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

First Author Name (Blank if Blind Review)
Affiliation (Blank if Blind Review)
Address (Blank if Blind Review)
e-mail address (Blank if Blind Review)
Optional phone number (Blank if Blind Review)

Second Author Name (Blank if Blind Review)
Affiliation (Blank if Blind Review)
Address (Blank if Blind Review)
e-mail address (Blank if Blind Review)
Optional phone number (Blank if Blind Review)

ABSTRACT

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

ACM Classification Keywords
H5.2 [User Interfaces]: Evaluation/methodology, interaction styles.

INTRODUCTION

A central concern in the field of information visualization (infovis) is to characterize how abstract 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. Subgraph highlighting to support path tracing applies to many real-world uses of graph visualization. Imagine a medical genetics investigator exploring a graph where nodes represent people and edges represent kinship, with the 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.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

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.

Determining the relative efficacy of visual cues 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 [Cleveland 84], many open questions remain [Heer 09]. To date, the currently prevalent approach to subgraph highlighting is to use a combination of static color and size encoding (for an example of static highlighting, see Fig 3). More recently, a new class of highlighting technique has been proposed [CHI09]: 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 (Figure 1). This class of technique has been studied in the context of an adaptive interface for menu selection [CHI09], and gained higher prominence when Google released a new home page featuring gradual onset in late 2009 [Mayer-2009].

Among the many data types used in infovis applications, node-link graphs are representative of the visual encoding and interaction issues faced in the field as a whole. For the purposes of testing highlighting techniques, we argue that the heavily studied area of interactive graph exploration (e.g. [van Ham 05]) is a good microcosm for infovis. Most graph exploration systems support highlighting subgraphs: a subset of the nodes and edges that are the elements of a graph. Tracing paths through the connections that make up the graph is a common task that users must perform when exploring this data type [Lee 06], and it has been previously

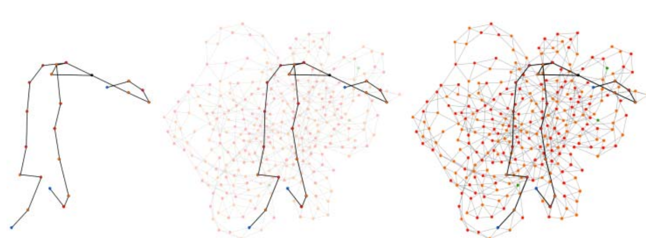

Figure 1. Time lapse from left to right shows a combination of ephemeral and static path highlighting to aid visual search. The a subset of paths is appears first, while the remainder of the graph fades in over a short period of time.

studied in controlled experiments [Ware 04] [Ware 05].

In this work, we explore the use of ephemeral techniques in an infovis setting, considering them as a type of visual cue to support path tracing in node-link graphs. We conducted experiments with 32 participants to compare path highlighting under three conditions: (1) an ephemeral encoding technique, (2) a static color and size coding, and (3) the combination of both techniques together. We chose static highlighting with color and size as the main competitor for ephemeral highlighting because, as mentioned above, it is the most commonly used approach in previous work. We included the combination of static and ephemeral because combining multiple redundant cues has often proved more effective than any single cue alone [Ware 05][Ware 08][Munzner 99]. We also included a control condition with no highlighting.

In our exploration, we accounted for prediction accuracy: whether the prediction of what to highlight was correct and thus helped the user perform the task, or whether it was incorrect and attending to the highlighted items would not help the user, or even be a distraction. Although this factor has gained prominence in recent investigations of adaptive interfaces [somebody: need good ref here], it has not been explicitly considered in the infovis community before. The implicit assumption of previous studies has been perfectly accurate prediction, whereas the reality of interactive visual exploration is that the current needs of a user are often only partially understood [tm todo: think about good ref here]. We hypothesized that ephemeral techniques may capture the known benefits of highlighting for accurate predictions, while mitigating the costs of highlighting the wrong items when the prediction is inaccurate.

Our results show that the combination of ephemeral and static encoding is overall the best option both in terms of performance, as well as self-reported workload and preference. We also found that when a neighbourhood subgraph is highlighted, the highlighting is beneficial regardless of whether it accurately captures the target node. In contrast, highlighting the set of shortest-paths to nodes of interest only offered a benefit when the target node itself was highlighted.

Our contribution is two-fold: (1) we are the first 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. In addition to the specific results for highlighting techniques that we present below, we 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. <something about impact of work, not just that we were first>

RELATED WORK

We divide the related work into previous studies of adaptive interfaces, and those in the infovis domain.

