

# Supporting Visual Queries on Medium Sized Node-Link Diagrams

Colin Ware  
Data Visualization Research Lab  
Center for Coastal and Ocean Mapping  
University of New Hampshire  
Durham, NH, 03824  
colinw@cisunix.unh.edu

Robert Bobrow  
BBN Technologies Inc.  
10 Moulton Street  
Cambridge, MA 02138  
rusty@bbn.com

## *Corresponding Author*

Colin Ware  
Data Visualization Research Lab  
Center for Coastal and Ocean Mapping  
University of New Hampshire  
Durham, NH, 03824  
colinw@cisunix.unh.edu

## **Abstract**

For reasons of clarity, a typical node-link diagram statically displayed on paper or a computer screen contains fewer than 30 nodes. However, many problems would benefit if far more complex information could be diagrammed. Following Munzner et al [15], we suggest that with interactive diagrams this may be possible. We describe an interactive technique whereby a subset of a larger network diagram is highlighted by being set into oscillatory motion when a node is selected with a mouse. The subset is determined by a breadth first search of the underlying graph starting from the selected node. This technique is designed to support visual queries on moderately large node-link diagrams containing up to a few thousand nodes. An experimental evaluation was carried out with networks having 32, 100, 320, 1000, and 3200 nodes respectively, and with four highlighting techniques: *static highlighting*, *motion highlighting*, *static+motion highlighting*, and *none*. The results show that the interactive highlighting methods support rapid visual queries of nodes in close topological proximity to one another, even for the largest diagrams tested. Without highlighting, error rates were high even for the smallest network that was evaluated. Motion highlighting and static highlighting were equally effective.

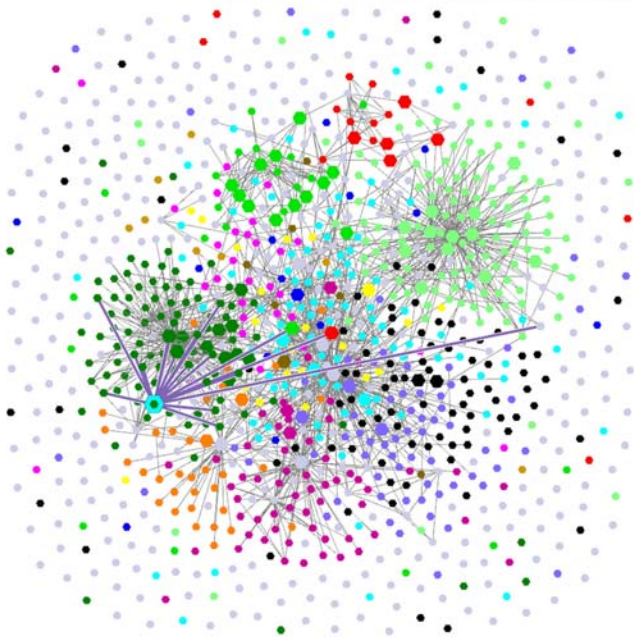
A second experiment was carried out to evaluate methods for showing two subsets of a larger network simultaneously in such a way that both are clearly distinct. The specific task was to determine if the two subsets had nodes in common. The results showed that this task could be performed rapidly and with few errors if one subset was highlighted using motion and the other was highlighted using a static technique. We discuss the implications for information visualization.

**Keywords:** Graph visualization, interactive animation, motion coding

## Introduction

The great majority of node-link diagrams contain fewer than 20 nodes and 30 links between them. This number falls far short of the millions of nodes and links that are contained in large graphs. Visualization of such large graphs in their entirety represents an almost insurmountable challenge. There is a common class of medium-sized graphs, however, containing between thirty nodes and a few thousand nodes. The graph representing all of the employees and their email traffic in a large, but not huge, company is an example (see Figure 1). The graph representing all of the parts in an automobile or machine of similar complexity is another. These may be loaded into the main memory of a computer and can be rendered in real-time on a screen. The problem is that this many nodes and the links between them cannot be drawn legibly in a static diagram.

The Constellation system of Munzner et al. [15] used an interactive technique to allow access to medium-sized graphs. To explore a graph, the user moved the mouse cursor over the nodes, this action triggered a local search of the graph and caused the links within one or two link radius to be highlighted by becoming brighter, wider and moved to the foreground layer. In this way, nodes in topological proximity could be brought to the attention of the user. In Constellation no attempt was made to draw links so that they did not cross nodes or other links or apply any of the layout technique normally applied to produce visual clarity. When not highlighted, links were drawn faintly and transparently over the nodes and there were so many that it was generally impossible to trace them; it was only when highlighted that the paths became clear.



**Figure 1** A graph containing over 1000 nodes. Only a subset of the edges are shown. The graph shows email traffic between individuals within a medium-sized corporation.

A number of other interactive techniques have been developed specifically to look at tree data [13,14,18,19,20]. In the case of Cone Trees, users must rotate a graph in a 3D space in order to explore the graph [19]. In the case of the Hyperbolic Tree Browser users must drag parts of the tree to the central region to see maximum detail [14]. Nicheworks [25] allowed users to interactively view large graphs but what was shown to the user was a subgraph consisting of a tree.

The problem of obtaining information from a visual display such as a graph can be characterized as a process of cognitively constructing and executing a series of *visual queries* [22]. These queries are executed both by means of eye movements and by selective tuning of the visual pattern finding mechanism so that the relevant features of the display are brought into visual working memory [22, 6]. Considered in this way, the effectiveness of a display can be enhanced in a number of ways. Improving the graphical design can make it easier to visually search for such patterns as nodes that are in close topological proximity to other nodes. Alternatively the visual search can be supported by means of an interactive technique, as in Constellation [15] of MEGraph[23]. In MEGraph (for motion enhanced graph) clicking on a node causes a breadth-first search of that graph beginning at that node. The entire subgraph within a certain specified topological radius is highlighted by being set into motion. An assumption implicit in this technique is that the connections between nodes in close topological proximity to the selected node are the ones most likely to be of interest when the diagram of a graph is used as an information display. We believe this assumption to be justified for common applications such as software engineering and social networks [8].

