Ephemeral Paths: Gradual Fade-In as a Visual Cue for Subgraph Highlighting

Jessica Q. Dawson, Joanna McGrenere, Tamara Munzner, Karyn Moffatt[†], Leah Findlater[‡]

Department of Computer Science
University of British Columbia
{jqdawson, joanna, tmm, kmoffatt, lkf}@cs.ubc.ca

ABSTRACT

Ephemeral highlighting uses the temporal dimension to draw the user's attention to specific interface elements through a combination of abrupt onset and gradual fade-in. This technique has shown promise in adaptive interfaces, but has not been tested as a dynamic visual encoding to support information visualization. We conducted a study with 32 participants using subgraph highlighting to support path tracing in node-link graphs, a task abstracting a large class of visual queries. The study compared multiple highlighting techniques, including traditional static highlighting (using color and size), ephemeral highlighting (where the subgraph is emphasized by appearing first, and the rest of the graph fades in gradually), and a combination of static and ephemeral. The combination was the most effective visual cue: it always performed at least as well or better than static highlighting. Ephemeral on its own was sometimes faster than the combined technique, but it was also more error prone. Self-reported workload and preference followed these performance results.

Author Keywords

Information visualization, subgraph highlighting, path tracing, visual onset, interactive graph exploration, evaluation.

INTRODUCTION

A central concern in the field of information visualization, or infovis, is to characterize how nonspatial information can be effectively represented using different cues for visual encoding. For example, highlighting a subset of elements can be done by changing their color, increasing their size, moving them in small orbits, or controlling their transparency. Many real-world uses of graph visualization are supported by the abstract task of path tracing when a subset of nodes and edges—a subgraph—is highlighted. Consider a medical genetics investigator exploring a graph where nodes represent people and edges represent kinship, with nodes colored according to whether that person has inherited the genetic markers correlated with certain diseases. Highlighting the edges in a two-hop neighbourhood around a node corresponds to focusing attention on everybody within two generations of the target. Highlighting the set of shortest paths between a person and all people with the markers for a particular disease focuses attention on potential inheritance routes. Network analyses of this type are gaining prominence in a variety of domains, from retailing to policing and counterterrorism [20].

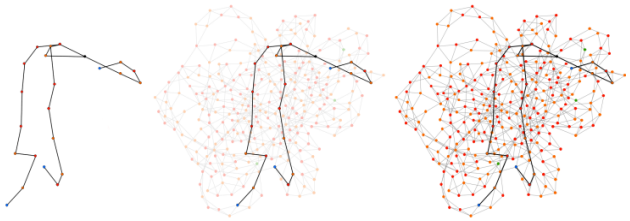

Figure 1. Time lapse shows the combined ephemeral+static highlighting technique, for a 5-hop shortest-path subgraph region. Left: The subgraph appears. Middle: The remainder fades in over a short period of time. Right: The static highlighting using color and size remains visible indefinitely.

Determining the relative efficacy of visual cues for operations such as subgraph highlighting has long been at the core of the infovis research agenda. While the traditional static cues such as position, color, size, and orientation have been under study for a long time [7], many open questions remain [16]. To date, the currently prevalent approach to subgraph highlighting is to use a combination of static color and size encoding, as shown in Figure 1 (right) and Figure 3. More recently, a new class of highlighting technique has been proposed [11]: ephemeral highlighting uses the temporal dimension to draw the user's attention to specific interface elements through a combination of abrupt onset and gradual fade-in, as shown in Figure 1. This class of technique has been studied in the context of an adaptive interface for menu selection [11], and gained higher prominence when Google released a new home page featuring gradual fade-in in late 2009 [19].

In this work, we explore the use of an ephemeral technique in an infovis setting, considering it as a type of visual cue for highlighting subgraphs to support path tracing in node-link graphs. The visual encoding and interaction issues faced when exploring node-link graphs are representative of the problems faced in the field of infovis as a whole. Most graph exploration systems support highlighting subgraphs.

[†] Moffat is now with the University of McGill

[‡] Findlater is now with the University of Maryland, College Park

Tracing paths through the connections that make up the graph is also a common task that users must perform when exploring this data type [18]. The task has been previously studied in controlled experiments, showing that dynamic highlighting techniques involving motion can outperform static visual cues [25,26].

We conducted an experiment with 32 participants to compare path highlighting under four conditions: (1) an ephemeral encoding technique, (2) a static color and size coding, (3) the combination of both techniques together, and (4) a control condition with no highlighting. We included static highlighting due to its prevalence and we included the combination of static and ephemeral because combining multiple redundant cues has often proved more effective than any single cue alone [22, 26, 27].

Our experiment also incorporated different accuracy levels for the highlighted subgraph, comparing cases where the highlighted subgraph was accurate in the sense of containing all relevant information to perform the path-tracing task, to those highlighting cases with incomplete or misleading information. Although such comparisons are now common practice in the adaptive interfaces domain (e.g. [10,12]), where this factor is known as predictive accuracy, they have not been explicitly considered in the infovis community before. Inaccurate, misleading predictions are known to have a cost in the context of predictive menus [10]. Accurate static highlighting of subgraphs is known to have a benefit in the context of infovis [26], but inaccurate highlighting has not been studied. We conjectured that ephemeral techniques may match or exceed the known benefits of static highlighting for accurate predictions, while mitigating the potential costs of highlighting the wrong items when the prediction is inaccurate.

The main contributions of this paper are (1) to propose the use of an ephemeral technique for an infovis task and (2) to study it in the context of a multi-factor controlled experiment. Our results show that the combination of ephemeral and static highlighting is the most effective visual cue: it always performed as well or better than static highlighting. Self-reported workload and preference followed these performance results. Ephemeral on its own was as fast and in some cases faster than when it was combined with static; however, our data suggests a speed-accuracy trade-off may be at play. As a secondary contribution, we also hope to encourage the infovis community to follow in the footsteps of the adaptive interface community and include predictive accuracy as a factor in future experiments.

RELATED WORK

Many interactive systems support path tracing in node-link graphs by highlighting subgraph regions. A number of these tools show the one-hop neighbourhood of direct connections to a node in response to clicking or hovering; one example is the Cerebral system [4]. Some tools support highlighting neighbourhoods of two or three hops [2], whereas larger neighbourhoods are not usually shown unless the graph is very sparse [21]. Similarly, many previous tools support highlighting the subgraph of all edges between some target node and a set of other nodes of interest, for example Tulip [2].

Much of the previous work on characterizing visual channels for encoding information has focused on static channels [7,24], establishing for example that position is the strongest cue for all data types, whereas color is more effective for nominal than for quantitative data. We focus here on the more relevant studies of dynamic channels. Bartram et al. characterized the effectiveness of different simple motions for a visual search task [5], and found that motion coding outperformed color and shape coding for detectability [6], and that anchored motions are less distracting than traveling motions [6].

