Eurographics Conference on Visualization (EuroVis) 2012
S. Bruckner, S. Miksch, and H. Pfister
(Guest Editors)

Volume 31 (2012), Number 3

Perception of Animated Node-Link Diagrams for Dynamic Graphs

S. Ghani, N. Elmqvist, and J. S. Yi

Purdue University, USA

Abstract

Effective visualization of dynamic graphs remains an open research topic, and many state-of-the-art tools use animated node-link diagrams for this purpose. Despite its intuitiveness, the effectiveness of animation in node-link diagrams has been questioned, and several empirical studies have shown that animation is not necessarily superior to static visualizations. However, the exact mechanics of perceiving animated node-link diagrams are still unclear. In this paper, we study the impact of different dynamic graph metrics on user perception of the animation. After deriving candidate visual graph metrics, we perform an exploratory user study where participants are asked to reconstruct the event sequence in animated node-link diagrams. Based on these findings, we conduct a second user study where we investigate the most important visual metrics in depth. Our findings show that node speed and target separation are prominent visual metrics to predict the performance of event sequencing tasks.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1. Introduction

Time-varying or *dynamic* graphs (graphs that change over time) are becoming increasingly important for understanding evolving mechanisms in engineering, biology, and the social sciences. For example, biologists study changes in phenotypes over time using dynamic graphs; crime analysts gather facts about the activities of individuals of interest over a period of time by representing their networks as dynamic graphs; and social networks such as those from Facebook or LinkedIn can also be represented in the form of dynamic graphs. The most common approach to visualize dynamic graphs is through animated node-link diagrams [MMBd05].

However, the effectiveness of animation has been questioned [TMB02], and several empirical studies have shown that animated visualization is not superior to static visualization. For example, showing trends using animation can be confusing to people, even when they are told where to look, and small multiples and trace lines may work better [RFF*08]. Several recent studies have shown that small multiples allow users to perform significantly faster than animation for many graph comprehension tasks [APP10,FQ11].

Several guidelines and metrics exist for how to optimize the readability of a static node-link diagram. Unfortunately, little such evidence exists that provides clear understanding of which aspects of dynamic (i.e., animated) node-link diagram cause these difficulties. Some of the suggested causes include movement speeds of nodes and edges, the distances between nodes and edges, and edge/node as well as edge/edge crossings. However, there could exist numerous additional factors that may also have an impact. If we knew which factors were the most influential, we could provide specific solutions to mitigate the impact of these factors.

Thus, the goal of this paper is to understand people's perception of dynamic graphs by characterizing the factors that affect user perception of animated node-link diagrams. Because we are not certain what those factors are, we begin by deriving visual metrics that characterize a dynamic graph using static graph metrics as a starting point. We then generate a large number of animations and calculate these metrics for each trial, selecting representative trials for each metric. We finally use these trials to perform an exploratory user study on correctness and cognitive load for addition and deletion of both nodes and links. Our study also includes two different graph layouts—a fixed layout and a force-directed layout—so that we can study how the movement of nodes affects user perception. We then use these results to run a second summative study that uses distinctive graph metrics as factors.

2. Background

A *dynamic graph* is a graph whose structure changes over time. These topological changes involve either nodes or edges being added to or removed from the graph. A *dynamic graph visualization technique* is therefore a graph visualization technique that is capable of visually conveying these changes over time. Note that this delimits our definition to not include graphs with node and edge attributes that change over time. Such graphs are outside the scope of this paper.

2.1. Dynamic Graph Visualization

Although there exist a few tools, such as TimeMatrix [YEL10], that use adjacency matrices for dynamic graphs, most tools that support dynamic graphs—such as Pajek [BM02], KrackPlot [KBM94], and UCINet [BEF02]—use node-link based representations. In general, conveying topological changes in time using node-link representations can be done in several different ways:

- **Animation:** Animation renders a sequence of successive static snapshots that create the illusion of movement. Whenever there is a temporal domain (like for dynamic graphs), animation is one of the most common choices to present the changes over time [MMBd05, TMB02].
- **Small multiples:** Small multiples show several snapshots of graphs at different points in time, often side-by-side or in a temporal sequence. However, choosing suitable and representative snapshots is a non-trivial problem, and aligning them side-by-side may cause the display space allocated to each individual snapshot to be small.
- **Traces:** Another common approach is to transform time into space by tracing the movement path of the visual objects [RFF*08]. This idea can also be applied to graphs, but may give rise to visual clutter and it is not obvious how to represent discrete events, such as additions or deletions.
- **3D:** Some graph visualization techniques use the third dimension (often height) for representing change over time [BC03]. However, 3D visualization is plagued by inherent problems such as distortion and occlusion effects.

2.2. Layout for Dynamic Graphs

Graph layout is critical for effective graph visualization in general [BETT99], and there exists a number of graph layout algorithms that are solely designed for dynamic graphs. Most strive to maximize the stability of the graph during the animation to reduce the need for the user to track nodes as they move around the graph. Several dynamic graph layout algorithms exist in the literature [BW97, CDT*92, DG02]. While one outcome for this work may be to help designers of such algorithms build better layouts for dynamic graphs, our work addresses more fundamental aspects of human perception than layout. For this reason, we adopt a specific layout—the lin-log force-directed algorithm [Noa05]—as a baseline in our evaluations instead of varying this as a factor.

2.3. Animation for Dynamic Graphs

Animation is used in user interfaces for a variety of purposes [BS90], and is also prominent in visualization research; examples include transitions of data [HR07], transitions of views [BTC*06], and trends over time [RFF*08]. Animation is also commonly used for dynamic graphs.

