Improving Revisitation in Graphs through Static Spatial Features

Sohaib Ghani* Niklas Elmqvist*

Purdue University

ABSTRACT

People generally remember locations in visual spaces with respect to spatial features and landmarks. Geographical maps provide many spatial features and hence are easy to remember. However, graphs are often visualized as node-link diagrams with few spatial features. We evaluate whether adding static spatial features to node-link diagrams will help in graph revisitation. We discuss three strategies for embellishing a graph and evaluate each in a user study. In our first study, we evaluate how to best add background features to a graph. In the second, we encode position using node size and color. In the third and final study, we take the best techniques from the first and second study, as well as shapes added to the graph as virtual landmarks, to find the best combination of spatial features for graph revisitation. We discuss the user study results and give our recommendations for design of graph visualization software.

Keywords: Revisitation, memorability, node-link diagrams.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1 INTRODUCTION

Revisitation in graphs is the task of remembering where nodes in the graph are located and how they can be reached [22], and is a common problem in navigating and understanding graphs [11, 14]. For example, a social scientist using a graph viewer to study a social network of Wikipedia contributors may need to remember the location of different cliques of contributors as he navigates the graph to answer specific research questions. However, revisitation is typically a complex task because standard node-link diagrams—with visual bubbles representing vertices and visual links their edges—often have a single-color background and lack significant graphical features. In contrast, graphs representing road networks are typically much easier to revisit because the geographical features of the map aids the viewer in remembering the locations of nodes.

More specifically, people often remember locations of objects in an environment with respect to spatial features [20]. On maps, this could take the form of remembering locations as being "near the river" or "close to the Eiffel tower." Even in daily life, people use features and landmarks to memorize locations and give directions, such as "turn left at the gas station" or "the house next to city hall".

In this paper, we evaluate how to add spatial features to node-link diagrams to improve graph revisitation. Just like for geographical maps, these spatial features are **static** and serve as cognitive anchors that help users build better mental maps of the graph. We define three orthogonal approaches for these *static spatial graph features*:

- SE **Substrate encoding:** Inspired by geographical maps, this approach adds graphical structure to the background, or *visual substrate*, of the node-link diagram, such as colored fields and textures in various spatial arrangements.
- NE **Node encoding:** The use of graphical attributes of the nodes to encode spatial information, such as varying their size, shape, and color depending on their location.
- LM **Virtual landmarks:** Adding discrete graphical objects of unique shape and color to the graph to serve as landmarks for remembering the local graph neighborhood.

*e-mail: {sghani, elm}@purdue.edu

To evaluate these approaches, we designed three user studies. In the first study, we explored different combinations of spatial configuration and graphical features for the **substrate encoding** approach. In the second, we did the same for graphical node attributes for the **node encoding** approach. In the third and final experiment, we studied the best of the substrate and node encoding techniques together with the **virtual landmarks** approach to find the best combination of static spatial graph features for graph revisitation.

Our results indicate that substrate encoding using solid colors arranged in a grid was the best technique from the first study. Similarly, encoding spatial information using the combination of node size and color was optimal in the second. For the final study, we found that substrate encoding coupled with landmarks resulted in optimal performance; surprisingly, node encoding did not have a significant impact on performance. Therefore, provided that the visual variables and the increased clutter associated with these spatial features do not conflict with a graph visualization, we recommend to use these strategies to aid the revisitation task.

2 RELATED WORK

We surveyed the literature to find evidence of adding spatial features to visual spaces. Below we present our findings within this domain. For particularly relevant work, we indicate which of the above strategies (SE, NE, or LM) that specific work employs.

2.1 Spatial Cognition

People commonly remember places in the real world with respect to spatial features, such as landmarks and other reference points, which serve as cognitive anchors and aid in building a mental map of the area [9, 17] (LM). Siegel and White [21] showed that people remember places better as their experience with those places increases (the more we visit a place, the better we remember it).

Spatial memory aids recall [1]. Robertson et al. [20] leveraged this by spatially organizing web bookmarks in the Data Mountain, and showed that spatial memory supports faster recall compared to standard bookmarks (SE). Spatial memory also persists over time. Czerwinski et al. [5] found that users can easily retrieve items that they placed in a spatial arrangement even after several months. Bateman et al. [2] study what Edward Tufte calls "chart junk", and show that charts with these seemingly unnecessary embellishments are easier to remember than those without, even after a few weeks (SE). The argument is that embellished charts have more spatial features that increase their memorability and ease of revisitation.

2.2 Spatial Features

Many visualization systems use spatial features to enhance the efficiency of a visualization. Havre et al. [10] propose the ThemeRiver, which has a characteristic sinuous shape for displaying patterns and relationships in large document collection (SE). Bubble Sets [4] use spatial features to show set memberships (SE). Graphical encoding of items in the form of color, shape, or size could also be used to convey information about the spatial position of items [16] (NE).

Research in virtual environments indicate that spatial features in 3D worlds aid navigation [7, 19] (SE, LM). For example, Darken and Sibert [6] showed that real-world wayfinding can be applied to virtual environments. They added structure to the 3D environment, such as paths, boundaries, and directional cues, and found that these help users to keep orientation and navigate in the world.

2.3 Navigation in Graphs

Navigation is an essential feature of graph visualization [11]. There are numerous techniques for navigation in graphs, each having its particular pros and cons. We review a representative sample below.

Scrolling is the simplest navigation technique, but quickly becomes cumbersome for exploring large spaces [13]. Pan and zoom [18] is commonly used for graphs, but can be tedious for exploring distant nodes in large networks. More complex techniques based on space distortion such as fisheye views [8] magnify the focus region while still showing context with less detail. However, these focus+context techniques are not visually stable and may confuse the user because the transition between focus and context is not uniform. Overview and detail is another commonly used technique that utilizes two windows to show the focus region as well as the whole visual space. It is effective for navigating large networks but requires additional interaction [12].

