

Evaluation of Animation Effects to Improve Indirect Manipulation

Bruce H. Thomas

School of Computer and Information Science
University of South Australia
Mawson Lakes, SA, Australia 5095
Bruce.Thomas@unisa.edu.au

Victor Demczuk

Land Operations Division
Defence Science and Technology Organisation
Salisbury, SA, Australia 5108
vic.demczuk@dsto.defence.gov.au

Abstract

This paper describes an experiment we have conducted to explore the effectiveness of animation in improving indirect manipulation operations. Indirect manipulation operations are those initiated by command menus and buttons, to perform a transformation on a graphical object or set of graphical objects. The particular improvement is an operation's ability to show both what would happen if the operation is committed and what would happen if it were cancelled, while an operation is being considered. The experiment required subjects to watch a simple alignment operation for a set of graphical objects. They were then asked to record the original placement of those graphical objects. Each task used one of four visual cues: modified telltale, wiggle, colour, or no visual cue. We found the modified telltale, wiggle, and colour visual effects significantly more effective than no visual feedback for cuing the user as to original position of the graphical objects. The modified telltale and colour effects were significantly more effective than the wiggle effect.

1 Introduction

We are interested in exploring how techniques borrowed from cartoon and computer animation may be used to enhance the experience of interacting with a computer. In particular, we wish to apply animation to the interface itself—to enhance or augment the effectiveness of human interaction with applications that present a graphical interface.

In previous work, we have used animation techniques to help smooth abrupt transitions in the visible state of the screen [14] and to reinforce the illusion of direct manipulation that underlies many human computer interfaces [13]. To demonstrate these ideas, we extended a graphical editor to incorporate animation into the direct manipulation of graphical objects. We also applied animation principles to interface operations where on-screen changes come about

indirectly as a result of commands initiated from menus and buttons [15].

This paper presents a new set of animation effects to provide visual cues to highlight information about changes that come about indirectly. The paper presents a set of results from an experiment carried out to show how effective these different visual cues are in improving a user's ability to visualise the pre and post positions of graphical objects during an indirect manipulation alignment operation.

1.1 Direct and indirect manipulation

In his seminal paper [9], Shneiderman identified the essential ingredients of direct manipulation: immediate response to actions, incremental changes, and reversible effects. Properly applied, these characteristics can convince users that they are directly interacting with application data, even though they know it is all an illusion [7].

Direct manipulation works best when there is a straightforward mapping from user actions to visible changes on the screen; typical examples include dragging an object to a new position, stretching it to make it larger, or rotating it around an origin. But some changes have no obvious mapping (like changing an object's colour), and others require more precision than can readily be achieved by direct action (like rotating an object by exact quarter turns or moving it to exactly align with another). One way to permit such operations is to provide commands that carry out the desired change, together with a way to invoke the commands as needed. In the context of graphical interfaces, the commands are usually invoked by pressing a button in a toolbar or selecting a choice from a menu. We call this style of interaction *indirect manipulation*, since an action on one object (a button, for instance) causes a change to another object.

Previously we have shown how animation techniques in our drawing editor can enliven the objects the user manipulates directly or indirectly, strengthening the sense that “things are happening.” For example, we display the be-