However, several studies question the effectiveness of animation over static graphics. Tversky et al. [TMB02] identified flaws in studies purporting to show the effectiveness of animation and claimed that it may not be superior to static graphics. Archambault et al. [APP10] found that small multiples is better than animation for dynamic graphs. Robertson et al. [RFF*08] showed that animation is less effective than both static representations and small multiples. Farrugia and Quigley [FQ11] similarly proved that static methods are better for representing dynamic graphs compared to animation. Interestingly, the effect of animation might not be linearly correlated with the animation speed in some tasks [PS08].

Nevertheless, animation is easy to implement, and is seemingly natural and useful. Perhaps for these reasons, animation is very prominent in graph visualization tools. However, there exists little knowledge as to *which* factors, in particular, contribute to making animation difficult to use for dynamic graphs. If we had this knowledge, we could design methods to mitigate the impact of these factors. This in turn would let us continue to use node-link diagrams, which after all are the most popular visual representations for graphs because of familiarity and graph task utility [LPP*06].

3. Dynamic Graph Metrics

Visual graph metrics are measures of a graph layout, such as average edge length and number of edge crossings, and there exists much work on defining such metrics [BFN85, BETT99] (note that this is different from pure graph metrics, which are invariant to layout). Most of these take an aesthetic approach [CP96], with the reasonable assumption that this will transfer to readability. In particular, Purchase [Pur02] proposed a set of seven aesthetic metrics that many tools use.

More recently, Dunne and Shneiderman [DS09] used Purchase's seven metrics as basis to give a set of new metrics for improving graph readability. They define metrics like node occlusion, edge crossing, edge crossing angle, and edge tunneling, and argue that these can be used to improve graph layout and presentation. However, all these metrics are designed for static and not dynamic graphs, which typically have many additional properties. Many of these metrics also focus on aesthetics rather than optimal perception of a graph.

In the below text, we extend these metrics to encompass dynamic graphs, as well as define some new metrics that are unique to dynamic graphs. In the discussion below, we define a *target* as a graph entity (node or edge) that is added or removed during the animated sequence. Some metrics are only defined when target nodes appear or disappear—*node metrics* (N)—while some are only defined when target edges appear or disappear—*edge metrics* (E). A third kind are defined for graphs or subgraphs—*graph metrics* (G). Also, depending on the size of the graph, it is possible to compute these metrics for a whole graph, or for smaller subgraphs.

- **Node Speed (N, G):** Average speed for nodes.
- **Target Speed (N, G):** Average speed for target nodes.
- **Target Separation (N, E, G):** We define this metric as the average distance between target nodes or edges (similar to the "minimizing temporal aliases" criterion [BBD09]).
- **Target Node Degree (N):** This metric yields the average degree (the sum of the in-degree and out-degree in a directed graph) of all target nodes in the graph.
- **Node-Node Distance (N):** This metric is motivated by Dunne's and Shneiderman's Node Occlusion [DS09], and for static graphs defines how much each node is occluded by any other node. For dynamic graphs, we define node occlusion as the minimum distance between a target node and any other node during the whole animation.
- **Node-Edge Distance (N):** This metric is motivated by the Edge Tunnel metric [DS09], which for static graphs describes nodes overlapping with edges. For dynamic graphs, we define this metric as the minimum distance between each target node and any edge of the graph (apart from its own edges) during the whole animation.
- **Edge-Edge Distance (E):** This metric is similar to the above two and is defined as the minimum distance between a target edge and any other edge during animation.
- **Edge Crossings (E):** Edge crossings is one of the most common graph metrics for readability [BETT99]. For dynamic graphs, we define this metric as the number of separate times that a target edge crosses any other edge.
- **Edge Angle Sweep (E):** Crossed edges are difficult to perceive as their incident angle decreases [DS09, WPCM02]. For dynamic graphs, we define this as the sum of the angle sweep of a target edge during an animation.
- **Edge Length (E):** Edge Length is an important metric for many graph layout algorithms since many algorithms try to reduce intra-node distances. For dynamic graphs, the length of the edge is not fixed, so we measure two values for each target edge: the minimum length and the maximum change, both of which are problematic to perceive.

There exists additional metrics for dynamic graphs, such as node size, color, and shape; labels; layout; and edge bends. However, these metrics are more indirect or abstract, and are thus likely to have a negligible impact on user perception and thereby outside the scope of our work.

4. Exploratory User Study

In the previous section, we defined a large collection of metrics for dynamic graphs represented as node-link diagrams. However, no empirical data exists on how accurately these different metrics predict the difficulty of perceiving dynamic changes in the graph. To begin investigating this relation between dynamic graph perception and the above metrics, we conducted an exploratory user study where we asked participants to observe node-link animations. We calculated the metrics for each graph and studied the relations between them and participant performance using statistical methods.

Another goal of this study is to determine the proper experimental settings for further studies. We assume that dynamic graphs (through force-directed layout) are more difficult to perceive than static graphs; detecting deletion of nodes are more difficult than addition of nodes; and the more nodes we have, the more difficult to perceive changes. Through this study, we would like to verify those aspects.

4.1. Participants and Apparatus

We recruited 16 paid volunteers (ten male and six female), screened not to be color blind at a compensation of $10 per hour. All participants were university students with ages varying from 20 to 31 years (average 25). Each participant used a 3.00 GHz dual-core PC with 4 GB of memory with the size of the graph viewport set to 1024×768 pixels.