2.4 Revisitation in Graphs

Our work is motivated by social scientists who use graphs in their daily work (see the next section). Many tasks require them to revisit a particular part of a social network [14]. This implies that effective revisitation mechanisms would help make overall navigation faster.

Skopik and Gutwin [22] introduce *visit wear*, defined as visual representations of places that have already been visited, and apply them to fisheye views for revisitation in graphs (NE, LM). These are *dynamic* spatial features that appear as the user interacts with and navigates in the graph. Our work, on the other hand, adds static features—fixed embellishments of a visual space—that are independent of interaction and can thus be applied to any graph visualization. Furthermore, Skopik and Gutwin's technique is dynamic (it adds visual cues to the graph as the user moves the mouse) and requires explicit activation, whereas our technique is passive and always active and thus better supports ad hoc revisitation tasks.

3 Motivation: Social Network Analysis

Our work is motivated by collaborations with social scientists who use visualization tools for social network analysis (SNA) as part of their regular duties. We observed that these scientists would often experience some trouble orienting themselves when returning to a previously studied social network. To get to the bottom of this issue, we performed structured interviews with two of our collaborators (both faculty members at our university and SNA experts).

Both scientists currently use UCINET [3] and the associated NetDraw application for all of their SNA needs. Depending on the project, they stated that they would use these tools 2-4 times a month in sessions spanning multiple (3+) hours, and in 50-60% of the cases the network they study would be one they had studied before. However, despite this intense usage, sessions would often be irregularly scheduled, sometimes with several days or a week in-between. Even though both scientists thought that they generally remembered the overall structure of a known graph, they conceded that the spring-embedding algorithm in NetDraw often causes slight variations in graph layout that make recall and revisitation more difficult. Moreover, ad hoc observations of social scientists performing SNA showed that more than 50% of all navigations in a node-link diagram was between previously visited parts of a graph.

4 Static Spatial Graph Features

The *memorability* of a visual space is a measure of a generic user's ability to remember information about the space. *Revisitation* is the task of remembering where objects in the visual space are located and how they can be reached. Memorability is thus intricately linked to revisitation. As discussed above, the purpose of our work is to study how to add spatial features to node-link representations of a graph to increase its memorability, thus aiding the revisitation task. Based on our survey of the literature, we study three different classes of static spatial graph features: substrate encoding, node encoding, and virtual landmarks. We describe these below.

4.1 Substrate Encoding

Substrate encoding mimics geographical maps by adding graphical features to the visual representation of the graph. In a map, these features are typically spatial regions, such as roads, city limits, state lines, etc. The regions themselves are generally identifiable through unique colors or textures. The features can then be used as reference points, but also increase the visual complexity of the map.

We identify two degrees of freedom for substrate encoding: the *partitioning* of the space into regions, and the *encoding* of identity into each region to allow the user to separate them (Figure 1):

- For partitioning, we can either use a space-driven or detail-driven approach: the former splits space into regions of equal size, whereas the latter splits space into regions with equal numbers of items (i.e. nodes). The advantage of a detail-driven approach is that if nodes are clustered in a small area of the whole graph, then we will allocate more partitions in that area. For uniform partitioning, a majority of the nodes may end up in the same partition, overloading its use for recall and potentially reducing the memorability of the visual space.

- For identity encoding, the simplest approach is to use a solid color, but a texture may yield better memorability and details that can serve as reference points [20].

4.2 Node Encoding

This approach uses the nodes (and potentially the edges) of a graph to encode information about the spatial position of the node. The graphical variables available for this include the size, shape, and color of a node. This approach has the advantage of not introducing a high degree of visual clutter. However, some of these graphical variables may already be utilized to convey underlying information about the data in many existing graph visualizations.

4.3 Virtual Landmarks

The basic idea with virtual landmarks mimics the role of landmarks in the real world—they serve as static reference points that can be used for orientation (e.g., the Eiffel tower in Paris). Unlike substrate encoding, which adds identifiable regions to the background of the graph, these landmarks are discrete objects that are evenly distributed in visual space. For this reason, landmarks typically give rise to less visual clutter than substrate encoding techniques without affecting the visual representation of the graph itself.

5 User Studies

We perform three studies to evaluate the above approaches to static spatial graph features. In the first study, we find the optimal substrate encoding technique. In the second, we find the optimal node encoding. Finally, in the third study, we compare the optimal substrate and node encodings with virtual landmarks to find the best overall combination of static spatial graph feature techniques.

In this section, we list the common features of all three experiments, including the apparatus, the experimental platform, the task, and the procedure followed for all studies.

5.1 Apparatus

The studies were conducted on a 3.00 GHz dual-core PC with 4 GB of memory, running Microsoft XP and equipped with a 24” monitor set to a resolution of 1600×1200 pixels.

5.2 Experiment Platform

For the purposes of these user studies, we built a node-link graph viewer in Java. We disabled node labels, as they are orthogonal to our approach. The viewer has both overview and detail windows, where the overview window shows a miniature of the whole workspace and the detail window shows the current focus. The overview includes a rectangle to show the location of the focus.

Participants panned the workspace either by dragging the mouse in the detail window or by dragging the cue rectangle in the overview window. Zooming was not possible. The overview window had a resolution of 300×300 pixels and the detail window was 1024×768 pixels. The overall size of the visual space was 3300×3300 pixels, i.e., approximately 3×4 detail screens.