Ware and Bobrow studied motion highlighting of subgraphs within a complex node-link graph. While a first study found that motion highlighting outperformed static highlighting with color and size [25], a second study that took interaction times into account found no difference between the two, but slight improvements when they were combined to redundantly code the information [26].

Although these studies shed light on the utility of the specific dynamic cues involving motion, the use of gradual onset as a dynamic visual cue has not been explicitly studied in an infovis context. Using an ephemeral technique to focus user attention was first proposed in the context of adaptive menus by Findlater et al. [11]. The premise behind their approach is that unlike abrupt onset, gradual onset does not draw attention [29]. Thus, predicted items appeared abruptly when the menu was opened, with the rest fading in gradually. Results were promising: ephemeral adaptation resulted in faster selection times than both no adaptation and adaptation with static color highlighting.

EXPERIMENTAL METHODOLOGY

We conducted a controlled experiment using path tracing tasks to compare variations of ephemeral and static subgraph highlighting and a control condition: participants reported the path length from a source node to the closest node of a specific color. We expected that the effectiveness of the techniques would differ depending on whether the highlighted subgraph region accurately captured the information required. To span performance across this space, we examined cases where the highlighted subgraph does not contain all relevant information but is partially complete, as well as cases where the highlighted subgraph is actively misleading.

To refine the task and experimental design, we conducted extensive pre-piloting and piloting. This is not uncommon for infovis experiments, because the parameter space of possibilities has not often been characterized in previous work. The pre-piloting consisted of 12 informal sessions with 8 users, for approximately 14 hours of observation. Our goal was to understand the impact of several factors

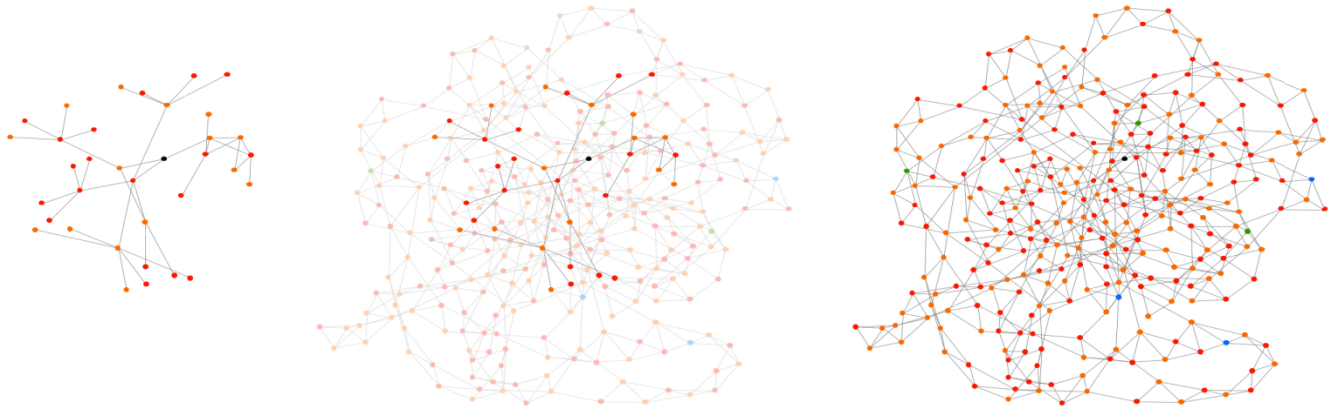

Figure 2. Time lapse shows the ephemeral subgraph highlighting technique of a 3-hop neighbourhood subgraph region. Left: The subgraph appears first. Middle: The remainder of the graph fades in over an onset delay of 10 seconds. Right: When fading is complete, no highlighting remains. Note that the rightmost graph is equivalent to the Control (no highlighting) condition.

including graph size, graph density, task difficulty, and ephemeral onset length on a user's ability to do the task. We describe key findings from the pre-piloting throughout the methodology description below. We also ran a small pilot study with 12 additional participants to get an early feel for the viability of the highlighting techniques, and to test our methodology. That study and its results are described directly before the results from the full study.

Task

The experimental task was a series of path-tracing trials where participants were presented with a laid-out small-world graph and asked to answer the question "How many hops from the source node is the closest blue node?" Each trial used a different synthetic 300-node graph with colored nodes and grey edges. See Figure 2 for an example sequence.

A black (source) node appeared first, giving participants 2 seconds to locate it before the start of the trial. Blue (target) and green (distractor) nodes had a frequency of 1% each, namely 3 nodes of each color; the remaining nodes were red or yellow, with equal frequency. There was only one *nearest* blue node and its distance from the source was between 2 and 5 hops. Participants could only answer once, and were not told if their answer was correct. After a time limit of 60 seconds, the screen was blanked and a dialog box prompted the participant for their best guess.

Rationale

Hop counting is a proxy task rather than being ecologically valid in and of itself. If the ultimate goal were to count hops, the interface described here would of course not be the most efficient approach. Rather, our goal is to measure the extent to which path tracing is affected by subgraph highlighting, and for the purposes of a laboratory experiment we needed a task with a simple answer space; that is, one for which the time to communicate the answer would not dominate in the results. Moreover, we needed to ensure that users could not answer the question based on preattentive popout alone; for example by spotting a node of a particular color, rather than actually tracing paths. We drew inspiration from the approach of Ware and Bobrow [26], who used questions such as "Is there a red node within two links of the target?" In our case we asked the user to give the numerical answer of the number of hops to a colored node, rather than a yes/no answer, to decrease the chance of a guess being correct.

We also echo the design philosophy of Ware and Bobrow that path tracing and subgraph highlighting are an abstraction that encompasses a large class of real-world problems: common visual queries where the user is conducting a visual search for some pattern of nodes and/or links [25]. Nearly every system that supports interactive graph exploration has some form of subgraph highlighting, ranging from generic systems for graphs with particular properties like small-world networks [23] to domain-specific systems such as MatrixExplorer [17] designed to meet the needs of social science researchers using participatory techniques. For example, MatrixExplorer uses static visual cues for highlighting interactively chosen selections to provide linking between a node-link and a matrix graph view.

During pre-piloting we assessed how large of a neighbourhood to highlight, how many hops to use for the target distance, and the number of the target and distractor nodes. We eliminated distances of 6 hops or more from consideration because participants often gave up or had very high error rates. Unsurprisingly, the task was easier at distances of 2 or 3 hops from the source node than at distances of 4 or 5 hops, and easier for nodes directly connected to the highlighted subgraph via 1 hop than those that were 2 or more hops away. When more than 3 target and distractor nodes were used the task time was dominated by double-checking the answer, whereas the task was too easy with just 1 or 2 candidates.