It is important to make a distinction between a visual query in the sense that we apply it here and the queries of Huang et al [12]. In their work, queries cause subgraphs of a much larger graph to be loaded and the spring layout changed dynamically based on the newly loaded data. Here we are using the word query in the sense that visual thinking involves queries on the visual environment [22]. The uploading involved is the uploading of information into the human brain. In our system, all the data is loaded into computer memory, the layout is only done once, and all the data is displayed graphically, although much of it is not “readable”. The active visual queries we are investigating thus involve a manual selection of a graph, an automatic highlighting of a subset of the displayed information, and a visual search for information.

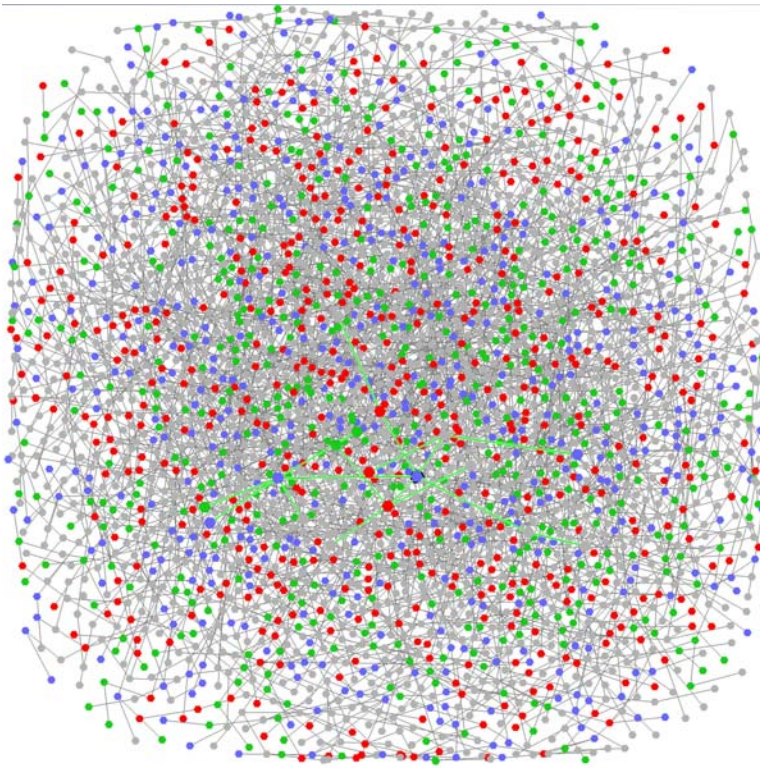
Giving a subgraph an oscillatory motion while the rest of the graph remains static should be a very effective way of highlighting information. Research on human perception suggests that effective highlighting of data should be done with *pre-attentive* cues [21]. A pre-attentive visual cue is something that stands out at a glance, no matter how complex the surrounding information. For example, red symbols in a display that is otherwise entirely black and white will stand out clearly. For the most part, only simple visual attributes stand out pre-attentively. Size, color, orientation, and oscillatory motion can all be seen pre-attentively. More complex patterns cannot be seen as easily. For example, a visual search for something that is both large and red (in the context of other large, but not red objects, and other red, but not large objects) will be difficult and error prone. This coupling of visual cues is called a

conjunction search. Motion is unlike most other visual cues in that visual conjunctions of motion and some other properties can be rapidly searched [9,10,16,17].

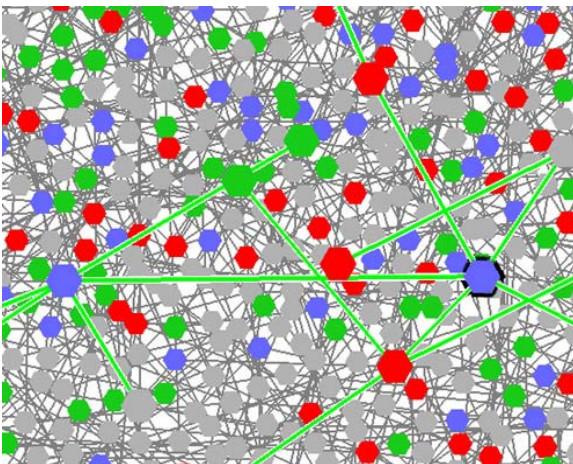
The fact that motion supports conjunction search makes it ideal for revealing subgraphs of larger graphs since it should allow a rapid visual search for particular patterns in the subgraph, based on other attributes such as target orientation. We have exploited this idea in MEGraph [23]. In MEGraph, rapid interactive querying, by clicking on nodes, triggers a breadth first search of the graph and this in turn causes highlighting of the nearby nodes and links through oscillatory motion of the subgraph. The effectiveness of this technique assumes that most visual queries on graphs will be for groups of nodes in close topological proximity, but this seems reasonable at least when those graphs represent such things as social networks and software architectures. In the case of social networks connections between nodes usually represent acquaintanceships. In the case of software architectures, nodes in near proximity might show which subroutines are used by a particular module.

Previously, in an empirical evaluation of MEGraph, we have shown motion enhancement to be more effective than static highlighting techniques with graphs having an average of 50 nodes and 82 edges. However, two significant criticisms could be leveled at our previous study. The first was that the graphs we used in the experiment were still quite small, although somewhat larger than can be seen clearly without an interactive highlighting technique. Yet we claimed that the technique should be useful for much larger graphs than this. The second criticism is that in the experiment we did not take the time of interaction into account. If we are to compare visual queries made by simply moving the eyes with queries made with the aid of an interactive technique, it is appropriate for interaction time to be taken into account as part of the cost of obtaining information from the display.