5.3 Task and Dataset

All three studies involved a revisitation task similar to that used by Skopik and Gutwin [22]. It consisted of two phases: an initial learning phase, followed by a revisitation phase.

1. **Learning:** Participants were shown N blinking nodes in sequence (visible in both overview and detail views) and were asked to visit each node and learn its position.

2. **Revisitation:** Participants were asked to revisit the nodes whose location they had learned in the previous phase, and in the same order as before.

Nodes to visit were selected randomly by the study platform from separate regions of the graph such that they had approximately the same total path length and thus similar a priori memorability for each trial. The order of nodes was randomly selected to be either clockwise or counter-clockwise to avoid the user having to move back and forth several times across the whole space.

Skopik and Gutwin [22] use $N = 6$ in their work, whereas our three studies use 4, 4, and 5, respectively. The reason for the difference is that Skopik and Gutwin used dynamic features to identify visited nodes, which clearly differentiate visited nodes from other nodes, while we only use static visual features. Our task is therefore more difficult, motivating a shorter sequence of nodes to remember.

We used a random graph with 108 nodes and 158 edges organized using a lin-log layout [15]. Because the lin-log algorithm is nondeterministic, the layout will be different for each trial, thus avoiding systematic learning effects. The algorithm also allows for optionally clustering nodes depending on the graph topology.

5.4 Procedure

Participants received training before each technique block to ensure that they knew how to solve trials using the technique. For each trial, participants clicked on a button to indicate they were ready to start the trial with the learning phase.

In the learning phase, participants were shown the full graph with any spatial features active in both detail and overview windows. They could navigate freely around by panning or moving the focus rectangle in the overview window. During this phase, one of the nodes in the graph would be blinking. This blinking node was visible in both windows. When the participant navigated the view and clicked on the node—thus learning its location—the node would stop blinking and another node would start blinking. The node could only be clicked in the detail window (the overview window was used only for showing the position of blinking nodes). After the participant had clicked N nodes, the learning phase would end. We did not record time or accuracy for this phase.

In the revisitation phase, the view was reset to the center of the workspace and participants were asked to revisit nodes in the same order they had visited them in the learning phase. A node was revisited by clicking it in the detail window.

The experiment platform recorded the following metrics:

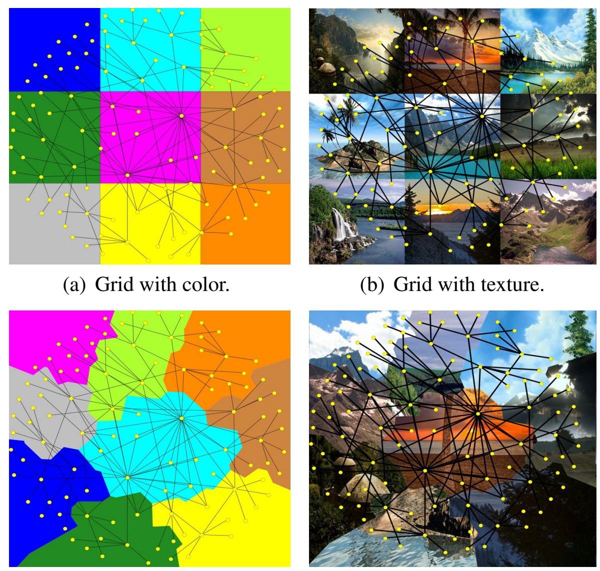
- **Completion time:** The time from the start of the revisitation phase to when the N:th node had been clicked.

- **ID errors:** Number of selected nodes that were not in the recall sequence at all.

- **Out-of-order errors:** Number of nodes in the recall sequence that were selected in the wrong order.

- **Overall correctness:** Whether a sequence was perfectly recalled (both ID errors and out-of-order errors zero).

Participants were allowed to rest between trials. After the experiment we collected user preferences and comments using a questionnaire and a short interview.



(a) Grid with color. (b) Grid with texture.

(c) Voronoi with color. (d) Voronoi with texture.

Figure 1: Matrix of substrate encoding methods for Study 1. Partitioning is mapped to the vertical axis, identity encoding to the horizontal.

6 Study 1: Substrate Encoding

In our first study, we add spatial features to the background of the graph. This study lasted approximately 50 minutes, including the initial training session and post-experimental questionnaire.

6.1 Participants

We recruited 16 paid volunteers (eight male and eight female), screened not to be color blind. All participants were university students with ages varying from 20 to 31 years (average 26), and all were regular computer users (more than 16 hours/week).

6.2 Experimental Conditions

6.2.1 Partitioning

Following our discussion above, we derive partitioning techniques for both equal-sized regions, as well as regions with equal numbers of items. For consistency, we fix the number of regions to 9 for both techniques (derived using an initial pilot study).

- **Grid.** A regular grid is the simplest space partitioning technique for equal-sized regions. We use a 3×3 grid to divide the space into the 9 regions (Figure 1(a) and (b)).

- **Voronoi diagram.** Partitioning space into regions with equal numbers of items requires us to group the graph nodes into 9 disjoint clusters based on their Euclidean distances. We then use a Voronoi diagram, summing up the cells for node in each cluster, to find the regions covered by these nodes. This yields an irregular partitioning focused on areas of high detail, presumably improving its memorability (Figure 1(c) and (d)).

6.2.2 Identity Encoding

We use two approaches for encoding identity into regions:

- **Color.** A solid color is the most straightforward way to differentiate between regions. We select 9 standard colors that are easily distinguishable (Figure 1(a) and (c)).

- **Texture.** A texture will yield more internal detail to a region, potentially increasing its memorability. We choose 9 landscape images, selected from the Web to have distinctive visual appearance (Figure 1(b) and (d)).

