Using Games to Investigate Movement for Graph Comprehension

John Bovey
Computing Laboratory
University of Kent
Canterbury, UK CT2 7NF
+44 1227 827688
J.D.Bovey@kent.ac.uk

Florence Benoy
Computing Laboratory
University of Kent
Canterbury, UK CT2 7NF
+44 1227 823817
P.M.Benoy@kent.ac.uk

Peter Rodgers
Computing Laboratory
University of Kent
Canterbury, UK CT2 7NF
+44 1227 827913
P.J.Rodgers@kent.ac.uk

ABSTRACT
We describe the results of empirical investigations that explore the effectiveness of moving graph diagrams to improve the comprehension of their structure. The investigations involved subjects playing a game that required understanding the structure of a number of graphs. The use of a game as the task was intended to motivate the exploration of the graph by the subjects. The results show that movement can be beneficial when there is node-node or node-edge occlusion in the graph diagram but can have a detrimental effect when there is no occlusion, particularly if the diagram is small. We believe the positive result should generalise to other graph exploration tasks, and that graph movement is likely be useful as an additional graph exploration tool.

Categories and Subject Descriptors
I.3.6 [**Computer Graphics**] Methodology and Techniques – *Graphics data structures and data types*

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *Animation*

General Terms
Experimentation, Human Factors.

Keywords
Graph Movement, Graph Drawing, Diagram Comprehension.

1. INTRODUCTION
Many application areas make use of graph diagrams as a visualization tool. These include software engineering diagrams, WWW mapping, database visualization, network visualization and social network analysis. Although, in an ideal world, all these diagrams would be automatically laid out to make them as clear as possible, in practice occlusion occurs. This may be because the occlusion is unavoidable in large, dense graphs, or because there are other constraints on the positioning of the nodes. For example, the graph may be superimposed on a geographic map. If the nodes represented UK universities, and the physical locations were maintained, then there would be node-node occlusion in cities such as of London. Also, it may not be desirable to apply occlusion avoidance methods to the data when merging two graphs with well understood layouts to avoid disturbing the user's mental map of the graph. In addition, the computational expense of applying drawing methods to larger data sets may be too high or the appropriate graph drawing methods might not even be present in the visualization system. In any case, graph drawing methods do not always remove every case of occlusion, particularly node-edge occlusion. For all of these cases, displaying the graph diagram with the nodes and edges moving rather than stationary could provide a computationally and cognitively lightweight method for increasing the comprehension of the data.

Graph movement could involve moving one, several or all nodes in a graph but in this paper we have restricted ourselves to diagrams in which either all the nodes or none of the nodes are moving. This is motivated by the need to improve the investigation of complex graphs without affecting their current layout. There are various sorts of movement possible, but here we move nodes in a circular manner about their static location. The conjecture that we have attempted to test is that this animation of the graph can aid understanding by revealing misunderstandings, in particular node-node and node-edge occlusion, present in the graph.

The need for evidence that movement helps in the comprehension of graphs means that an empirical investigation was required. This involved subjects completing graph investigation tasks. To motivate the subjects we decided to make the investigation part of a game that could only be played successfully by subjects who understood the structure of the displayed graph

Whilst there is previous work in animation [2,3,4] and user controlled movement of 3D graphs [7], we can find no evidence of other research groups developing systems to aid comprehension by the automated regular movement of graph or similar data structures. Our paper at IV03 [1] introduced movement for the understanding of graphs. The initial study described in that paper showed some promising results, but the outcome was ambiguous. After this study we hypothesised that occlusion in graphs was a layout issue for which movement is beneficial. This paper describes a study that clearly shows this hypothesis to be true in the circumstances we have tested, and

hence is the first work demonstrating the utility of movement in comprehending graphs.

Section 2 of this paper describes the background to this research and motivates the choice of movement and games that were used for the study. Section 3 details the methodology of the empirical study, describes the software used, presents the data and gives our analysis of the results. Section 4 summarises the paper and lists possible further work possible in both graph movement and using games for empirical study.

2. GRAPH MOVEMENT

There are two major ways in which adding movement to a graph diagram may make it more comprehensible.

- Adding motion to a graph diagram could make the graph structure clearer. This is particularly true if there is occlusion present. If the graph is moving slightly then parts that are connected will move together whereas parts that look connected but, in fact, are not, will usually move independently.
- It should be possible to use motion in order to add additional information to a graph diagram in much the same way that colour or rendering is used. The simplest way to do this would be to use movement to highlight, and draw attention to, individual nodes – people are naturally alert to movement against a still background. Movement could also be used to add more complex information to graphs. For example, numerical parameters of nodes could be shown by using the speed or amplitude of the motion, communication could be shown by motion that ripples through the graph, and so on.

This paper is mainly concerned with showing that motion can be useful in the first of these two ways – clarifying graph diagrams in which there is some occlusion. Ideally, graph diagrams should not contain occlusion and there has been a great deal of research into algorithms that can automatically lay out graphs, but, for reasons listed in the introduction, it is not always possible to remove all occlusion from a graph diagram.