4.2. Experimental Platform

We built a dynamic graph viewer in Java for this experiment. The viewer shows a node-link representation of the graph so that it is fully contained within the viewport. Mouse or keyboard interaction inside the viewport was disabled.

4.3. Experimental Factors

We included the below three factors in our study:

- **Task Types (T):** Nodes being added (NA) or deleted (ND), or edges being added (EA) or deleted (ED).
- **Number of Entities (E)** Animations become more difficult to perceive and remember as the number of events increases. Therefore, we include the number of entities E as a condition for how many target entities are added or deleted from the node-link diagram during the animation. Guided by a pilot study, we use 2, 3, and 5 entities.
- **Layout (L):** For dynamic graphs, the graph layout is one of the most important parameters to choose. As discussed in the related work section, there exists several different layout algorithms solely designed for dynamic graphs where the main purpose is to dampen the impact on the whole graph when entities are added or deleted. This presumably makes these events easier to perceive. For this experiment we choose two different layouts (which represent realistic upper and lower bounds of stability) to find out whether dynamic graph layouts, which minimize node movement, cause more difficulty than static layouts:
  - **Fixed Layout:** We calculate the layout for the complete graph (including added or deleted entities) before

the animation so that entities will be added or deleted without any movement in the layout. This layout condition was chosen since we think it is the ideal dynamic graph layout with no node movement at all, i.e., it is a lower bound on layout stability.
– **Force-Directed Layout.** We calculate a force-directed layout (lin-log [Noa05]) for every intermediate graph in the animation, i.e., every time step of the graph when an entity has been added or removed. Node positions will be linearly interpolated between these intermediate layouts during the animation.

It is important to note that the purpose of this experiment is *not* to validate existing dynamic graph layout algorithms, but rather to determine the factors affecting dynamic graph perception. For this reason, we explicitly chose *not* to use such an optimized layout algorithm in this experiment, but instead used the above layouts as references.

4.4. Study Design

Given the above factors, we used a within-subject full-factorial design: 4 task types (T) $\times$ 3 number of entities (E) $\times$ 2 layouts (L) $\times$ 3 repetitions = 72 unique conditions. We counterbalanced the order of layout and blocked on this factor. Inside each layout block, we counterbalanced the order of the task types and blocked on task types. With 16 participants, we collected total of 1152 trials. The experiment platform recorded the following metrics:

- **Correctness:** A binary response to indicate whether the entity order given by the participant was fully accurate.
- **Cognitive Load:** 1-9 scale representing the participant's self-reported mental effort for the trial [Paa92].

We debated using correctness measures based on order or position. However, it is difficult to justify which metric to choose, so we decided to instead use a binary measure.

4.5. Task

We wanted to choose a representative task where participants would not only have to perceive changes in the graph topology, but also maintain the overall mental map of the graph as well as the event chronology throughout the animation. We therefore asked our participants to reconstruct the causal order of nodes or edges being added or removed from the graph during the animation, which is an intrinsic task for revealing temporal relationships in dynamic graphs. This is a realistic task: consider a social network where a user is analyzing the friends added over a time period. In this case, the user may need to know which friend was added first, and how this caused other friends to be added later on.

A trial starts with a node-link diagram being shown in our graph viewer, with all nodes drawn in gray and edges in black. During the animation, the graph will successively change, one entity at a time. The nature of the change depends on the task type T: for addition tasks, nodes or edges are added, whereas for deletion tasks, they are removed. The number of entities added or removed depends on the experimental factor E, and they are randomly chosen from the graph before the start of the trial. Participants will not know in advance how many entities will be added or removed.

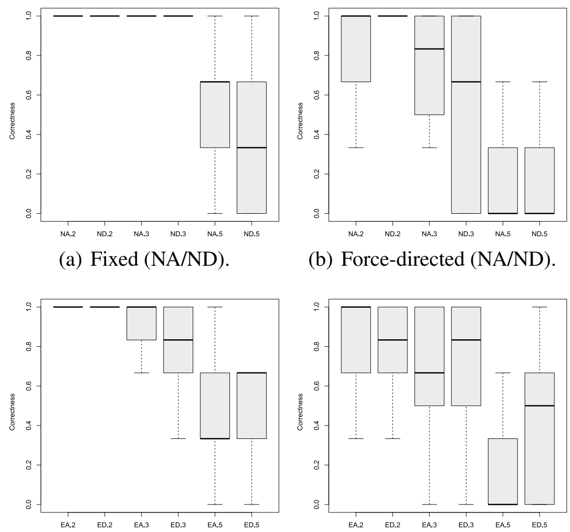
The animation is performed as a linear interpolation of node positions. The time between the entities being added or removed is fixed to 1 second (derived using an initial pilot study). The other option was to fix the overall trial time—say to 3 seconds—causing all changes to happen during this time. However, this would cause different cognitive load on participants depending on the number of entities added or removed; for example, when the number of entities is equal to 2, there will be a 1.5 second gap between the two events, whereas for 5 entities, there will be 600 ms gap between events. By keeping the time between events constant, we tried to balance the cognitive load regardless of the number of entities. Below we describe the task types:

- **Node Addition (NA):** In this task, new nodes are successively added to the graph in a random position on the visual space and with the same gray color as other nodes. When adding a new node, the edges associated with it are also added to the graph. For each new node, we then calculate a layout for the graph with this node added and animate the nodes to their new positions in a 1-second animation. The trial ends when all entities have been added.
- **Node Deletion (ND)**: Node deletion tasks successively remove nodes from the graph, causing a new layout to be calculated and an animation to the new positions to be performed. When removing a node, we remove all of its edges. When the trial ends, we again show the deleted nodes (without their edges, since the layout has changed) in the same position from where they were deleted.
- **Edge Addition (EA):** Edge addition successively adds edges to the graph during the animation. New edges have the same black color as other edges, and are added to already existing nodes in the graph. After an edge has been added, a new layout is calculated and the nodes are then animated as they move to their new positions.
- **Edge Deletion (ED)**: Finally, the edge deletion task, analogously to the other tasks, proceeds by removing edges from the graph. The nodes connected to the edge are not removed, and edges which are the only edge for a particular node are never removed. After each edge is removed, the layout is recalculated and the nodes are animated to their new positions. When all edges have been removed, we again show the deleted edges (without their nodes, since the graph layout may have changed) in the original position from where they were deleted.