Adaptive Interfaces

Using an ephemeral technique to focus the user's attention on a subset of items in the display was first discussed in the context of adaptive interfaces. Adaptive interfaces, which automatically tailor the interaction to suit an individual user's needs, cover a broad range of tasks and contexts; see Jameson [jameson09] for an overview. A familiar commercial example is the Microsoft Windows XP Start menu, which moves a small number of predicted programs to the first level to ease their selection. Although spatially adaptive interfaces such as the Start menu are theoretically beneficial, evaluations have reported mixed results (e.g., [sears94][gajos06][findlater08]). Another line of research is to draw the user's visual attention to adaptively predicted elements, thus reducing visual search time in a complex interface. Most efforts have focused on color highlighting [gajos05][tsandilas05][tsandilas07], but ephemeral adaptation has recently offered more promising results [findlater09]. Among the more important factors impacting the effectiveness of adaptive interfaces is the level of predictive accuracy; that is, the accuracy with which the adaptive algorithm can predict the user's needs. Higher predictive accuracy can improve performance and satisfaction [findlater08][gajos06].

Information Visualization

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 (e.g. Cerebral [Barsky 08]). Highlighting neighbourhoods of two or three hops is also common, whereas larger neighbourhoods are not usually shown unless the graph is very sparse. Similarly, many previous tools support highlighting the subgraph of all edges between some target node and a set of other nodes of interest (e.g. Tulip [Auber 03]).

Much of the previous work on characterizing visual channels for encoding information has focused on static channels [Cleveland 84][Ware 04]. We focus here on studies of dynamic channels. Bartram et al. characterized the effectiveness of different simple motions for a visual search task [Bartram 02], found that motion coding outperformed color and shape coding for detectability [Bartram 03], and found that anchored motions are less distracting than travelling motions [Bartram 03].

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 [Ware 04], 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 [Ware 05].

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. Findlater et al. [Findlater09]

Figure 2. Ephemeral subgraph highlighting of a 3-hop neighbourhood subgraph region. Time span is from left to right as the full graph fades in; an onset delay of 10 seconds was used in our study.

discuss the rationale of why abrupt and gradual onset hold promise as perceptually appropriate techniques, which we do not repeat here.

EXPERIMENTAL METHODOLOGY

We conducted a controlled experiment to compare variations of ephemeral subgraph highlighting, static subgraph highlighting and a control condition using path tracing tasks: participants reported the path length from a source node to the closest node of a certain colour. We expected the effectiveness of the techniques would differ depending on the subgraph region highlighted, and whether or not that subgraph contained the target node (accurate vs. inaccurate predictions). Returning to the neighbourhood and shortest-path example from the Introduction, we might expect users to respond differently to mispredictions in these two cases, causing more difficulty in the shortest-path case. Thus, we included both of these subgraphs conditions in our investigation, which allowed us to more fully capture the range of performance across this space.

To refine the tasks and experimental design, we first conducted 12 informal sessions with 8 users, for approximately 14 hours of observation. Our goal was to understand the impact of several factors including graph size, graph density, task difficulty, and ephemeral onset length on a user's ability to do the task. Infovis experiments often require fairly extensive pre-piloting because the parameter space of possibilities has not been characterized in previous work; for example, understanding the factors that affected task difficulty required significant experimentation, whereas Findlater et al. [Findlater 09] already knew many of these factors in advance for menu selection.

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 were regular computer users (minimum 3 hours/week). They received $10 per hour of participation.

Task

The experimental task was a series of path-finding 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 trial.

A black node (source) appeared first, giving participants 2 seconds to locate it before the start of the trial. Blue (target) and green (distracter) nodes had a frequency of 1% each; the remaining nodes were red or yellow, with equal frequency. There was only one *nearest* blue node and its distance (h) 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 the participant was prompted for their best guess.

This task was designed to compare the effectiveness of visual cues for subgraph highlighting, rather than being ecologically valid in and of itself. In a real usage scenario, users would not typically directly count hops, but rather use the highlighted subgraph in service of their main task. For the purposes of a laboratory experiment, however, we needed task with a simple answer space (i.e., 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 [Ware 05], 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 the true/false answer for a given number of hops, to decrease the chance of a guess being correct.

We did extensive testing during pre-piloting of the parameters for factors that had a large impact on task difficulty: how large of a neighbourhood to highlight, and how many hops to use for the target distance. 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. A more subtle issue is that the amount of intersection between the target path and highlighted area varies; for example, for the `SPaths/WrgP` case, sometimes the wrong path to the green nodes intersects with what would be the path to the blue nodes for a few hops. The task is then easier than if there were no intersection.