Since we want to use motion that is additional to an existing graph layout, the nodes should move in a small area and stay close to their original position. Various kinds of graph movement are possible (see [1] for a further discussion), however in this paper we used circular motion in which the nodes are out of phase with each other. That is, each node moves in small, smooth, circles of the same size and speed, but with different nodes at different points on the circle so that the nodes move relatively to each other. This decision was based on our experience and experimentation using the software.

2.1 Choice of Task

Often, the goal of empirical graph layout studies is to determine which of two or more different kinds of graph presentation is easer to comprehend. The presentations may differ in the kind of layout, the rendering or, in our case, the presence or absence of movement. In order to do this, users are shown graphs presented in different ways and are asked to perform a task that requires them to understand the structure of the graph in some way. We can then measure whether they succeed or fail in the task and, if they succeed, how long it takes. When designing such a study, the choice of task is quite important – it needs to have a number of properties:

- Success cannot be achieved just by looking at individual nodes – it depends on comprehending the overall structure of the graph in some way.
- It can be easily understood by subjects who are not familiar with graph theory or graph terminology.
- Success and failure need to be easy to measure and easy to recognise by the subject. If this is not the case then we can end up with the subjects trying to do one thing while we are measuring their success at something else.
- The task should not be too onerous and, ideally, should even be enjoyable. This is because we need to keep the subjects motivated to do as well as they can at the task. If the sequence of trials is boring or frustrating then we run the risk of having subjects who do not care how well they do but just want to get the study over with.

It is possible to think of quite a wide range of tasks that could potentially be used in graph comprehension experiments. One natural family is that of locating simple sub-graphs such as triangles, squares or tetrahedrons. The difficulty with these is in striking the balance between searches that are too difficult and frustrating, and searches that are too easy.

Another fairly natural task is that of finding a path between two pre-selected nodes. This does have the advantage of being easy to understand but it also has some drawbacks. If the subject is asked to find any path, without any restriction on its length, then they may be tempted to select nodes at random until they get lucky. If we attempt to motivate them to find a short path (with a reward, say) then we run into difficulties comparing performance – is a longer path found quickly better than a shorter path that takes longer to find? If we require them to find the shortest path then it is often not obvious to the subject whether they have succeeded or not. Also, the difficulty of finding the shortest path in a graph diagram seems to be critically dependent on the length of the path. Paths with 4 or less edges are easy to find by selecting neighbours of the end-points but longer paths can be very difficult and frustrating to find.

One promising family of tasks is games that the subject can play against the computer. Although games are more difficult to set up than simpler tasks, they do have a number of attractive properties.

- They generally have an unambiguous outcome: win or lose.
- They are easy to explain and understand, particularly by our most plentiful pool of potential subjects, undergraduate students.
- They are, at least potentially, easy to motivate. People naturally want to win games they play and winning is a reward and a source of motivation. It should also be possible to devise games that give feedback about how well the player performed.

2.2 Examples of Games on Graphs

These are just some examples of possible graph games. The first group are ones in which the players (subject and computer) alternately select nodes. This can be visualized as putting pieces on the nodes as in Go or Hex.

Path Finding The subject has to form a path between two pre-selected end-point nodes while the computer has to block the path.

Path Blocking The computer tries to construct the path while the subject does the blocking.

Triangles The winner is the first player to select three nodes that form a triangle.

Graph Othello The goal is to trap a path full of your opponent's pieces between the piece that you are playing and another of your pieces – then, the trapped pieces become yours. The winner is the one with the most pieces on the board at the end.

Another family of games is those in which the players each occupy one or more nodes and can move their pieces to adjacent nodes. For example:

Treasure Hunt Some nodes have treasure that can be claimed by visiting the node before your opponent. The winner would be the player who has the most treasure at the end of the game. This game has the motivational advantage that rewards happen during the game as well as at the end.

Pursuit The goal is to corner your opponent and prevent him moving or possibly to take his pieces by moving on top of them.

One potential drawback of using games is that it is necessary to implement a playing program for the computer. In practice, though, this is not too difficult since the program does not need to play perfectly, just well enough to provide a challenge for a novice human. The most important thing is that the human subjects should not be able to adopt very trivial strategies in order to win. For example, a user who is trying to block a path between two nodes should not be able to do so by simply picking off all the neighbours of one end point.

3. EMPIRICAL PROCESS AND ANALYSIS

The study was designed to answer the research questions "Does adding motion to graph diagrams increase users' comprehension?" and "If movement does have benefits, how do they depend on the presence of occlusion and on the size of the graph?"

3.1 The Graph Movement Software

The controlled display of moving graph diagrams is a new research area and so we have had to develop our own software. Our primary tool is a program called *Gran* that runs on Unix, is written in C and uses the Gtk+ graphics and user interface library. Gran is a dual-purpose program that is used both to create moving and still graphs, and also to present them for game playing by subjects in an investigation. In its editing mode, Gran has controls that allow graph diagrams to be laid out, edited, and made to move in different ways. It also allows the experimenter to try out games on the graphs as they are constructed. At present, the only games we have implemented are *Path-finding*, *Path-blocking* and *Triangles,* but it will not be difficult to add other games in the future. Graphs are stored in an XML format that includes information about the graph's motion and default game as well as the lists of nodes and links.

When it is used in experiment-mode, Gran does not display the editing controls but, instead, displays a message inviting the subject to start playing, along with a single button that can be used to cancel a game or start the next one. In this mode the program also automatically logs the outcome of the game, how long it took and the times of all the moves.

