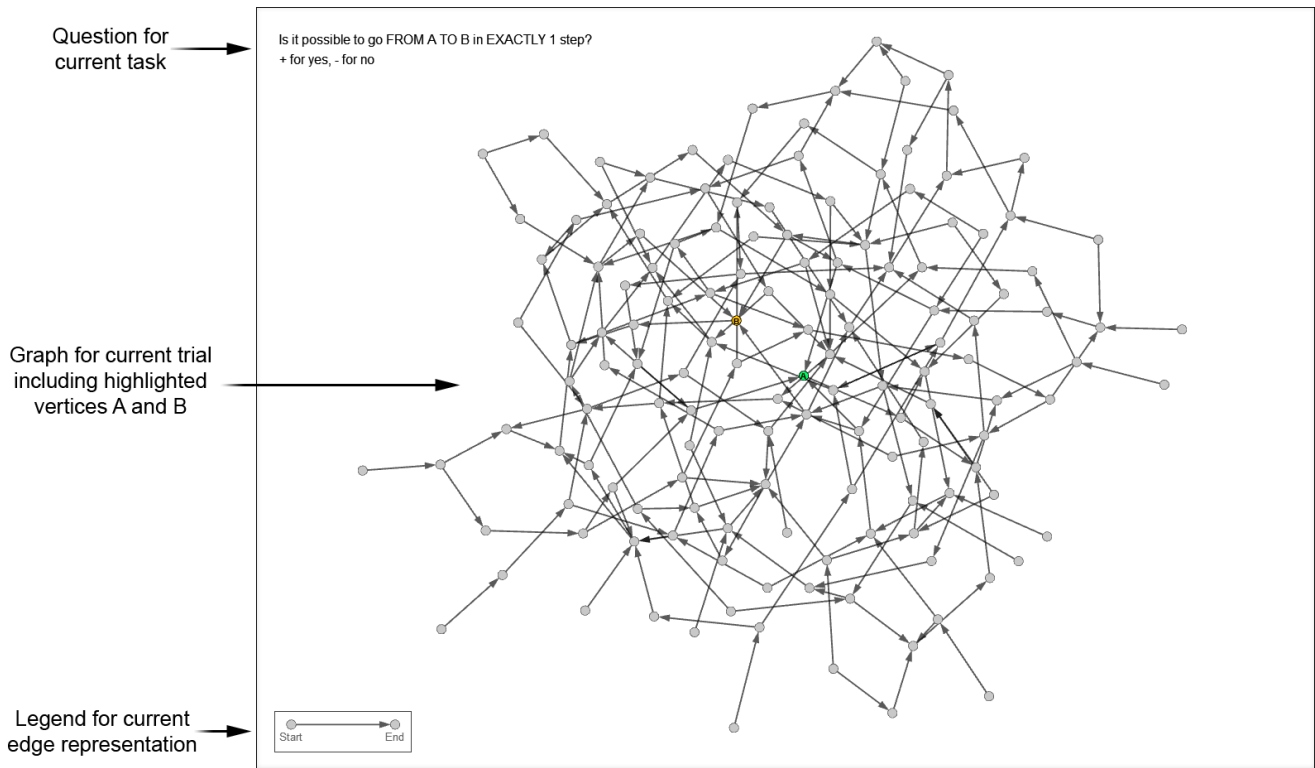


Figure 5. The screen contents as presented to a subject during a trial. The task-relevant question is displayed in the upper-left corner, a legend describing how to interpret the current edge representation is displayed in the lower-left corner, and the graph for the current trial is displayed in the center, including the highlighted vertices A (green) and B (orange).