Dataset and Graphs

Following the arguments of Auber et al. [3] and others, we used the Watts-Strogatz model to create small-world graphs [28]. We tuned the Watts-Strogatz parameters during pre-

piloting; we used degree-4 nodes in the initial circle lattice, and a 10% probability of random reattachment. We chose a graph size of 300 nodes and 600 edges as the best balance of difficulty and density [21]. To lay out graphs, we used the very straightforward force-directed placement built into the Prefuse toolkit [15]. We ran the force-directed layout for 5 seconds for each graph. To ensure all graphs were similarly sized on the display, we accepted only those with an aspect ratio of 0.8–1.12, discarding the rest.

Rationale

While we considered using real-world data, we wanted to use a fresh graph for each trial to avoid undesired learning effects. Since it would have been difficult to find sufficiently isomorphic datasets for this type of repetitious laboratory experiment, we chose to use synthetic data. Some previous experiments have used random synthetic graphs [14]; however, we wanted to use graphs with properties more characteristic of real infovis applications. Hence we used the Watts-Strogatz model.

In pre-piloting, we tested graphs ranging in size from 200 to 1000 nodes. We wanted to avoid the problem reported by Ware and Bobrow [26], where the difficult tasks were too difficult, and had shorter times than the easier tasks because the users gave up. With a size of 300 nodes and 600 edges, participants could typically complete the most difficult no-highlighting cases without giving up and within one minute; in the easier cases where the answer was highlighted in some way, users could typically answer within 20 seconds.

Although many more sophisticated methods than force-directed layout have been proposed, such as multilevel [1] or constraint-based [9] approaches, for data sets of sufficiently large size even the most cutting-edge techniques still suffer from extreme visual clutter from overlaps and crossings between the nodes and edges. Our usage scenario is that the laid-out graph suffers from enough visual clutter that highlighting a subgraph helps the user track some path of interest through the graph. This scenario holds for both large graphs laid out with sophisticated methods, or for smaller graphs laid out with more straightforward methods. We chose the latter to simplify the experiment. We thus argue that although a graph of 300 nodes and 600 edges may sound small compared to the size of real-world datasets, the complexity of its visual appearance was carefully tuned to adequately represent the information density of complex situations while still allowing for controlled experimentation.

Experimental Factors

We included three experimental factors: subgraph region, highlighting technique, and predictive accuracy.

Subgraph Region

With the neighbourhood subgraph (`Nhood`) condition, nodes and edges within three hops of the source were highlighted, as shown in Figure 2. In the shortest-path subgraph (`Spath`) condition, the nodes and edges between the source and nodes of a particular color (blue or green) were highlighted, as shown in Figure 3.

Highlighting Technique

We included four techniques: Control (`Ctrl`) had no highlighting; Static (`Stc`) emphasized the predicted area by circling nodes and making edges thicker and darker; Ephemeral (`Eph`) emphasized the predicted area by having it appear first, with the rest of the graph appearing gradually over 10 seconds; and Ephemeral+Static (`Eph+Stc`) combined the two cues. Figure 1 shows an example of `Eph+Stc`, Figure 2 shows `Eph`, Figure 3 shows `Stc`, and the rightmost graph in Figure 2 shows `Ctrl`. The onset time of 10 seconds was determined through pre-piloting; onset times of 12 to 15 seconds were found to be disruptive because non-subgraph regions took too long to become distinguishable, and onset times of less than 8 seconds were deemed too fast to be helpful.

Predictive Accuracy

In the accurate prediction (`AccP`) condition, the highlighted subgraph contained all information required to complete the task, while for the inaccurate prediction condition (`WrgP`), the answer was outside the emphasized subgraph. Thus, for `Spath`, an accurate prediction meant that all paths to blue nodes were highlighted, whereas all paths to green nodes were highlighted for an inaccurate prediction. For an accurate prediction with `Nhood`, the blue node was within the highlighted 3-hop neighbourhood. Studies with adaptive interfaces have shown that predictive accuracy can impact user behavior and performance [10,12]. Testing an equal number of accurately predicted and inaccurately predicted trials results in an overall prediction accuracy of 50% from the user's point of view, and is a sensible threshold for a first exploration of accuracy in an infovis context.

The two subgraph region factors were designed to test two different types of inaccurate predictions: incomplete vs. misleading information. In the `Nhood` case, the number of hops highlighted in the inaccurate condition represents an underestimate of the neighborhood size required to carry out the task. In this case, the inaccurate prediction provides partial information that could accelerate the user's search despite being incomplete, but less so than the accurate prediction. In the `Spath` case, the inaccurate prediction was intended to have a higher cost, actively misleading the user by highlighting paths to the wrong places.

Interface

Users were not allowed to interact with the graph, for example by zooming or panning, because we did not want interaction time to be a confounding variable in the experiment. Pre-piloting tests showed that node-edge crossings caused confusion because it was ambiguous whether the edge terminated at the node or continued underneath it. In many interactive graph exploration systems, users resolve this well-known visual ambiguity by briefly moving the node to see whether the edges stay attached to it, or are left behind. To resolve the ambiguity

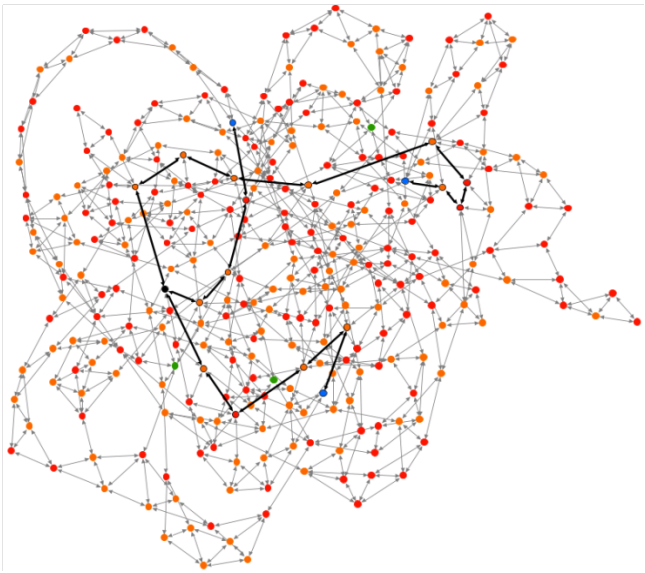

Figure 3. Static highlighting technique, where the entire graph appears at once. A 5-hop shortest-path subgraph with arrowheads visible is shown here.

without introducing noticeable interaction time costs, we allowed users to toggle on or off arrowheads showing the ends of each edge. Typical usage was that users turned them on briefly to remove ambiguities, but left them off most of the time to minimize visual clutter.

