# Difference Map Readability for Dynamic Graphs

Daniel Archambault, Helen Purchase, Bruno Pinaud

## To cite this version:

Daniel Archambault, Helen Purchase, Bruno Pinaud. Difference Map Readability for Dynamic Graphs. 18th International Symposium on Graph Drawing, Sep 2010, Constance, Germany. pp.50-61, 10.1007/978-3-642-18469-7_5 . inria-00514150

**HAL Id: inria-00514150**
**https://hal.inria.fr/inria-00514150**

Submitted on 1 Sep 2010

# Difference Map Readability for Dynamic Graphs

Daniel Archambault[1], Helen C. Purchase[2] and Bruno Pinaud[3]

[1] Clique Strategic Research Cluster, University College Dublin
daniel.archambault@ucd.ie
[2] Department of Computing Science, University of Glasgow
hcp@dcs.gla.ac.uk
[3] Université de Bordeaux I, LaBRI UMR CNRS 5800 & INRIA Bordeaux Sud-Ouest
bruno.pinaud@labri.fr

**Abstract.** Difference maps are one way to show changes between timeslices in a dynamic graph. They highlight, using colour, the nodes and edges that were added, removed, or persisted between every pair of adjacent timeslices. Although some work has used difference maps for visualization, no user study has been performed to gauge their performance. In this paper, we present a user study to evaluate the effectiveness of difference maps in comparison with presenting the evolution of the dynamic graph over time on three interfaces. We found evidence that difference maps produced significantly fewer errors when determining the number of edges inserted or removed from a graph as it evolves over time. Also, difference maps were significantly preferred on all tasks.

## 1 Introduction

**Dynamic graph drawing** deals with the problem of depicting a graph that evolves over time. Dynamic graph drawing algorithms typically represent the evolving graph as a series of timeslices. A **timeslice** encodes the structure of the graph at a given time. The timeslices, also known as the sequence of graphs, are often placed in chronological order, demonstrating graph evolution.

A few visualization systems [3, 6] have exploited difference maps to show the evolution of dynamic series of graphs. A **difference map** does not present the actual timeslices. Rather, for each pair of adjacent timeslices, it presents the union of the nodes and edges in both graphs. The nodes and edges are coloured one of three colours depending on whether they were added, removed, or persisted in the graph over that timeslice. Despite the use of difference maps in visualization systems, the effectiveness of this presentation method has yet to be evaluated.

Many different user interfaces have been used to present dynamic graphs to a user. In an **animation** of the dynamic graph sequence, nodes and edges that are added and removed from the drawing are faded in and out of the display. Node movement is smoothly interpolated so that the user of the system can more easily follow how the data has changed. One could also picture a **slide show** of the data whereby the data is presented like a Powerpoint presentation. No smooth transitions exist between drawings in the sequence, and the arrow keys are used to cycle through the graphs in the series. In a **small multiples** [21] interface, all timeslices are presented on the screen at once with each timeslice placed inside its own window. The user scans the matrix of windows to see how the graph evolves. All three interfaces could be used to present a series of difference maps, but we currently don't know which is the most effective.

This work presents a user study which investigates two research questions:

1. Do difference maps help improve the readability of dynamic graphs?
2. Under what interface do they help the most: animation, slide show, or small multiples?

We found that difference maps can help answer questions about large scale changes in terms of the number of edges in a graph. Also, difference maps were preferred over simply presenting the dynamic graph series as it evolves over time.

## 2 Previous and Related Work

Previous and related work is divided into three subsections. First, we present some of the work on difference maps in section 2.1. Section 2.2 presents work in dynamic graph drawing. Finally, section 2.3 presents a few user studies with results on dynamic graph drawing readability.

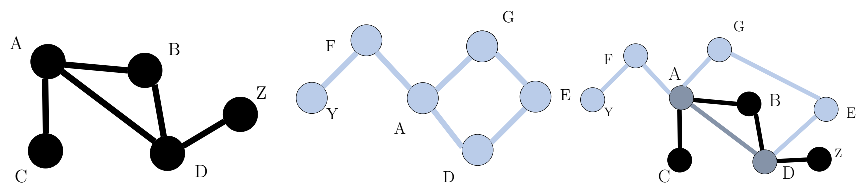
### 2.1 Difference Maps

Difference maps were designed to show the differences, in terms of nodes and edges, between a pair of graphs. They do so by taking the union of the nodes and edges in both graphs and colouring them by presence in one graph, the other, or both. Assuming that there is a unique identifier for each node of the graph, a one-to-one correspondence is available and the difference map can be computed in linear time. Fig. 1(c) shows a difference map computed from two graphs. The black nodes are only present in Fig. 1(a). Similarly, the light grey nodes are only present in Fig. 1(b). The grey nodes in Fig. 1(c), however, are present in both graphs.