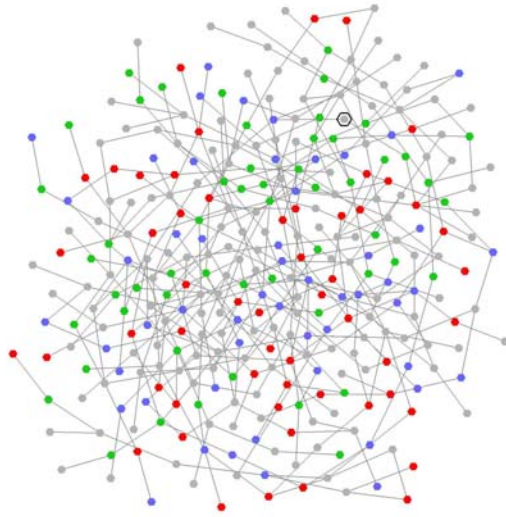
The first of the two studies we report here answers both of these criticisms. The graphs are much larger, containing up to 3200 nodes and 3700 edges. And we also measure the time cost of interaction. The goal of the second experiment was to look at the problem of simultaneously highlighting two subgraphs of a larger graph.



**Figure 2** A 3200 node network diagram laid out using spring forces.



**Figure 3** A part of Figure 2 showing static highlighting.



**Figure 4** A 320 node graph of the kind used in our study.

## **Experiment 1: Effectiveness of Motion Highlighting with Medium-Sized Graphs**

The first experiment we report here was designed to find the relative values of motion highlighting and static highlighting on a set of graphs of sizes between 32 and 3200 nodes. Highlighting was done using breadth first search of a graph around a selected node. We used the simple task of visually searching for red nodes within the radius of two edges from the selected nodes. This is a very specific task, but we believe that the results should be representative of results for many similar tasks involving visual search for simple patterns.

### **Task**

On each trial the subject's task was to determine if there was a red node within two links of a particular node designated by a bold circle drawn around it. In the highlighting conditions, the subject had to move the mouse over the highlighted node, click on it, then visually search the highlighted subset to find out if there was a node satisfying the criterion. The subject then pressed one of two keys on the keyboard -- the 'M' key for *yes* and the 'N' key for *no*. These keycaps had 'yes' and 'no' labels placed on them.

### **Conditions and Trial Blocks**

Four kinds of highlighting were used.

- 1) No Highlighting.

- 2) Static Highlighting
- 3) Motion Highlighting
- 4) Static and Motion Highlighting combined

There were four graph sizes used: 32, 100, 320, 1000, and 3200 nodes. The product of highlighting methods and graph sizes gave 20 conditions. The experiment was run as a within subjects design.

### **The Node-Link diagrams**

One sixth of the nodes were colored red, one sixth were green and one-sixth were blue and the remaining one half were gray. The node diameter was 2 mm when not highlighted and the link width was 2 pixels (approx 0.25 mm).

Static highlighting increased node diameter to 3mm (more than doubling the area) and increased the link width to 4 pixels (approx 0.5mm). Highlighted links were given a white border by first drawing a 7 pixel width line, followed by a centered 3 pixel wide line centered on it. Figure 2 illustrates a 3200 node diagram and Figure 3 shows part of it enlarged to make the highlighting more visible. Figure 4 illustrates a 320 node diagram.

The graph was generated by the following algorithm:

*For each node in the graph:*

*Form a link to either 1 or 2 other nodes, randomly selected, with a generator biased so that a single connection occurred 83.33 % of the time and two connections occurred 16.67% of the time.*

The result was a graph with the following statistical properties (empirically determined):

- 26 % of the nodes have degree one
- 35% of the nodes have degree two
- 23% of the nodes have degree three
- 10% of the nodes have degree four
- 4% of the nodes have five connections
- <1% of the nodes have 6 or more connections

The graph layout was done by spring forces [7] using an iterative approach. All nodes repelled one another according to the inverse of their squared separation. Connected nodes were attracted to one another with a spring force having a constant such that the resting length of the spring was 1.8 cm. A force proportion to the distance from the center cubed was applied to keep the outer fringes of the graph from spreading out of the display area. This was applied independently in x and y directions and is the reason for the compression seen in the edges of the graph in Figure 2. To speed up the layout process a spatially defined regular grid containing a linked list of the nodes was used to find near neighbors and calculate the repulsion forces between nodes [11].



A Viewsonic VP2290b display was used for the experiment. This is capable of displaying 3840x2400 pixels with approximately 81 pixels per cm. All graphs were displayed at a 20 Hz update rate. Diagrams were drawn in a square region having 2200 pixels (27 cm) on a side.

## **Procedure**

At the start, subjects were given a practice session with a few trials in each of the conditions. This was continued until the experimenter felt certain that the subject fully understood the task. Following this, the experiment was run using blocks of trials where the same highlighting technique was used within a block. For each block subjects were first shown a single graph having 50 nodes and asked to select a particular highlighted node and respond appropriately. This was intended to re-acquaint them with that particular highlighting technique. Next they were presented with a new graph generated based on one of the size conditions and they had to make a series of 6 responses (3 with 'yes' as a correct response; 3 with 'no' as a correct response randomly interspersed). Following this they were presented with a differently sized graph and again made six responses. This was repeated until they had seen all graph sizes under that highlighting condition with the sequence of differently sized graphs randomly determined. Subsequently the subjects were given blocks of trials with each of the other highlighting methods. There was a different random ordering of the conditions given to each of the subjects.