Design

We ran two experiments in parallel for the subgraph conditions and analyzed their data separately. Statistical comparison would not have been appropriate due to inherent differences in the subgraph conditions, which would have hindered meaningful interpretation of the results: notably, for `Nhood`, accurate predictions have path lengths of 2 or 3 and inaccurate predictions have path lengths of 4 or 5; in contrast, answers are evenly distributed for `Spath`.

Thus, we assigned half the participants to each of the subgraph conditions (`Nhood`, `Spath`), and used, for each, a 2-factor within-subjects design with four levels of *highlighting technique* (`Ctrl`, `Stc`, `Eph`, `Eph+Stc`) and two levels of *predictive accuracy* (`AccP`, `WrgP`). Presentation order of the highlighting techniques was counterbalanced using a balanced Latin square, and an order was randomly assigned to each participant. Target nodes were spread evenly across the answer space of 2 to 5 hops from the source node.

Dependent Measures

Our quantitative measures were task completion time and errors. Time was recorded as the median trial time from the initial graph appearance to the keystroke entry of an answer. Error rate was calculated as the percentage of incorrect answers. Our qualitative measures were self-reported confidence, workload, and a comparative ranking. Confidence was recorded after each trial using a scale from 1-low to 3-high. Workload was assessed after each highlighting technique using the 20-point NASA-TLX subscales for mental demand, physical demand, temporal demand, effort, performance, and frustration.

Analysis

We analyzed trial completion time using a 2×4×4 (`accuracy × highlighting × presentation order`) repeated measures ANOVA for each subgraph condition. For the error data, ANOVAs were not appropriate because the data violated assumptions of normality. Thus, for error, preference, and confidence data we performed separate non-parametric analyses for each factor of interest, using Friedman tests with Wilcoxon Signed Ranks tests for pairwise comparisons. We applied Bonferroni adjustments to all pairwise comparisons to protect against Type I errors. In addition to statistically significant results ($p < .05$), we note areas where a possible trend ($p < .10$) warrants further investigation. We also report partial eta-squared (η^2), a measure of effect size. As a guideline, $.01 < \eta^2 \leq .06$ is a small effect; $.06 < \eta^2 \leq .14$ medium; and $\eta^2 > .14$ large [8].

The notion of predictive accuracy does not apply to `Ctrl` because `Ctrl` provides no highlighting. In the `Spath` condition, target node distances were evenly spread across `AccP` and `WrgP` trials, so we used the overall average of `Ctrl` when comparing it to the highlighting conditions. For the `Nhood` subgraphs, `AccP` and `WrgP` trials used different path lengths, so we averaged only those `Ctrl` trials with the corresponding path lengths for each level of predictive accuracy: `AccP` (2-3 hops) and `WrgP` (4-5 hops).

Procedure

The study was designed to take no more than 2.5 hours. To start, participants filled out a background questionnaire. They were then given an overview of the task. For `Spath`, participants were told that the system would highlight the shortest-path to either all the blue nodes or to all the green nodes. For `Nhood`, they were told that the system would always highlight a 3-hop neighbourhood around the source and that target node may or may not be inside this neighbourhood. Participants were not told how frequently these behaviors would occur, but were told that the answer would always be between 2 and 5.

The experimenter then briefly explained the highlighting behavior for each condition, and had participants perform two training trials with each highlighting technique. After each practice trial, participants were told whether or not they answered correctly, and were shown the correct path to the answer. After training, participants completed 4 blocks of trials with each technique. Before each new highlighting technique, participants were given an additional 2 practice trials as a refresher. Within each block, there were 2 trials for each possible distance for a total of 8 randomly ordered trials. Each participant thus did 32 trials per highlighting technique, 128 trials in total.

Participants took a 1-minute break halfway through and a 2-minute break at the end of each highlighting technique

condition. Between techniques they also completed subjective questionnaires, including the NASA-TLX. At the end of the study, they ranked all four highlighting techniques, and completed a post-experiment interview.

Apparatus

The experiment was coded in Java using Prefuse [15]. It was conducted on a 2.53 GHz Intel Dual Core Apple laptop with 4 GB of RAM, using an external keyboard and 27" monitor with 1920×1200 resolution. The system recorded all timing and error data, and self-reported confidence levels. All graphs were generated and laid out in advance, for the twin benefits of reduced wait times for participants and reproducibility. All participants saw the same set of graphs: presentation order was randomized across subjects with the exception of the training set, which was presented in the same order for everybody. We pre-generated 128 graphs for the study, plus 16 graphs for training.

PILOT STUDY

This proof-of-concept study followed a shortened form of the above methodology, and fit into a 1.5-hour session. The 12 participants completed 2 blocks with each highlighting technique (instead of 4), using the `Nhood` subgraph condition only, and a separate pre-generated set of 64 graphs.

Results

The results were promising, suggesting that highlighting increased speed, decreased errors, and was preferred to `Ctrl`. Neither `Eph` nor `Eph+Stc` performed significantly worse than `Stc` on either speed or error rate. Overall, `Stc` was faster than `Ctrl`, and a trend suggested that `Eph+Stc` was faster as well. `Eph's` speed was not statistically different from any of the interfaces. However, Figure 4 suggests a tradeoff: its speed appears comparable to the other highlighting conditions for accurate predictions, but was somewhat slower for inaccurate ones, as such, it was no faster than `Ctrl` overall. In terms of errors, the highlighting techniques generally reduced errors compared to `Ctrl` (though for `Eph` vs. `Ctrl`, this was only a trend).

The statistics to support the claims above are as follows. For time, there was a main effect of highlighting technique ($F_{3,24} = 7.31$, $p = .001$, $\eta^2 = .477$) and a main effect of accuracy ($F_{1,8} = 81.3$, $p < .001$, $\eta^2 = .910$), but no interaction between highlighting technique and accuracy (nor any main or interaction effects with presentation order). Pairwise comparisons with `Ctrl` confirmed a significant difference for `Stc` ($p = .047$) and a trend for `Eph+Stc` ($p = .087$). Non-parametric analysis of the error data revealed a main effect of highlighting technique for `AccP` ($p < .0001$), but not for `WrgP` ($p = .136$). Pairwise comparisons with `Ctrl` on `AccP` trials showed significant differences for `Stc` and `Eph+Stc` (both $p = .018$), and a trend for `Eph` ($p = .078$).

