The “Mental Map” versus “Static Aesthetic” Compromise in Dynamic Graphs: A User Study

Peter Saffrey and Helen Purchase

Computing Science Department
University of Glasgow
G12 8QQ, Glasgow, UK

{pzs,hcp}@dcs.gla.ac.uk

Abstract

The design of automatic layout algorithms for single graphs is a well established field, and some recent studies show how these algorithms affect human understanding. By contrast, layout algorithms for graphs that change over time are relatively immature, and few studies exist to evaluate their effectiveness empirically. This paper presents two new dynamic graph layout algorithms and empirical investigations of how effective these algorithms are with respect to human understanding. Central to each algorithm is the “mental map”: the degree to which the layout supports continuous understanding. This work aims to evaluate the importance of the mental map, alongside traditional static graph aesthetics, in answering questions about dynamic graphs. We discover that a simple concept of the mental map is not sufficient for increasing understanding of the graph.

Keywords: Dynamic graph layout, mental map, empirical study.

1 Introduction

Research on algorithms for the effective layout of graphs has been active for many years [12]. These algorithms have typically been valued for their computational efficiency and the extent to which they conform to pre-defined layout principles. Only recently has any empirical work been conducted to determine the effect of conformance to these principles on user understanding [11]. The main focus of the Graph Drawing research community is highly computational and algorithmic (e.g. [9,12]); the research reported here (and its predecessors (eg.[11])) is rooted within this community, while taking a user-centric approach.

While static graphs remain applicable in a variety of applications, recent developments in graph layout research have concentrated on the layout of dynamic graphs, representing relational information that changes over time. Examples of applications of such dynamic processes include software engineering visualizations where graphs depict execution time behavior, changing Internet usage, and changes in social network structures. Research on the effects of dynamic layout principles on user understanding is therefore timely.

In our previous study[10], we conducted an empirical study into dynamic graph layout algorithms applied to hierarchical data. Here, we extend this work to include more generic, undirected graphs. Until now, no such research exists in the literature.

1.1 Dynamic Graph Layout Systems

Existing research on dynamic graph layout algorithms focuses on the implementation of the algorithms themselves, rather than empirical evaluation (e.g. [5]).

Empirical investigation of a dynamic graph layout system requires that the system have changeable parameters, so that comparative tests can be performed.

A number of existing dynamic graph systems are available, including Gevol [3], GraphAEL [5] and others [1]. However, none of these systems provide continuous parametric adjustment. We have therefore developed and implemented our own interactive dynamic graph drawing system, GDG, which includes several novel dynamic graph layout algorithms based on the concept of maintaining the user’s mental map. These algorithms incorporate parameters which allow us to adjust the extent of mental model applied. Two of these algorithms are evaluated here.

1.2 The User’s “Mental Map”

A dynamic graph layout algorithm is typically based on two separate criteria. The first is traditional static aesthetics, such as node overlapping and edge crossings, applied to each time-slice. The second is a dynamic criterion known as “preserving the mental map” [2], the structural cognitive information maintained by a user as the graph changes between time-slices.

The mental map is manifested in a variety of features, including the preservation of orthogonality, clusters and topology [8]. In our work, we use the interpretation broadly taken in [8] and treat the mental map as the *preservation of node position* and by association, preserving the position of the adjoining edges.

There is often a contradiction between the static criteria and the mental map: preserving node position may force later time-slices to have a poor layout according to static criteria. The experiment reported in this paper addresses this contradiction. We attempt to determine whether, from the point of view of user understanding, it is preferable to maintain the mental map by preserving node position, conform to static layout aesthetics at each time-slice, or make a compromise between the two.

2. Experimental Overview

We conducted two studies to investigate the influence of the mental map on human understanding. The major difference between the two studies was the size of the graphs: the first study focused on graphs with approximately 20 nodes and 25 edges; the second, on graphs with approximately 80 nodes and 100 edges: these will be referred to as the small and large studies respectively. In the following discussion, methodological details of each study are the same unless stated otherwise.

2.1 Experimental Procedure

The experimental procedure used was based on former static graph layout experiments (e.g. [11]), using an online system to present the graphs, asking the participants to enter their answers to questions on the graphs and collecting error and time response data. Our work was also informed by our experience from our previous study on a dynamic layout algorithm [10].

At the start of the experiment, tutorial and worked example material was presented to each participant to familiarize them with the experimental tasks; at the end, the participants completed a ranking and qualitative questionnaire. A within-subjects methodology was used to reduce any subject variability, with the inclusion of practise tasks and randomization controlling for the learning effect. User-controlled rest breaks were included throughout the duration of the experiment to address any problem of fatigue.