3.2 Games and Movements Used

There is plenty of scope for experimenting with different kinds of graph movement and different games but, for this experiment, we wanted a conclusive result, which meant that we needed to keep the experiment simple. This meant that we had to make some arbitrary choices. The style of motion that we settled on is circular with a diameter of 23 pixels and a circumference of 64 pixels. In the moving graphs, the nodes move at a speed of 25 pixels per second which means that they do about one revolution every 2.5 seconds. Each node in the graph was a assigned one of eight equally spaced phases. Figure 1 illustrates an example of how node movement reveals occlusion. The node shown with a white background is shown on the left at the top of its cycle (12 O'clock). The middle diagram shows it at approximately 8 O'clock and the right diagram shows it at approximately 4 O'clock. The other two nodes are also moving and all three nodes have different phases.

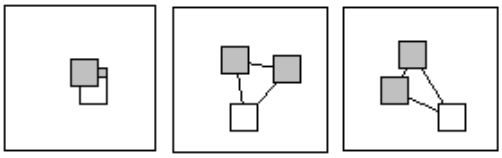

Figure 1. A snapshots of three occluded nodes rotating.

When it came to the choice of game, a primary requirement was for a short learning time since each subject attends a single session and there was not much time for practice. We also wanted to use a game with a simple outcome that would be easy to analyse statistically, ruling out games like Triangles that could end in a draw. In the end, we chose the Path Blocking game since it proved relatively simple to choose suitable graphs that were neither too difficult nor too easy.

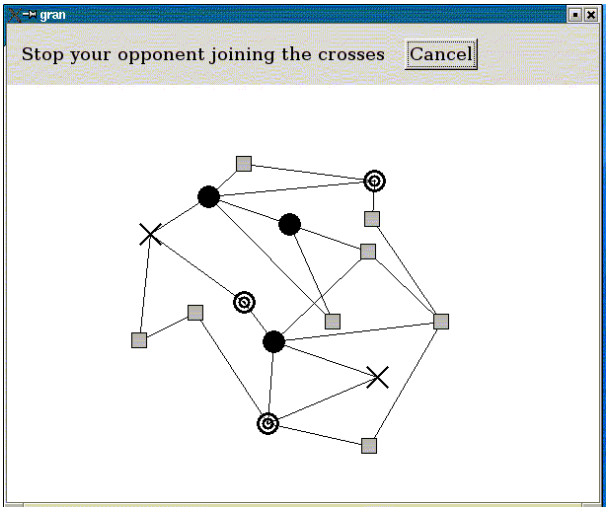

Figure 2. A game in progress in graph M2. The nodes shown with double circles are those chosen by the user, the black filled circles are those chosen by the computer. The square nodes have not yet been selected by either player.

Figure 2 shows a game in the process of being played. The computer is attempting to find a path between the two crosses and the subject is attempting to block all paths. In all the games played in the study the user played first. In this game, it is the subject's turn to play. In the current state of the game shown there is one node that the subject can select to win the game. If the subject selects any other node, then the computer will select it on it's next turn and so win the game.

3.3 The Context of the Trial

The subjects for our investigation were 56 undergraduate and postgraduate students who were mostly studying Computer Science. The subjects were given £5 for attending and a further £5 for succeeding in at least 40 out of all 50 games (the 10 training graphs and the 40 data graphs). In the event 38 out of the 56 subjects gained the extra amount. The subject with the largest number of wins was given an additional £10. The best score was by one subject who got 47 out of 50. Since there were 20 distinct graphs, each presented moving and then still or vice versa, there were 56 x 20 = 1120 pairs of data. Of the 20 graphs 4 had node-node occlusion, 4 had node-edge occlusion and the other 12 graphs were categorised by size, 4 small, 4 medium and 4 large. There were also 10 training graphs, of which 5 were presented moving and 5 still. The data from the training graphs was discarded for the purposes of analysis.

The nature of the data was such that a subject could either win or lose each game and each subject saw each graph both moving and still. The graph order and whether the subject played a game with a moving graph first or a still graph first was varied by presenting each subject with one of 24 different graph sequences. Each sequence started with 10 training graphs that were followed by the 40 test graphs, (20 different graphs, each moving and still). The sequences were devised so that each subject saw half the graphs moving first and half still first, with the still and moving versions of each graph being well separated by other graphs. In addition, each graph appeared first moving in half the sequences and first still in the other half.

The subjects performed the tasks in one of four sessions, with between 4 and 18 subjects in each session. An introduction to the game and interface was presented to the subjects and a walkthrough of the first three training graphs. They were informed that if they did well (gaining over 40 on the test) they would get an extra £5, and that the subject with the greatest number of wins would get a further £10, with the time taken for the test as a tie break, should two or more subjects get the same highest number of wins. The students were then told to perform the tests. Once they had finished they were asked to read a de-briefing document explaining the nature of the research and to complete a questionnaire before leaving the room.

The graphs were laid out by hand and several are shown below. They had around 15 nodes and 20 edges, with two nodes (shown as crosses) being pre-selected as the path end-points. Where occlusion was present, there were two instances of occlusion, and both instances of occlusion involved a path between the two end-point nodes. The graph displays that we used for the experiments had red nodes and green edges on a black background but, for the sake of clarity, they are shown here using black and grey on white.