Dataset and Graphs

While we considered using real-world data, we wanted to use a fresh graph for each trial to avoid undesired learning effects. Since would also have been difficult to find sufficiently isomorphic datasets for this type of repetitious laboratory experiment, we chose to use synthetic data. Although some previous experiments have used random synthetic graphs [Ghoniem 04]), we wanted to use graphs with properties more characteristic of real infovis applications. Following the arguments of Auber et al. and others [Auber 04], we used the Watts-Strogatz model to create small-world graphs [Watts 04]. Many real-world graphs have the small-world property, namely short average path lengths and high clustering coefficients, including gene networks, social networks, and the Internet. The Watts-Strogatz [Wts98] algorithm parameters were degree-4 edges in the initial circle lattice, and a 10% probability of random reattachment.

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 [Ware 05], where the difficult tasks were too difficult, and had shorter times than the easier tasks because the users gave up. In the difficult cases, error rates tended to be no better than chance with graphs of more than 500 nodes, and users would often quickly give up completely for graphs of over 750. Conversely, participants sometimes found the easy cases too trivial for graphs of 200 nodes. We thus chose a graph size of 300 nodes and 600 edges as the best balance of difficulty and density, as it typically allowed participants to complete the task in the most difficult control case in under 1 minute with a cap of 25% for the error. For this size, in the easier cases where the answer was highlighted in some way, users could typically find it within 20 seconds and with much smaller rate of error.

Experimental Factors

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

Subgraph Region

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

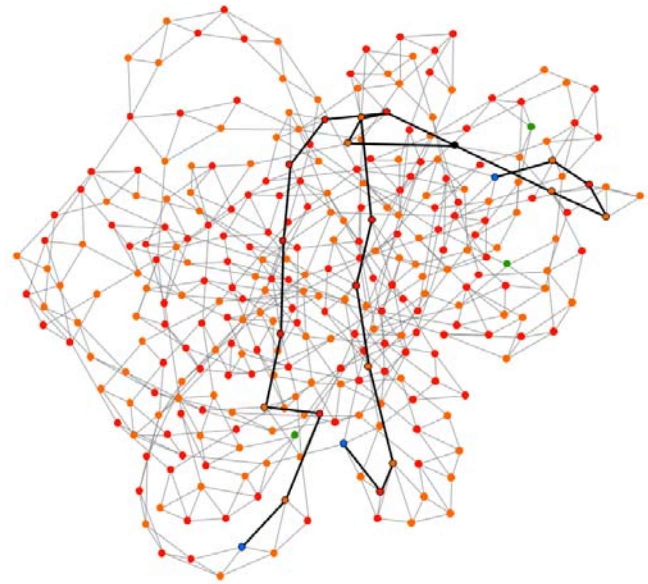

Figure 3. Static highlighting, where the entire graph appeared at once. This example highlights a shortest-path subgraph.

Highlighting Technique

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

Predictive Accuracy

In the accurate prediction (`AccP`) condition, the highlighted subgraph contained all information required to complete the task, and for the wrong prediction condition (`WrgP`), the answer was outside the 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 a wrong prediction. For an accurate predicition with `NHood`, the blue node was within the highlighted 3-hop neighbourhood.

Design

We ran two parallel experiments, assigning half the participants to each of the subgraph conditions (`Nhood`, `SPaths`). Each experiment was 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). We ran these two experiments in parallel because we wanted to gather data for multiple subgraph conditions, but were not interested in making statistical comparisons between them. 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 an answer space from 2 to 5 hops from the source node.

Measures

Our quantitative measures were task completion time and errors. Time was recorded as the median time from the initial graph appearance to the keystroke entry of an answer. We also recorded the amount of time that arrowheads were visible. Error rate was calculated as the percentage of incorrect answers. Our qualitative measures were self-reported confidence, workload, and a comparative ranking. After each trial, confidence was recorded using a scale from 1-low to 3-high. was assessed after each highlighting technique using the 20-point NASA TLX subscales for mental demand, physical demand, temporal demand, effort, performance and frustration. At the end of the study, participants were asked to comparatively rank the four highlighting techniques.

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 data 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 $\eta^2 \geq .14$ large [cohen-73].

The notion of predictive accuracy does not apply to Ctrl because it provides no highlighting. In the SPaths 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 Spaths, 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(s) 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, trials consisted of 2 trials for each possible distance, h, 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 each highlighting technique condition, and a 2-minute break at the end of the condition. Between techniques they also completed the subjective questionnaires, including the NASA TLX. At the end of the study, they ranked all four highlighting techniques, and completed a post-experiment interview.