After each trial, the target entities were randomly colored red, green, blue, yellow, and pink. Using these colors, the participant was then asked to reconstruct the order of addition or deletion. Because our focus is on investigating the impact of changing graph topology on perception, we specifically do *not* use any visual technique—e.g., fading, ghosting, highlighting—to draw the user's attention to changes.

4.6. Trial Generation

We use randomly generated graphs in our evaluation in order to cover the full spectrum of possible variations of different graph metrics. We specifically avoided using an actual small-world social network so as not to bias the perceptual phenomena studied in the experiment. Accordingly, all of our trials used a relatively small base graph with 25 nodes and 50 edges. However, generating representative trials was a challenge because no empirical data exists on what constitutes a typical node-link animation. We therefore used the graph metrics to guide our trial generation and selection.



(a) Fixed (NA/ND). (b) Force-directed (NA/ND).

(c) Fixed (EA/ED). (d) Force-directed (EA/ED).

Figure 1: *Correctness for task and number of entities.*

Unfortunately, our graph metrics are descriptive rather than generative, and we also do not know the standard distribution for each metric given typical trials. In other words, we have no straightforward way of generating a trial with specific metric values, and we also do not have a good understanding of which metric values are representative.

To solve these problems, we began by generating a large number of random trials by calculating new graph layouts and metrics for each trial. More specifically, we generated 10,000 trials for each task type, for each number of entities, and for each layout (240,000 trials in total) and analyzed the results statistically. We found significant correlations ($p <$ .05) only between the Node Speed and Target Speed metrics (not surprising); no other metrics were correlated.

Given all of these randomly generated trials, we wanted to randomly select three trials (one per repetition) for each experimental condition. For these trials to be representative, however, we must ensure that we choose only trials that are not abnormal (i.e., outliers) for any metric. To achieve this, we choose trials that fall within the .7 confidence interval for each metric; for the correlated metrics, we use the .7 confidence region in correlated space. Furthermore, we also want our trials to cover the full domain within each metric distribution (but inside the .7 confidence interval); we therefore randomly select differing trials. This yields 72 unique trials for our study that we use for all participants in varying order.

4.7. Procedure

Participants received training before each technique block to ensure that they knew how to solve each task type using the viewer. For each trial, participants clicked on a button to indicate they were ready to start the trial. After that they were shown the graph in its initial phase (before animation) and double-clicked on the viewport to start the animation.

As the animation ended, a dialog box appeared on the screen where the user was asked to give the order of added or removed entities using their colors. After recording the order, another dialog appeared on the screen asking the user about how much mental effort that trial required. We use a 9-point Likert scale for this question where 1 represents very low mental effort and 9 represents very high mental effort [Paa92]. This gives us another perspective of what makes these graphs difficult to perceive [HEH09].

Participants could rest between trials. A full experiment lasted 75 minutes, after which we conducted an interview.

4.8. Hypotheses

Based on our motivation (Section 4), we formulate the following hypotheses as a foundation for the follow-up studies:

- H1 *Force-directed layout will be more difficult than fixed layout.* The Force-directed layout will cause higher cognitive load and lower accuracy due to moving nodes and edges.
- H2 *Deletion tasks will be more difficult than addition tasks.* Added entities will continue to exist as time progresses, whereas deleted entities disappear permanently. Added entities are thus easier to detect, and deletion will therefore yield significantly higher load and lower accuracy.
- H3 *High numbers of entities will cause higher difficulty.* We think that higher numbers of entities will cause a higher cognitive load on the user, thereby decreasing correctness.

4.9. Results

4.9.1. Correctness

We defined correctness as whether or not event order was correctly reconstructed by the participant. We analyzed the effect of the conditions and metrics on correctness using logistic regression. We found a main effect of layout technique on correctness ($F(1,1127) = 71.64, p < .0001$); fixed layout was significantly more correct than force-directed layout. We studied graphically (Figure 1) and found the L = Fixed condition had relatively small effects, whereas L = Force-Directed had larger effects, particularly for $E = 5$.

Not surprisingly, we also found a significant main effect of entity number on the correctness ($F(2,1127) = 130.19, p < .0001$). A Tukey's HSD test revealed that all differences were pairwise significant ($p < .05$) in the order $2 > 3 > 5$ for decreasing correctness. Finally, we found no significant main effect of addition/deletion ($F(1,1127) = .98, p = .3218$), node/edge ($F(1,1127) = .09, p = .7650$), and no interaction effects on the correctness. In other words, participants were not doing significantly better or worse with any of the four task types we studied in our experiment.