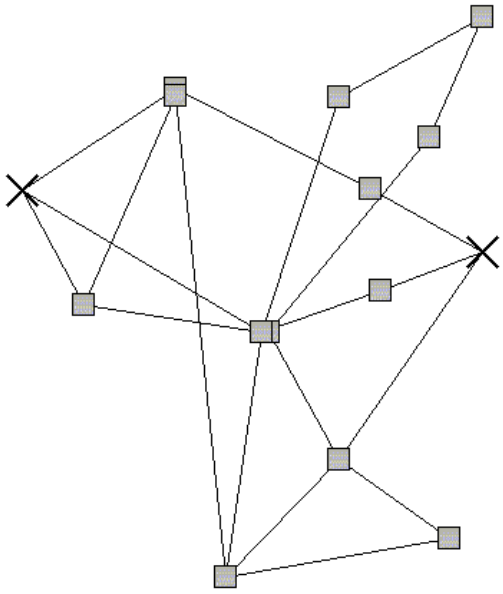

Figure 3. Graph NN1, which contains node-node occlusion

Figure 3 shows a graph with node-node occlusion. The occlusion used was partial, so that a fraction (approximately $1/5^{th}$) of the rear node is visible. The visibility of the rear node was intended to show the user that there is an additional node present, but the overlap is sufficient to make it difficult to discern which node is connected to which edge in the still diagrams.

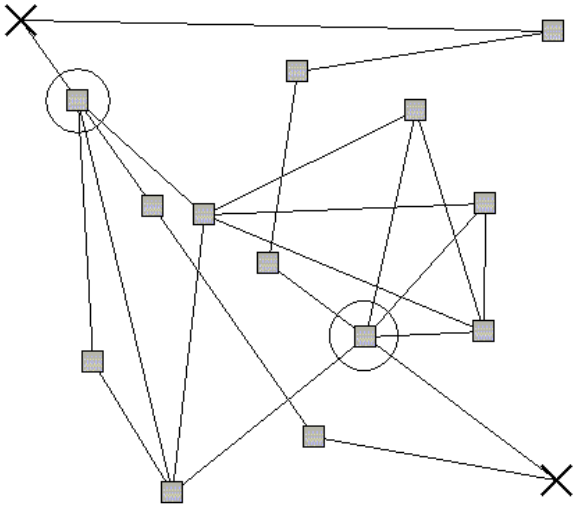

Figure 4. Graph NE3, which contains node-edge occlusion

Figure 4 shows a graph with node-edge occlusion. The circled nodes occlude edges, which means they lie on top of the edges without connecting to them. All the instances of node-edge occlusion were exact, that is the edge crossed the centre of the nodes.

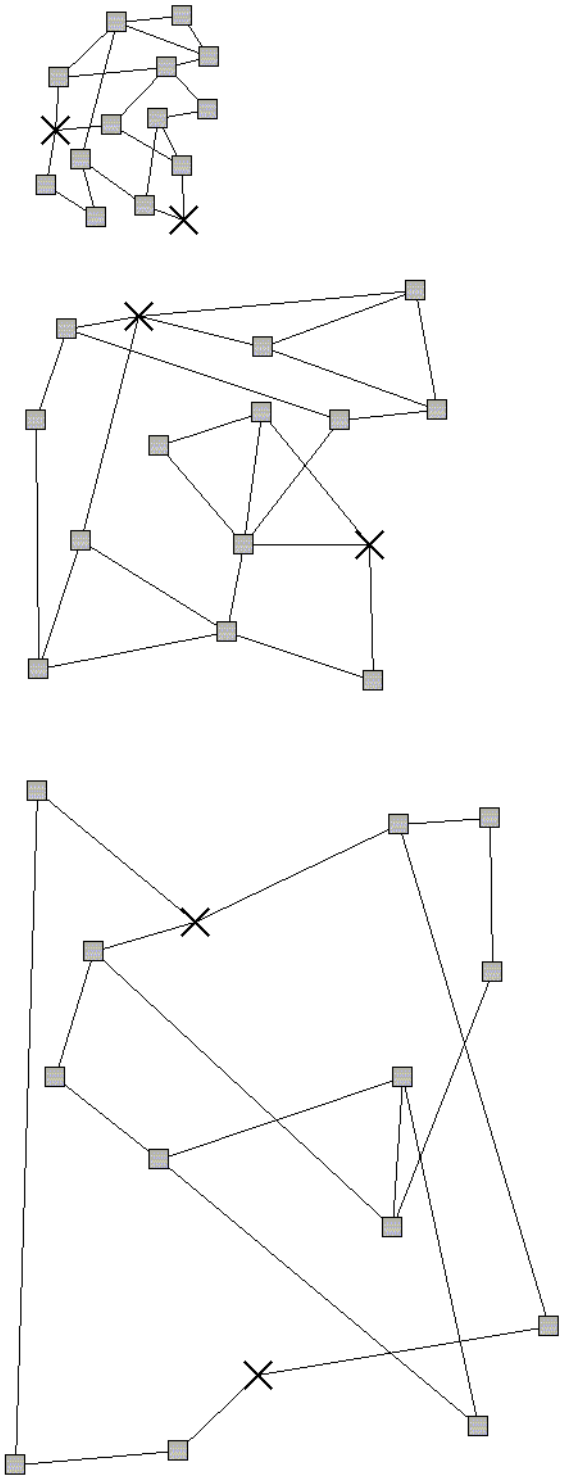

Figure 5. A small, medium and large graph (graphs S2, M4 and L4)

Figure 5 shows the relative proportions of small, medium and large graphs. The number of nodes and edges does not change, only the size of the bounding box of the graph. The relative proportions of the graphs was decided by the subjective view of the investigators after initial experimentation with the games and movement. All the occluded graphs have the proportions of the medium graph.