Interface

To lay out the graph data, we used the very straightforward force-directed placement built into the Prefuse toolkit [Heer 06]. Although many more sophisticated methods have been proposed, such as multilevel [Archambault 07] or constraint-based [Dwyer 06] 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 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 0.8–0.12, discarding the rest.

Users were not allowed to interact with the graph at all, 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, this well-known visual ambiguity is resolved by the user briefly moving the nodes to see whether the edges stay attached to them, or are left behind. To resolve the ambiguity without introducing 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.

Apparatus

The experiment was coded in Java using Prefuse [Heer05]. It was conducted on a 2.53 GHz Intel Dual Core Apple laptop with 4 GB of RAM, using an external keyboard and 28” 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: the 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

Before running the full study, we began with a small pilot study to get an early feel for the viability of the highlighting techniques, and to test our methodology. This proof-of-concept study followed a shortened form of the above methodology, and fit into a 1.5-hour session. Participants completed two blocks with each highlighting technique (instead of 4), using the `Nhood` subgraph condition only. For this pilot study, we recruited an additional 12 participants (7 female, aged 19–26, median = 21), and pre-generated a separate set of 64 graphs.

Results

Overall, the results were promising, and suggested that highlighting increased speed, decreased errors, and was preferred to `Ctrl`. More specifically, `Stc` was faster than `Ctrl` independent of predictive accuracy, and resulted in fewer errors than `Ctrl` for accurate predictions. Encouragingly, neither `Eph` nor `Eph+Stc` performed significantly worse than `Stc`. Relative to `Ctrl`, `Eph+Stc` reduced errors (for `AccP`) and showed a trend of being faster (for both `AccP` and `WrgP`), while `Eph` showed only a trend of reducing errors (for `AccP`). `Eph`’s speed was comparable to the other highlighting techniques for `AccP`, but somewhat slower for `WrgP`; as such, it was no faster than `Ctrl` overall.

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` ($p = .078$).

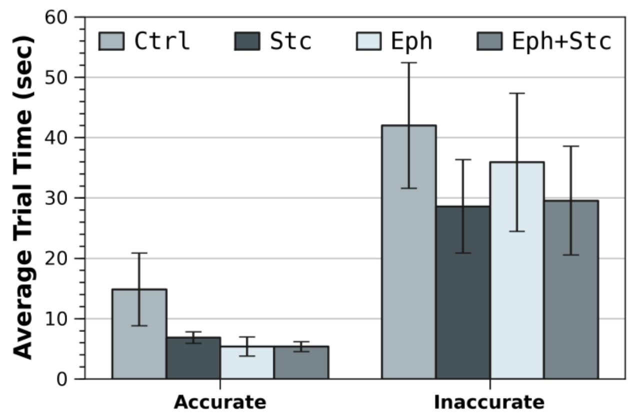

Figure 4: Average trial time by highlighting technique and predictive accuracy for `Nhood` task in the pilot study. Error bars show 95% Confidence Intervals.

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 highlighting region. One 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, that not everyone adopted an optimal strategy might explain the somewhat poorer performance of `Eph` for `WrgP`.

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, but did not detect differences between the highlighting techniques (main effect: $p < .0001$; all three pairwise comparisons with `Ctrl` were $p = .012$; all others were not significant).

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 to ensure all participants began with a comparable understanding. 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 might be more effective.

Hypotheses

Our premise was that `Eph` and `Eph+Stc` would offer benefits over `Stc` when highlighting predictions were accurate, but they would not hinder performance as much as `Stc` in cases where an inaccurate prediction could be detrimental to performance. 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 classic finding in Infovis that shows static highlighting improves performance over no highlighting, 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 (H1-A) and inaccurate (H1-I) cases.

H2. `Eph+Stc` and `Eph` result in better performance than `Stc` for the accurate case (H2-A), and are not worse for the inaccurate case (H2-I).

`Spath`:

H3. `Stc` results in better performance than `Ctrl` for the accurate case (H3-A), but worse performance than `Ctrl for the` inaccurate case (H3-I).

H4. `Eph+Stc` and `Eph` result in better performance than `Stc` in the accurate case (H4-A), but in the inaccurate case (H4-I): (1) `Eph+Stc` performs the same as or worse than `Stc`, and (2) `Eph` performs better than `Stc`.

H1 and H2 are based on the pilot study results and our initial rationalization 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 tasks, we make no formal predictions about `Nhood` versus `Spath`.

Results

The average speed and error rates for each highlighting technique and subgraph conditions are shown in figures <<X -- Y>>. We report the neighbourhood and shortest-path subgraph results here for ease of presentation.

Speed

With the neighbourhood subgraph, all highlighting conditions were faster than `Ctrl`. 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 prediction accuracy and highlighting technique, $F_{3,36} = 2.6$, $p = .067$, $\eta^2 = .178$). There were no significant interaction effects with order.