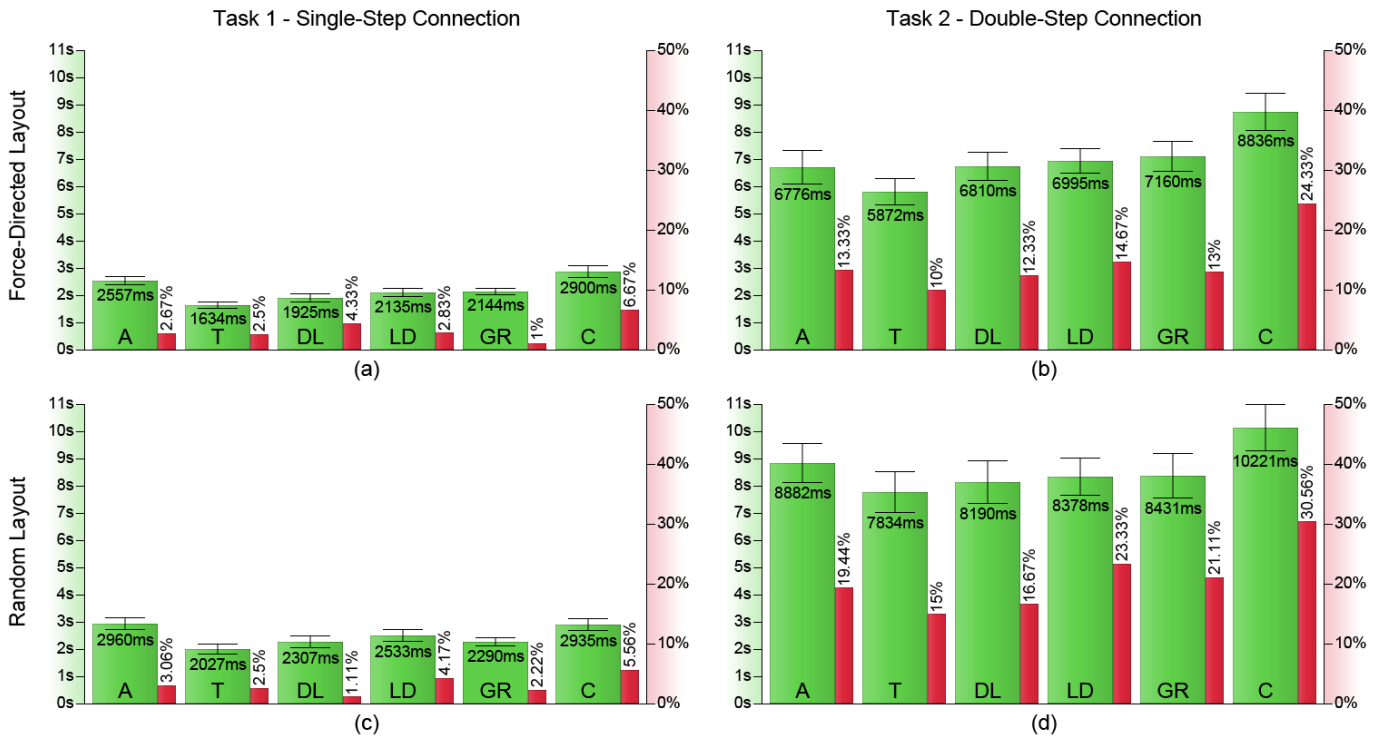


Figure 6. Results of first user experiment. Average response times per directed-edge representation are grouped by layout and task and are shown with a 95% confidence interval. Response error per representation is shown as well.

there is a clear overall pattern for the response times across all four combinations of task and layout: 1) “curved” comes out last and is followed by “arrow”, 2) the color-/intensity-based representations “dark-to-light”, “light-to-dark”, and “green-to-red” group together closely, and 3) “tapered” has the lowest overall response time.

Based on these observations, we can facilitate analysis by only analyzing response time using one-way ANOVA and Tukey’s HSD post-hoc test, since the results for response error follow a similar pattern. Furthermore, we can perform the analysis on the combined result of all tasks and layouts instead of a separate analysis for each of the four combinations of task and layout because of the presence of the aforementioned overall pattern. The combined results are shown in Figure 7.

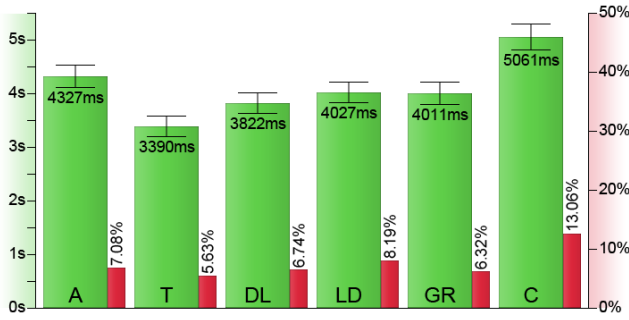


Figure 7. Combined results of first user experiment. Average response times per directed-edge representation are shown with a 95% confidence interval. Response error per representation is shown as well.