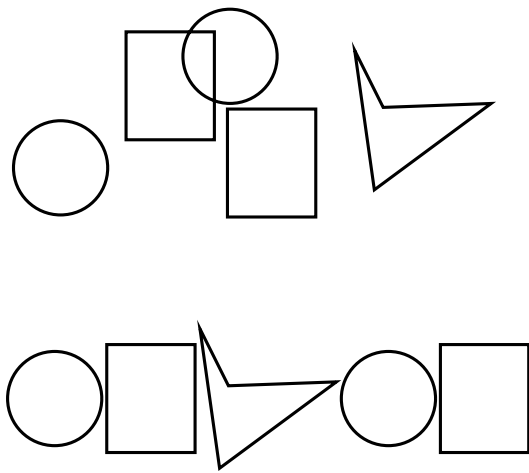


Figure 1. Before/after alignment

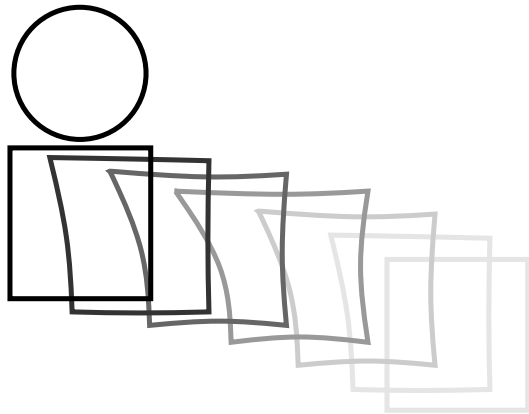


Figure 2. Animating alignment

haviour of a manipulated object as a cartooner might: an object dragged by its corner distorts slightly, as if it has inertia. In the case of an indirect manipulation causing an object to move (perhaps to align it to another object), we would portray the change as if the object is dragged by one of its corners.

1.2 The problem

This research addresses two problems that often occur in indirect manipulation: an abrupt visual change can be confusing (and sometimes ambiguous), and the final result of a complex change can be difficult to predict.

The visual changes caused by direct manipulation are easy for users to understand because the user focuses directly on the affected object and tracks the change as it occurs. But for changes caused by indirect manipulation, where the user's attention is focussed elsewhere and the change is instantaneous, the effect can be confusing.

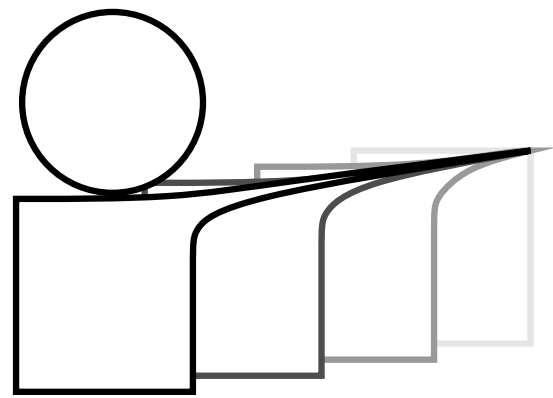


Figure 3. Previewing alignment

Sudden visual changes can be disruptive because the user must subconsciously digest the new state of the screen; it takes a brief but significant moment before the new state is assimilated and the user can proceed with another task. Moreover, if the initial and final state of the screen are similar (perhaps an operation rotated a square through a quarter turn or flipped an object about an axis of symmetry), it may not be obvious just what has changed or how the change came about. To understand the changes, the user has to remember the initial state of the screen. The user's memory may be taxed by this, and this extra load on their memory may increase the user's cognitive load for the task.

For example, Figure 1 shows an operation where objects have been rearranged so that they abut left-to-right. By inspection, it is not obvious which objects in the initial configuration (top diagram) correspond to the objects in the final configuration (bottom diagram). If the screen instantaneously changes from one configuration to the other, the user may well be confused.

1.3 Human memory

We are presenting in this paper user interface techniques to improve a user's ability to remember the initial state of an indirect manipulation operation and understand final state of that operation. An overview of key factors affecting a user's ability to remember this form of information is presented here.

Memory can be considered to have two components: working memory and long-term memory [16]. Working memory is the temporary store used to retain new information. This is the memory that is used to calculate mental arithmetic. It is used to hold information until it can be encoded into long-term memory. Information stored in working memory is often encoded in either spatial or verbal forms, and is usually associated with a semantic representation [10] in that awareness of meaning is associated with

appearance. Wickens states that the two modes can operate in parallel and either compete or cooperate for attention.

The decay of visual information is faster than for verbal. Thus visual cues are forgotten more easily than verbal cues. This can be reduced by a longer-lasting visual cue, but at the expense of visual clutter. Wickens lists six factors that influence the probability of forgetting in working memory: time, attention, capacity demands, chunking, interference, and similarity confusion.

Working memory is limited in capacity, and this limit interacts with time. Faster decay is observed when more items are held in working memory. Miller [8] identified the limit of memory span as “the magical number seven plus or minus two.”