6.2.3 Layout

Graph layout was added as a factor to determine whether detail-driven (Voronoi) or space-driven (grid) partitioning perform differently for different layouts. Towards this end, we used two separate lin-log [15] layouts: one yields uniform node distribution with uniform edge lengths (more readable but with no clustering), and the second clusters similar nodes based on the graph topology.

6.3 Study Design

Crossing two partitioning and two encoding techniques, we tested four techniques T in this study (Figure 1): grid with color (GC), grid with texture (GT), Voronoi diagram with color (VC), and Voronoi diagram with texture (VT). We used a within-subject full factorial experimental design: 2 partitionings (P) $\times$ 2 encodings (E) $\times$ 2 layouts (L) $\times$ 3 repetitions (R) = 24 total trials. We counterbalanced the order of the techniques and blocked on technique; the layout factor was administered in random order. With 16 participants, we collected time and correctness for a total of 384 trials.

6.4 Hypotheses

H1 Voronoi diagram will be faster and more accurate than grid for spatial partitioning.

H2 Texture will be more accurate than color for identity encoding.

6.5 Results

We analyzed the average of all repetitions for each condition as the final measurement—see Figure 3 for boxplots of the completion times and errors using each technique. Table 1 summarizes the effects on time and errors using a repeated-measures analysis of variance. The completion time violated the normality assumption, so we analyzed its logarithm (which conformed to normality).

Posthoc tests (Tukey's HSD) showed that GC was significantly faster than all other techniques ($p < .05$), and GT was significantly more accurate than VT for ID errors ($p < .05$).

Figure 3(c) shows subjective ratings for techniques in terms of efficiency, enjoyability, and clutter (1 = low, 5 = high). Differences in ratings were significant for efficiency and visual clutter (Friedman test, $p < .05$), but not for enjoyability.

Factors	DF	Time	ID Errors
Partitioning (P)	1, 15	**13.28	*4.98
Encoding (E)	1, 15	**18.60	1.19
Layout (L)	1, 15	0.18	0.33
P * E	1, 15	*5.61	2.26

* = $p \leq 0.05$, ** = $p \leq 0.001$.

Table 1: Effects of factors on time and error for Study 1 (RM-ANOVA, F-values in columns). Nonsignificant interactions elided.

6.6 Summary

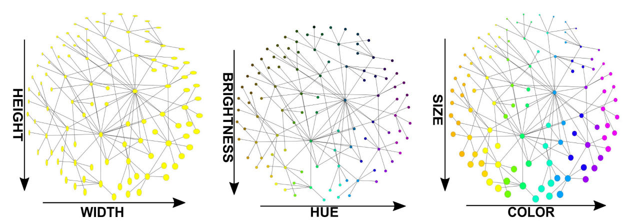
We can summarize the above results in the following way:

- Space-driven partitioning using a grid yields significantly faster and more accurate performance than detail-driven partitioning using a Voronoi diagram (rejecting H1);

- Encoding regions using a solid color yields significantly faster performance, with no significant difference in errors, than encoding using a texture (rejecting H2); and

- There is no significant effect of graph layout on completion time or accuracy.

These results suggest that the combination of grid with color (Figure 1(a)) is the optimal substrate encoding technique for this particular revisitation task. The findings also mean that we need not study graph layout in further studies, and we thus use the uniform lin-log layout [15] for the remainder of this paper.



(a) Size (b) Color (c) Size & Color

Figure 2: Overview of node encoding techniques for Study 2.

7 Study 2: Node Encoding

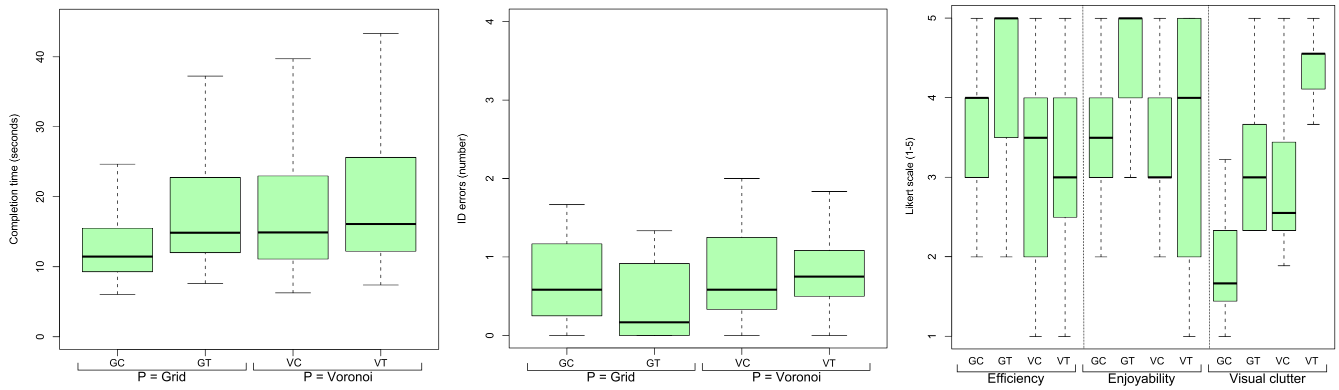
In the second study, the spatial position of nodes was encoded in their size and color. This study was conducted in conjunction with Study 1, and used the same participants as that study. Because we used the same basic task and scenario in both studies, we counterbalanced the order of the studies to mitigate systematic effects of practice. This study lasted approximately 40 minutes.

7.1 Experimental Conditions

We included a single factor: the type of node encoding.