We performed a one-way ANOVA analysis with response time as the single factor and the six single-cue directed-edge representations as the groups for which we have multiple observations. The ANOVA analysis indicated that the differences in response time between the six representations are statistically significant ($F(5, 1440) = 25.54, p < 0.05$). However, it does not show us between which representations the differences in response time are significant. We used Tukey’s HSD post-hoc test to perform a pairwise comparison to see where the actual significant differences reside. The result of this is shown in Figure 8. To facilitate interpretation of the results in Figure 8, a graphical depiction is shown in Figure 9 (although “arrow” was found to be a poor candidate, we nevertheless use this intuitive representation in Figure 9, since occlusion and visual clutter will not be a problem in this case due to the limited number of nodes and edges).

The result shows that “curved” significantly differs from all other representations; “curved” is clearly worst as far as performance is concerned. “tapered”, on the other hand, significantly differs from all but one of the others. Overall, “tapered” can be regarded as the best representation in terms of performance. Tukey’s HSD test also shows that the color-/intensity-based representations “dark-to-light”, “light-to-dark”, and “green-to-red” can indeed be grouped together: there are no significant differences between any

combination of these representations. Finally, “arrow” performs significantly worse than two and significantly better than one of the five remaining representations.

	A	T	DL	LD	GR	C
A		8.56	4.56	2.71	2.85	6.61
T	8.56		3.89	5.75	5.61	15.07
DL	4.56	3.89		1.86	1.71	11.18
LD	2.71	5.75	1.86		0.14	9.32
GR	2.85	5.61	1.71	0.14		9.47
C	6.61	15.07	11.18	9.32	9.47	

Figure 8. Result of performing Tukey’s HSD test on the combination of all tasks and layouts. For $\alpha = 0.05$, 6 treatments, and a within-groups degrees-of-freedom value > 1000 (8634 in our case), all values larger than 4.03 (green) show a statistically significant difference.

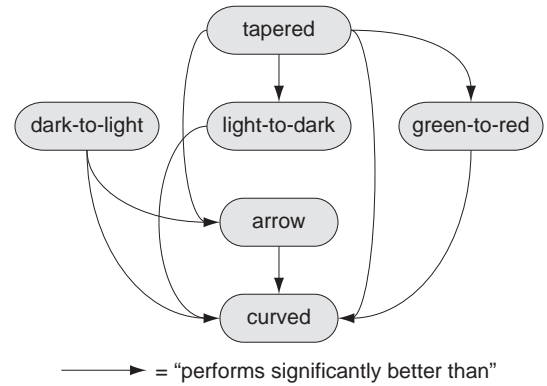


Figure 9. Graphical depiction of performing Tukey’s HSD test on the combination of all tasks and layouts. An edge is present for each statistically significant difference in Figure 8. Edge direction indicates that the origin representation performs significantly better than the destination representation.

The following holds with regard to our initial hypotheses. The intuitive “arrow” representation performed a lot worse than we expected; although we indeed expected “tapered” to perform better than “arrow”, we did not expect this to be true for the color-/intensity-based representations as well. The use of an arrowhead might indeed have caused the aforementioned occlusion problems and visual clutter, which consequently led to a decrease in performance. Although additional, more specific tests are needed to ensure that the poor performance can indeed be attributed to the presence of the arrowheads, various subjects did state that they experienced the arrowheads as somewhat distracting. As far as “dark-to-light” and “light-to-dark” are concerned, subjects had a (slight) preference for the former, indicating that the needle-interpretation seems to be more intuitive. This is in line with the fact that the needle-interpretation is visually more compatible with an arrow pointing in a specific direction, especially if a generalized arrow representation, such as “tapered”, is considered. The “green-to-red”- and “curvature”-representation indeed performed as we hypothesized. Finally, “tapered” performed clearly better than “arrow”; not – as we noted before – just slightly better.