Archambault [3] used difference maps and graph hierarchies to show where areas of a large graph changed. In this work, graph hierarchies or cluster trees, were used to simplify a large difference map into areas of similar evolution. The work also presented coarsening methods to make the diagrams simpler to read.

Bourqui and Jourdan [6] accentuated areas common to pairs of biological networks using difference maps. In the study of biological networks, structurally similar pathways often have similar function. By emphasizing structural similarities, both structure and context of functionally similar elements can be compared.

Both papers present techniques to visualize difference maps. However, neither of them have a user study to evaluate their effectiveness. These techniques



(a) Timeslice 1 (b) Timeslice 2 (c) Difference Map

**Fig. 1.** Two timeslices and the resulting difference map. **(a)** The graph at time 1. **(b)** The graph at time 2. **(c)** The difference map. Nodes and edges at time 1 are coloured black. Nodes and edges at time 2 are coloured light grey. Nodes that appear in both timeslices are coloured grey. The difference map encodes the nodes and edges that are added, removed, and persist over the dynamic graph series.

use other graph visualization techniques, such as graph hierarchies and fisheye views, that we do not test in this experiment. In this study, we are interested in evaluating the overall difference map approach in a dynamic graph context.

### 2.2 Dynamic Graph Drawing and the Mental Map

A number of dynamic graph drawing algorithms have looked at effective ways of preserving the mental map [16, 8, 7, 9, 14, 11, 13, 5], and various experiments have investigated the effect of preserving the mental map [17, 20, 18, 1].

In this study, we use the GraphAEL algorithm [11]. In this approach, inter-timeslice edges exist between nodes that are the same across timeslices. All timeslices are placed into the same plane and laid out using a force directed algorithm. The assigned strength of these inter-timeslice edges controls the amount of mental map preservation between timeslices: the higher the strength of the inter-timeslice edges, the shorter the distance nodes can move, increasing the degree of mental map preservation. Informed by previous experiments [18, 1], we use a relatively low level of mental map preservation over all conditions and factors in this experiment.

### 2.3 Animation vs Small Multiples for Dynamic Data

Several experiments have evaluated the performance of interfaces on dynamically evolving data. Most of these experiments have compared animation to small multiples on various types of data, and a pair of experiments have looked at this question in the context of dynamic graphs.

Griffen *et al.* [15] found that for clusters of moving hexagons against background noise, animation could be faster and more accurate than small multiples. Robertson *et al.* [19] compared animation, trace line, and small multiples visualization techniques on animated multi-dimensional data. The authors found that animation was the least effective form for analysis. Both small multiples and trace lines were significantly faster than animation, and small multiples was significantly more accurate. These varied results may indicate that the effectiveness of animation or small multiples strongly depends on the data to be visualized.

A pair of experiments have compared animation and small multiples in the context of dynamic graphs. Archambault *et al.* [1] compare small multiples and animation in addition to the effect of the mental map. In this experiment, small multiples was significantly faster overall and for most tasks. For tasks that involved the simultaneous appearance of nodes or edges in the data, animation was significantly more accurate, but it was not the case that more time led to fewer errors. Farrugia *et al.* [12] compared animation and small multiples on two dynamic graph series. The experiment found that small multiples was significantly faster for most tasks.

In this experiment, we focus on the effectiveness of the difference map and determining the appropriate interface for it. We also compare three interfaces, small multiples, animation, and slide show, in this context.

## 3 The Experiment

To test the effectiveness of difference maps with respect to interface, we performed a within subject experiment. We employed a 2 condition (no difference map (ND) vs with difference maps (WD)) $\times$ 3 factor (animation (Anim) vs slide show (SD) vs small multiples (SM)) $\times$ 2 data set (`threads2` and `van de Bunt`) $\times$ 4 question design. The following subsections provide the details of this design.