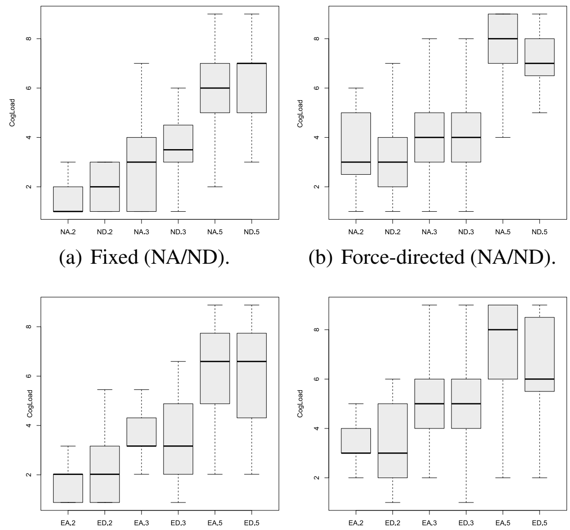
4.9.2. Cognitive Load

Recall that cognitive load was self-reported by each participant for each trial. We analyzed data using a linear mixed model [KNNL05] for analysis, treating the cognitive load measure as a continuous variable (standard for 9-point Likert scales). We found a significant main effect of layout on cognitive load ($F(1,1127) = 224.56, p < .0001$). This mirrors our results on the layout effect on correctness. We also draw the graphical plots to confirm these results (Figure 2). Similarly, we found a significant main effect of number of entities ($F(2,1127) = 629.14, p < .0001$). A Tukey's HSD test revealed that all pairwise differences between the levels were significant ($p < .05$), in the order $5 > 3 > 2$ (listed in descending cognitive load, i.e., $E = 5$ had the highest load).

One interesting finding is that there is an interaction effect between whether the task is addition or deletion and layout ($F(1,1127) = 7.13, p = .0077$). A Tukey's HSD test confirms that when the fixed layout is used, the addition task is perceived easier than the deletion task, but when force-directed layout is used, the addition task is perceived more difficult than the deletion task. Except for this interaction effect, there were no significant main and interaction factors.

4.9.3. Visual Graph Metrics

While our results above found significant main effects of the factors on both correctness and cognitive loads, the results were much more muddled for our dynamic graph metrics. We analyzed the effects on correctness using logistic regression with the metrics as part of the model, and the effects of cognitive load using linear regression. We found no significant effects for any of the metrics for any of the analyses. We did, however, find some marginally significant results ($p < .1$), which, while not strictly reliable according to inferential statistics practice, at least gives us an indication of the impact of these metrics on the graph perception task. More specifically, both of the Node Speed and Target Separation metrics have marginally significant effects on the user correctness measurements. These findings were also strongly supported by the subjective feedback from the participants.



(a) Fixed (NA/ND). (b) Force-directed (NA/ND).
(c) Fixed (EA/ED). (d) Force-directed (EA/ED).

Figure 2: *Cognitive load for task and number of entities.*

For the Node Speed metric, many participants thought that trials with a large degree of node movement are harder to perceive than ones with lesser node movement. This is an intuitive result, since it is easy to see that for two animations with the same number of entities, the one where the entities move the most (and thus the fastest) is harder to perceive than the one where they move the least.

For the Target Separation metric, many participants noted that trials where targets were scattered across the whole screen were easier to perceive and recall than those where they were grouped close together. This is somewhat counterintuitive given our previous argument where we thought that a high Target Separation would increase the difficulty because it would force the users to divide their attention over a large surface on the screen. Further interviews and observations of participants gave rise to a possible explanation: high Target Separation means that the targets each have unique spatial locations on the screen, such as "left side", "top right", "middle", etc, which allowed users to easily memorize spatial locations. For low Target Separation, the same areas on the screen often ended up being overloaded for more than one target, which made recall more difficult.

4.10. Discussion

We can summarize our findings as follows:

- Force-directed layout caused topology events to become significantly more difficult to perceive (confirming H1);

- There was no significant difference on cognitive load or correctness for the different tasks (rejecting H2); and
- The number of entities had a significant impact on both correctness and cognitive load (confirming H3).

This initial exploratory study gave us basic information on dynamic graph perception. Guided by these findings, our next step is to design a summative user study where we choose the most important graph metrics as factors, selecting particular levels from the value ranges we explored here.

Because the effects on correctness were most significant for force-directed layout, we restrict our summative study to these conditions. This is not surprising, since these were also the only trials with animated nodes and edges.

Furthermore, we also found that correctness was significantly impacted by the number of entities, with $E = 5$ seeing the most difference. This is again not surprising, and it suggests that our chosen difficulty levels were indeed representative of the full range of easy to hard tasks. Regardless, these results allow us to focus our summative study on these situations and discard conditions where $E = 2$ or $E = 3$.

Finally, our findings on the graph metrics indicate that only a few of the long list of metrics that we derived actually have a significant impact on graph perception, at least given the constraints of the present study. Based on the discussion above, we found that the target separation and the node speed metrics were the dominant ones that best predicted dynamic graph perception performance. In other words, these metrics are good candidates for further investigation.

5. Summative User Study

We follow up with a summative study where we evaluate those metrics that played a significant role in user perception for dynamic graphs in the exploratory study. For this study, the apparatus, experimental platform, task, dataset, and procedure were the same as in the previous study.

5.1. Participants

We recruited 12 paid volunteers (six male and six female), screened not to be color blind. All participants were university students with ages varying from 20 to 26 years (average 22). None of the participants from the exploratory study participated in this second summative study.