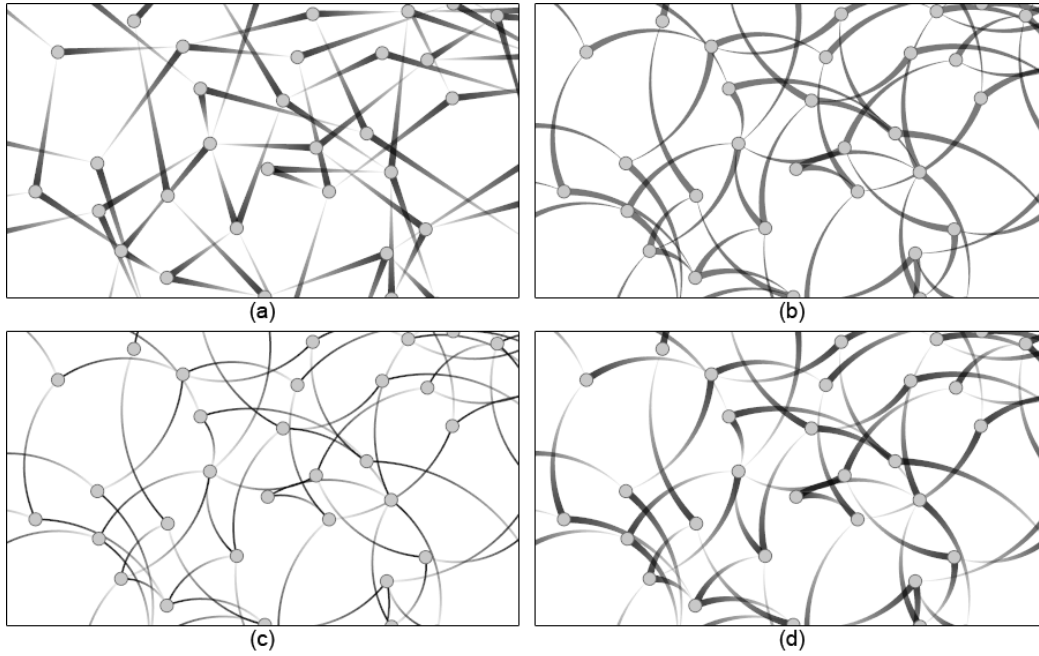


Figure 10. The four multi-cue directed-edge representations used in the follow-up user experiment. (a) “TI” (tapered-intensity); (b) “TC” (tapered-curved); (c) “IC” (intensity-curved); (d) “TIC” (tapered-intensity-curved).

5 FOLLOW-UP USER EXPERIMENT

This section describes the follow-up user experiment involving four multi-cue directed-edge representations.

5.1 Multi-Cue Directed-Edge Representations

For the follow-up user experiment we needed to limit the possible combinations of representations as much as possible, since testing would otherwise become infeasible. We therefore selected a subset of the six single-cue representations from the first user experiment and subsequently constructed multi-cue representations by taking all possible combinations of the representations within this subset.

We did this by observing that the color-/intensity-based representations could be grouped together and that subjects had a slight preference for the “dark-to-light”-representation. We therefore chose “dark-to-light” as being representative for this group of representations. We furthermore kept the “tapered”- and “curvature”-representation because we wanted to see what influence the presence of the best and worst representation during combining would be. Finally, the “arrow”-representation was omitted since we had already seen that its performance was quite low, we were mainly interested in the alternative representations, and because the “arrow”-representation would increase the number of possible combinations from 4 to 11.

This resulted in a subset of three representations – “intensity”, “tapered”, and “curved” – based on which we created the four combinations depicted in Figure 10, i.e., “TI” (tapered-intensity), “TC” (tapered-curved), “IC” (intensity-curved), and “TIC” (tapered-intensity-curved).

Furthermore, visual density was again kept as constant as

possible by varying alpha-blending and the visualizations were shown and tested on the same setup as before.

5.2 Hypotheses

We speculated that combining would probably lead to one of three possibilities: 1) combined performance would roughly be the average of the performances of each of the comprising representations, 2) the comprising representations would positively reinforce each other by providing a user with multiple visual cues, which would increase performance, or 3) the comprising representations would negatively interfere and/or would overload a user with too many visual cues, which would lead to a decrease in performance.

Based on our own experience with the four multi-cue representations while informally working with the testing program, we expected the second option to be the most feasible, i.e., combining would probably lead to increased performance due to the presence of multiple visual cues that would positively reinforce each other.