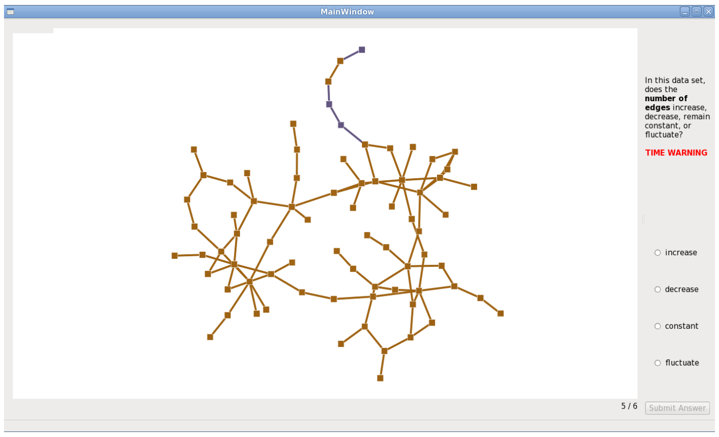
### 3.1 Interfaces

The animation interface is similar to a movie player. The current view of the graph takes up the entire screen and smooth transitions morph the graph from one timeslice to another. Nodes that are added to the data or removed from it are faded in or out respectively. The positions of nodes and edges are linearly interpolated between frames. At any time, the participant could stop the animation and drag the slider at their own rate. No other form of interaction, including zooming, is allowed. This interface was used previously in Archambault *et al.* [1].

The slide show interface is very similar to a Powerpoint presentation. In this interface, each timeslice takes up the entire screen as in the animation condition. However, no smooth transition exists between pairs of timeslices. At the bottom right corner of the interface, the current slide number and the total number of slides is indicated. The participant uses the arrow keys to advance to the next timeslice or rewind to the previous one. No other form of interaction, including zooming, is allowed. This interface is shown in Fig. 2.

In the small multiples interface, all timeslices are presented in a matrix ordered left to right and top to bottom. The participant scans the windows to determine the right answer. No other form of interaction, including zooming, is allowed. This interface was used previously in Archambault *et al.* [1].[1]

[1] Examples of each interface in operation under each condition $\times$ factor pairing for all questions are available at `http://www.labri.fr/perso/bpinaud/diffmap/`



In this data set, does the **number of edges** increase, decrease, remain constant, or fluctuate?

TIME WARNING

○ increase

○ decrease

○ constant

○ fluctuate

5 / 6

**Fig. 2.** Experiment interface. The question appears on the right along with four multiple choice answers. The participant selected the appropriate radio button and clicked "submit answer" to respond. The slide show interface under the difference map condition for `threads2` is shown in this figure.

| | | | | | | avg. | | | max | | | min | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Set | $\|N\|$ | $\|E\|$ | $d$ | max $d$ | min $d$ | $\|N\|'$ | $\|E\|'$ | $d'$ | $\|N\|'$ | $\|E\|'$ | $d'$ | $\|N\|'$ | $\|E\|'$ | $d'$ |
| `threads2` | 70.14 | 83.43 | 2.38 | 8 | 1 | 2.50 | 3.83 | 0.06 | 4 | 4 | 2 | 0 | 0 | 0 |
| `van de Bunt` | 23.14 | 32.43 | 2.56 | 9 | 1 | 3.33 | 6.17 | 0.78 | 11 | 25 | 6 | -6 | -24 | -6 |

**Fig. 3.** Graph statistics where $|N|$, $|E|$, and $d$ are the average number of nodes, average number of edges, and average degree respectively. Primes indicate changes over time. The values for max $d$ and min $d$ are the maximum and minimum degrees observed over all timeslices. The columns labeled avg. correspond to the average change in values over all timeslices. The columns labeled max and min are the maximum and minimum changes observed between any pair of successive timeslices in the data set.

## 3.2 Difference Map Encoding

A difference map, as previously described, is the union of a pair of adjacent timeslices in the dynamic graph sequence. Thus, given a sequence of $t$ graphs, there would be $t-1$ difference maps, depicting graph evolution. If a node is deleted between a pair of timeslices, it is light blue. If it is added, it is purple. Nodes that persist are brown. The same colour scheme is applied to the edges. In the non difference map condition, all $t$ timeslices are presented using the interface. All nodes, except those pertaining to the question, are coloured grey.