3.4 Data Analysis

The data was effectively gathered over two distinct sets of graphs, those that were characterised by occlusion and those characterised by size. The occluded graphs were of 2 types those with node-node occlusion, see Figure 3 and those with node-edge occlusion, see Figure 4. There were 4 graphs of each type.

The graphs characterised by size did not have any occlusion and had similar numbers of nodes and edges. This meant that those characterised as large had considerably more 'white space' between the structural components (edges and nodes) than those characterised as small. There were 4 graphs in each of the 4 categories small, medium and large.

Our analyses were considered in two ways:

Paired data Since each subject was presented with all graphs, both moving and still, in the first instance we consider a comparison of winning times. For the data to be paired we require that each subject has a winning time for both the moving and still versions of the graph.

For both the occluded graphs and those characterised by size, the null hypothesis was that there was no difference between the times over the moving graphs and the still graphs. The chosen test was the student-t test over paired data. To suggest that there was a difference in performance we would look for a T-statistic with an absolute value greater that 2 and a p-value less than 0.05.

Non-paired data We include more of the gathered data by generalising over average times and the number of wins for a particular graph, but also with a more general notion of performance. We consider better performance not only by comparison over times but in terms of a won game against a lost game; that is, for each graph a subject performed better with the moving (still) graph if i) both games were won and the moving (still) time was better or ii) if the moving (still) game was won and the still (moving) game was lost. Only when a subject lost both games was the data discarded.

This last, along with comparisons of average times and the number of wins, is presented in straightforward comparisons over the individual graphs and depicted in bar charts.

All of the bar charts presented here concern comparisons between the results over moving and still graphs. The charts are all constructed to allow ease of reading. The bars are in groups of three: the leftmost bar is always the result for the still version of the graph, the middle bar for the moving version and the rightmost bar for the difference between the two.

The difference is always calculated so that a positive difference, that is, above the horizontal axis, indicates a more favourable outcome for the moving graphs and a negative difference a more favourable outcome for the still graphs. Note that times are measured in milliseconds.

Particular graphs are named along the horizontal axes. The graphs are named with the following conventions:

- NE1 indicates the first of four graphs distinguished by node-edge occlusion
- NN1 indicates the first of four graphs distinguished by node-node occlusion
- L1, M1 and S1 denote the first of four graphs distinguished by size: large, medium and small respectively.

3.4.1 Observations over the occluded graphs

Paired data A two tailed student-t test was carried out over the paired data for the occluded graphs with a null hypothesis that there was no difference between performance over the moving and still versions of the graphs. The difference between performance over moving and still graphs was characterised as *ln(st/mt)*, that is, the log of still-time divided by the moving-time, allowing a normal distribution of the data.

The outcome of the test with a p-value (the possibility of rejecting the null hypothesis when it is true) so small, that correct to 3 significant figures it was 0, and a t-statistic of 6.17 indicates that the null hypothesis be rejected and that there is a significant difference in the performance over the moving and the still graphs. The accompanying confidence interval for the difference, see Table 1, has upper and lower bounds that are positive indicating that performance over the moving graphs is better than over the still.

Graph Type	Mean	SD	p	T	95%CI
Occluded	0.1204	0.2882	<0.0005	6.17	0.0819, 0.1589

Table 2

Non-paired data

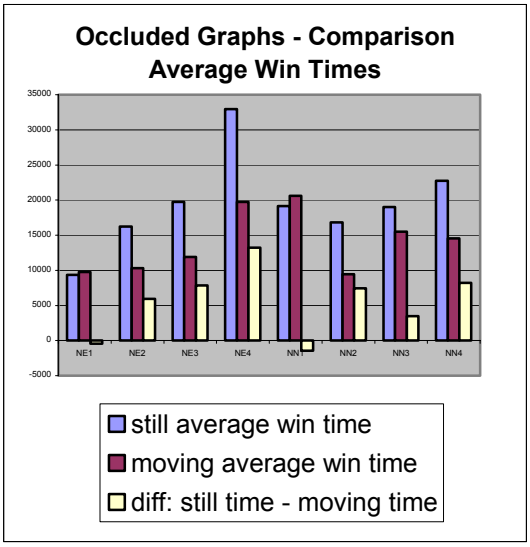

Figure 6

Figure 6, above, shows the outcome over all data where the only exception to faster times for the moving graphs is with a very small negative difference for NE1 and NN1.