An *item* in working memory can be an image, a number or a word made up of a number of letters. The word is a *chunk* of information, defined by Miller [8] as a set of adjacent stimulus units that are closely tied together by associations in the subject’s long-term memory. Thus words such as cat, dog etc. are tied together in long-term memory as chunks of information rather than individual letters.

Card, Moran and Newell [2] showed that memory decay depended on the number of chunks. As an example three chunks might be stored for only seven seconds, while one chunk could be stored for 70 seconds.

The organisation of information by chunking can be exploited to aid recall. An association in the stimuli must be learnt. The new set of higher-order information is stored as one chunk. On recall, only the sequence of new chunks must be remembered. When each chunk is activated in turn, its contents are *unpacked* and recalled in correct order.

In addition to forgetting due to the passage of time, material is also lost due to interference and confusion. Other information may be encoded that *uses-up* storage capacity. Similar stored stimuli may cause confusion. Thus if colour is used to code, two shades of a colour may cause confusion, compared to using two distinct colour codes.

1.4 Structure of the paper

This paper starts with a description of our original visual cues for indirect manipulation, and we proceed to present the three new visual cues investigated in this paper. We then define the experimental design used in the investigation. The results and discussion of this investigation are then described. Finally, a set of concluding remarks and future directions are presented.

2 Our Original Visual Cues

Our previous investigation [12] into the effectiveness of applying animation to direct manipulation interfaces

showed that computer-literate users like the idea of cartoon-style animation in direct manipulation drawing editor interfaces. In particular, they felt that animation improved the visual feedback for constrained operations such as movement under the influence of “gravity”, and they rated the animation enjoyable and intuitive to use for a range of editing tasks. Moreover, when given a choice, such users chose to set the magnitude of the animation effects to be clearly evident.

For indirectly initiated changes, we demonstrated firstly, that animation provides visual continuity and hence reduces confusion and ambiguity, secondly, that cartooning techniques can help avoid surprises for complex operations by suggesting the final outcome of the change before the operation is committed [15]. To demonstrate these effect, we added animations to indirect manipulation in a prototype drawing editor.

2.1 Animating changes

Animation can help smooth abrupt changes in appearance by more gradually changing the screen’s appearance from the old state to the new. These forms of smoothing animations have been investigated by a number of researchers. Whizz [5, 6] provides tools for building animated interactive applications. The Self animated widgets [4] provide menus that grow, dialog boxes that dissolve, arrows that grow and shrink smoothly, and objects that can be “hit” and bounce about from moving arrows. In the “buttonfly” 3D graphical menu system for Silicon Graphics workstations [11], activating a submenu button brings up a new menu by smoothly flipping the button over, exposing a new set of buttons.

Other workers have shown [1] that avoiding abrupt changes by smoothly changing the visual appearance of the screen over a short time interval improves understanding of the operation and can even reduce the total time needed to carry out a given task. For example, the Cone Tree and Perspective Wall visualisations (part of the Information Visualiser project [3]) use smooth transitions to display changing views of complex data; objects slide or rotate into new configurations and views. The improvement occurs largely because a smooth change can be tracked by the user’s perceptual system, whereas an abrupt change must be processed as a cognitive task.

2.1.1 “Direct” indirect manipulation

Our approach to smoothing a change caused by indirect manipulation is to consider what the operation would look like if it was carried out (albeit less quickly and precisely) by direct manipulation. For example, if an indirect manipulation causes an object to move (perhaps to align it to another

object), then our editor portrays the change as if the object is dragged by one of its corners. As with direct manipulation, we use cartoon-style animation techniques to suggest the effect of the interaction on the changing object; the object is distorted to suggest that it is slightly rubbery and that it is somewhat reluctant to change.

Figure 2 shows a situation where the square is moving to the left so that it is aligned with the circle. Of course, the static figure cannot fully convey the dynamic feel of the interaction. We have superimposed several successive frames from the sequence to suggest the effect (the shading of frames is intended to imply the passage of time). In use, the distortion of the manipulated object gives the impression that the object is being towed along by its corner; without the distortion the impression is more like the object is moving spontaneously. The net result is that the cartooning animation helps convey the impression that the action, although indirect, is still under the control of the user (or perhaps a willing agent who will do the user's bidding).