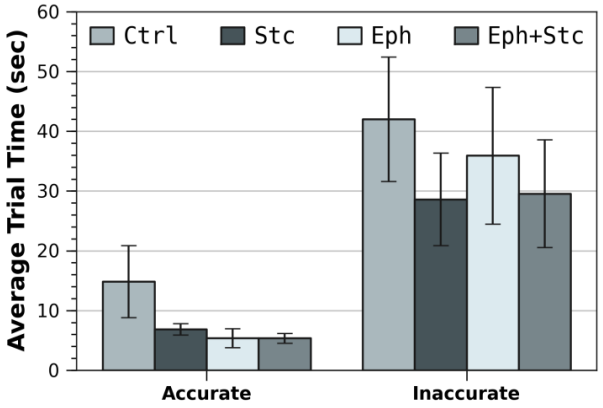

Figure 4: Average median trial time by highlighting technique and predictive accuracy for `Nhood` task in the pilot study ($N = 12$). Error bars show 95% Confidence Intervals.

As shown in Figure 4, variability was large for inaccurate predictions, suggesting individual differences were at play. From our post-study interviews, we found that participants used different strategies when the target was outside the neighbourhood (i.e., for `WrgP` trials), particularly for `Eph`. Most reported counting back from blue nodes towards the highlighted area, which worked well for `Stc`, but `Eph` faded too quickly for this to be helpful (as noted by one participant). With `Eph` a few participants reported either counting out from or trying to memorize the edges of the highlighted region. One participant explained that the onset helped with the counting-out approach because it gradually "added to the search area." Participants who adopted these alternatives did better with `Eph`. Moreover, widespread adoption of a non-optimal strategy for `WrgP` might explain the somewhat poorer performance of `Eph` in that case.

Finally, we note that subjective preferences were consistent with the performance results, but particularly encouraging for `Eph+Stc`. All participants ranked `Ctrl` last. Between the highlighting techniques, `Eph+Stc` was most preferred by 7 participants, `Stc` by 4, and `Eph` by 1. Non-parametric analysis confirmed `Ctrl` was least preferred (main effect: $p < .0001$; all three pairwise comparisons with `Ctrl` were $p = .012$; all others were not significant).

In addition to informing our final hypotheses, the pilot results triggered a change in our training procedure as described in the next section. We expected that this change in training would increase the effectiveness of the ephemeral technique on its own.

FULL STUDY

In light of the individual differences in strategy observed in the pilot study, we chose to instruct participants on the most effective strategies for each highlighting technique. Specifically, participants were told that, for `Stc`, counting back from blue nodes was an effective strategy, while for `Eph` and `Eph+Stc`, they were told that counting out from the highlighted region was likely to be more effective.

Participants

We recruited 32 participants from fliers posted on campus (20 female, aged 19–56, median = 25). All had normal or corrected-to-normal vision and regular color vision, and all used a computer for at least 3 hours per week. They received $10 per hour of participation.

Hypotheses

Our premise was that Eph and Eph+Stc would offer benefits over Stc when highlight predictions were accurate, but that Eph on its own would not hinder performance as much as Stc when predictions were wrong. Although we speculated about how Eph and Eph+Stc would compare to each other, for simplicity we formalize only our strongest predictions here, which compared the two new techniques to Stc. To replicate the previous finding in infovis that shows accurate static highlighting improves performance over no highlighting [25,26], we structure our hypotheses to compare Stc to Ctrl, then Eph and Eph+Stc to Stc, for both accurate and inaccurate cases:

Nhood:

H1. Stc results in better performance than Ctrl in both accurate and inaccurate cases.

H2. Eph+Stc results in better performance than Stc for the accurate case, and is not worse for the inaccurate case.

Spath:

H3. Stc results in better performance than Ctrl for the accurate case, but worse performance than Ctrl for the inaccurate case.

H4. Eph+Stc and Eph result in better performance than Stc in the accurate case, but in the inaccurate case: (1) Eph+Stc performs no better than Stc, and (2) Eph performs better than Stc.

H1 and H2 are based on the pilot study results and our initial rationale for testing ephemeral highlighting. For the Spath task, we predicted that persistent inaccurate highlighting could be detrimental to performance (i.e., with Stc and Eph+Stc), but that Eph should mitigate that negative effect. Since we never intended to directly compare the two subgraph conditions, we make no formal predictions about Nhood versus Spath.

Results

The average speed and error rates for each highlighting technique and subgraph condition are shown in Figure 5. We interleave the subgraph results for ease of presentation.

Speed - Nhood

All highlighting conditions were faster than Ctrl. When the target node appeared inside the neighbourhood subgraph, Eph and Eph+Stc were both faster than Stc. As expected, highlighting technique and accuracy both impacted the speed with which participants completed the task (main effect of highlighting technique, $F_{3,36} = 31.8$, $p < .001$, $\eta^2 = .726$; main effect of predictive accuracy, $F_{1,12} = 114.1$, $p < .001$, $\eta^2 = .905$). There was also a trend suggesting that whether the target node was inside the highlighted subgraph impacted speed differently based on the highlighting technique used (interaction of predictive accuracy and highlighting technique, $F_{3,36} = 2.6$, $p = .067$, $\eta^2 = .178$). There were no main or interaction effects with order.

We examined the pairwise comparisons for prediction accuracy and highlighting technique to test our main hypotheses (for a justification for pairwise comparisons on trend-level effects, see Games [13]). The comparisons revealed: (1) all highlighting conditions were significantly faster than Ctrl (AccP: all $p < .001$; WrgP: all $p < .02$), and (2) for AccP, Eph and Eph+Stc were faster than Stc (both $p < .01$). No other significant differences were found.

Speed - Spath

Eph was fastest and Ctrl slowest when the correct path was highlighted, with no differences for inaccurate highlighting. Similar to the Nhood results, highlighting technique and accuracy both impacted the speed at which participants completed the task (main effects of highlighting technique, $F_{3,36} = 13.6$, $p < .001$, $\eta^2 = .532$, and predictive accuracy $F_{1,12} = 103.4$, $p < .001$, $\eta^2 = .896$). Also, as hypothesized, speed with the highlighting techniques varied depending on whether the target node was highlighted (interaction between predictive accuracy and highlighting technique, $F_{3,36} = 29.0$, $p < .001$, $\eta^2 = .707$).