## 3.3 Data Sets

In this experiment, two graph series of similar size were used to gauge the readability of difference maps. Fig. 3 reports the graph series parameters.

`Threads2`, used in the work of Frishman and Tal [13], is a graph series representing online newsgroup discussions. Nodes are authors of newsgroup articles, and an edge exists between two authors if one replied to the posting of another. As authors and postings are never deleted, this data set always grows in size. We selected seven timeslices from this data: timeslices ten through sixteen.

The `van de Bunt` data set [10] is a network that was used previously in the experiment of Farrugia *et al.* [12]. The nodes in the graph are undergraduate students. An edge exists between two undergraduates if they self-reported in a survey that they had a relationship. In the original graph, there were a number of edge types which encoded if the relationship was best friends, friends, friendly neutral, or troubled. In our experiment, we used only the links that were best friends or friends. This data set fluctuates in size over the timeslices.

### 3.4 Tasks

Our tasks aim to test the readability of both local and global structure in the graph. More importantly, each question should require the participant to look at all timeslices, because the full dynamic evolution of the graph should be taken into account. We chose four questions.

The first question tests the evolution of node degrees in the graph and is a local, topology-based question. It is similar to the types of questions posed in Purchase *et al.* [18]. Four nodes were highlighted different colours, and participants were asked to select the colour of the node as the answer to the question. The remaining nodes and edges were coloured as specified by the condition.

1. Node degree changes. One of the following questions was asked:
   (a) Which vertex **increases** its **degree** over time?
   (b) Which vertex **decreases** its **degree** over time?
   (c) Which vertex **keeps** its **degree constant** over time?

The second question explores when specific edges appear in the graph and gauges if participants can see when a specific pair of edges is added to the data set. The question is local and is one of the most basic questions related to the dynamism of the graph. Four to six nodes were highlighted one of four different colours and participants were asked if a pair of thick edges simultaneously appeared adjacent to a node of a specific colour. The remaining nodes and edges in the graph were coloured as specified by the condition.

2. Which **edges** appear together **exactly once** over all timeslices?

The third question tests the ability of the participant to notice global trends in the graph. Specifically, the question tests if overall trends, in terms of the number of edges in the graph, can be perceived. All nodes and edges, for this question, were coloured as specified by the condition.

3. In this data set, does the **number of edges** increase, decrease, remain constant, or fluctuate?

Finally, we would like to use a question that tests the more global, topology-related readability of a graph. In this case, it involves reading a path through the graph between two nodes. In this question, a black focus node and four coloured nodes appeared in the graph. Participants were asked to select the coloured node that became closer, in a graph connectivity sense, to the black node. The remaining nodes and edges, for this question, were coloured as specified by the condition.

4. When a path exists between the black node and a node of each of the four colours, which path **only decreases** in length?

Nodes pertinent to each question were highlighted with colours to eliminate additional cognitive cost for searching for nodes or reading labels. Keywords appeared in bold, as shown above, so that the participant could easily recognize each question. Multiple choice questions, with four answers, were used. Asking participants to select nodes directly on the screen would put the animation condition at a disadvantage, because in this case, nodes often move on the screen.

## 3.5 Experimental Design

Each condition × factor pairing (ie: a pairing of difference map and presentation method, for example, difference maps with slide show) was placed in its own block, giving the experiment a total of six blocks. Each block started with a demonstration of the interface, allowing the participant to ask questions, find out about the experiment, and see how the answers could be found.

The blocks each had eight experimental tasks: 2 data sets × 4 questions. These eight tasks were prefixed with a practice block of four questions. During this practice block, each of the four questions was asked exactly once with two on `threads2` and two on `van de Bunt`. Eight versions of each question were found on both data sets. The first six versions were used as experimental data and the last two versions were only used in the practice blocks. Thus, for experimental data, the same version of the same question was never asked twice to the participant. For practice block data, the same version of the same question was never asked under the same condition. For each participant, the order of the questions within each block was randomized with versions of each question randomly selected.

To minimize the cognitive shift incurred by moving from the difference map condition to the non difference map condition, participants answered all questions on one condition followed by the other. However, conditions were counterbalanced by presenting the non difference map condition first to even participants and the difference map condition first to odd participants. The order of the interface blocks within each condition was randomized such that each participant had a unique interface order.