This entire process was repeated three times. This yielded a possible 9 'yes' and 9 'no' correct responses for each highlighting/graph size combination yielding a total of  $(4 \times 5 \times (9+9)) = 360$  trials for each subject. The entire experiment typically took 45 minutes to an hour.

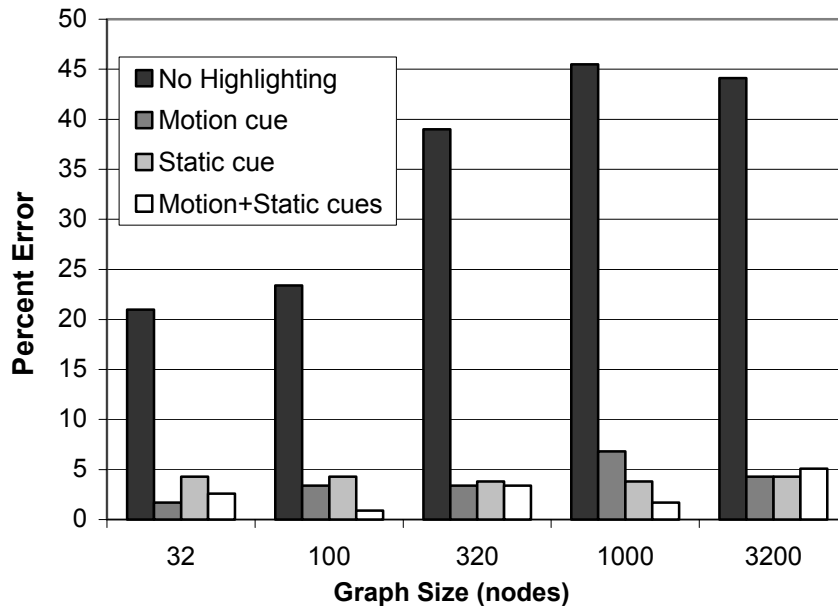
## **Subjects**

The 13 subjects were mostly undergraduate students at the University of New Hampshire, paid to participate. One was a research assistant in the lab.

## **Results from Experiment 1**

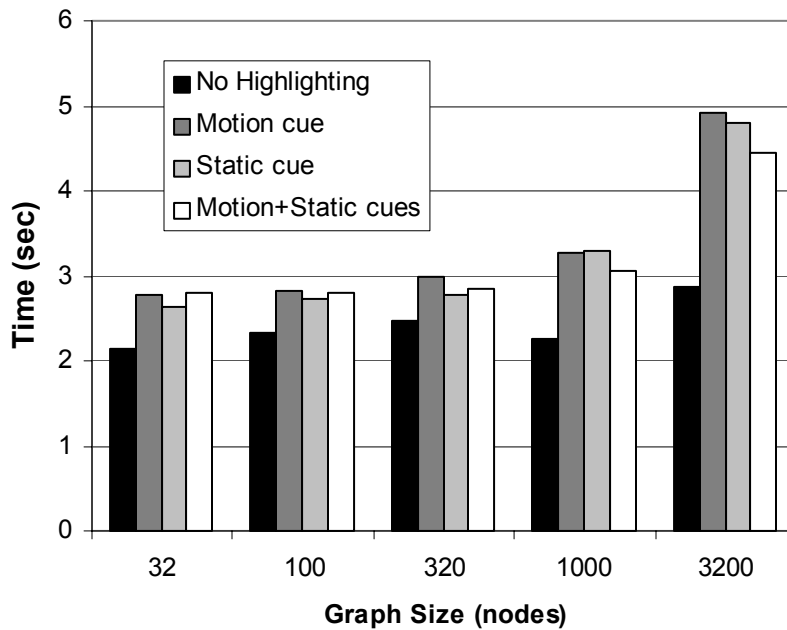
The error rate results are summarized in Figure 5. The most obvious effect is that all of the interactive highlighting conditions resulted in much lower error rates compared with the *No Highlighting* condition. A three-way ANOVA was carried out for the following factors: graph size, highlighting method, target present vs target absent. There was a main effect for highlighting method ( $F(3,4640) = 366.0, p < .0001$ ). The error rate was lowest with both motion and static cues (2.7%) next came the static cue (3.9%) and the motion cue (3.9%). The *No Highlighting* condition had a mean error rate of 34.7%. A Tukey HSD test applied to the highlighting method revealed only two groups: one group containing the high error rate *No Highlighting* condition and the other consisting of the other three conditions. There was also a significant effect of graph size ( $F(1,4640) = 13.8, p < 0.001$ ) and an interaction between these two factors ( $F(12,4649) = 9.22, p < 0.001$ ). Most of the source of this interaction is the *No Highlighting* condition. Subjects performed at close to chance levels when the graphs were 320 nodes and above. There was also

a main effect for target present vs target absent ( $F(1,4640) = 2.29, p < 0.001$ ). Subjects made more errors overall (13.6%) when the target was present, than when the target was absent (9.1%).



**Figure 5** Summary of error rates for different graph sizes and highlighting conditions.

The response time results are summarized in Figure 6. This shows that there was little variation in response times for the first four graph sizes with a marked increase for the largest, 3200 node graph. There were significant effects for highlighting method ( $F(3,4640) = 46.6, p < 0.001$ ), graph size ( $F(4,4640) = 100.8, p < 0.0001$ ), and also an interaction between the two ( $F(12,4640) = 4.322, p < 0.0001$ ). The nature of this interaction appear to be that although the longest response times occurred for the 3200 node graph for all conditions, the time increase was less in the No Highlighting condition. There was also a main effect for target present vs target absent ( $F(1,4640) = 19.7, p < 0.0001$ ). Examining Figures 5 and 6 suggests that conditions with high error rates also had faster responses. It was thought that this might be due to subject giving up early on the more difficult conditions. A t-test showed that, on average, mean response time for conditions where the error was greater than 10% was shorter than for conditions where the error rate was less than 10% ( $p < 0.01$ ).