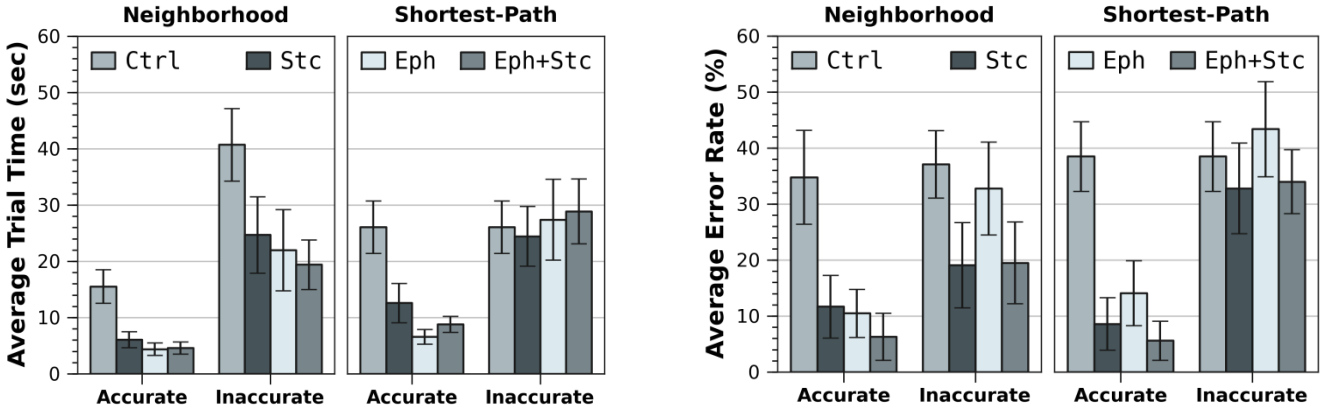

Figure 5: Performance by highlighting technique and predictive accuracy for Nhood ($N = 16$) and Spath ($N = 16$). Left: Average median trial time. Right: Average error rates. Error bars show 95% Confidence Intervals.

Pairwise comparisons showed that for AccP, Ctrl was significantly slower than Stc, Eph, and Eph+Stc (all $p < .001$), and Eph was significantly faster than both Stc and Eph+Stc (both $p < .01$). None of the other pairwise comparisons for AccP or WrgP were significant.

Unexpectedly, our results also showed an interaction of order and highlighting technique ($F_{9,36} = 2.76$, $p = .015$, $\eta^2 = .408$), and a trend for an interaction between accuracy, order, and technique ($F_{9,36} = 2.13$, $p = .052$, $\eta^2 = .348$). We investigated the 3-way interaction and found that for accurate predictions, the overall pattern of results presented above held. For inaccurate predictions, however, Eph performed more poorly when presented first, but there is nothing in the data to suggest that this is more than a fluke.

Errors

The error results largely followed the results for speed for both Nhood and Spath, with the exception of Eph having more errors than the other highlighting conditions some of the time. Where the highlighting techniques were faster than Ctrl, they also tended to have fewer errors. One deviation from this pattern is with Eph: though Eph was faster than Ctrl for Nhood-WrgP, it did not result in fewer errors. Further, there was a trend of Eph having more errors than the other two highlighting conditions for Nhood-WrgP, and trend for Eph having more errors than Eph+Stc for Spath overall (collapsing across accuracy), as shown in Figure 5. This does raise concerns. The fact that Eph was the fastest technique in the Spath-AccP case, but has more errors than Eph+Stc, suggests that a speed-accuracy trade-off may be at play. Additional research is needed to confirm this trend and characterize its nature.

The statistics follow. For Nhood, there was a main effect of highlighting technique on errors for both AccP and WrgP (both $p < .001$); for Spath, there was only a main effect for AccP ($p < .001$). Pairwise comparisons showed Ctrl had more errors than Stc, Eph, and Eph+Stc for Nhood-AccP (all $p < .02$) and Spath-AccP (all $p < .005$), and more errors than Stc and Eph+Stc for Nhood-WrgP (both $p < .01$). In addition, there was a trend suggesting that for Nhood-WrgP, Eph resulted in more errors than Stc ($p = .067$) and Eph+Stc ($p = .055$). Collapsing across accuracy for Spath shows a main effect of highlighting technique ($p < .001$), with a trend of Eph having more errors than Eph+Stc ($p = .06$).

Confidence

Highlighting led to greater confidence except when predictions were wrong in Spath, as expected. Highlighting technique and accuracy both impacted participants' confidence in their answers: main effects of highlighting technique on confidence for Nhood-AccP, Nhood-WrgP, and Spath-AccP (all $p < .001$). In all of these cases, pairwise comparisons showed that Ctrl resulted in lower confidence than each highlighting technique. (Except for Stc in Nhood-WrgP, where there was only a trend, $p = .072$, all other comparisons with Ctrl were $p < .02$.) There were no differences among the highlighting techniques themselves except that Eph resulted in significantly lower confidence than Eph+Stc ($p = .012$), and a trend of lower confidence than Stc ($p = .066$), both for Spath-AccP.

Preference and Workload

Preference and workload did not differ appreciably between the two subgraph conditions; thus, for brevity, we collapse them in the following analyses.

Eph+Stc was most often preferred, while Ctrl was the least preferred. When participants were asked to rank order the highlighting techniques, Ctrl was selected as least preferred by 27 of the 32 participants, with Eph selected by the remaining 5. The most preferred choice was split: 20 preferred Eph+Stc, 9 Stc, and 3 Eph. Statistical analysis confirmed a partial ordering, with Eph+Stc and Stc ranked first, Eph second, and Ctrl last. (A Friedman test showed a significant main effect of highlighting technique, $p < .001$. All pairwise comparisons were significant, $p < .01$, except between Eph+Stc and Stc, $p = .141$).

In terms of workload, Eph+Stc and Stc were best, while Ctrl was worst. We collapsed the NASA-TLX subscales into a single measure of workload (Cronbach's alpha = .815). As expected, our analysis showed that highlighting technique had an impact on participants' reported workload, and showed no impact of order. Ctrl had a significantly higher workload than each highlighting technique, and Eph had a higher workload than both Stc and Eph+Stc. (A 4x4 (highlighting × presentation order) repeated measures ANOVA showed a main effect of highlighting, $F_{2.3,64.3} = 28.6$, $p < .001$, $\eta^2 = .505$. All pairwise comparisons were significant, $p < .01$, except between Stc and Eph+Stc, $p = .287$). Although the pattern of results was consistent across the subgraph conditions, the average workload for Spath was 3 points higher than for Nhood.

Summary

We summarize the results in terms of our hypotheses.

Nhood:

H1. *Supported*. Replicating previous work [25,26], Stc resulted in better performance than Ctrl in terms of speed and error rate. Our new finding is that this result holds regardless of predictive accuracy.

H2. *Supported.* When the target node was within the subgraph, Eph+Stc was faster than Stc and no different in terms of errors. As predicted, there were no differences for the inaccurate case. Also, Eph followed the same pattern as Eph+Stc, except for a trend on errors in the inaccurate case suggesting it performed worse than Eph+Stc and Stc.

Spath:

H3. *Partially supported.* As hypothesized, when the path to the target node was highlighted, Stc was faster and had lower error rates than Ctrl. Although we expected

Stc to perform worse than Ctrl in the inaccurate case, it did not.