All three interfaces were rendered in real time using the Tulip framework [4]. No time limit was enforced per question or for the experiment overall. However, a warning label appeared on the screen after forty seconds had elapsed for each question, and participants were encouraged to finish their work quickly after that point. The animation started playing automatically after a delay of three seconds had elapsed and took about ten seconds to play in its entirety.

Each experiment was conducted individually with the researcher and took approximately 1 hour and 20 minutes, including the pre-experiment training, practice tasks, experimental tasks for both experimental conditions, and post-experiment questionnaire. Overall, there were twenty-five participants used in the final results. Participants were drawn from members of the Complex and Adaptive Systems Laboratory of University College Dublin (UCD CASL). All but one had computer science experience.

## 4 Results

In this section, we present the results for our experiment overall, divided by question, and divided by interface. We compare the difference map presentation (WD) to the standard representation (ND). A Shapiro-Wilk test, with a significance level of $\alpha = 0.05$, was used to determine whether or not the data was normally distributed. We found that the error rate data was not normally distributed whereas response time data was. As a consequence, we used an exact Wilcoxon signed rank test on the error rate data and a paired t-test on the response time data. For both tests, a significance level of $\alpha = 0.05$ was used. When we divided the data by question, we applied a Bonferroni correction, thus reducing the significance level to $\alpha = 0.025$.

### 4.1 WD vs ND

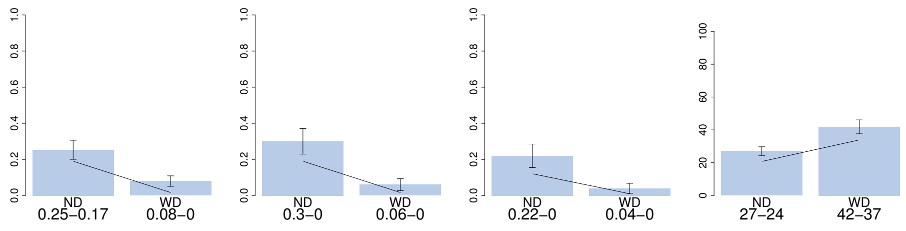
Overall, we did not find a significant difference, either in terms of error rate or response time, between the difference map condition (WD) and the simple graph timeslice condition without difference maps (ND) independent of interface.

**By Question** On questions 1, 2, and, 4, we did not find a significant difference in terms of error rate or response time when comparing WD to ND independent of interface. However, on question 3, we discovered that WD produced significantly fewer errors (WD 0.08, ND 0.25, $p = 0.0035$) as shown in Fig. 4(a). Neither presentation method was significantly faster.

### 4.2 WD vs ND Divided by Interface

We subsequently divided the data by interface to determine if, within interface, there were differences between the two presentation methods.

**Animation** When considering only the animation interface, we found no significant difference between either condition (WD and ND) overall or on a per question basis.



(a) Q3 Error All (b) Q3 Error SD (c) Q3 Error SM (d) Q2 Time SM

**Fig. 4.** Significant differences found in response time and error rate for this experiment. Error rate is percentage error and time is seconds. The mean and median values, separated by a dash, are written below each bar of the chart. SD and SM are the slide show and small multiples interfaces respectively.

**Slide Show** On slide show, we did not find a significant difference between WD and ND overall. However, for question 3, we found that WD produced significantly fewer errors than ND (WD: 0.06, ND 0.30, $p = 0.013$) as shown in Fig 4(b). We did not find a significant difference in terms of response time.

**Small Multiples** Under the small multiples interface, we did not find a significant difference overall. However, for question 3, we found that WD produced fewer errors than ND (WD 0.04, ND 0.22, $p = 0.008$). For question 2, we found that ND was significantly faster than WD (ND 27s, WD 42s, $p = 0.007$). These results are shown in Figs. 4(c) and 4(d) respectively. We did not find a significant difference in terms of response time.

### 4.3 Preference Data

A summary of our findings from the post-experiment questionnaire are shown in Fig. 5. Participants were asked to rank the six condition × interface pairs from 1 to 6, with 1 indicating the most preferred.

When we aggregated the results across interface, WD was preferred to ND for all questions and significantly so for questions 1, 3, and 4. For question 3, *In this data set, does the number of edges increase, decrease, remain constant, or fluctuate?* most participants remarked qualitatively that WD made this question much easier to answer. For question 2, some participants noted that the two colours used in the question made it more difficult to answer.