We also needed to select experimental parameters to determine the timing of the dynamic graph animations. Through pilot tests we determined appropriate speed and time parameters. Adjustments were made to the difficulty of the questions used to avoid ceiling and floor effects; this will be discussed in section 3.3.

The only distinction between the small and large studies was a longer delay for answering the question in the large study, dictated by results from pilot experiments.

To display the experimental tasks we used DynaGUESS, a generic system for information visualisation experiments developed during our previous work. It facilitates easy preparation and customization of online experiments using svg files, enabling the experimenter to set parameters controlling timing, randomization and rest breaks. The data collected are the accuracy of the question answers and the response time.

2.2 The Mental Map: Restricting Node Movement

Our studies evaluate two related generic graph layout algorithms. Both algorithms (the *proportional restriction algorithm* and the *geometric restriction algorithm*) are based on combining an implementation of the spring embedder algorithm [7] with the principle of the mental map based on *preserving node position* between time-slices, described in section 1. They proceed as follows:

1. Lay out the first time-slice according to the spring embedder algorithm.
2. For each remaining time-slice
   a. Derive a candidate layout according to the spring embedder algorithm.
   b. Allow the nodes to move from the previous position towards the new candidate position, but restrict the movement of the nodes either by *proportion* or by *geometric distance*.

The difference between the proportional and geometric algorithms is at step 2b and is illustrated in Figures 1, 2 and 3. Figure 1 shows the movement of the nodes under an unrestricted force-directed algorithm; Figure 2 shows their movement when restricted by proportion; figure 3 shows their movement when restricted by geometric distance.

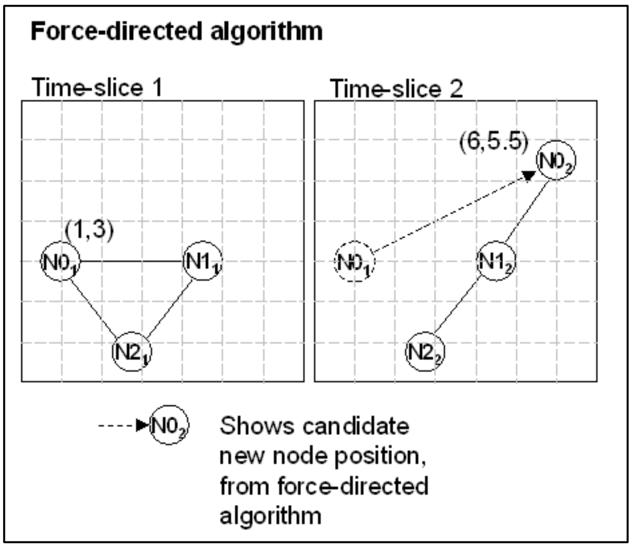

Figure 1: **Illustration of force directed algorithm node placement.**

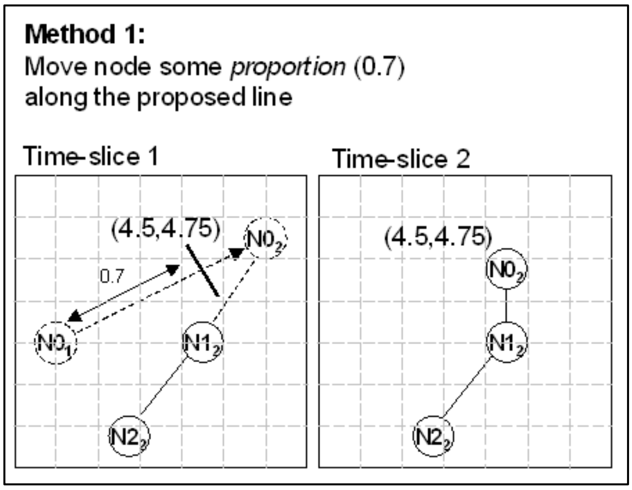

Figure 2: Restricting movement by proportion.

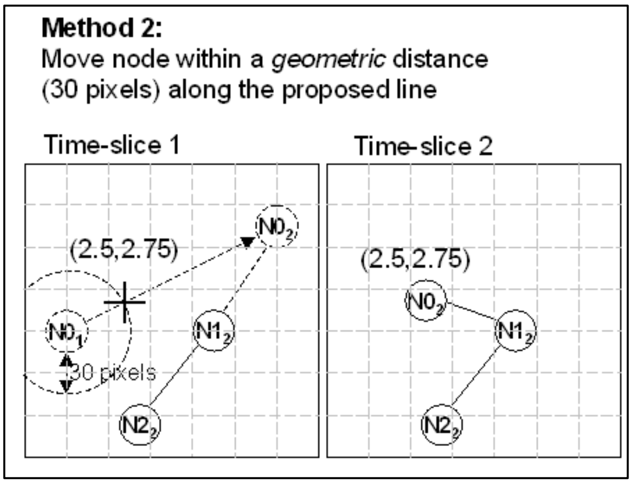

Figure 3: Restricing movement by geometric distance.