H4. *Partially supported.* Eph was faster than Stc (with no difference in errors) in the accurate case, but contrary to our hypothesis, Eph-Stc was not different from Stc. Also, while Eph was overall fastest in the accurate case, Eph+Stc had less errors than Eph. Although we had expected differences in the inaccurate case, none were found.

DISCUSSION

Ephemeral and static combined is the best choice; ephemeral on its own is fast, but sometimes more error prone. Combined highlighting (Eph+Stc) performed no worse than static highlighting (Stc) in any condition, and outperformed it for accurate predictions in the neighbourhood subgraph condition. Participants also reported a strong preference, lower workload, and more confidence using the combined technique relative to the others. On its own, ephemeral highlighting (Eph) performed just as well as the combined technique in the accurate neighbourhood subgraph condition, but while it was as fast as the combined technique in the inaccurate condition, it trended towards more errors. Ephemeral was actually the fastest technique for accurate predictions in the shortest-path subgraph, but again, it trended to more errors than the combined technique, suggesting a speed-accuracy tradeoff.

Ephemeral highlighting on its own holds promise as a visual cue for the infovis domain. Even though ephemeral highlighting alone did not clearly outperform the other two highlighting techniques, it still has promise as a viable visual cue for encoding abstract information. First, we note that the combination of multiple cues has often been found to outperform a single cue [25, 27]. Clearly Eph+Stc uses more than one cue, but Stc is also a combination of cues, namely, color and shape. Thus it remains as future work to see if Eph is competitive when compared to a single cue. Second, visual cues do not have to outperform all known cues to be of interest since the goal in an infovis context is typically to encode multiple visual variables at once; if the strongest cues such as position, color, and size are already in use for the most important variables, then an ephemeral technique could still be used for a less important variable, or as interactive highlighting that does not impede the use of color coding to show something else. Future work thus should include characterizing whether the ephemeral technique that we tested is in fact a separable rather than integral channel with respect to the other major visual cues [24].

Misleading highlighting did not impair performance. The result that accurate highlighting improves performance for all three highlighting types was an expected baseline. Indeed, our finding that static highlighting outperforms no highlighting for accurate predictions, specifically replicates previous work [26]. Our finding that incomplete yet correct highlighting (Nhood-WrgP) accelerates performance for all three techniques is a new result, but was also expected. However, we were surprised that misleading predictions (Spaths-WrgP) did not impair performance in this infovis-oriented experiment, even though they did for predictive menus [11]. For misleading predictions, our hypothesis that Stc would perform worse than Ctrl was not supported: all three highlighting techniques resulted in performance equivalent to, but no worse than, no highlighting.

Predictive accuracy needs consideration. The surprising result for misleading predictions underscores our contention that predictive accuracy should be considered a factor of interest by the infovis community in future work. It raises an intriguing question that extends beyond ephemeral visual cues in particular, of whether misprediction using visual cues for infovis in general might be less dangerous than with adaptive interfaces. Future work could also explore to what degree the benefits of the highlighting conditions we studied here will differentially change when using higher or lower predictive accuracies than the 50% threshold of our study.

CONCLUSIONS AND FUTURE WORK

We ran a controlled experiment to examine user performance, confidence, and preference when tracing a path through a node-link graph, a representative task for the infovis domain. We compared three subgraph highlighting conditions to a no-highlight control condition: (1) using the traditional static visual cues of color and size, (2) a dynamic ephemeral cue of abrupt onset for the subgraph followed by the gradual fade-in for the rest of the graph, and (3) the combination of both. We also incorporated the factor of predictive accuracy, where an accurate choice of the region to highlight emphasized all of the information required to accomplish the task, versus an inaccurate choice of region that was either incomplete or actively misleading.

Our findings show that the combination of ephemeral and static highlighting outperformed static highlighting on its own. Ephemeral highlighting alone was sometimes faster, yet we caution that a speed-accuracy tradeoff may be at play.

In addition to the specific extensions outlined above, there remain many avenues for further extending this research.

Future work should explore user interaction. In our study, users could not interact with the graphs, but in infovis systems for data exploration, highlighting is typically done in response to user selections. One open question is how to use this particular ephemeral technique effectively in an interactive system; for example, many participants requested the ability to pause fade-in on demand. Allowing for user-controlled immediate completion of the fade-in may also be useful. Another question worth exploring is whether alternate techniques could be designed that incorporate abrupt onset and gradual fade-in in a different way than our experiment.

An additional avenue for exploration is the combination of ephemeral and static highlighting in adaptive interfaces. Findlater et al. [11] ephemerally highlighted a subset of items in pull-down menus and found that it improved performance over a standard menu and over persistent color highlighting of menu items. Our results suggest that a combination of ephemeral and persistent highlighting would be even more effective than ephemeral on its own.

Finally, though we have argued throughout this paper that path tracing and subgraph highlighting constitute a representative microcosm of infovis issues, future work should of course explore a broader range of tasks and data types to verify the generalizability of our results.

REFLECTIONS

After carrying out this study, we have begun to understand and appreciate the significance of a few key differences between adaptive interfaces and information visualization systems. In this section we discuss these lessons learned, and why successfully applying ephemeral highlighting to an interactive visualization system may not be as straightforward as we originally imagined.

Menus vs. exploration

Ephemeral highlighting was originally developed as a technique to solve problems common to adaptive interfaces [11]. Typical adaptive menu tasks are repetitive and tend to assume that the user has a clear, specific target in mind. For example, in the menu task used in [11] a user would need to repeatedly select the same specific menu items in order to perform common actions. In theory, this sort of repeated use of the menus both helps the users to learn the location of items in the menu, and allows the system to build a reliable predictive model based on what the user selects most frequently. This scenario is starkly different from the common infovis task of exploration. Users typically use a visualization system to explore their data when they do not yet know what specific item or feature it is that they are trying to find. Exploration tasks therefore only last for as long it takes the user to find something of interest and such a task rarely bears repeating.

Spatial location

An important part of ephemeral highlighting is the consistency of spatial location. In the menu example users are working with absolute locations, enabling them to build a static model of the locations of menu items onscreen. Here ephemeral highlighting helped users draw on this spatial memory to find mispredicted items quickly, thereby reducing the impact of misprediction [11]. In contrast, spatial navigation of a data set tends to be in support of larger and more abstract learning goals like making comparisons. These sorts of tasks heavily rely on the relative spatial locations of data points, but have little use for absolute locations. As a result, spatial consistency does not support exploration tasks to the same degree as tasks where quickly and repeatedly locating an item is central.

Implications of temporal duration

In the context of menus [11] the ephemeral highlighting fade-in occurred very quickly, over just 250ms, and was only employed to facilitate a smooth transition between two stages: the filtered set of items and the non-filtered set. However, the path tracing task in the present study took the user much longer to complete, forcing us to increase the length of the fade-in to 10 seconds. The longer fade-in created a third stage where participants worked during the animation, using both the filtered set and the transparent non-filtered set at the same time. As a result, transparency became a significant component of the technique where it was not before.