- **Size.** We vary node width based on horizontal position on the substrate and node height based on their vertical position (Figure 2(a)). The idea behind this technique is to allow users to remember nodes using their dimensions. The color of the nodes remains fixed in this technique.

- **Color.** Here we vary the color of nodes by changing hue based on horizontal position and brightness based on vertical position (Figure 2(b)). Nodes are thus assigned a unique color depending on their position in the graph, thereby increasing the memorability of the whole graph. The size of the nodes remains fixed in this technique.



(a) Completion time. (b) ID errors. (c) Subjective ratings (clutter / 2).

Figure 3: Participant performance and ratings for different substrate encoding techniques (Study 1).

- **Size and Color.** We also study the combination of size and color to encode position (Figure 2(c)). We allow the hue of the node (saturation and brightness is fixed at 100%) to change depending on horizontal position, and node radius to change depending on vertical position. We varied the hue because this is most easily distinguishable.

7.2 Study Design

We used a within-subject full factorial design: 3 techniques (T) $\times$ 5 repetitions (R) = 15 total trials. We again counterbalanced the order of the techniques and blocked at each technique. With 16 participants, we collected time and errors for a total of 240 trials.

7.3 Hypothesis

H3 Size and color combined will be the best node encoding technique in terms of both time and accuracy.

7.4 Results

We analyzed the collected data using a repeated-measures ANOVA (completion time violated the normality assumption, so we analyzed its logarithm). Figures 4(a) and 4(b) show completion times and errors. The main effect for completion time between different techniques T was significant ($F(2,30) = 12.00, p < .0001$), as was the main effect for ID errors ($F(2,30) = 5.48, p < .01$).

We further analyzed these effects using a posthoc Tukey's HSD and found that Size+Color was significantly faster and more accurate than both Size or Color alone ($p < .05$), but there was no significant difference between these other two.

Figure 4(c) shows average ratings for the techniques in terms of efficiency and enjoyability (1 is low, 5 high). Differences in ratings were significant for both metrics (Friedman test, $p < .05$).

7.5 Summary

Our results confirm H3: the combination of size and color for encoding position is both faster and more accurate than each of these techniques separately. This suggests we choose this combination as the optimal node encoding technique.

8 Study 3: Combinations of Techniques

In this last study, we select the best techniques from Study 1 (Grid with Color) and Study 2 (Size and Color) as well as the virtual landmarks approach to find the optimal combination of static spatial features for the revisitation task in graphs. Because we combine several techniques in this experiment, we opted to increase the difficulty of the task to $N = 5$ (instead of $N = 4$) to get more separation between the individual combinations of techniques.

8.1 Participants

We recruited 16 paid volunteers (eleven male and five female), screened not to be color blind. This pool of participants was different from Study 1 and 2. All participants were university students with ages varying from 20 to 32 years (average 24). All participants reported that they used computers more than 16 hours/week.

8.2 Experimental Conditions

This study included three factors that governed whether a particular spatial feature was active or not: spatial encoding SE, node encoding NE, and virtual landmarks LM.

Virtual landmarks are created by adding 9 discrete graphical objects of different shapes and colors as reference points to the graph background (Figure 5(b)). Just like landmarks in the real world, we hope these will aid users in remembering node locations. Because Study 1 showed that space-driven partitioning yields better performance than detail-driven partitioning, we opted to space out these 9 landmarks in the centers of the 9 cells of a regular grid.

8.3 Study Design

Crossing the three spatial feature factors results in eight different techniques T evaluated in this study (Figure 5): SE, NE, LM, SE+NE, SE+LM, NE+LM, SE+NE+LM, and simple graph (SG) (all features disabled). We used a within-subject full factorial design: 2 levels of spatial encoding (SE) $\times$ 2 levels of node encoding (NE) $\times$ 2 levels of landmarks (LM) $\times$ 5 repetitions (R) = 40 total trials. We counterbalanced and blocked on the technique T. With 16 participants, we collected data for 640 trials.

8.4 Hypotheses

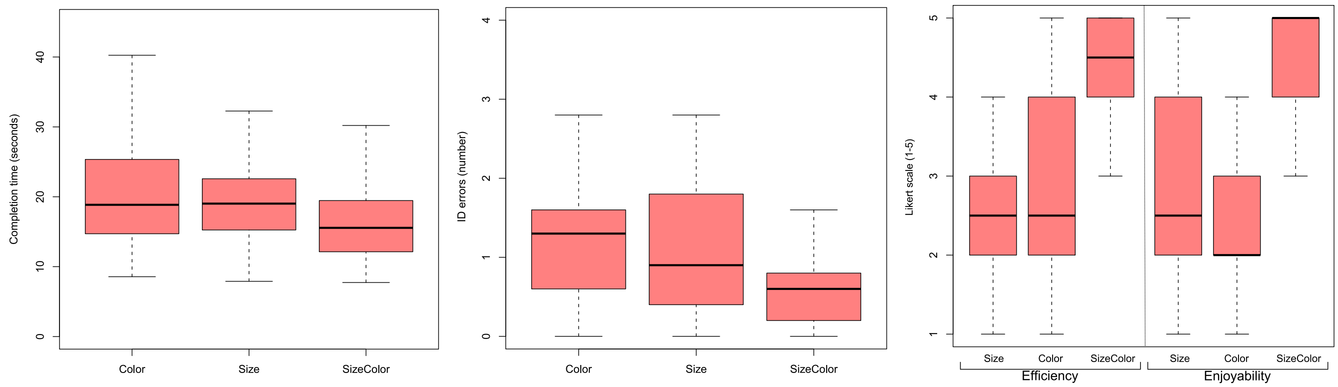
H4 Techniques utilizing substrate encoding will be faster and more accurate than node encoding and landmarks.