When the target node appeared inside the neighbourhood subgraph, `Eph` and `Eph+Stc` were both faster than `Stc`. We examined the pairwise comparisons for prediction accuracy and highlighting technique to test our main hypotheses (for a discussion of posthoc comparisons on trend-level effects see Games [Games-1971]). The comparisons revealed: (1) all the highlighting conditions were significantly faster than Ctrl (`AccP`: all $p < .001$; `WrgP`: $p = .022$, $p = .002$, $p < .001$, for `Stc, Eph, Eph+Stc`, respectively), and (2) `Eph` and `Eph+Stc` were faster than `Stc` (p=.009, .008, respectively). No other significant differences were found.

With the shortest-path subgraphs, `Eph` was fastest and `Ctrl` slowest for accurate predictions, with no differences for inaccurate predictions. 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 inside or outside the subgraph (interaction between predictive accuracy and highlighting technique, $F_{3,36} = 29.0$, $p < .001$, $\eta^2 = .707$). 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` ($p = .003$, $.006$, respectively). None of the other pairwise comparisons were significant.

Unexpectedly, our results also showed an interaction of order and highlighting technique with the shortest-path subgraphs ($F_{9,36} = 2.76$, $p = .015$, $\eta^2 = .408$), and a trend suggesting an interaction between accuracy, order, and highlighting 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.

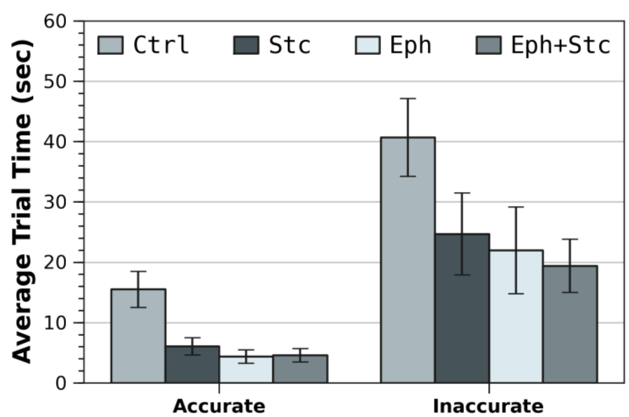

Fig X: Average trial time by highlighting technique and predictive accuracy for Nhood. Error bars show 95% CI.

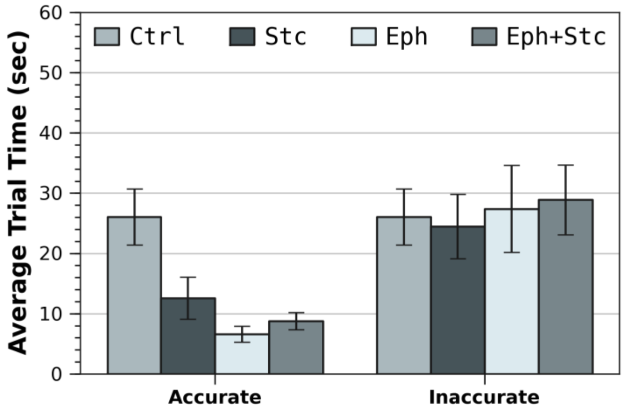

Fig X: Average trial time by highlighting technique and predictive accuracy for Spath. Error bars show 95% CI.

Errors

The error results largely followed the results for speed for both Nhood and Spaths: where the highlighting techniques were faster than Ctrl, they also tended to have fewer errors. One deviation from this pattern was with Eph (for Nhood-WrgP). Though Eph was faster than Ctrl for Nhood-WrgP, it did not result in fewer errors (see Fig X), and in fact, there was a trend of Eph having more errors than the other two highlighting conditions. There were otherwise no differences in errors among the highlighting techniques.

The statistics follow. For Nhood, there was a main effect of highlighting technique on errors for both AccP and WrgP (both $p < .001$); for Spaths, 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, (p = .015, .004, .002, respectively) and Spaths-AccP, (p = .002, .004, .002, respectively), and more errors than Stc and Eph+Stc, for Nhood-WrgP (p = .005, .008, respectively). In addition, there was a trend suggesting Eph resulted in more errors than Stc and Eph+Stc for Nhood-WrgP (p = .067, .055, respectively).