2.2 Previewing consequences

Some indirect manipulation operations cause complex changes whose effects are difficult to predict. For example, aligning a group of objects to a common edge can cause many objects to move, all by differing amounts. Furthermore, many editors offer a wide range of alignment options (by centres, by sides, equally spaced between endpoints, or abutted) in either or both the horizontal or vertical dimension. Often when the user performs the operation, the final configuration of the objects is not quite what was intended; the user might make several attempts before achieving the desired effect.

One way to deal with such operations is to adopt a trial-and-error approach, with a heavy reliance on an 'undo' facility; if the final result is incorrect, it can be undone and a new attempt made. At best, such a working style is cumbersome.

2.2.1 Trying out operations

We use cartooning techniques to illustrate the consequences of an operation before it is committed. In our prototype editor, indirect manipulation operations are initiated by buttons that use a two-stage action. Pressing and holding a button ('arming' the button) carries out the operation (using animation, of course) but does not commit the change. Moving the mouse pointer off the button ('disarming') cancels the change and restores the objects to their original positions. The operation is only committed if the mouse pointer is still on the button when it is released ('activating').

The technique is a natural extension to the usual workings of a button. A user who is confident of the effects of

an operation can simply click the button in the usual way. However, where the consequences are less certain, the user can press and hold the button, observe the result, then either commit or cancel the operation as appropriate.

2.2.2 Suggesting the options

Our editor exploits one further technique to animate indirect manipulations involving changes to an object's geometry. While an operation is being considered (but before it is committed) we wanted to show both what would happen if the operation was committed and what would happen if it were cancelled.

Our solution uses an extension of the cartoon technique we have previously used to show the effect of constraints on objects. For example, where an attempt is made to move an object that is pinned in place, a cartooner might show it distorted in the direction of pull, as if it is made of a rubbery material. We use a similar technique to suggest that an operation has not yet been committed—we show the object in the position it will adopt if the change is committed, but with a corner stretched back to its original position, as if it is attached by an elastic *telltale*. The effect suggests both possible outcomes to the trial operation: If the change is committed, the telltale will retract into the object, showing that the new position has been accepted. But if the change is cancelled, the object will be released and the telltale will drag it back to its original position.

Figure 3 illustrates the idea in the context of an alignment operation; again, the square is moving to the left to align with the circle. The figure shows the situation at the point where the animation for the trial operation has been completed, but before it has been committed. The position of the square clearly shows what the final alignment would look like, but its top right corner is stretched back to its original position to show that the object has not yet released its hold on that place. If the operation is committed, the stretched telltale will retract, leaving the square correctly aligned. But if the operation is cancelled, the square will spring back to its original position.

3 Our New Visual Cues

When performing an alignment task, the original visual cue of using a telltale and smooth animation provides the user with a clear indication of the current and previous positions of the graphical objects. In pilot testing, subjects were able to discern the previous position of aligned objects with close to an 100% accuracy. This is not a surprising fact, given the telltale points directly back to the previous position. As an example of a typical alignment task, Figure 4(a) shows a set of graphical objects before an alignment operation, and Figure 4(b) shows the completion of the align-