H5 The combination of all three spatial graph feature techniques will be fastest and most accurate.

8.5 Results

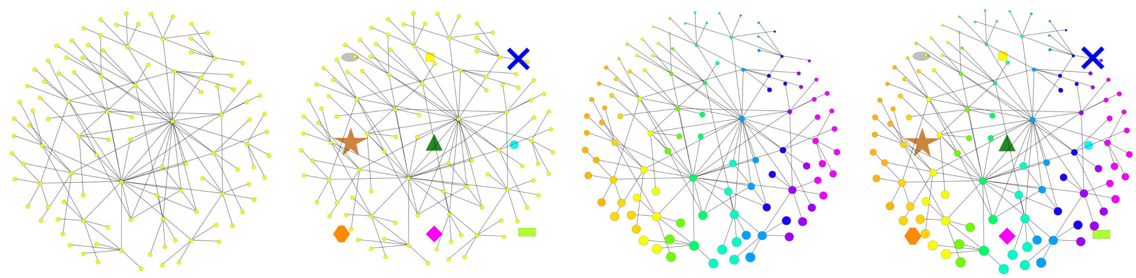
We analyzed the average of all repetitions for each condition and participant—see Figure 6 for plots of completion time and errors per technique. Table 2 summarizes the effects on completion time and errors using a repeated-measures ANOVA (completion time violated the normality assumption, so we analyzed its logarithm).

We analyzed both of these effects using a Tukey HSD. Figure 7 depicts pairwise significant differences ($p < .05$) for completion time. For ID errors, Tukey HSD showed that SE, LM, SE+LM,

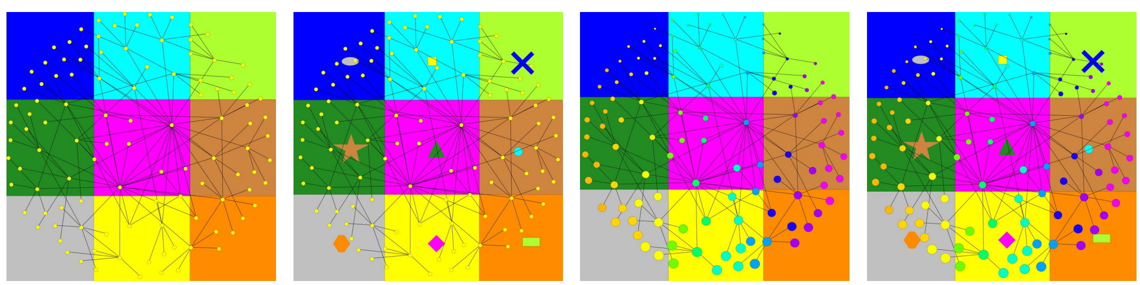


(a) Completion time. (b) ID errors. (c) Subjective ratings.

Figure 4: Participant performance and ratings for different node encoding techniques (Study 2).



(a) SG (b) LM (c) NE (d) NE+LM



(e) SE (f) SE+LM (g) SE+NE (h) SE+NE+LM

Figure 5: Overview of the eight techniques used in Study 3.

Factors	DF	Time	ID Errors
Spatial encoding (SE)	1, 15	**33.35	**27.55
Node encoding (NE)	1, 15	2.94	0.64
Landmarks (LM)	1, 15	*6.04	**21.42

* = $p \leq 0.05$, ** = $p \leq 0.001$.

Table 2: Effects of factors on completion time and errors for Study 3 (RM-ANOVA, F-values in columns). No interactions were significant.

SE+NE, and SE+NE+LM all were significantly more accurate than SG ($p < .05$); no other pairwise difference was significant.

Finally, Figure 6(c) shows ratings for the techniques in terms of efficiency, enjoyability, and clutter (1 = low, 5 = high). Differences were significant for all metrics (Friedman test, $p < .05$).

8.6 Summary

We can summarize the above results as follows:

- Techniques with substrate encoding were significantly faster, and were not less accurate (partially confirming H4);
- SE+NE+LM was **not** significantly faster and more accurate than all other techniques (rejecting H5); and
- Virtual landmarks is a generally promising strategy, performing second only to substrate encoding.

These results suggest that the best approach to static spatial graph features is substrate encoding, followed by virtual landmarks. Both of these techniques—especially in combination—performed significantly better than the competing techniques. Node encoding seems not to make much difference either way, which is perhaps why the combination of all three approaches is good, but not significantly better than others—a somewhat counterintuitive result.

9 Discussion

We can summarize our overall findings as follows:

- Substrate encoding is the dominant strategy, in particular using space-driven partitioning with a solid color.

- Virtual landmarks help significantly in revisitation.
- Node encoding does not perform as well as the other two.
- The combination of virtual landmarks and substrate encoding is the optimal technique for the revisitation task.

Many of our results are surprising and do not fully obey our intuition about graph navigation. We discuss these below.

For substrate encoding, we initially hypothesized (H1) that detail-driven partitioning would be better than space-driven partitioning for two reasons: (1) the detail-driven approach will adapt the graphical features depending on the data clustering, and (2) the irregular boundaries arising from this partitioning will increase graph memorability. However, our results disproved this hypothesis. For reason (1), we believe this is because graphs in Study 1 did not have pathological situations with all nodes clustered in a few places—our layout factor did not introduce clustering to the level needed to affect the results. For reason (2), some of the participants commented during the exit interview that the irregular boundaries of the detail-driven approach confused rather than helped them.