Confidence

Highlighting led to greater confidence, except when predictions were wrong in Spaths. There was a main effect of highlighting technique on confidence regardless of predictive accuracy with the Nhood subgraphs, but only for AccP with the Spaths subgraphs (all $p < .001$). In all of these cases, pairwise comparisons showed that Ctrl resulted in lower confidence than each highlighting technique (except that for Nhood-WrgP there was only a trend with Stc). (For Stc, Eph, & Eph+Stc, respectively: Nhood-AccP, p = .006, .012, .012; Nhood-WrgP: p = .072, .006, .006; Spaths-AccP: p = .002, .002, .002.) Among the highlighting techniques, there were no differences except that for Spaths, AccP, Eph resulted in significantly lower confidence than Eph+Stc (p = .012), and a trend of lower confidence than Stc (p = .066).

Subjective Measures

Eph+Stc and Stc were most preferred, while Ctrl was the least preferred. When participants were asked to rank order the highlighting techniques, Ctrl was most frequently selected as least preferred (27/32), with Eph selected by the rest (5/32). The most preferred choice was split among Eph+Stc (20/32), (Stc 9/32) and (Eph 3/32). Statistical analysis show that both Stc and Eph+Stc were significantly preferred to Ctrl, and Eph+Stc was significantly preferred to Eph. (A Friedman test showed a significant main effect of highlighting technique preference for both Nhood and Spaths, both $p < .0001$. Pairwise comparisons showed that, for Nhood, Ctrl was significantly less preferred to Stc (p = .002) and Eph+Stc (p = .002), with a trend suggesting it was also less preferred than Eph (p = .054). In addition, Eph+Stc was significantly preferred to Eph (p = .006) and a trend suggests Stc was preferred to Eph (p = .078). The results for the Shortest-path view mirror these results.)

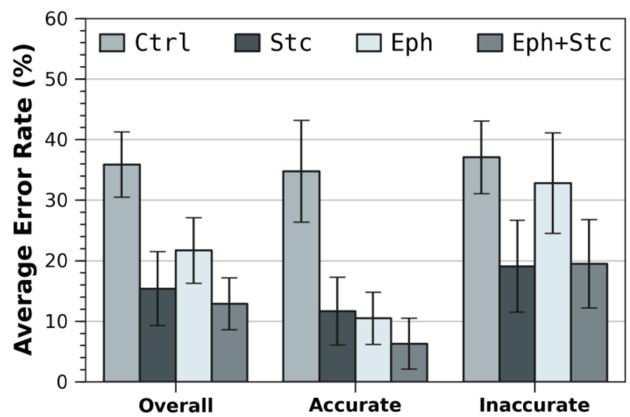

Fig X: Average error-rates by highlighting technique and predictive accuracy for Nhood. Error bars show 95% CI.

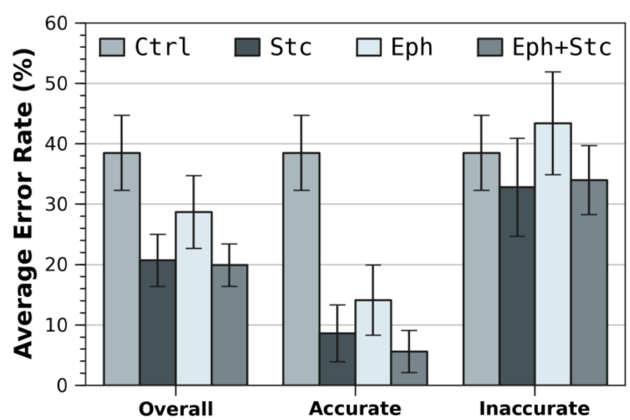

Fig X: Average error-rates by highlighting technique and predictive accuracy for Spath. Error bars show 95% CI.

In terms of workload, Eph+Stc and Stc were best, while Ctrl was worst. We collapsed the NASA-TLX subscales for both views into a single measure of workload (Cronbach's alpha = .815) and applied an RM ANOVA (highlighting technique × order). We found a main effect of highlighting technique on workload for ($p < .001$), and a main effect of subgraph ($p < .001$). Pairwise comparisons showed that `Ctrl` had a significantly higher workload than all of the highlighting conditions (`Stc`, $p < .001$; `Eph`, $p = .005$; `Eph+Stc`, $p < .001$). In addition, `Eph` had a higher workload than `Stc` ($p = .001$) and `Eph+Stc` ($p < .001$). There was no significant difference between `Stc` and `Eph+Stc` ($p = .288$). The average workload for `Spaths` was 3 points higher than the average workload for `Nhood`.

Summary

We summarize the results in terms of our hypotheses.

`Nhood`:

H1. *Supported*. Replicating previous work [refs], `Stc` resulted in better performance than `Ctrl` in terms of speed and error rate, regardless of predictive accuracy.