In each case, the first time-slice is laid out with the spring embedder algorithm. The second time-slice is also initially laid out with this algorithm but the movement of each node towards its new position is restricted along the line of movement. In *proportional restriction* (PR), this is a proportion [0..1] of the proposed movement. In *geometric restriction* (GR), this is up to a maximum geometric distance. In each case, a smaller parameter setting allows less movement and therefore a higher mental map. Similarly, in each case a value that imposes no restriction (a proportion of 1 or a very high geometric limit respectively) is equivalent to simply applying the spring embedder to each time-slice.

3 Experimental Design

Experimental design decisions were identical for the small and large studies.

For each algorithm, we used four conditions corresponding to *high*, *medium* and *low* mental map and a *zero* mental map condition. For convenience, we presented the four conditions for both algorithms together in a single experiment, but separated the small and large graphs into two separate experiments.

We randomly generated three dynamic graphs for each algorithm. Each dynamic graph was then laid out according to the four mental map settings (high, medium, low and none). The generation of the graphs will be described in section 3.2.

For each graph/condition combination we asked two questions to evaluate the ability of our subjects to understand different features of those graphs: identifying nodes with the highest degree and identifying the time-slice that contains the shortest path between two nodes. These questions were adjusted for difficulty based on our pilot studies. These questions will be described in section 3.3.

In each complete experiment, there were 48 tasks (2 algorithms x 4 settings x 3 graphs x 2 questions = 48), preceded by 16 practise tasks drawn randomly from those 48 with practise data discarded at the analysis stage.

3.1 Layout Conditions

These layout conditions were identical for the small and large studies.

- **PR** We chose to set the proportional restriction parameter to 0.2, 0.4, 0.6 and 1.0 to represent high, medium, low and no mental map respectively.
- **GR** We chose to set the geometric restriction parameter to 50, 100, 150 and 800 pixels to represent high, medium, low and no mental map respectively.

3.2 Graph Generation

The generation algorithm begins by generating the first time-slice with the specified number of edges randomly distributed between the specified number of nodes. This corresponds to an Erdos-Reyni graph with random model G(n,p) [4]. In successive time-slices, a random number of nodes and edges are added and removed up to the maximum specified by the relevant parameters. If a node is removed, all its edges are also removed. Graphs that are not fully connected are permitted and orphan nodes were removed.

Since we presented both proportional and geometric restriction algorithms together, we generated three different graphs for each algorithm. This was to prevent familiarity with any particular graph.

Graphs were generated in advance, and each participant in the study was shown the same set of generated graphs[2].

The graphs used were randomly generated based on the parameters shown in Table 1:

Table 1: Graph generation parameters for small and large studies.

	Small Study	Large Study
Number of time-slices	10	10
Initial number of nodes	20	80
Initial number of edges	25	100
Max. nodes added per time-slice	1	2
Max. nodes removed per time-slice	1	2
Max. edges added per time-slice	3	8
Max. edges removed per time-slice	2	6
Delete orphan nodes	Yes	Yes

[2] The graphs used are available at http://www.dcs.gla.ac.uk/~hcp/edge/ syntaxsvgs.html

3.3 Questions.

Each layout was tested with two questions.

For the small study these were:

- *Which of these nodes has the most edges: [yellow, green, red, purple]?*
- *In which time slice is the shortest path between these two nodes: [yellow,green]?*

For the large study these were:

- *Which of these groups of nodes of has the most edges: [yellow, green, red, purple]?*
- *In which time slice is the shortest path between these two groups of nodes: [yellow,green]?*

It is not possible to anticipate all the uses of dynamic graphs. These questions are intended to represent both node and path based properties of graphs.
In both cases, and for both experiments, we needed to control the difficulty of these questions to avoid ceiling and floor effects[3].

Highest Degree. The "most edges" question asks the user to find the node or group of nodes with the highest degree over all the time-slices.

For the small study, we chose four nodes from the graph, coloured these yellow, green, red and purple and asked users to identify the coloured node with the highest degree.

For the large study, we choose four groups of nodes where each group contained five nodes each. Each node group was coloured giving yellow, green, purple and red groups. We asked the users to identify the group with the most degree overall: the sum of the degrees of all the nodes in that group.

In both cases, we controlled the difficulty of the question by varying the difference in degree between the answer options. If the nodes or groups of nodes have similar degree, this makes the question more difficult.

Shortest Path. The shortest path question asks users to identify the time-slice where there is the shortest path between two nodes or groups of nodes.