We also hypothesized (H2) that texture will yield better accuracy than color. However, the results show no significant difference in accuracy. In interviews, participants stated that they clearly preferred texture, because it helped them remember nodes with respect to visual features in the photographs such as lakes, mountains, grass, etc. We believe the reason for this discrepancy is that in our experiment, the grid cells are sufficiently small that the color and boundary of a cell provides adequate memorability for accurate recall. In support of our belief, some participants stated that they memorize nodes first by background color and then with respect to region boundaries, and thus if the node is somewhere in the center of the region, it is difficult to remember. For large regions, we think that texture encoding will be more accurate, but another approach to improve color can be to add landmarks as reference points in each region. This is also what the results from Study 3 suggest.

The subjective ratings (Figure 3(c)) indicate that participants prefer space-driven partitioning and texture encoding both in terms of efficiency and enjoyability. Most of the participants commented that with texture background, the task was much more enjoyable compared to other techniques. However, they also thought that texture adds more visual clutter. Color is not as enjoyable but has much less visual clutter. Overall, all sixteen participants stated that they liked having access to substrate encoding.

For node encoding, we hypothesized (H3) that Size+Color would be the fastest and most accurate technique, and results also confirm this. Size+Color was also favored by participants in interviews and ratings. They thought that size and color alone was difficult to remember, whereas Size+Color gives a more memorable identity to nodes. Thirteen out of sixteen participants stated that they would like node encoding to be present to aid graph revisitation.

For the last study, we hypothesized (H5) that the combination of all three techniques would perform the best, but this was not the case. Results show that node encoding is not particularly effective for this task, and therefore, the combination of SE+LM+NE does not perform significantly better than all other techniques (Figure 7). Several participants commented that when they used this technique, they did not utilize the node encoding for recall, possibly because it is more difficult to use compared to the other two. Substrate encoding, on the other hand, performed well (H4) because background color and graphical boundaries were easy to remember, according to participants. As for landmarks, participants said they were useful as reference points. Participants also stated that the combination of substrate encoding and landmarks was the optimal technique because it allowed them to remember the nodes on two levels: *globally* by background color, and *locally* by landmarks and boundaries.

Figure 6(c) shows subjective ratings for Study 3. The key insight here is that participants thought SE+LM and SE+NE+LM are the most efficient and most enjoyable techniques. However, SE+NE+LM has more perceived visual clutter. All sixteen participants overall stated that they liked spatial features for revisitation.

While we measured out-of-order errors as the number of *correct* nodes that were selected in the *wrong* order, our results showed that this type of error was very uncommon. This is probably because our sequence of nodes to recall was chosen in a regular order. Therefore, our results above do not report these errors.

Note that we also analyzed the overall correctness (trials with both zero ID and zero order errors) using logistic regression and found nearly identical effects to the ANOVA results ($p < .05$). We also analyzed pairwise differences in subjective ratings using Wilcoxon signed-rank tests, and found the expected trends in ratings. However, due to limited space, we elide these results here.

9.1 Limitations

As it is often the case with controlled experiments, we had to make a large number of decisions on how to design our studies. For example, we decided to include an overview of the visual space. We did this so that the visual space could be larger than the screen, preventing participants from remembering nodes by absolute positions on the screen rather than by spatial features. Furthermore, the overview was scaled down to a factor of about 10, making it difficult for participants to remember nodes using just the overview; also, nodes could not be clicked or selected in the overview.

Our choice of textures for Study 1 may also be open to debate. We selected landscape images because they are neutral and have many spatial features that improve their memorability. However, it is plausible that other textures, such as regular or stochastic (e.g., Perlin noise) patterns, would yield different results. The same could be said for our color choices, which are somewhat oversaturated.

We should note that our static approach to spatial graph features relies on the graph layout being static, and that a dynamic layout (such as a force-directed one) may cause the user's conceptual map of the graph to become inaccurate. This is an intrinsic limitation of our approach, but is also true of dynamic spatial features (e.g. [22]).

Finally, we do not evaluate virtual landmarks in a separate study to save space and time. However, there may exist additional design variations of landmarks that warrant separate studies of their own.

9.2 Generalization

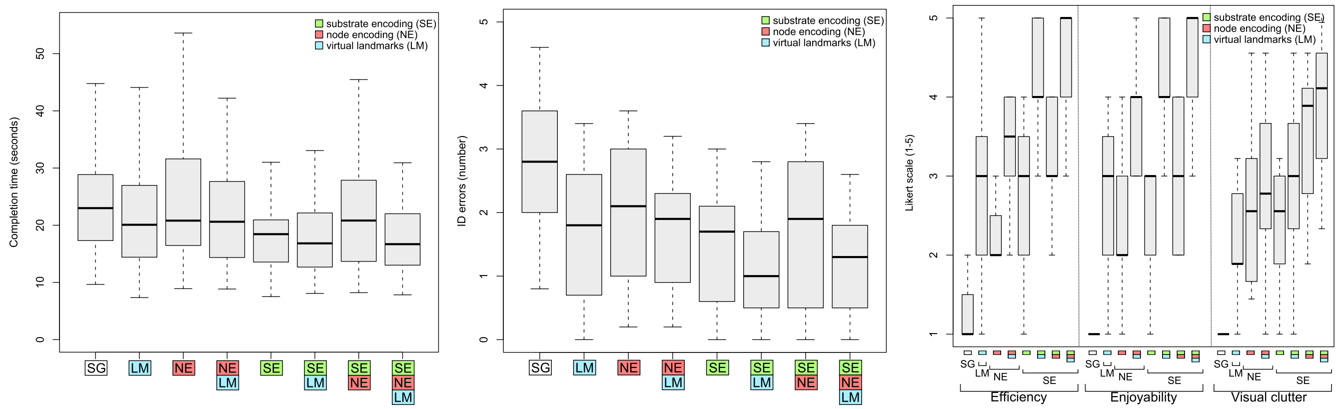
Overall, we think that our results can be generalized for virtually any graph, and our recommendation is to use the combination of substrate encoding and virtual landmarks. These features also tend to make graph navigation more enjoyable (Figure 6(c)), perhaps approaching the enjoyability of geographical map navigation that motivated our work in the first place. However, the techniques **will** add visual clutter to a visualization, and some graph features may also conflict with existing visual variables used for a different purpose.