H2. *Supported.* When the target node was within the subgraph, `Eph+Stc` and `Eph` were faster than Stc and no different in terms of errors. As predicted, there were no differences for the inaccurate case.

`Spath`:

H3. *Partially supported.* As hypothesized, when the target node was within the subgraph `Stc` was faster and had lower error rates than `Ctrl`. Although we had expected differences in the inaccurate case, none were found.

H4. *Partially supported.* `Eph` was fastest in the accurate case, but contrary to our hypothesis, `Eph-Stc` was not different from `Stc`. Although we had expected differences in the inaccurate case, none were found.

DISCUSSION

Our results are encouraging, showing that ephemeral highlighting can improve performance with visual search tasks in complex graphs. Overall, the combination of ephemeral and static highlighting (`Eph+Stc`) showed the most promise, since it performed no worse than `static` highlighting in any condition, and offered an improvement in terms of speed when the target node was within the `Nhood` subgraph. Confidence and preference also favour combining ephemeral and static highlighting. We also replicated previous findings [ref?], showing that static highlighting improves performance over no highlighting.

We have introduced predictive accuracy as a concept that infovis researchers should explicitly consider, and we have demonstrated that whether or not the highlighting is accurate impacts performance, even with static highlighting. Our study examined an effective predictive accuracy of 50%. Future work should explore not only to what degree the benefits of the highlighting conditions we studied here will improve with higher predictive accuracy, but also how much benefit may even be retained for lower accuracy.

Studies on predictive accuracy and adaptive user highlighting techniques have shown that incorrect predictions can negatively impact performance compared to no predictions at all [refs], and we had expected this to be the case with the shortest-path subgraphs. Unlike the neighbourhood subgraphs, if the target node was not in the highlighting shortest-path subgraph, the highlighting seemed unlikely to aid the task and could potentially cause distraction. However, the fact that our hypotheses were unsupported in this respect is a positive finding. Further, although we did not explicitly set out to examine differences between different types of subgraph highlighting in our study, the varying impact of predictive accuracy depending on the subgraph chosen is noteworthy.

Although the combination of ephemeral and static highlighting is the most effective of the techniques we studied, static highlighting may already be in use in an infovis setting. As such, ephemeral highlighting would often need to be used on its own. This is an important consideration, because despite the performance improvements that ephemeral highlighting offered in some conditions, ephemeral on its own did not fare as well on subjective measures. Further, although reported confidence suggested users were in general aware of their actual performance, this was not always the case with the ephemeral highlighting (with accurate highlighting of shortest-path subgraphs, ephemeral offered the best performance but not the highest confidence).

Application / generalizability to other infovis tasks?

The combination of ephemeral and static highlighting should also have implications for adaptive GUIs as well. Findlater et al. [CHI2009] ephemerally highlighted a subset of items in pull-down menus and found the technique 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 for adaptive GUIs.

One limitation of our study is that users did not interact with the graphs, unlike in many infovis tasks. Static highlighting, for example, often appears when the user clicks on a node of interest as they are exploring the graph. Although our results should generalize to the visual search aspect of highlighting even with more user interaction with the visualization, potential improvements to the techniques could be explored. For example, in our experiments, the user had no control over the speed of onset for the ephemeral conditions, but allowing the user to pause or complete the fade-in could be useful.

Another limitation of our study?

CONCLUSIONS

We have presented ephemeral highlighting as a viable technique for use in information visualization. Through a controlled laboratory study we examined the effect of static and ephemeral techniques on two seperate types of subgraph highlighting. In both cases we found strong evidence that a combination of ephemeral and static highlighting improves both performance and user satisfaction over static highlighting alone.

We also directly considered the impact of prediction accuracy in an infovis setting. We were surprised to find that static highlighting did not negatively influence performance as we had expected. However, the consistent differences between accurate and inaccurate conditions provide evidence that predictive accuracy should be given stronger consideration by the infovis community.

ACKNOWLEDGMENTS

Blank for blind review.