## 5 Discussion

### 5.1 Does the Difference Map Help?

Overall, we did not find that difference maps helped on all tasks. As our tasks are varied, this fact may not be all that surprising as the presentation method may not be suitable for all types of questions.

| | Q1 | | Q2 | | Q3 | | Q4 | |
|---|---|---|---|---|---|---|---|---|
| | ND | WD | ND | WD | ND | WD | ND | WD |
| Mean | 12.96 | 8.08 | 11.16 | 9.64 | 13.40 | 7.40 | 12.04 | 8.96 |
| Std. Deviation | 2.61 | 2.63 | 2.93 | 3.15 | 2.50 | 2.58 | 3.27 | 3.27 |

**Fig. 5.** Table of preference data comparing WD to ND aggregated across interface. WD was preferred in all cases and significantly so on questions 1, 3, and 4 ($p = 0.001$, $p = 0.000$, and $p = 0.028$ using a Wilcoxon signed ranks test).

When we divided the data by question, however, we did find that difference maps were able to significantly reduce the number of errors for the question which asked if the number of edges increased, decreased, remained constant, or fluctuated (question 3). These results were supported by the survey data which found difference maps were preferred significantly on this question. As difference maps highlight, using colour, if the edge has been added or removed, the participant can easily see where and by how much the graph has changed. Thus, difference maps may be helpful when trying to gauge how much the graph has changed between timeslices at a large scale.

## 5.2 Does it Help for All Interfaces?

There was no significant difference between the two conditions overall for any of the interfaces individually. However, on the slide show interface, WD produced significantly fewer errors than ND for question 3 (change in total number of edges). Using small multiples, ND was significantly faster than WD on question 2 (simultaneous appearance of edges). In terms of error rate for question 3 on this interface, WD produced significantly fewer errors. No other significant differences were found.

The results for question 3 are unsurprising considering that we saw an overall benefit of difference maps on this question globally. It seems that the benefit was achieved mostly using the small multiples and slide show interfaces. As these interfaces do not smoothly fade edges in and out, it may be the case that colour helped gauge when something was inserted or deleted. However, further study is required to confirm this conjecture.

The result on question 2 for the small multiples interface, that ND is significantly faster than WD, may be related to the fact that colour encoded both the answer to the question and graph structure changes. In the WD condition, participants had to contend with two sets of colours for each edge and reason about what the combination meant. Thus, the results suggest that the participants were able to perform the task equally as well, but it took them much longer to find the solution.

The difference map was significantly preferred for questions 1, 3, and 4 according to the survey data. It is surprising to get such strong preference data for one presentation method that does not match the corresponding performance data (which found little benefit in the use of difference maps when performing tasks). This result suggests that even if the use of difference maps may not improve performance, users might feel more comfortable with this presentation method when performing tasks on a dynamic graph sequence.

### 5.3 Limitations

We have tried to collect data that allows us to make generalizations for each interface, by including more than one data set, and more than one question. However, it would have been impossible to test a wider range of data sets and even more questions. The generalization of these results are therefore limited by these parameters. It was necessary for our participants to have some knowledge of graphs, meaning that our results only hold for this particular population. Running the experiment in a laboratory situation, on context-free graphs (even if based on real data sets) means that these results may not extrapolate to the visualization of graphs within an application context.

## 6 Conclusions and Future Work

In future work, it would be interesting to see if graph hierarchies can help improve the performance of difference maps. Hierarchies have been used in systems that present difference maps [6, 3], and a recent experiment [2] has provided some evidence that hierarchies can improve graph readability for some tasks. It would be interesting to see if difference maps can benefit from these representations.

We performed a user study to gauge the benefit of using difference maps rather than presenting the timeslices directly in a dynamic graph. We tested the readability of difference maps using three interfaces and four questions. In this study, we found that difference maps can help answer questions about large scale changes in a dynamic graph in terms of changes in the number of edges. Also, difference maps were strongly preferred by participants.

## Acknowledgments

The first author would like to acknowledge the support of the Clique Strategic Research Cluster funded by Science Foundation Ireland (SFI) Grant No. 08/SRC/I1407. We would also like to thank John Hamer for writing the dynamic graph layout algorithm based on the GraphAEL model. Finally, we would like to thank all of the participants who took part in this experiment.