**Figure 6** Summary of response times for different graphs sizes and highlighting conditions.

### Discussion of Experiment 1

The main practical result from Experiment 1 is that interactive highlighting can make medium sized node-link diagrams accessible to queries about the nodes in close topological proximity. In the no-highlighting conditions the error rates were such that even the smallest graphs would not be usable in a practical application. It should be recognized that the task was monotonous and subjects may have cared little about the outcome and thus error rates could undoubtedly be reduced given a higher level of motivation. Nevertheless, it is unlikely that the larger diagrams could be used effectively without one of the interactive highlighting techniques. The shorter response times for the No Highlighting conditions were not expected. A possible explanation for this result is that subjects developed a presumption through the course of the experiment that when they encountered a No Highlighting condition they would not be able to produce a correct response, so they produced a rapid guess as a response.

Both the static highlighting and the motion highlighting methods proved to be equally effective. Thus the experiment failed to confirm our previous finding that motion highlighting can be more effective than static highlighting. This suggests that the main advantage of motion highlighting can be as an addition highlighting technique intended to supplement static highlighting methods, rather than replace them. Complex data requires complex visual codings of data attributes. It may well be that visual cues useful for highlighting (e.g. node or line width) will have already been used. Thus the main advantage of motion can be that it can provide an additional coding channel independent of the already heavily used static coding methods. Alternatively, motion can provide support for highlighting two different subsets of a complex diagram.

## Experiment 2: Supporting Visual Queries with Two Highlighted Subgraphs

Problem solving with real applications is likely to demand support for considerably more complex queries than those demonstrated in Experiment 1. For example, it may be useful to select two different subgraphs of a larger diagram for the purpose of understanding how they might be related. In the case of a social network we might wish to look at two subnetworks representing two teams working on different projects. Nodes representing individuals working on both projects might be of special interest. The problem of distinctly highlighting two subgraphs of a larger graph is the subject of our second study. To identify a task suitable for empirical assessment we first considered that highlighting two subgraphs should (at least) be able to support any of the following queries. Can it be seen that a particular node belongs to:

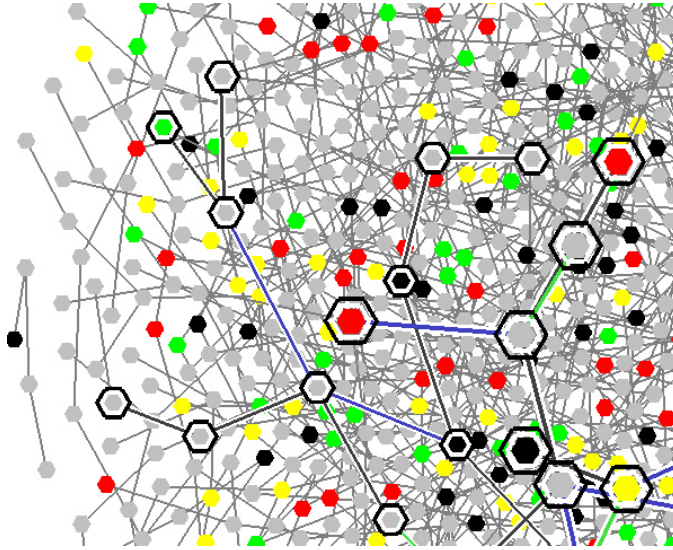
- (a) Subgraph A,
- (b) the intersection of subgraphs A & B,
- (c) neither A nor B.

The third category (c) is the hardest to support because it requires a visual conjunction search. It is necessary to clearly perceive the conjunction of two different highlighting methods. We therefore designed an experiment to look at how different design alternatives can support this kind of query and we reduced the task to the simple visual query – do highlighted subgraphs A and B intersect (i.e do they have nodes in common)?

We hypothesized that having static highlighting on one subgraph and motion highlighting on the other would be the most effective way of supporting the *intersection of two subgraphs* query. As discussed in the introduction, pre-attentive conjunctions can be formed by using motion features in conjunction with static features. Consider a case in which one subgraph was set in motion, while a second subgraph was highlighted by putting borders around the nodes and edges. Finding out if the two subgraphs had nodes in common would involve visually searching for nodes that are both moving and have borders around the nodes and edges. This is a visual conjunction of motion and non-motion cues.

We were also interested in the question of whether it was possible to use two different kinds of motion to highlight different subgraphs; for example, by highlighting one subgraph with vertical motion and the other with horizontal motion. We also used this experiment to evaluate this possibility. As part of an iterative design phase we developed four different motion patterns attempting to make them clearly distinct from sinusoidal *horizontal* motion (2 Hz sinusoidal) of the first subgraph. Three of them used *vertical* motion of the second subgraph; the first of these used the same frequency (2 Hz); the second doubled the frequency to 4 Hz; a third used bursts of 8 Hz motion interspersed with no motion. In the fourth motion pattern the nodes sinusoidally grew and shrank in size. Since visual searches for conjunctions for two kinds of motion are not known to occur pre-attentively, we did not expect that two motions could easily support our visual search task. But we thought it worth while to try to find pairs of

motions that supported rapid conjunctive searches. If we found them we would have made a discovery of both theoretical and practical interest.



**Figure 7** A section of a 1000 node graph used in Experiment 2. Condition 7 is illustrated. The smaller highlighted nodes and links show the static highlighting cues used for most conditions. The larger nodes and links are in pulse highlighting mode.