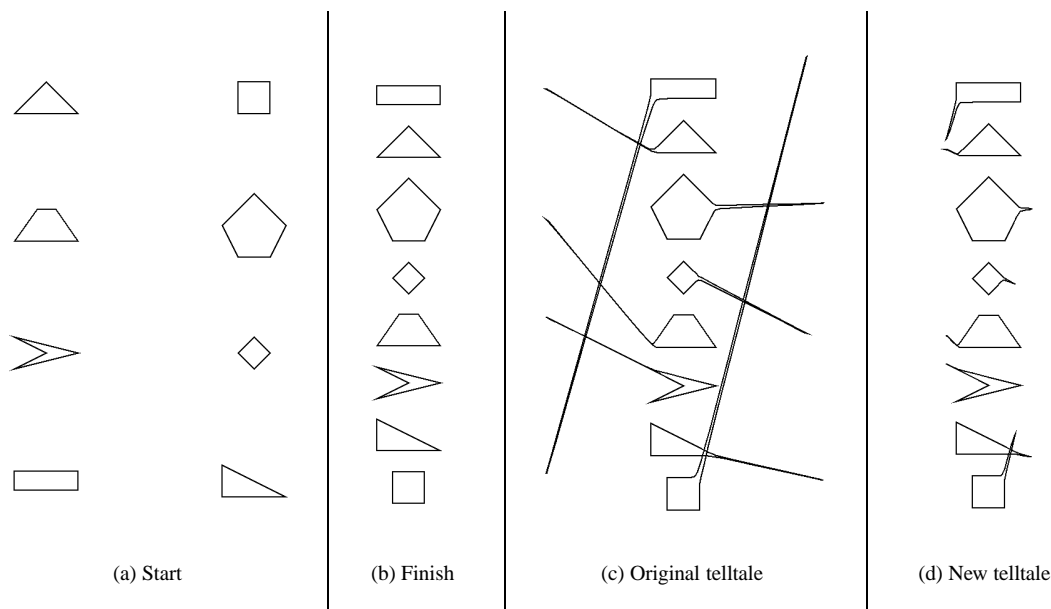


Figure 4. Example of the starting and finishing configurations for an alignment operation

ment. The original telltale cues for the same alignment operation are shown in Figure 4(c). These telltales dominate the visual features of the drawing.

In this paper, we investigated a set of three more subtle visual cues to highlight the previous positions of aligned graphical objects. These new and novel visual effects for the user are as follows:

- **new telltale:** Objects are animated as if one of the corners is being dragged by a mouse while the remainder of the object lags behind, and a telltale is stretched back 10% towards its original position.
- **colour:** Colour is used to cue the original position of the graphical object, and the object is snapped to its new location. There is a unique colour for each of the original positions.
- **wiggle:** A unique small cycling animation, or *wiggle*, is used to cue the original position of the graphical object. There is a different wiggle effect for each of the original positions.

The first new visual cue is a more subtle telltale effect, as shown in Figure 4(d). The telltale only stretches back 10% towards its original position. When compared to the original telltale visual cue depicted in Figure 4(c), the new telltale effect does not obscure other graphical objects and the final positions are still clear to the user. The telltale methods cue the user by explicitly referring to its previous position.

A second method is to uniquely code the original position of each graphical object. We investigated two coding strategies, colour and canned animations.

For the colour effect, the graphical objects are drawn on a white background. We coded the eight positions depicted in Figure 4(a) with the eight colours shown in Figure 5, all of which had a large separation in the RGB colour space. The choice of colour is an area still to be investigated.

Canned animations or *wiggles* are applied as a second coding strategy. Figure 6 shows the eight animations used to uniquely code the original positions. The rotate wiggle changes the orientation of the object by 5° . The move wiggle translates the object by one of the following: 5 points left to right, 5 points top to bottom, or a 5 point diameter circle motion. (One printer's point or *point* is a $\frac{1}{72}$ of an inch.) The scale wiggle changes the size of the object by one of the following: making the width larger and smaller by 40%, making the height larger and smaller by 40%, or making the width and height larger and smaller by 40%. The bottom right graphical object was kept still as a contrast animation effect. Once again the choice of canned animation is an area still to be investigated.

4 Experiment

To test the effectiveness of the different visual cues, we modified our editor to use one of the four feedback techniques for alignment operations, new telltale, colour, or wiggle feedback, or no feedback. During the no feedback effect, objects align with no visible feedback at all; they snap suddenly to their aligned positions.