5.2. Experimental Factors

We use the following factors for this study:

- **Task Types (T):** We only used one task (node addition, NA) in this study since all the result from the exploratory study showed no significant difference for different tasks.
- **Number of Entities (E):** Based on our reasoning above, we chose the number of entities to be 5 (the level that had the highest variance in the exploratory study).
- **Layout (L):** Since the fixed layout had significantly better performance, we only used the force-directed layout here.
- **Target Separation (S):** From the exploratory study, we found that when targets are separated on the screen they are easier to perceive and recall compared to when they are close. Therefore, we adopted three different target separation ranges relative to the screen size: low (20-25% of screen size), medium (35-40%), and high (50-55%).
- **Node Speed (V):** The last study showed that the farther the nodes travel, the more difficult they are to track. Based on this argument, we introduced three ranges to be traveled for this condition relative to the size of the screen: low (10-20%), medium (30-40%), and high (60-75%).

5.3. Study Design

We used a within-subject full-factorial design: 3 separations (S) $\times$ 3 node speeds (V) $\times$ 3 repetitions = 27 conditions. We counterbalanced target separations and node speeds. With 12 participants, we collected a total of 324 trials.

5.4. Trial Generation

We used the same approach for trial generation as in the earlier study, but we selected trials using the above intervals for S and V. By using widely separated intervals for each of the factors, we hoped to be able to include very different trials in our study, which would allow us to find significant effects on the correctness and cognitive load. We used the same set of trials for all participants, but in varying order.

5.5. Hypotheses

H4 *High target separation will cause more correctness and lower cognitive load than low separation.* Our marginally significant results from the exploratory study suggest that target separation is influential for spatial recall.

H5 *High node speed will cause less correctness and higher cognitive load than low speed.* Our preliminary findings as well as intuition suggest that the amount of movement in the animation will significantly impact user perception.

5.6. Results

5.6.1. Correctness

We again defined correctness as whether the participant exactly replicated the sequence of events, and analyzed the data using logistic regression. Figure 3 shows boxplots for correctness as a function of the factors Target Separation S and Node Speed V. Target Separation had a significant main effect on the correctness ($F(2,304) = 14.99, p < .0001$).

We analyzed the pairwise differences between values for Target Separation and found that the large separation was significantly better (i.e., higher correctness) than both the

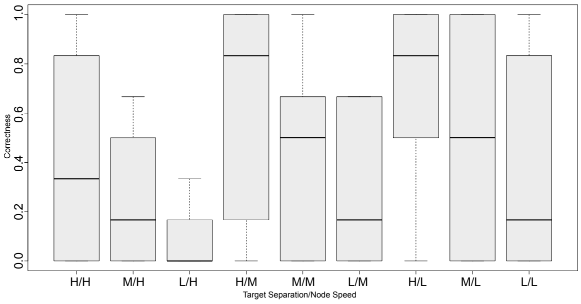


Figure 3: *Correctness as a function of Target Separation S and Node Velcity V (presented as S/V in labels).*

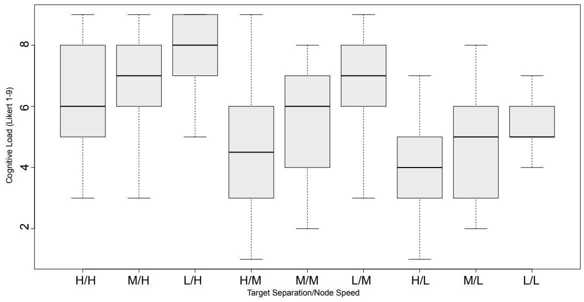


Figure 5: *Cognitive load as a function of Target Separation S and Node Speed V (presented as S/V in labels).*

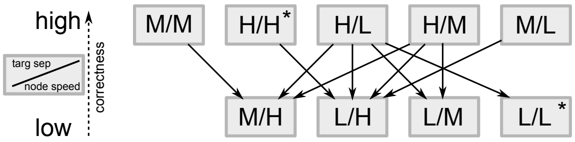


Figure 4: *Significant pairwise differences ($p < .05$) for correctness. Each box represents a combination of Target Separations and Node Velocities. Arrows signify that the source has significantly more correctness than the destination.*

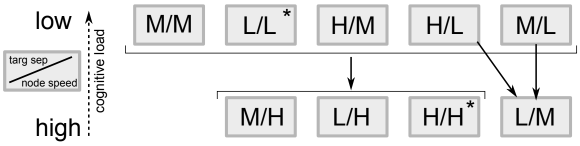


Figure 6: *Significant pairwise differences ($p < .05$) for cognitive load. Each box represents a combination of Target Separations and Node Speeds. Arrows signify that the source has significantly less cognitive load than the destination.*

low and medium values (Tukey's HSD test, $p < .05$); in addition, medium separation were significantly more correct than low ($p < .05$). Node Speed also had a significant main effect on the correctness ($F(2,304) = 11.26, p < .0001$).

A Tukey's HSD test revealed that high speeds caused significantly less correctness than both low and medium speeds ($p < .05$). Medium and low speeds showed no significant difference. We found no significant interactions between S and V. However, analyzing the pairwise differences using a Tukey's HSD test indicated that some combinations were significantly more correct than others (see Figure 4).

5.6.2. Cognitive Load

We analyzed the self-reported cognitive load measure (Likert 1-9 scale) using a linear mixed model [KNNL05]. Figure 5 shows boxplots for cognitive load as a function of S and V. There was a significant effect of Target Separation T on cognitive load ($F(2,304) = 6.17, p = .024$).