### Conditions

All conditions had two different subgraphs highlighted. These were selected so that they always overlapped spatially. In half the trials the two subgraphs had two or three nodes in common; in half they were disjoint. Of the eight conditions four had static highlighting of one subgraph and motion highlighting of the other. In the other four conditions, subgraphs were highlighted with both static and motion cues, although each had a different motion.

### Conditions with one subgraph moving and the other statically highlighted

Subgraph A moved with the following motions (no static highlighting cues).

- 1) Vertical sinusoidal: 2 hz. Amplitude 0.333 cm.
- 2) Vertical sinusoidal: 4 Hz.
- 3) Vertical burst motion: 8Hz sinusoidal in alternating bursts, 0.5 sec moving 0.5 sec static.
- 4) Pulse motion: 1 Hz size changes. This sinusoidally changes the size of the nodes and the width of the links by a factor of 2 at a rate of 1 Hz. This condition is illustrated in Figure 7.

Subgraph B did not move but was statically highlighted.

### Conditions with two moving subgraphs

Subgraph A moved with one of the four motions listed above. Subgraph B moved horizontally with a sinusoidal oscillatory motion at 2 Hz. Both subgraph A and subgraph B were highlighted with additional static cues.

**Table 1 Summary of Conditions for Experiment 2**

		Subgraph B	
		Static highlighting for subgraph B only. Subgraph A has motion highlighting but no static highlighting	Horizontal motion for subgraph B.(both graphs move and also use static highlighting cues)
<i>Subgraph A</i>	Vertical	1	5
	Vertical double frequency	2	6
	Vertical Burst motion	3	7
	Pulse motion	4	8

### Properties of the Graphs

The graphs were constructed and laid out using the same method employed for Experiment 1. For Experiment 2 graphs always had 1000 nodes. The two subgraphs were constructed by means of a breadth first searches from two randomly selected nodes to a level of 3. These were used in a trial if the following conditions were met: a) Each subgraph contained at least 8 nodes. b) Subgraphs overlapped spatially by at least 40% (using as a measure the overlap between bounding boxes of each subgraph as a percentage of the area of the bounding box of both subgraphs). c) For trials with common nodes the two subgraphs had either 2 or 3 shared nodes. The method of construction involved repeatedly creating subgraphs until a pair was found that met the criteria.

### Other display parameters

We used a CRT monitor for this experiment because it had better temporal characteristics than the ultra high resolution LCD monitor used in the first experiment. The update rate for all conditions was 60 Hz. The viewport in which all patterns appeared was 27 cm (1100 pixels) square. The nodes were hexagons 0.4 cm across. The static highlighting was bolder than that used for experiment 1. Each statically highlighted node had a white ring around it (0.5 cm in dia) surrounded by a black ring (0.7 cm in diameter). Links had a line width of 2 pixels (approx 0.5 mm). When highlighted they were first drawn with a 7 pixel wide white line then with a 3 pixel wide line (Figure 7 illustrates).

### Task

On each trial a large graph containing two highlighted subgraphs was presented to the subject. The subject was required to press the right mouse key, if there were common nodes between the two subgraphs (a 'yes' response) and the left mouse key if there were no common nodes (a 'no' response). In the overlap conditions there are between 2 and 5 common nodes.

## Subjects

The 12 subjects were a combination of graduate and undergraduate students. Most were paid for participating.

## Procedure

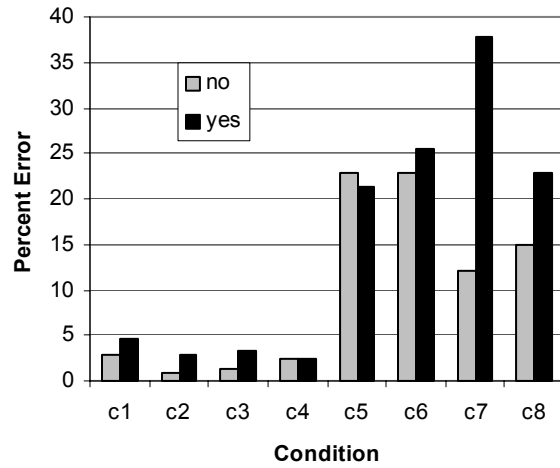
Subjects were first given a training session in which they were shown all 8 conditions, and an example with overlap and without overlap for each one. Next conditions were given in a different random order for each subject with 20 trials presented in each condition. In half the trials the two highlighted subgraphs had nodes in common (requiring a yes response); in half they did not. Once the subjects had run through all of the conditions they were given a rest for a few minutes and then they repeated a second set of trials. There were a total of 40 trials per condition.

## Results from Experiment 2

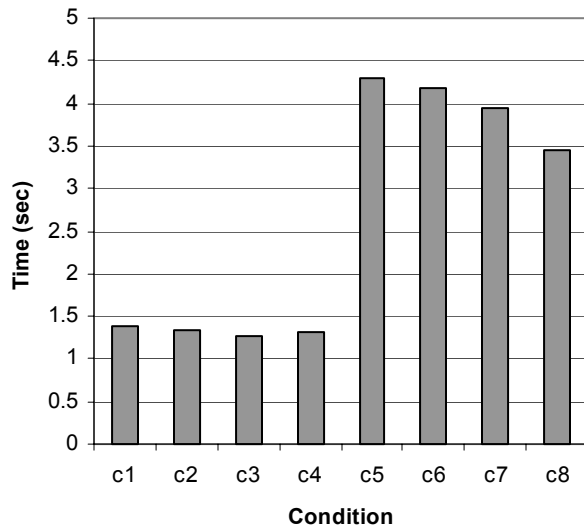
The main results are summarized in Figures 8 and 9 which show error rates and response times respectively. As shown in Figure 9 the use of static highlighting for one subgraph and motion cues for the other resulted in dramatically reduced error rates (2.6% versus 22.5%). A two-way ANOVA was carried out on errors, where the 8 conditions listed in Table 1 were one factor and the target present versus target absent was the other factor. The highlighting factor was highly significant ( $F(7,77) = 32.2$ ;  $p < 0.0001$ ). There was no significant main effect for target present vs target absent, but there was an interaction ( $F(7,77) = 4.1$ ;  $p < 0.01$ ) between the two factors. An examination of Figure 7 shows this to be due to condition c7 where there was a much higher error rate for target present conditions. In this condition there were far fewer *yes* responses than *no* responses. As was the case for Experiment 1, there was a relationship between error rates and response times. Only for this experiment the relationship was reversed. Response times were longer for conditions where there were higher errors.