5.3 Experiment Setup

We already saw that there was a clear overall pattern for the response times across all four combinations of task and layout in the first user experiment. We therefore decided to only perform one task consisting of single-step connection trials on force-directed layouts during the follow-up experiment.

Subjects performed this task on each of the four multi-cue directed-edge representations. The order in which the representations were presented was again randomized for each subject. This amounted to 20 trials per representation \times 4 representations = 80 trials per test instance. Completing a

test instance took subjects on average between 5 to 10 minutes.

Individual trials were presented to a subject in the same way as during the first user experiment and the choice of vertex pairs during each individual trial was performed in the same way as well.

Fifteen test subjects with normal or corrected-to-normal vision participated in the follow-up user experiment. Twelve of them were (under)graduate students at our university; three were part of the scientific staff. They were not paid for participating in the experiment.

5.4 Statistical Analysis and Results

We performed a one-way ANOVA analysis with response time as the single factor and the four multi-cue directed-edge representations as the groups for which we have multiple observations. The ANOVA analysis indicated that the differences in response times between the four representations are statistically significant ($F(3, 300) = 4.25, p < 0.05$).

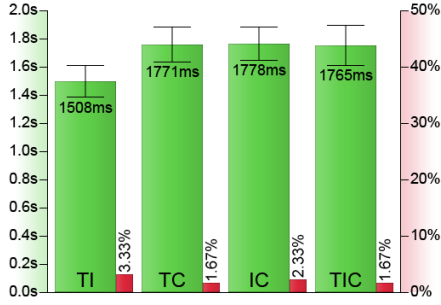


Figure 11. Results of follow-up user experiment. Average response times per directed-edge representation are shown with a 95% confidence interval. Response error per representation is shown as well.

The result of performing Tukey’s HSD post-hoc test to see where the actual significant differences reside is shown in Figure 12. The result shows that only “ti” significantly differs from all other representations; “ti” can be regarded as the best representation in terms of performance.

	TI	TC	IC	TIC
TI		4.11	4.22	4.03
TC	4.11		0.11	0.08
IC	4.22	0.11		0.19
TIC	4.03	0.08	0.19	

Figure 12. Result of performing Tukey’s HSD test. For $\alpha = 0.05$, 4 treatments, and a within-groups degrees-of-freedom value > 1000 (1196 in our case), all values larger than 3.63 (green) show a statistically significant difference.

Because the single-cue and multi-cue representations cannot be compared directly and since only fifteen instead of thirty subjects participated in the second experiment, we did not perform a statistical analysis to compare the results of the first experiment to those of the second experiment. However, when comparing the results depicted in Figure 11 to those in Figure 6a, it is clear that our initial hypothesis does not fully

hold, i.e., combining does not necessarily lead to increased performance due to the presence of multiple visual cues that positively reinforce each other. It is indeed the case that “ti” (1508ms) performs better than “tapered” (1634ms) and “dark-to-light” (1925ms) separately, but the improvement is small when compared to the best of its comprising representations (“tapered”). The same holds for “ic” (1778ms), which performs better than “dark-to-light” (1925ms) and “curved” (2900ms) separately, but the improvement is again small when compared to the best of its comprising representations (“dark-to-light”). Furthermore, “TC” (1771ms) and “TIC” (1765ms) even perform somewhat worse than the best of their comprising representations, i.e., “tapered” (1634ms) in both cases.

These results indicate that the way in which visual cues interact in the context of multi-cue directed-edge representations is more complex than being a clear case of 1) averaging, 2) positive reinforcement, or 3) negative interference.

6 RECOMMENDATIONS

Based on the results in Sections 4.4 and 5.4, our recommendations regarding the visualization of directed edges within node-link-based graph layouts are the following:

- A standard arrow representation should be avoided whenever possible. Although this representation is straightforward and intuitive to most users, the performance of this popular representation is quite low, which is probably due to the use of arrowheads that cause occlusion problems and visual clutter;
- A tapered representation in which the width of an edge is gradually varied along its length – wide at the start and narrow at the end – proved to be the best representation in terms of performance. We therefore recommend the use of this representation to visualize directed edges;
- If an intensity-based representation is chosen to visualize directed edges (which is still a better choice than the standard arrow representation mentioned above as far as performance is concerned) then a dark-to-light representation has a performance advantage over a light-to-dark representation. This hints at a preference of users for the aforementioned needle-interpretation with respect to intensity-based directed-edge representations;
- There does not seem to be a clear performance benefit associated with the multi-cue directed-edge representations that we tested. We therefore recommend the use of single-cue directed-edge representations instead.