We studied the pairwise differences further using a Tukey's HSD test, and found that only high separations were perceived as having significantly less cognitive load than low separations ($p < .05$). Furthermore, we also found a significant effect of Node Speed V on cognitive load ($F(2,304) = 55.69, p < .0001$). Analysis showed that all three pairwise differences were significant, i.e., low > medium > high for increasing cognitive load (Tukey's HSD, $p < .05$).

We again found no significant interactions between the factors, but a Tukey's HSD test showed that some combinations of separation and speed were perceived as having less cognitive load than other combinations (see Figure 6).

5.7. Discussion

We can summarize our findings from above as follows:

- Target Separation S had a significant effect on both correctness and cognitive load (confirming H4); and
- Node Speed V had a significant effect on both correctness and cognitive load (confirming H5).

The latter of these confirmed hypotheses—on average node speed—is perhaps not so surprising. Higher average speed makes tracking individual nodes more difficult because the visual flux in the scene increases. This effect is reflected in both correctness and cognitive load measurements.

That increasing distance between targets makes perceiving the graph animation easier is more surprising. As noted earlier in this paper, it also runs somewhat counter to the hypotheses that low target separation would make it easier to perceive targets being added or removed because they are grouped closer together. Our explanation is the same as before: a more dominant effect is that high target separation makes it easier for participants to recall temporal events because their spatial location will become less ambiguous and easier to recall. In other words, if targets are spaced far apart, they will each have a unique position on the screen.

The stars in Figures 4 and 6 indicate the two speed vs. separation combinations—H/H and L/L—that shift places for correctness and cognitive load, i.e., they belong to the top tier in one comparison (H/H for correctness, L/L for cognitive load) and the bottom tier in the other (H/H for cognitive load, L/L for correctness). This is consistent with our observations: low speed seems to predict low cognitive load, whereas high target separation predicts high correctness.

It is interesting that the cognitive load measure clearly shows the impact of node speed, but not target separation. In other words, participants reported that high node movement caused their cognitive load to increase, which also indeed impacted correctness. However, their cognitive load ratings shows that they were *not* aware that low target separation had the same effect, as evidenced by the correctness measure.

6. Implications for Design

Our work in this paper has two main contributions: (1) the definition of visual graph metrics for dynamic graphs that form the design space for graph animation, and (2) results from empirical studies showing that two specific metrics—node speed and target separation—have a particularly strong impact on perception of animated dynamic graphs. Below we discuss what implications these findings will have on graph visualization and how these results generalize (or not).

That node speed would have a dominant effect on dynamic graph perception (H5) is already a well-known fact for animation in general [TMB02], and animated graphs in particular [BW97, CDT*92]. This result confirms the current emphasis on minimizing node movement in layout algorithms for dynamic graphs. Maintaining the stability of a dynamic graph will certainly facilitate perception of change.

The target separation finding (H4) was not previously known, and represents a useful guideline that designers of graph layout algorithms may want to take into account. In other words, beyond minimizing edge crossings and maintaining uniform edge length, algorithms may now also want to maximize the distance between entities that will appear or disappear during the animation. Doing so will make it easier for users to distinguish between these additions or deletions for when they want to reconstruct the temporal sequence. On the other hand, this would require at least some knowledge of future changes to the graph, which is not always available.

Our levels for node speed and target separation may also be useful. We found these intervals through careful data exploration after generating thousands of representative trials, and our split results clearly prove their validity: low speeds yielded high correctness, whereas high speeds yielded low correctness. The same was true for target separation. Therefore, an informal rule of thumb for node speed would be that nodes in an animated dynamic graph should ideally move less than 5% of the viewport dimension per second. For target separation, we found that even medium separation gave rise to significant improvement, so a rule of thumb would be that targets in an animated dynamic graph should ideally be separated by at least 35-40% of the viewport dimension.

7. Limitations

No evaluation can cover all aspects of a realistic task, and we had to delimit our study to make these experiments feasible to perform. For example, we did not study how visualization can be used to draw attention to the addition and deletion events in the animation. Examples include highlighting, fading, and phosphor-style effects [BTC*06]. The reason for omitting these was that we wanted this study to be focused on the purely perceptual aspects of dynamic graphs. Using visualization to improve awareness comes later.

Another potential limitation is that the format of our evaluation lended itself to recruiting only a relatively small number of participants (28 in total) in order to keep the time investment manageable. This may be the reason why some effects on dynamic graph metrics were not significant. Further studies are needed to investigate these phenomena in more detail with more participants, e.g., using crowdsourcing.

Furthermore, many of our findings are not specific to graphs and hold true for general animation. For example, that the amount of visual movement affects perception of the graph is a straightforward, if useful, result. We were disappointed not to find strong correlations between many of our graph metrics and the user performance. However, these findings are indicative of the drawbacks of using animation across the board, drawbacks that other authors have observed as well [TMB02]. While we strived to isolate our findings to those specific to dynamic graphs only, we suspect that more controlled studies are necessary in the future to find the more subtle effects that metrics such as edge sweep, edge-edge distance, and edge crossings may have on the animation.

8. Conclusion and Future Work

We have performed an in-depth quantitative evaluation of dynamic graph perception using animated node-link diagrams. Our findings suggest significant effects of node movement and target separation on user performance. These findings are consistent with existing work in the literature and suggest future ways for how designers of dynamic graph visualization software can improve their tools.

Our future work will use these findings to design better graph visualizations and to study alternative visual representations, either separate from node-link diagrams altogether, or incorporate additional visual elements to aid perception.

9. Acknowledgements

This work was supported in part by the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0003.