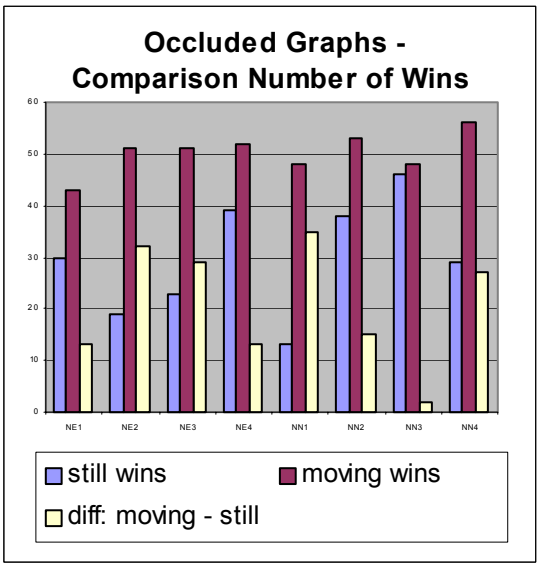

Figure 7

Figure 7 shows a comparison of the number of wins for each game. Note that for graphs NE1 and particularly NN1, where the average time over wins was slower with movement, that there are more overall wins over the moving graphs.

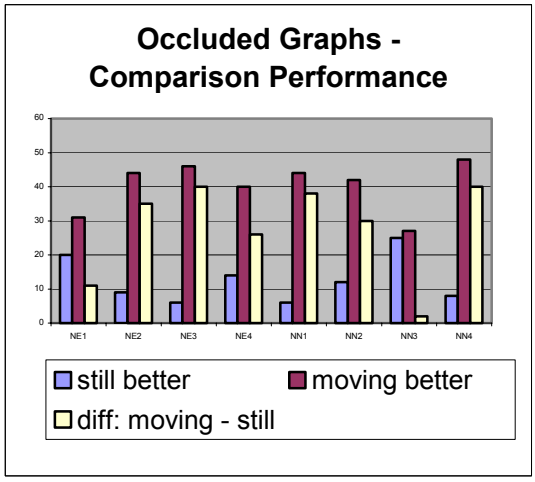

Figure 8

Figure 8 presents a third view of the data characterised not simply by win or lose the game but by performance as defined in the preamble to this section (3.4). The moving graphs out perform the still as differences are uniformly both positive and substantial with the exception of NN3 where there is only a small, but nevertheless, positive difference.

3.4.2 Observations categorised by size

Paired data A two tailed student-t test was carried out over the paired data for the graphs categorised by size in the same way as

with the occluded graphs. The difference between performance over moving and still graphs was again characterised as *ln(st/mt)*, that is, the log of still-time divided by the moving-time, allowing a normal distribution of the data.

The outcome of the test with a p-value (the possibility of rejecting the null hypothesis when it is true) so small, that correct to 3 significant figures it was 0, and a t-statistic of -4.65 indicates that the null hypothesis be rejected and that there is a significant difference in the performance over the moving and the still graphs. The accompanying confidence interval for the difference, see Table 1, has upper and lower bounds that are negative indicating that performance over the still graphs is better than over the moving.

Graph Type	Mean	SD	p	T	95%CI
Sized	-0.0483	0.2314	<0.0005	-4.65	-0.0686, -0.0279

Table 1

Non-paired data

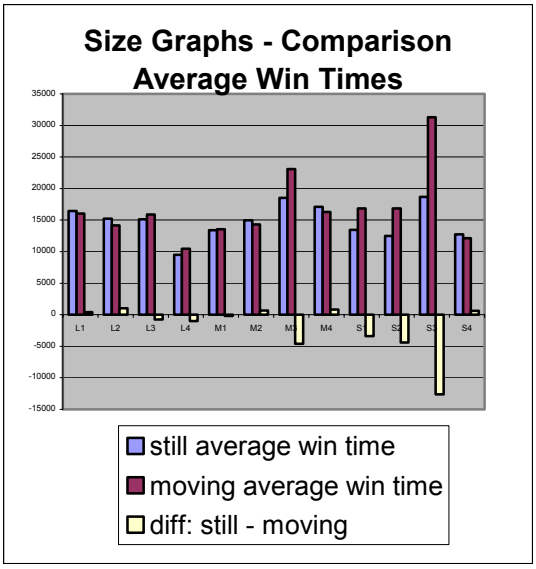

Figure 9

Figure 9 shows average times for both moving and still graphs, note the longer moving times with M3, see Figure 14, and all of the small graphs, except for S4, see Figure 13.

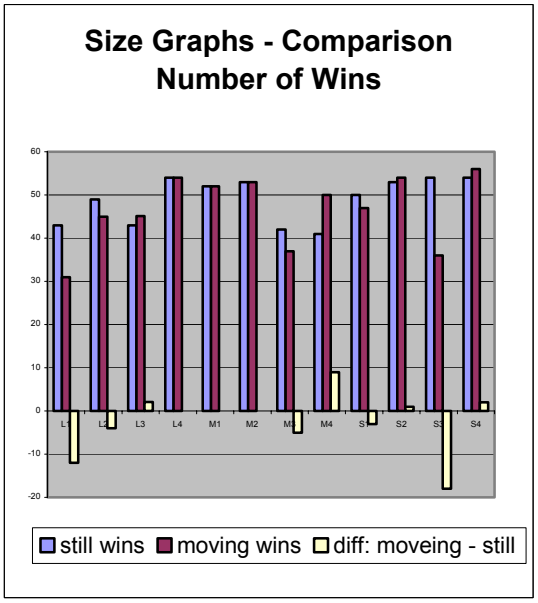

Figure 10

Figure 10 shows more wins overall for the still graphs but with the exception of L1, see Figure 12, and S3, see Figure 15, the actual discrepancy is small.