## References

1. Archambault, D., Purchase, H.C., Pinaud, B.: Animation, small multiples, and the effect of mental map preservation in dynamic graphs. IEEE Trans. on Visualization and Computer Graphics (2010), to appear
2. Archambault, D., Purchase, H.C., Pinaud, B.: The readability of path-preserving clusterings of graphs. Computer Graphics Forum (Proc. EuroVis) 29(3), 1173–1182 (2010)

3. Archambault, D.: Structural differences between two graphs through hierarchies. In: Proc. of Graphics Interface. pp. 87–94 (2009)
4. Auber, D.: Tulip : A huge graph visualization framework. In: Mutzel, P., Jünger, M. (eds.) Graph Drawing Software, pp. 105–126. Mathematics and Visualization, Springer-Verlag (2003)
5. Boitmanis, K., Brandes, U., Pich, C.: Visualizing internet evolution on the autonomous systems level. In: Proc. of Graph Drawing (GD '07). LNCS, vol. 4875, pp. 365–376 (2007)
6. Bourqui, R., Jourdan, F.: Revealing subnetwork roles using contextual visualization: Comparison of metabolic networks. In: Proc. 12th Int. Conf. on Information Visualisation (IV'08). pp. 638–643 (2008)
7. Brandes, U., Wagner, D.: A bayesian paradigm for dynamic graph layout. In: Proc. of Graph Drawing (GD '97). LNCS, vol. 1353, pp. 236–247 (1997)
8. Cohen, R.F., Battista, G.D., Tollis, I.G., Cohen, R.F., Tamassia, R., Tollis, I.G.: A framework for dynamic graph drawing. ACM Symp. on Computational Geometry pp. 261–270 (1992)
9. Diehl, S., Görg, C.: Graphs, they are a changing – dynamic graph drawing for a sequence of graphs. In: Proc. of Graph Drawing (GD '02). LNCS, vol. 2528, pp. 23–31 (2002)
10. van Duijn, M., Zeggelink, E., Huisman, M., Stokman, F., Wasseur, F.: Evolution of sociology freshmen into a frendship network. The Journal of Math. Sociology 27(2), 153–191 (2003), `www.stats.ox.ac.uk/~snijders/siena/vdBunt_data.htm`
11. Erten, C., Harding, P.J., Kobourov, S., Wampler, K., Yee, G.V.: GraphAEL: Graph animations with evolving layouts. In: Proc. of Graph Drawing (GD '03). LNCS, vol. 2912, pp. 98–110 (2003)
12. Farrugia, M., Quigley, A.: Effective temporal graph layout: A comparative study of animation versus static display methods. Journal of Information Visualization (2010), to appear
13. Frishman, Y., Tal, A.: Online dynamic graph drawing. IEEE Trans. on Visualization and Computer Graphics 14(4), 727–740 (2008)
14. Görg, C., Birke, P., Pohl, M., Diehl, S.: Dynamic graph drawing of sequences of orthogonal and hierarchical graphs. In: Proc. of Graph Drawing (GD ' 04). LNCS, vol. 3383, pp. 228–238 (2004)
15. Griffen, A.L., MacEachren, A.M., Hardisty, F., Steiner, E., Li, B.: A comparison of animated maps with static small-multiple maps for visually identifying space-time clusters. Annals of the Association of American Geographers 96(4), 740–753 (2006)
16. Moen, S.: Drawing dynamic trees. IEEE Software 7(4), 21–28 (1990)
17. Purchase, H.C., Hoggan, E., Görg, C.: How important is the "mental map"? – an empirical investigation of a dynamic graph layout algorithm. In: Proc. Graph Drawing (GD '06). LNCS, vol. 4372, pp. 184–195 (2006)
18. Purchase, H.C., Samra, A.: Extremes are better: Investigating mental map preservation in dynamic graphs. In: Diagrams 2008. LNAI, vol. 5223, pp. 60–73 (2008)
19. Robertson, G., Fernandez, R., Fisher, D., Lee, B., Stasko, J.: Effectiveness of animation in trend visualization. IEEE Trans. on Visualization and Computer Graphics (Proc. Vis/InfoVis '08) 14(6), 1325–1332 (2008)
20. Saffrey, P., Purchase, H.C.: The "mental map" versus "static aesthetic" compromise in dynamic graphs: A user study. In: Proc. of the 9th Australasian User Interface Conference. pp. 85–93 (2008)
21. Tufte, E.: Envisioning Information. Graphics Press (1990)