7 CONCLUSIONS AND FUTURE WORK

We have presented six single-cue and four multi-cue directed-edge representations as alternatives to the standard arrow representation for the visualization of directed edges in graphs drawn as node-link diagrams. To investigate whether or not these alternative representations suffered less from problems related to occlusion and visual clutter, we performed two user studies in which subjects performed different tasks on two classes of node-link-based graph layouts.

We provided initial hypotheses regarding the performance, i.e., speed and accuracy, of each of the representations and compared these to the outcome of the user studies. Based on the analysis of both user studies we provided recommendations for the representation of directed edges in the context of graphs drawn as node-link diagrams.

Based on the experience gathered while informally working with our testing tool and while performing the user studies, our recommendations for possible future work are as follows:

- Additional single-cue directed-edge representations (and combinations thereof) could be tested to determine whether a certain type of representation might perform better than the ones that we have already tested. For instance, animated representations might be worth investigating as well;
- Alternative tasks relevant to graph exploration could be tested instead of single- and double-step connection trials to determine whether the relative performance of directed-edge representations within the context of graph exploration is task-dependent or not. An example of such an alternative task is the determination of the fan-in (number of incoming edges) and/or fan-out (number of outgoing edges) of a vertex;
- As mentioned before, the way in which visual cues interact in the context of multi-cue directed-edge representations is more complex than being a clear case of averaging, positive reinforcement, or negative interference. To be able to create a generic model of the way in which representations interact (if possible at all), additional multi-cue directed-edge representations need to be tested to collect more information on the effects of combining representations.

ACKNOWLEDGEMENTS

This project is funded by the Netherlands Organization for Scientific Research (NWO) Jacquard program under research grant no. 638.001.408 (*Reconstructor Project*). We would furthermore like to thank the participants in our user study for their time.

REFERENCES

1. 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.
2. J. Ellson, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull. Graphviz and Dynagraph – Static and Dynamic Graph Drawing Tools. In M. Jünger and P. Mutzel, editors, *Graph Drawing Software*, pages 127–148. Springer-Verlag, 2003.
3. P. Erdős and A. Rényi. On Random Graphs. *Publicationes Mathematicae*, 6:290–297, 1959.
4. D. Holten. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics (TVCG; Proc. of INFOVIS'06)*, 12(5):741–748, 2006.
5. N. D. Jean-Daniel Fekete, David Wang, A. Aris, and C. Plaisant. Overlaying Graph Links on Treemaps. In *Proceedings of the 2003 IEEE Symposium on Information Visualization (INFOVIS'03), Poster Compendium*, pages 82–83, 2003.
6. B. Kerr. Thread Arcs: An Email Thread Visualization. In *Proceedings of the 2003 IEEE Symposium on Information Visualization (INFOVIS'03)*, pages 211–218, 2003.
7. 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.
8. G. Melançon. Just How Dense are Dense Graphs in the Real World? A Methodological Note. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'06)*, pages 75–81, 2006.
9. C. North. Toward Measuring Visualization Insight. *IEEE Computer Graphics & Applications (CG&A)*, 26(3):6–9, 2006.
10. C. Ware and R. Bobrow. Supporting Visual Queries on Medium-Sized Node-Link Diagrams. *Information Visualization*, 4(1):49–58, 2005.
11. C. Ware and P. Mitchell. Visualizing Graphs in Three Dimensions. *ACM Transactions on Applied Perception (TAP)*, 5(1):1–15, 2008.
12. M. Wattenberg. Arc Diagrams: Visualizing Structure in Strings. In *Proceedings of the 2002 IEEE Symposium on Information Visualization (INFOVIS'02)*, pages 110–116, 2002.
13. N. Wong, S. Carpendale, and S. Greenberg. EdgeLens: An Interactive Method for Managing Edge Congestion in Graphs. In *Proceedings of the 2003 IEEE Symposium on Information Visualization (INFOVIS'03)*, pages 51–58, 2003.
14. 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 2005 IEEE Symposium on Information Visualization (INFOVIS'05)*, pages 73–80, 2005.