References

[APP10] ARCHAMBAULT D., PURCHASE H., PINAUD B.: Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Trans. on Visualization and Computer Graphics 7*, 4 (2010), 539–552. 1, 2

[BBD09] BECK F., BURCH M., DIEHL S.: Towards an aesthetic dimensions framework for dynamic graph visualisations. In *Proceedings of the International Conference on Information Visualization* (2009), pp. 592–597. 3

[BC03] BRANDES U., CORMAN S. R.: Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization 2*, 1 (2003), 40–50. 2

[BEF02] BORGATTI S., EVERETT M., FREEMAN L.: UCINET for windows: Software for social network analysis. *Harvard Analytic Technologies 6* (2002). 2

[BETT99] BATTISTA G. D., EADES P., TAMASSIA R., TOLLIS I. G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999. 2, 3

[BFN85] BATINI C., FURLANI L., NARDELLI E.: What is a good diagram? A pragmatic approach. In *Proceedings of the International Conference on the Entity-Relationship Approach* (1985), pp. 312–319. 2

[BM02] BATAGELJ V., MRVAR A.: Pajek—analysis and visualization of large networks. In *Proceedings of the International Symposium on Graph Drawing* (2002), pp. 8–11. 2

[BS90] BAECKER R., SMALL I.: Animation at the interface. *The art of human-computer interface design* (1990), 251–267. 2

[BTC*06] BAUDISCH P., TAN D. S., COLLOMB M., ROBBINS D. C., HINCKLEY K., AGRAWALA M., ZHAO S., RAMOS G.: Phosphor: explaining transitions in the user interface using afterglow effects. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (2006), pp. 169–178. 2, 9

[BW97] BRANDES U., WAGNER D.: A bayesian paradigm for dynamic graph layout. In *Proceedings of the International Symposium on Graph Drawing* (1997), pp. 236–247. 2, 9

[CDT*92] COHEN R., DIBATTISTA G., TAMASSIA R., TOLLIS I., BERTOLAZZI P.: A framework for dynamic graph drawing. In *Proceedings of the Symposium on Computational Geometry* (1992), pp. 261–270. 2, 9

[CP96] COLEMAN M., PARKER D.: Aesthetics-based graph layout for human consumption. *Software: Practice and Experience 26*, 12 (1996), 1415–1438. 2

[DG02] DIEHL S., GÖRG C.: Graphs, they are changing–dynamic graph drawing for a sequence of graphs. In *Proceedings of the International Symposium on Graph Drawing* (2002), vol. 2528, pp. 23–30. 2

[DS09] DUNNE C., SHNEIDERMAN B.: *Improving graph drawing readability by incorporating readability metrics: a software tool for network analysts*. Tech. Rep. HCIL-2009-13, University of Maryland, 2009. 2, 3

[FQ11] FARRUGIA M., QUIGLEY A.: Effective temporal graph layout: a comparative study of animation versus static display methods. *Information Visualization* (2011). 1, 2

[HEH09] HUANG W., EADES P., HONG S.: Measuring effectiveness of graph visualizations: a cognitive load perspective. *Information Visualization 8*, 3 (2009), 139–152. 5

[HR07] HEER J., ROBERTSON G. G.: Animated transitions in statistical data graphics. *IEEE Trans. on Visualization and Computer Graphics 13*, 6 (2007), 1240–1247. 2

[KBM94] KRACKHARDT D., BLYTHE J., MCGRATH C.: Krackplot 3.0: An improved network drawing program. *Connections 17*, 2 (1994), 53–55. 2

[KNNL05] KUTNER M. H., NACHTSHEIM C. J., NETER J., LI W.: *Applied linear statistical models*. McGraw-Hill, 2005. 6, 8

[LPP*06] LEE B., PARR C. S., PLAISANT C., BEDERSON B. B., VEKSLER V. D., GRAY W. D., KOTFILA C.: TreePlus: interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1414–1426. 2

[MMBd05] MOODY J., MCFARLAND D., BENDER-DEMOLL S.: Dynamic network visualization. *AJS 110*, 4 (2005), 1206–1241. 1, 2

[Noa05] NOACK A.: Energy-based clustering of graphs with nonuniform degrees. In *Proceedings of the International Symposium on Graph Drawing* (2005), pp. 309–320. 2, 4

[Paa92] PAAS F.: Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of Educational Psychology 84*, 4 (1992), 429. 4, 5

[PS08] PURCHASE H. C., SAMRA A.: Extremes are better: Investigating mental map preservation in dynamic graphs. In *Proceedings of the International Conference on Diagrammatic Representation and Inference* (2008), vol. 5223 of *Lecture Notes in Computer Science*, pp. 60–73. 2

[Pur02] PURCHASE H. C.: Metrics for graph drawing aesthetics. *Journal of Visual Languages and Computing 13*, 5 (2002), 501–516. 2

[RFF*08] ROBERTSON G. G., FERNANDEZ R., FISHER D., LEE B., STASKO J. T.: Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1325–1332. 1, 2

[TMB02] TVERSKY B., MORRISON J. B., BÉTRANCOURT M.: Animation: can it facilitate? *International Journal of Human-Computer Studies 57*, 4 (2002), 247–262. 1, 2, 9

[WPCM02] WARE C., PURCHASE H. C., COLPOYS L., MCGILL M.: Cognitive measurements of graph aesthetics. *Information Visualization 1*, 2 (2002), 103–110. 3

[YEL10] YI J. S., ELMQVIST N., LEE S.: TimeMatrix: analyzing temporal social networks using interactive Matrix-Based visualizations. *International Journal of Human-Computer Interaction 26*, 11 (2010), 1031–1051. 2