The difficulty of this question is controlled by the length of the shortest paths that appear in each time-slice, since long paths are more difficult to follow. The difficulty is also dictated by the variation between the lengths of the shortest path in each time-slice, because it is easier to distinguish between shortest paths that have a greater variation in length. We controlled both these factors in both studies. In the large study, we ensured that nodes from the two groups were connected, otherwise, users would have to find shortest paths between a large number of scattered nodes.

Finally, we made sure that the correct answer only occurred in a maximum of two time-slices, to reduce the possibility of achieving a correct answer randomly.

The restrictions on the graphs were dictated by the desired difficulty of the questions and meant that several graphs were generated and subsequently discarded as unsuitable. The most suitable graphs for the shortest path question tended to be where the shortest path time-slices occur at the start or the end. We selected graphs for both the small and large study with shortest path questions that had answers at the beginning, the middle and the end of the set of time-slices.

3.4 Graph Display

Graph animations were generated in SVG format [6] and rendered using the Adobe SVG browser plugin for Internet Explorer. Text at the top of the display informed the user which time-slice is currently shown, or when in transition between time-slices.

Figure 4a and 4b show screenshots from the small and large graph experiments respectively.

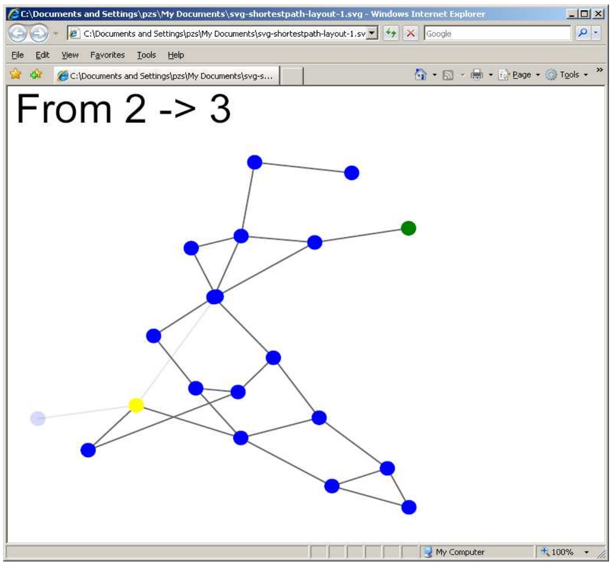

Figure 4a: Screenshot from the small study, showing the display between timeslices.

[3] Ceiling and floor effects occur when questions are too easy or too difficult respectively. If too many user responses are correct or incorrect, it can be difficult to distinguish between the influences of the different conditions on user performance.

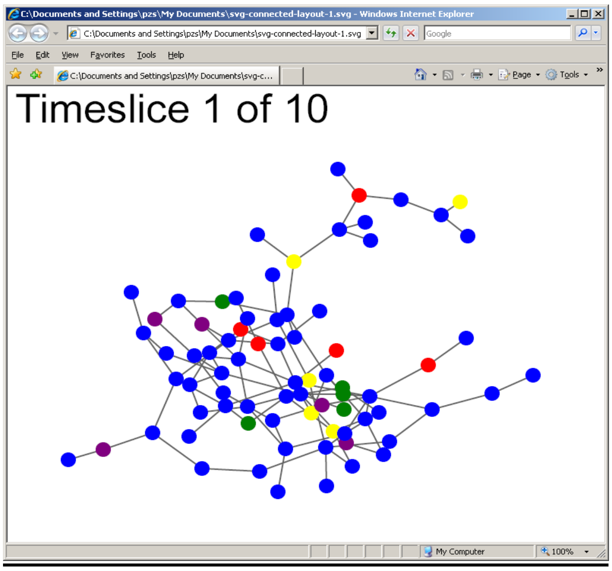

Figure 5b: Screenshot from the large study, showing the display at a specific time-slice.

3.5 Data Collection.

Participants were recruited from undergraduate and postgraduate courses in the Computing Science department, University of Glasgow. Each study used 21 participants at approximately one hour per participant. No significant problems were encountered during this process.

Our dependent variables were accuracy and response time. After the main study, we presented questionnaires asking for user preference on different layout conditions, as well as other qualitative feedback.

4 Results

The assumption is that preserving the mental map is important in understanding dynamic graphs [2,8]. However, as mentioned in section 1, preservation of the mental map could lead to poor layouts of later time-slices. We predicted that the best results would be achieved by a medium mental map, where a compromise is made between preserving the mental map and good layouts at each time-slice.

To analyse each set of these results, we used a standard two-tailed ANOVA analysis, based on the critical values of the F distribution, with α=0.05. In all cases, conservative readings of the distribution were used. Where significance was found, we compared each condition using Tukey's pairwise analysis.