REFERENCES

1. [Archambault 07] Archambault, D., Munzner, T., and Auber, D. TopoLayout: Multi-Level Graph Layout by Topological Features. IEEE Trans Visualization and Computer Graphics (TVCG) 13(2):305--317, 2007.
2. [Auber 03] Auber, D. Tulip: A huge graph visualisation framework. In Graph Drawing Software (P. Mutzel and M. Jünger, eds), p 105-126, Springer-Verlag 2003.
3. [Auber03] Auber, D., Chiricota, Y., Jourdan, F., Melancon, G. Multiscale Visualization of Small World Networks. Proc. IEEE Symp. Information Visualization (InfoVis) 2003, p 75-84.
4. [Barsky 08] Barsky, A., Munzner, T., Gardy, J.L., and Kincaid, R. Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008) 14(6):1253-1260, 2008
5. [Bartram 02] Bartram, L., and Ware, C., (2002). Filtering and Brushing with Motion. Information Visualization. Vol 1 (1) 66-79.
6. [Bartram 03] Bartram, L., Ware, C. and Calvert, T. (2003). Moticons: Detection, Distraction and Task. International Journal of Human-Computer Studies, special issue on Notifications and Interruptions, 58(5), pp. 515-545.
7. Cleavland, 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. Eta-squared and partial eta-squared in communication science. *Human Communication Research 28*, (1973), 473–490.
9. [Dwyer 06] Dwyer, T., Koren, Y., Marriott, K. IPSep-CoLa: an incremental procedure for separation constraint layout of graphs. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2006), 12(5):821-828, 2006.
10. Jameson, A., 2008. Adaptive interfaces and agents. In: Sears, A., Jacko, J., editors. Human-Computer Interaction Handbook. 2nd edition. New York: Lawrence Erlbaum.
11. Findlater, L., McGrenere, J., Modjeska, D., 2008. Evaluation of a role-based approach for customizing a complex development environment. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems , 1267-1270.
12. Findlater, L., Moffatt, K., McGrenere, J., Dawson, J., 2009. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 1655-1664.
13. Gajos, K., Christianson, D., Hoffmann, R., Shaked, R., Henning, K., Long, J.J., and Weld, D.S. Fast And Robust Interface Generation for Ubiquitous Applications. In Proc. UBICOMP'05, (2005), 37–55.
14. Gajos, K.Z., Czerwinski, M., Tan, D.S., and Weld, D.S. Exploring the design space for adaptive graphical user interfaces. In Proc. AVI'06, (2006), 201–208.
15. Games, P.A. Multiple Comparisons of Means. *American Educational Research Journal,* 1971, 8(3): 531–565.
16. [Ghoniem04] Ghoniem, M., Fekete, J.-D., Castagliola, P. A comparison of the readability of graphs using node-link and matrix-based representations. Proc. IEEE Symp. Information Visualization (InfoVis), p 17-24, 2004
17. [Heer 05] Jeffrey Heer, Stuart K. Card, James A. Landay prefuse: a toolkit for interactive information visualization. Proc. Human Factors in Computing Systems (CHI) 2005, p 421-430.
18. [Heer 09] Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations. Jeffrey Heer, Nicholas Kong, Maneesh Agrawala. Proc. ACM Human Factors in Computing Systems (CHI), pp. 1303-1312, 2009
19. [Lee 06] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and Nathalie Henry. Task taxonomy for graph visualization. Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV), article 14, ACM, 2006
20. [Mayer-2009] 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.
21. [Munzner 99] Munzner, T., Guimbretière, F., and Robertson, G. Constellation: A Visualization Tool For

Linguistic Queries from MindNet. Proc. IEEE Symp. Information Visualization (InfoVis), 1999, pp 132-135.

22. Sears, A., Shneiderman, B., 1994. Split menus: effectively using selection frequency to organize menus. ACM Transactions on Human- Computer Interaction 1 (1), 27–51.
23. Tsandilas, T. and schraefel, m.c. An empirical assessment of adaptation techniques. In CHI'05 Extended Abstracts, (2005), 2009–2012.
24. Tsandilas, T. and schraefel, m.c. Bubbling menus: A selective mechanism for accessing hierarchical drop-down menus. In Proc. CHI'07, (2007), 1195–1204. [We can cut reference #10 if needed for space.]
25. [van Ham 05] Frank van Ham and Jarke van Wijk. Interactive Visualization of Small World Graphs. Proc. IEEE Symp. Information Visualization (InfoVis), p 199-206, 2004.
26. [Ware 05] C. Ware and R. Bobrow (2005) Supporting visual queries on medium sized node-link diagrams. Information Visualization. 4,-58.
27. [Ware 08] Ware, C. and Mitchell, P. (2008) Visualizing Graphs in Three Dimensions. ACM Transactions on Applied Perception. 5(1). 2:1-15
28. [WareBook 04] Colin Ware. Information Visualization: Perception for Design, 2nd ed. Morgan Kaufmann, 2004 [Ware 04] C. Ware and R. Bobrow (2004). Motion to support rapid interactive queries on Node-Link diagrams. ACM Transactions on Applied Perception.1, 1-15.

[Watts98] Watts, D.J., Strogatz, S.H. (1998). Collective dynamics of 'small-world' networks. Nature 393 (6684): 409–10.

The columns on the last page should be of approximately equal length.