Scalability is a concern for larger graphs. We speculate that a hierarchical spatial subdivision may be appropriate here, and it is also possible that the more detail provided by a texture encoding will yield better memorability for such graphs.

Our results from the Study 1 notwithstanding, a truly general approach to static spatial graph features should probably be detail-driven to be able to deal with pathological cases with high amounts of node clustering. Since our participants indicated that irregular boundaries confused them, one idea may be to combine detail-driven partitioning with rectangular shapes to yield more regular and predictable boundaries. This is left for future research.

10 Conclusion and Future Work

We present several ways of adding static spatial features to graphs and show that they help revisitation in graphs by performing three orthogonal studies. In our first study, we evaluate different substrate encoding techniques and show that grid with color is optimal. In the



(a) Completion time. (b) ID errors. (c) Subjective ratings (clutter / 2).

Figure 6: Participant performance and ratings for different combinations of static spatial graph feature techniques (Study 3).

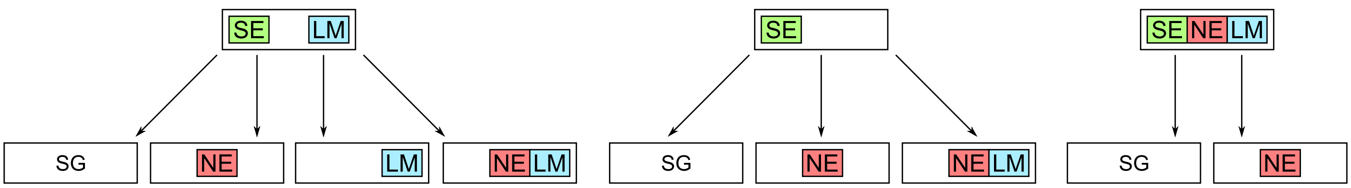


Figure 7: Significant pairwise differences in completion time (Tukey's HSD, $p < .05$). Arrows indicate which technique was faster than another.

second study, we found that encoding spatial location in node size and color is the best node encoding technique. Finally, in the last study, we show that substrate encoding, landmarks, and their combination are optimal techniques for graph revisitation in general.

Revisitation is important in many other scenarios—such as finding icons on the screen, traversing file systems, navigating web browsers history, etc—so our future work entails designing techniques for aiding these scenarios as well.

References

[1] F. Attneave. Symmetry, information, and memory for patterns. *The American Journal of Psychology*, 68(2):209–222, 1955.

[2] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. A. Brooks. Useful junk?: The effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 2573–2582, 2010.

[3] S. Borgatti, M. Everett, and L. Freeman. UCINET. `http://www.analytictech.com/ucinet/`. Last accessed December 2010.

[4] C. Collins, G. Penn, and M. S. T. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009.

[5] M. P. Czerwinski, M. van Dantzich, G. Robertson, and H. Hoffman. The contribution of thumbnail image, mouse-over text and spatial location memory to web page retrieval in 3D. In *Proceedings of INTERACT*, pages 163–170, 1999.

[6] R. P. Darken and J. L. Sibert. Navigating large virtual spaces. *Int. Journal of Human-Computer Interaction*, 8(1):49–71, Oct. 1995.

[7] R. P. Darken and J. L. Silbert. A toolset for navigation in virtual environments. In *Proceedings of ACM User Interface Software and Technology*, pages 157–165, 1993.

[8] G. W. Furnas. Generalized fisheye views. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 16–23, 1986.

[9] S. E. Goldin and P. W. Thorndyke. Spatial learning and reasoning skill. In *Spatial Orientation: Theory, Research, and Application*. Plenum Press, 1983.

[10] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[11] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, Jan./Mar. 2000.

[12] K. Hornbæk, B. B. Bederson, and C. Plaisant. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM ToCHI*, 9(4):362–389, 2002.

[13] T. Igarashi and K. Hinckley. Speed-dependent automatic zooming for browsing large documents. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 139–148, 2000.

[14] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proc. of BELIV*, pages 82–86, 2006.

[15] A. Noack. An energy model for visual graph clustering. In *Proceedings of the Symposium on Graph Drawing*, pages 425–436, 2003.

[16] L. T. Nowell, R. S. Schulman, and D. Hix. Graphical encoding for information visualization: An empirical study. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 43–50, 2002.

[17] R. Passini. *Wayfinding in Architecture*. Van Nostrand Reinhold, New York, NY, 1984.

[18] K. Perlin and D. Fox. Pad: An alternative approach to the computer interface. In *Computer Graphics (ACM SIGGRAPH '93 Proceedings)*, pages 57–64, 1993.

[19] A. E. Richardson, D. R. Montello, and M. Hegarty. Spatial knowledge acquisition from maps and from navigation in real and virtual environments. *Memory & Cognition*, 27(4):741–750, 1999.

[20] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich. Data Mountain: Using spatial memory for document management. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 153–162, 1998.

[21] A. W. Siegel and S. H. White. The development of spatial representations of large-scale environments. In *Advances in child development and behavior*, pages 9–55. Academic Press, 1975.

[22] A. Skopik and C. Gutwin. Improving revisitation in fisheye views with visit wear. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 771–780, 2005.