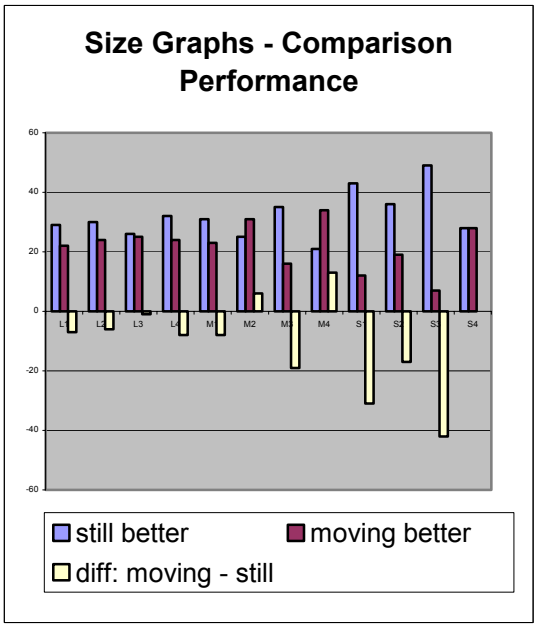

Figure 11

Figure 11 shows better performance overall for the still graphs particularly the small graphs and M3, see Figure 14. The notable exception is S4, see Figure 13.

Summary: By observation over the charts and the outcome of the t-test statistics above, there is strong evidence to suggest that movement is clearly an aid to understanding for occluded graphs but movement is not an aid to understanding for still graphs.

However, the above results raise several questions about particular graphs and further questions concerning the way movement is added to a graph.

Performance with graph L1 (see Figure 12) is not as good as with the other large graphs and it might be reasonable to assume that a fairly well spaced graph could not be hindered by movement, but some effect is apparent. Inspection of the graph reveals that the two crosses between which the path must be blocked are almost as far apart as they could be, this is not the case with the other large graphs. It is possible that this is a contributory cause.

Similarly performance over graph M3 (see Figure 14) is markedly better for the still graphs, but there is no apparent reason for the disparity between M3 and the remainder of the medium sized graphs.

It may seem not unreasonable to expect that with small graphs with no occlusion that the still graphs would perform better as the movement may hinder the understanding of the graph structure. With graphs S1-S3 performance favours the still graphs, but performance with graph S4 (see Figure 13) is virtually the same over both moving and still graphs alike and yet there is no obvious characteristic to distinguish S4 from the other small graphs