We found no significant results in timing data. Therefore, all the results presented here are based on user errors, comparing different mental map conditions.

4.1 Small Study

Figures 5a and 5b show the overall errors for both algorithms in the small study. Each bar represents the average number of errors out of six over all participants: three graphs each presented for two different questions.

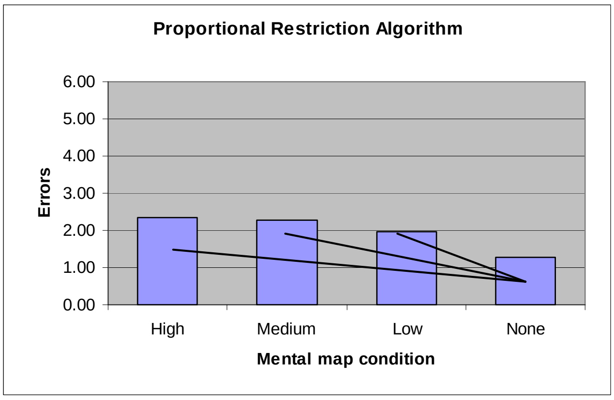

Figure 6a: Errors for small study, average out of 6, proportional restriction algorithm.

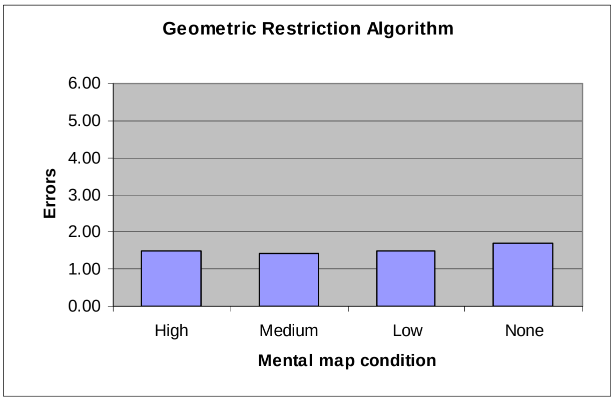

Figure 7a: Errors for small study, average out of 6, geometric restriction algorithm.

- The PR algorithm shows significance (F=3.6210 > F(4, 80)): the no mental map condition produces fewer errors than the other three conditions.
- The GR algorithm shows no significance.

Figures 6a and 6b show the errors for the PR algorithm only, separated into questions. Each bar represents the average errors out of three: three graphs for each question.

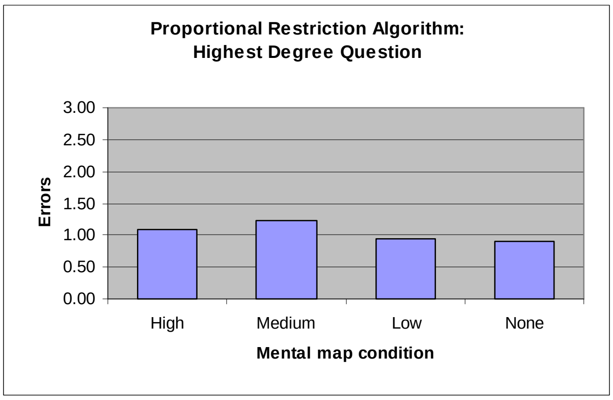

Figure 8a: Errors: small study, proportional restriction, average out of 3, highest degree question

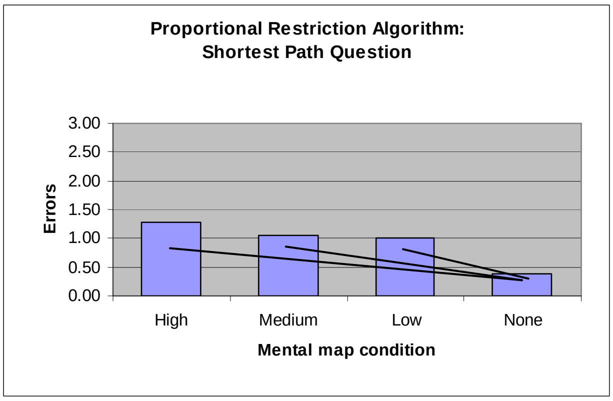

Figure 9b: Errors: small study, proportional restriction, average out of 3, shortest path question

- The highest degree question shows no significance.
- The shortest path question shows significance (F=5.4189 > F(4, 80)): again, the no mental map condition produces fewer errors than the other three conditions.

Figures 7a and 7b show the errors for the GR algorithm only, divided by question. Each bar represents the average number of errors out of three: three graphs for each question.