Ephemeral vs. transparency as a visual dimension

The original definition of ephemeral highlighting was the use of abrupt onset for highlighted content followed by automatic fade-in of the rest of the material. When we consider this technique through the lens of infovis, in terms of visually encoding abstract information through visual dimensions including color and transparency [24], the question arises how the ephemeral technique is different from the use of transparency as a visual dimension. This distinction becomes troublingly murky when we consider directions for future work that would integrate user interaction.

Participant requests for more control over the transparent stage of the technique in our study suggests that they found the transparent stage the most useful, and also calls into question the benefits of an automatic fade-in. Such evidence certainly seems to favor introducing interactivity by replacing the automatic fade-in with user control. But if we follow this route and allow users to interact with the technique by pausing the fade-in, then one could argue that the technique is the same as interactive static transparency. Or if the user could cause the fade-in to automatically complete, one could similarly argue that the technique is the same as interactive filtering. Without the automatic aspect, ephemeral highlighting becomes difficult to separate from transparency in a visualization context.

Predictive accuracy in infovis

Although predictive accuracy is a central concern in adaptive interfaces, it had not been previously studied in an infovis context. Our study made the simplifying assumption that a predictive model could highlight neighborhoods or shortest paths based on some knowledge of what the user would need. In the adaptive interface community, such predictive models are standard. At first glance it might seem straightforward to build a similar model for information visualization systems, but the broad scope of interactive exploration makes this job a very difficult one. Can a predictive infovis system determine what path users need when they rarely repeat the same task twice? In this highly mutable context effective predictions in response to a user selection would require a very complex model of user behavior and cognition, an aspirational goal that may

require decades of work from the infovis research community before it reaches fruition.

ACKNOWLEDGMENTS

This work was supported in part by an NSERC USRA.

REFERENCES

1. Archambault, D., Munzner, T., & Auber, D. 2007. TopoLayout: Multi-level graph layout by topological features. *IEEE TVCG*, 13(2): 305–317.
2. Auber, D. 2003. Tulip: A huge graph visualisation framework. In (Mutzel, P. & Jünger, M. eds) *Graph Drawing Software* , 105–126.
3. Auber, D., Chiricota, Y., Jourdan, F., & Melancon, G. 2003. Multiscale visualization of small world networks. In *Proc. InfoVis'03*, 75–84.
4. Barsky, A., Munzner, T., Gardy, J., & Kincaid, R. 2008. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE TVCG (Proc InfoVis'08)*, 14(6): 1253–1260.
5. Bartram, L. & Ware, C. 2002. Filtering and brushing with motion. *Inf Vis J,* 1(1): 66–79.
6. Bartram, L., Ware, C., & Calvert, T. 2003. Moticons: Detection, distraction and task. *Int J Hum Comput Stud*, 58(5): 515–545.
7. Cleveland, W. & McGill, R. 1984. Graphical perception: Theory, experimentation and the application to the development of graphical models. *J Am Stat Assoc*, 79(387): 531–554.
8. Cohen, J. 1973. Eta-squared and partial eta-squared in communication science. *Hum Comm Res,* 28: 473–490.
9. Dwyer, T., Koren, Y., & Marriott, K. 2006. IPSep-CoLa: An incremental procedure for separation constraint layout of graphs. *IEEE TVCG (Proc InfoVis'06)*, 12(5): 821–828.
10. Findlater, L. & McGrenere, J. 2008. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *Proc CHI'08*, 1247–1256.
11. Findlater, L., Moffatt, K., McGrenere, J., & Dawson, J., 2009. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. In *Proc CHI'09*, 1655–1664.
12. Gajos, K., Czerwinski, M., Tan, D., & Weld, D. 2006. Exploring the design space for adaptive graphical user interfaces. In *Proc AVI'06*, 201–208.
13. Games, P.A. 1971. Multiple Comparisons of Means. *Am Educ Res J,* 8(3): 531–565.
14. Ghoniem, M., Fekete, J.-D., & Castagliola, P. 2004. A comparison of the readability of graphs using node-link and matrix-based representations. In *Proc InfoVis'04*, 17–24.
15. Heer, J., Card, S., & Landay, J. 2005. Prefuse: A toolkit for interactive information visualization. In *Proc CHI'05*, 421–430.
16. Heer, J., Kong, N., & Agrawala, M. 2009. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Proc CHI'09,* 1303–1312.
17. Henry, N. & Fekete, J.-D. 2006. MatrixExplorer: A Dual-representation system to explore social networks. In *IEEE TVCG (Proc InfoVis'06)*, 12(5): 677–684.
18. Lee, B., Plaisant, C., Parr, C., Fekete, J.-D., & Henry, N. 2006. Task taxonomy for graph visualization. In *Proc BELIV'06 (AVI'06 Workshop)*, 14:1–5.
19. Mayer, M., Horn, K., and Wiley, J. 2009. *Official Google Blog: Now you see it, now you don't.* Retrieved 16/09/10 from http://googleblog.blogspot.com/2009/12/now-you-see-it-now-you-dont.html.
20. Mining social networks: Untangling the social web. *The Economist Technology Quarterly,* Sep 04 2010, 16–17.
21. Melancon, G. 2006. Just how dense are dense graphs in the real world? A methodological note. In *Proc BELIV'06 (AVI'06 Workshop)*, 13:1–7.
22. Munzner, T., Guimbretière, F., & Robertson, G. 1999. Constellation: A visualization tool for linguistic queries from MindNet. In *Proc InfoVis'99*, 132–135.
23. van Ham, F. & van Wijk, J. 2004. Interactive Visualization of Small World Graphs. In *Proc InfoVis'04*, 199–206.
24. Ware, C. 2004. Information Visualization: Perception for Design, 2nd ed. Morgan Kaufmann.
25. Ware, C. & Bobrow, R. 2004. Motion to support rapid interactive queries on node-link diagrams. *ACM TAP,* 1(1): 1–15.
26. Ware, C. & Bobrow, R. 2005. Supporting visual queries on medium sized node-link diagrams. *Inf Vis J*, 4(1): 49–58.
27. Ware, C. & Mitchell, P. 2008. Visualizing Graphs in Three Dimensions. *ACM TAP*, 5(1): 2:1–14.
28. Watts, D. & Strogatz, S. 1998. Collective dynamics of 'small-world' networks. *Nature* 393(6684): 409–10.
29. Yantis, S. & Jonides, J. 1984. Abrupt visual onset and selective attention: Evidence from visual search. *J Exp Psychol Hum Percept Perform*, 10(5): 601–621.