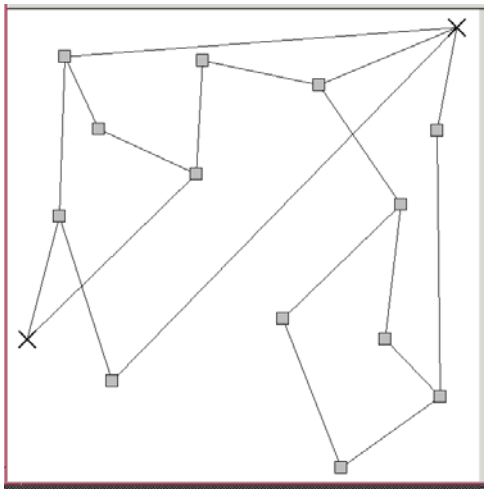

Figure 12. Graph L1

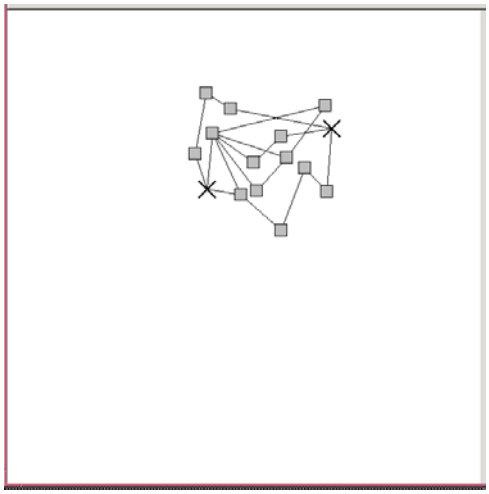

Figure 13. Graph S4

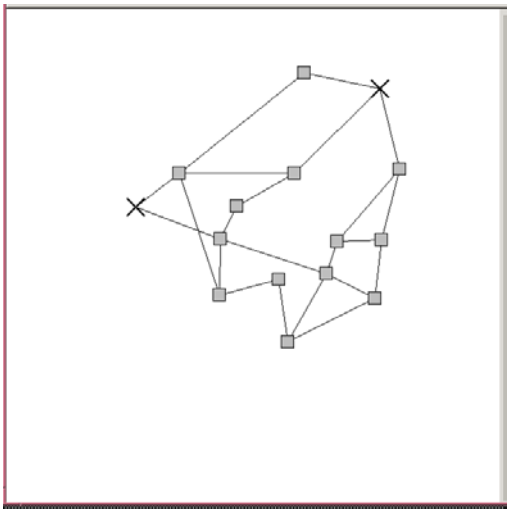

Figure 14. Graph M3

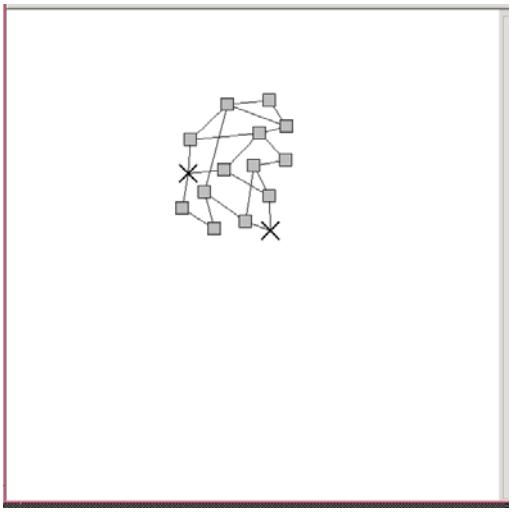

Figure 15. Graph S3

3.4.3 The questionnaire

All of the subjects were asked to fill in a questionnaire after the trials. Of the 56 subjects 12 commented, in one way or another, on the fact that node-edge or node-node occlusion was present it made it difficult to discern the underlying structure of the graph. Table 3 presents their appraisal of the effectiveness of movement in helping them in their task, and given that the subjects were not informed that the presented graphs came from two distinct groups their appraisal is not at variance with the outcome of the data analyses.

	Easier with movement	**Easier still**	**Depended on the graph**	**No real difference**
Blocking your opponent	15	15	26	0

Table 3

4. SUMMARY AND FURTHER WORK

The investigations that we have carried out show that adding motion to graphs with node-node occlusion or node-edge occlusion makes them easier to comprehend. They also appear to show that adding movement to smaller graphs makes them harder to comprehend, suggesting that there is a trade-off between the positive effect of motion in resolving occlusion, and a negative effect caused by adding relatively large amounts of motion compared to the size of the graph. This is a useful first step but it leaves a lot of questions unanswered, for example:

- What is the minimum motion that can be applied to a graph and still significantly help with resolving occlusion?
- What is the maximum movement that can be applied before it makes the graph harder to comprehend?
- Graph motion has several parameters governing: speed, distance, the path followed by individual nodes and the way in which nodes move relative to each other. How do these parameters affect graph comprehension with and without occlusion?
- Is there any kind of graph motion that makes a non-occluded graph easier to comprehend?
- There are ways in which motion might be used to add extra information to a graph diagram, do any of these work?

The only way to try and determine the answers to questions like these would be to do a large number of empirical investigations. However, this is not feasible with the kind of investigation procedure we have described in this paper. Our investigations were aimed at getting clear statistical evidence for a single hypothesis – that adding motion to graph diagrams helps with comprehension, particularly if the diagram contains occlusion. We feel that we have succeeded in that but we only tried one kind of motion and the study was time consuming to set up and fairly expensive to run (we paid out nearly £500 to the subjects). What is needed is some way to do relatively lightweight exploratory investigations which attempt to get tentative answers to some of the questions above. Hopefully, these would generate some hypotheses that could be tested using more formal controlled studies like the one described in this paper.

One approach might be to develop graph games that are sufficiently rewarding to play so that a significant number of people would want to play them for enjoyment. We could then re-implement the game program as a Java applet and provide the games on a web site. Players would see a series of games on graphs of different kinds, some moving and some still, and the applet would return the results to the server so that they can be logged for analysis. If such a system can be made to work then it could also be used as an experimental platform for other kinds of graph layout and graph rendering techniques.

5. REFERENCES

1. J.D. Bovey, P.J. Rodgers, and P.M. Benoy. Movement as an Aid to Understanding Graphs. In 7th International Conference on Information Visualization (IV03), pages 472-478. IEEE, July 2003.
2. S. Feiner, D. Salesin and T. Banchoff. Dial: A Diagrammatic Animation Language. IEEE Computer Graphics and Applications 2,9 pp. 43-54. 1982.
3. C. Friedrich and P. Eades. Graph Drawing in Motion. Vol. 6, no. 3, pp. 353-370. 2002.
4. F. Höfting, E. Wanke, A. Balmoŝan and C. Bergmann. 1st Grade - A System for Implementation, Testing and Animation of Graph Algorithms. LNCS 665, pp. 706-707. 1993.
5. H.C. Purchase, R.F. Cohen, and M. James. Validating Graph Drawing Aesthetics. GD95, LNCS 1027, 435-446. 1995.
6. P. J. Rodgers and N. Vidal. Graph Algorithm Animation with Grrr. In Agtive99: Applications of Graph Transformations with Industrial Relevance, LNCS 1779, pages 379-394. 2000.
7. C. Ware, G. Frank. Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions. ACM Transactions on Graphics Vol.15, no. 2, pp. 121-140. 1996