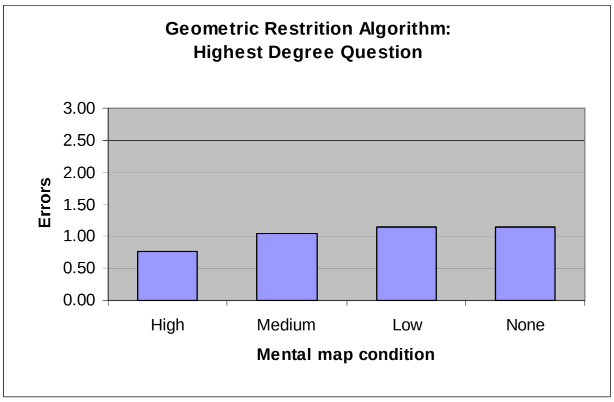

Figure 10a: Errors: small study, geometric restriction, average out of 3, highest degree question.

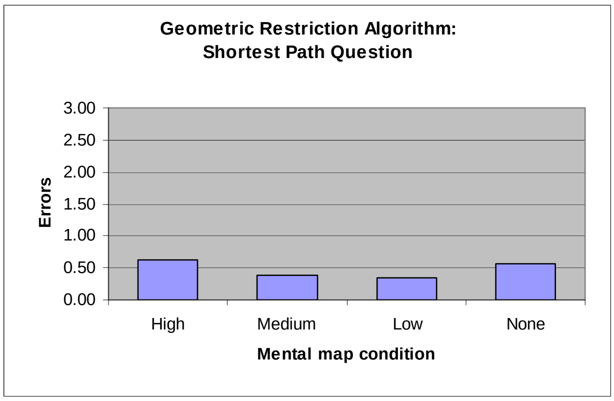

Figure 11b: Errors: small study, geometric restriction, average out of 3, shortest path question.

- Neither graph shows any significance.

4.2 Large Study

Figures 8a and 8b show the overall errors for both algorithms in the large study. Each bar represents the average number of errors out of six: three graphs each presented for two different questions.

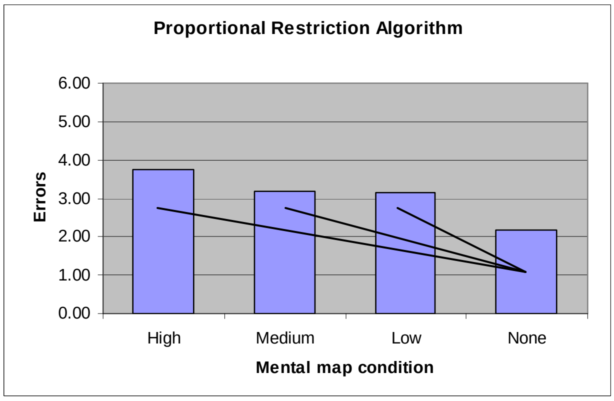

Figure 12a: Errors for the large study, proportional restriction, average out of 6.

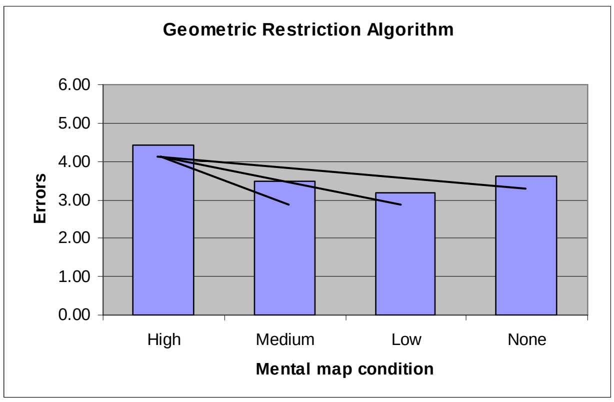

Figure 13b: Errors for the large study, proportional restriction, average out of 6.

- The PR algorithm shows significance (F=5.1303 > F(4,80)): the no mental map condition

produces fewer errors than the other three conditions.
- The GR algorithm shows significance (F=5.4675 > F(4,80)): the high mental map condition produces more errors than the other three conditions.

Figures 9a and 9b show the errors for the PR algorithm only, divided by question. Each bar represents the average number of errors out of three: three graphs for each question.