A second ANOVA with response times as the dependent variable revealed a main effect for condition ( $F(7,77) = 43.2$ ;  $p < 0.001$ ) and a Tukey HSD test revealed three groups [ $\{c1,c2,c3,c4\} < \{c8\} < \{c5,c6,c7,c8\}$ ]. All of the conditions that used motion highlighting for one subgraph and static highlighting for the other were in the group with the shortest response times (mean 1.64 sec.). Condition c8 (pulse motion for one subgraph, horizontal motion for the other) was the next fastest with a mean response of 3.48 sec. The three remaining conditions formed a third group with a mean time of 4.15 sec. Overall the results strongly support the idea that the conjunction of motion and non-motion cues can be rapidly searched visually, and we failed to find pairs of motions that supported rapid visual searches.

On average, responses on trials when the two subgraphs contained common nodes (target present) were faster than responses when they did not (no target present) (2.49 sec vs 3.13 sec ( $F(1,11)=18.05$ ,  $p < 0.001$ ). There was no significant interaction between this factor and the display condition.



**Figure 8** A summary of the error rate results from Experiment 2. Grey bars show conditions where there was no target present. Black bars show conditions where there was a target present.



**Figure 9** A summary of the response time results from Experiment 2. See Table 1 for a description of the conditions.



## Discussion of Experiment 2

The results show that it is possible to independently highlight two subgraphs of a larger graph and clearly perceive nodes that belong to both subgraphs if one is highlighted using motion and the other is highlighted using static cues. This agrees with the theory that the conjunction of motion and non-motion cues can be searched rapidly. Our attempts to support the same task with two different motion codings were far less effective. In all the cases using motion cues in both subgraphs the error rates were greater than 20%. The bias towards *no* responses rather than *yes* responses for condition 7 is intriguing although we have no explanation for it.

## Conclusion

With our first experiment we showed that a technique of interactively highlighting subgraphs can efficiently support visual queries on graphs containing up to 3200 nodes. Motion highlighting proved to be as effective as static highlighting and the combination of motion and static highlighting was the most effective. Although this result showed that motion highlighting is an effective technique, we failed to confirm our previous finding that in some cases it can be better than static highlighting [23]. This is hardly surprising; a highlighting cue is not all or nothing. Even weak highlighting cues will increase the rate of visual search strong cues will make a search faster. Thus, whether or not static highlighting or motion highlighting is more effective is a matter of degree. Factors such as size and degree of contrast will be critical for static cues and factors such amplitude and frequency will be critical for motion cues. We can easily tune either static and motion cues to be the most salient. These are properties that the display designer must consider and the exact settings will depend on the requirements of the application.

The question arises as to the ultimate limitations of interactive graph highlighting in making large graphs accessible to visual queries. Part of the problem is simply space for the nodes. We used a very high resolution screen for the first experiment and it would probably be undesirable to pack nodes much more closely than this (see Figure 1). However, only half of the screen area was used suggesting that it might be possible to support interaction with node-link diagrams having double the size (6400) using the same techniques. In the future, we may expect wall sized displays having hundreds of millions of pixels of resolution. Such displays should make it possible to show substantially larger graphs, possibly containing tens or hundreds of thousand of nodes. However, unless highlighting methods are made even more distinctive, visual searches will inevitably take longer. Figure 5 already shows a marked increase in the time to respond with the largest (3200) node graphs and as graph size increased the visual search time could be expected to increase further. We already know that motion is an especially effective technique for highlighting in peripheral vision [24,2,3]. The motion highlighting methods we have developed are likely to become even more beneficial with large high resolution displays. Research is also needed to determine the limits on graph that can be interactively viewed on much smaller and more portable screens.

The main advantage of motion highlighting over static highlighting may be that it adds to the available lexicon of graphical techniques for making information available to rapid visual queries. Experiment 2 showed that motion highlighting in addition to static highlighting allows us to support visual queries on two subgraphs of a larger graph.

Although we have not tested all combinations of static and motion highlighting, we know from the extensive literature on pre-attentive processing that most conjunctions of static cues are not rapidly perceived [21]. However there is one conjunction of visual cues that might work, namely using stereoscopic depth in addition to static cues instead of motion [17,22]. Unfortunately, displaying stereoscopic depth requires technologies such as frame sequential glasses or lenticular screens that have significant drawbacks.

In Experiment 2 we only tested the visual query for nodes in common between two subgraphs. We did this because it seemed likely to be the most difficult case to support. It would be useful to empirically test support for the other queries we discussed in the introduction. It is possible, although we regard it as unlikely, that the motion highlighting for one subgraph and static highlighting for the other subgraph, would make it difficult to see whether nodes belonged to one subgraph or the other, as opposed to considering only the conjunction.

The results on motion highlighting should apply equally to other visual interfaces besides graph visualization. There may be a broad range of applications where motion highlighting can be used to highlight one attribute of a data set and a static technique can be used for another attribute in such a way that each attribute, or the conjunction of attributes, can be visually queried. In the visualization field, one obvious use is the interactive querying of scatterplots using techniques such as dynamic queries [1] or brushing [5].