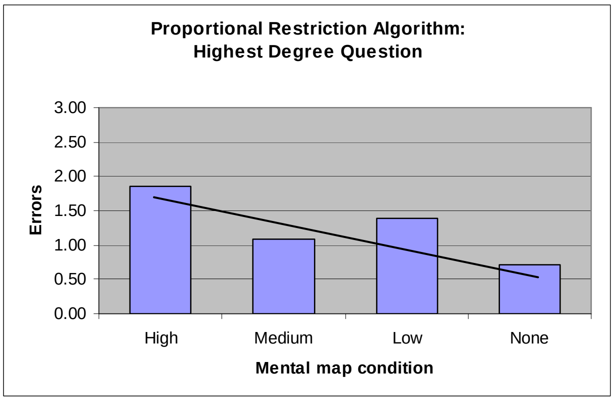

Figure 14a: Errors: large study, proportional restriction, average out of 3, highest degree question

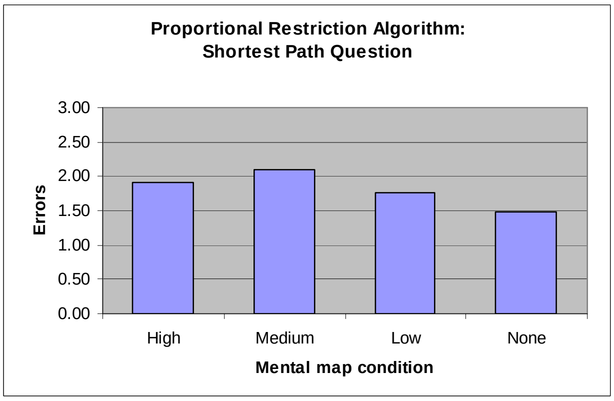

Figure 15b: Errors: large study, proportional restriction, average out of 3, shortest path question

- The highest degree question shows significance (F=4.5973 > F(4,80)): The high mental map condition produces more errors than the no mental map condition.
- The shortest path question shows no significance.

Figures 10a and 10b show the errors for the GR algorithm only, divided by question. Each bar represents the average number of errors out of three: three graphs for each question.

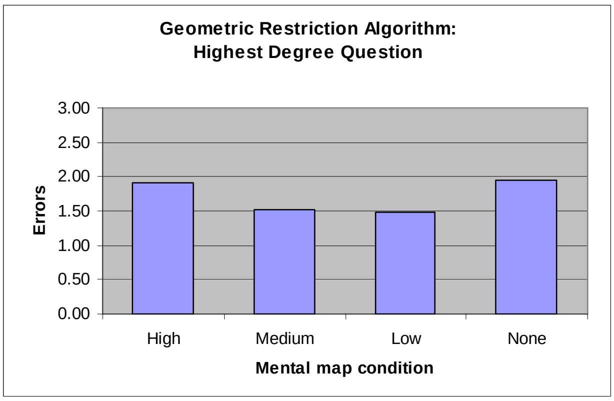

Figure 16a: Errors: large study, proportional restriction, average out of 3, highest degree question

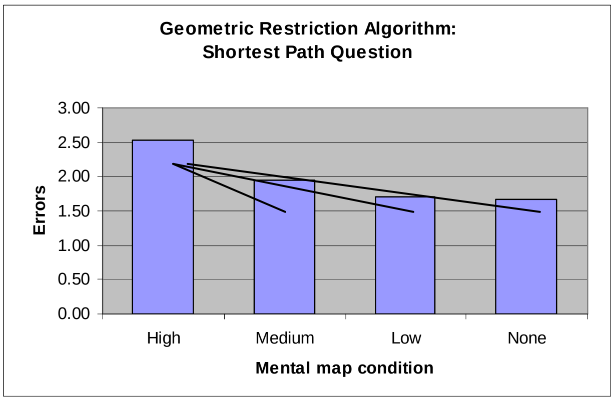

Figure 17b: Errors: large study, proportional restriction, average out of 3, shortest path question

- The highest degree question shows no significance.
- The shortest path question shows significance (F=6.6423 > F(4,80)): the high mental map condition produces more errors than the other three conditions.

5 Discussion and Analysis

At first glance our results are disappointing. Given the definition of the mental map as restricting the movement of nodes [2,8], our algorithms seem to be a sensible choice.

However, in all cases where there is significance, high mental map conditions produced the most errors, or no mental map conditions produced the least.

We can also make the following observations:

- It seems anomalous that the no mental map condition performed differently under the PR and GR algorithms. However, as described in section 3.3 the two algorithms were applied to different sets of graphs, and differences between the graphs may account for the difference in performance.

- This variation in graphs could also account for the difference in difficulty between graphs in the small study. Graph variety was important to make the results from our experiment generic to different graph structures.

One explanation for the poor performance of the high mental map condition is that both the proportional and geometric restriction algorithms can cause node overlapping, because restricting node movement interferes with the mechanisms used by the spring embedder algorithm to prevent node overlapping. This hypothesis is supported by several remarks in the participant questionnaires which cite overlapping nodes as the most important source of difficulty, particularly in the large study.

Another explanation could be that both the questions we have chosen are focused on the identification of edge properties in the graph. In [10] we found that preserving the mental map was only helpful in identifying particular nodes by name. If a task requires a user to track all nodes, the mental map may be important to associating a node between time-slices. However, our study made nodes in a task easy to track by applying a distinctive colour, which may have reduced the need for the mental map.