Finally we wish to comment on the artificiality of the experiments we have conducted. The purpose of this paper has been to focus on the particular properties of effective interactive highlighting and to accomplish this we used artificial diagrams and a display that leaves out many of the requirements of a real application. A real application, for example would have to have a mechanism for displaying text associated with selected nodes, and allow for graphs having a much less homogeneous structure than the ones we artificially constructed. Additional techniques for showing the history of interaction would undoubtedly be required also. Still, our experiences with the email traffic graph shown in Figure 1 has convinced us that these techniques can be applied to real social network data. It is also possible to enhance the range of interactive highlighting techniques over the simple breadth first search we used in this study. For example, MEGraph contains an option to automatically show shortest paths in the graph when two nodes are selected. Ultimately, we envision a range on interactive queries that could be applied rapidly to a graph, incrementally building, or pruning, a selected subgraph depending on the task.

### **Acknowledgements**

This study was supported and monitored by the Advanced Research and Development Activity (ARDA) and the National Geospatial-Intelligence Agency (NGA) under Contract Number (NMA401-02-C-0019). The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official Department of Defense position, policy, or decision, unless so designated by other official documentation.

## References

- 1 Ahlberg C, Williamson C, Shneiderman B. Dynamic queries for information exploration. *CHI'92*. 1992 (Monterey) ACM Press; 619-626.
- 2 Bartram L, Ware C, Calvert T. Moticons: detection, distraction and task. *International Journal of Human-Computer Studies, Special Issue on Notifications and Interruptions* 2003; 58(5): 515-545.
- 3 Bartram, L. and Ware, C. Filtering and brushing with motion. *Information Visualization* 2002; 1(1): 66-79.
- 4 Becker RA, Eick SG, Wilks AR. Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics* March 1995; 1(1): 16-28.
- 5 Becker RA, Cleveland WS. Brushing scatterplots, In: Cleveland WS, McGill ME, (Eds). *Dynamic Graphics for Statistics*. Wadsworth & Brooks/Cole Advanced Books and Software 1988; 111-120.
- 6 Cutzu, F., and Tsotsos, (2003) The selective tuning model of visual attention. Testing the predictions arising from the inhibitory surround mechanism. *Vision Research*. 43(2) 205-219.
- 7 di Battista G, Eades P, Tamassia R, Tollis IG. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall Englewood 1999; Cliffs, NJ.
- 8 Brandes Y., Raab J. and Wagner D. Exploratory network visualization: Simultaneous display of status and connections. *Journal of Social Structure* 2001; 2(4): 1-28.
- 9 Driver J, McLeod P, Dienes Z. Motion coherence and conjunction search: Implications for guided search theory. *Perception and Psychophysics* 1992; 51(1): 79-85.
- 10 Driver J, McLeod P. Reversing Visual Search Asymmetries with conjunctions of movement and orientation. *Journal of Experimental Psychology, Human Perception and Performance* 1992; 18: 22-33.
- 11 Fructerman T, Reingold E. Graph drawing by force-directed placement, *Software Practice and Experience* 1991; 21(11): 1129-1164.
- 12 Huang ML, Eades, P, Wang, Online animated visualization of huge graphs using a modified spring algorithm. *Journal of Visual Languages and Computing* 1998; 9(6): 623-645.
- 13 Kleiberg E, van d Wetering van Wijk, JJ. Botanical visualization of huge hierarchies. *IEEE Symposium on Information Visualization*, 2001 (San Diego, CA), 87-94.
- 14 Lamping J, Rao R. 1994 Laying Out and Visualizing Large Trees Using a Hyperbolic Space. *UIST ACM Press*; 13-14
- 15 Munzner T, Guimbretière F, Robertson G. Constellation: A visualization tool for linguistic queries for MindNet. *IEEE Symposium on Information Visualization* 1999 (San Francisco, CA), October 25-26, 132-135.
- 16 McLeod P, Driver J, Crisp J. Visual search for a conjunction of movement and form is parallel. *Nature* 1988; 332, 154-155.
- 17 Nakayama K, Silverman GH. Serial and parallel processing of visual feature conjunctions. *Nature* 1986; 320, 264-265.

- 18 Plaisant C, Grosjean J, Bederson BB. SpaceTree: Supporting exploration in a large node-link tree, design evolution and empirical evaluation. *IEEE Symposium on Information Visualization 2002* (Boston, MA). 57-64.
- 19 Robertson G, Mackinlay JD, Card SW. Information Visualization using 3D interactive animation. *Communications of the ACM*. 1993; 36(4) 57-71.
- 20 Teoh ST, Ma L. RINGS: A technique for visualizing large hierarchies. *International Symposium on Graph Drawing 2002*. (Irvine CA) Springer, 268-275.
- 21 Triesman A, Gormican S. Search, similarity and integration of features between and within dimensions, *Journal of Experimental Psychology, Human Perception and Performance* 1991. 17(3), 652-676.
- 22 Ware C, *Information Visualization: Perception for Design*, 2<sup>nd</sup> Edition, Morgan Kaufman, San Francisco. 2004;
- 23 Ware C, Bobrow R. Motion to Support rapid Interactive Queries on Node-Link diagrams. *ACM Transactions on Applied Perception*. 2004; 1, 1-15.
- 24 Ware C, Bonner J, Cater R, Knight W. Simple animation as a human interrupt. *International Journal of Human-Computer Interaction* 1992; 4(4) 341-348.
- 25 Wills GJ. NichWorks: Interactive visualization of very large graphs. *Journal of Computational and Graphical Statistics* 1999; 8(2) 190-212.