A further explanation is that several users commented that large node movements actually helped answer the questions, because the movement helped them to track the behavior of the node in relation to other nodes.

This analysis leads us to the following conclusions:

1. A notion of the mental map as a restriction on node movement is too narrow. The mental map implementation should consider the overall relation between nodes and edges, rather than merely position.
2. The mental map should work more harmoniously with static criteria, in particular node overlapping. Our studies show that preserving node position without also preventing node overlapping leads to poor layouts.
3. The mental map should also take into account that users do not necessarily find static nodes to be helpful in maintaining their mental map of a graph, because the movement helps them to track node position.
4. Exactly how the mental map should be represented in a dynamic graph layout will depend on the specific task.

The analysis of our study suggests that dynamic graph layout algorithms should be based on a more sophisticated concept of the mental map than simply restricting node movement. For example, algorithms intended to preserve the mental map could integrate more closely with static aesthetics such as node overlapping or edge crossings, and may include other dynamic layout criteria like preserving the angle or length of edges. These concepts would need to be applied in layout algorithms and evaluated using a similar study to those discussed in this paper. Our results would also suggest that the concept of the mental map should be developed in the context of the specific task to be addressed.

5.1 Conclusions

We have presented two new algorithms for dynamic graph layout, with parameters based on preserving the user’s mental map by restricting node movement. We have evaluated these algorithms on different mental map conditions using two different graph sizes. These results have been analysed to investigate the importance of the mental map on user understanding.

Our results seem to disagree with the accepted understanding of the mental map [2,8]. We found that preserving the user’s mental map by restricting node movement does not always contribute positively to understanding, at least where this causes significant node overlapping. In both studies, layout based purely on a static layout algorithm applied to each time-slice, with no restriction on node movement, produced the least errors. We have identified some possibilities as to why this has occurred, and suggest the development of a more sophisticated notion of the mental map.

6 Acknowledgements

Thanks to Eve Hoggan and Lauren Dempster for developing the original versions of the software tools used (DynaGUESS and GDG), students from the University of Glasgow for participating in our studies, and the EPSRC (EP/D077982). Ethical approval was given by the University of Glasgow (FIMS00177).

References

1 Brandes, U. and Corman, S. R. (2002): Visual unrolling of network evolution and the analysis of dynamic discourse. In INFOVIS ‘02: Proceedings of the IEEE Symposium on Information Visualiztion (InfoVis ’02), page 145, Washington, DC, USA, 2002. IEEE Computer Society

2 Coleman, M. K. and Parker, D. S. (1996): Aesthetics-based graph layout for human consumption. Software—Practice & Experience, - portal.acm.org

3 Collberg, C., Kobourov, S., Nagra, J., Pitts, J., and Wampler, K. (2003): A system for graph-based visualization of the evolution of software. In SoftVis ’03: Proceedings of the 2003 ACM symposium on Software visualization, pages 77-ff, New York, NY, USA, 2003. ACM Press

4 Erdős, P. and Rényi, A. (1961): On the strength of connectedness of a random graph. Acta Mathematica Hungarica, 1961

5 Erten, C., Harding, P. J., Kobourov, S. G., Wampler, K. and Yee, G. V. (2003): Graphael: Graph animations with evolving layouts. In Giuseppe Lotta, editor, Graph Drawing, volume 2912 of Lecture Notes in Computer Science, pages 98-110.

6 Ferraiolo, J. (2003): Scalable Vector Graphics (SVG) 1.0 Specification. w3.org

7 Fruchterman, T. M. J. and Reingold, E. M (1991): Graph Drawing by Force-directed Placement.

Software—Practice and Experience, Vol. 21(1 1), 1129-1164 (November 1991)
8 Lee, Y., Lin, C., and Yen, H. (2006): Mental map preserving graph drawing using simulated annealing. 2006 Asia-Pacific Symposium on information Visualisation - Volume 60. K. Misue, K. Sugiyama, and J. Tanaka, Eds.
9 Misue, K., Eades, P., Lai, W., Sugiyama, K. (1995) Layout Adjustment and the Mental Map. Journal of Visual Languages and Computing, 1995
10 Purchase, H. C., Hoggan, E. and Görg, C. (2006): How Important is the “Mental Map” – an Empirical Investigation of a Dynamic Graph Layout Algorithm. Graph Drawing 2006
11 Purchase, H. C. (2000): Effective information visualization: a study of graph drawing aesthetics and algorithms. Interacting with Computers, 13(2):147-162, 2000.
12 Tamassia, R., Di Battista, G., and Batini, C (1988): Automatic graph drawing and readability of diagrams. Systems, Man and Cybernetics, IEEE Transactions on, 1988