The experiment measures the subject's ability to remember the placement of graphical objects on a computer screen, in a simulated alignment operation. The subject is

Blue	Green
Red	Black
Olive Green	Aqua Blue
Purple	Magenta

Figure 5. Colour effect

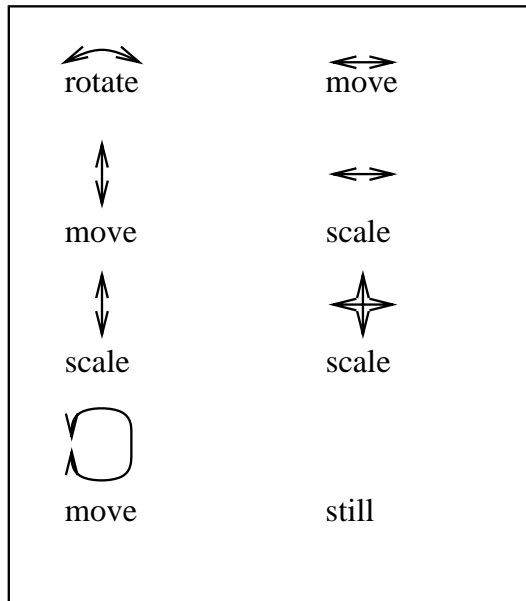


Figure 6. Wiggle animation effect

shown a screen of four to eight graphical objects. After a predefined time period (five seconds), the objects are automatically moved to a new location. The objects are moved with one of the four different animation effects. The subject then draws the graphical objects on the top sheet in front of them. The subject then repeats these procedures. We measured the users' performances by comparing the drawn graphical objects with their original positions, and we measured the drawing time for each task. We also asked users to complete an exit survey on their attitude to the various forms of feedback.

The experiment tested the following hypotheses:

1. Animation and colour cues will help subjects remember the original position of objects before an alignment operation compared with feedback that does not use such cues.
2. Users will prefer performing the task with visual cues

over performing the task without visual cues.

4.1 Experimental design

The experimental task was to remember the positions of the objects from the configuration show in Figure 4(a) and draw them after they had aligned to the positions shown in Figure 4(b).

Subjects were tested individually, working on a Silicon Graphics Indy workstation in a quiet isolated office. The experiment supervisor read from a script to explain the requirements of the experiment and gave the following instructions:

1. The subjects are shown a screen of graphical objects.
2. They look at the objects, and try to remember where they are placed.
3. After a five second time period, the objects are moved to a new location.
4. The objects' movement is performed with one of four different animation effects, no effect, new telltale, wiggle, or colour.
5. The subjects then draw the graphical objects on paper in their original position.
6. The task is completed by pressing the "Quit" button.

The subjects were asked to read an information sheet and a set of instructions, then asked if they had any questions. Then they were asked to sign a consent form.

To familiarise themselves with the editor and the feedback effects, each subject performed eight trials of the task, a four object and an eight object task with each feedback effect. Then each subject performed four sets of trials, each consisting of ten repetitions of the task for one of the four feedback effects. To minimise ordering artifacts, the feedback effects were presented in a random order for each subject. The graphical objects were also presented randomly in one of eight regions of the application's window. They were placed starting from the upper left corner, and the order of filling was left to right and top to bottom. The experiment took about 40 minutes for each subject to complete.

The exit survey asked the subjects about their impressions of the animation effects and the task. They recorded their responses on a 7 point scale, with 1 corresponding to negative reactions ("terrible", "frustrating", "dull", "difficult") and 7 corresponding to positive reactions ("wonderful", "satisfying", "stimulating", "easy").

The experiment was completed by nine subjects (mean age 28, age range 20 - 40 years). On average, subjects had eight years computing experience with mouse-based interfaces. Subjects were self-selected (on a first-come basis) from a pool of computer-literate staff and students drawn from the School of Computer and Information Science at the University of South Australia.

Number of items	Percent correct
4	97
5	88
6	85
7	76
8	73

Table 1. Number of memory items vs. percent correct

Effect Type	Percent correct
no effect	62
wiggle	82
new telltale	96
colour	96

Table 2. Effect types versus percent correct

5 Results and Discussion

The experiment was broken down into two major components, objective and subjective. The objective component measured the subjects ability to remember the correct original position of a set of graphical objects after they had been moved in an alignment style indirect manipulation operation. These results are presented first. The subjective component measured the subjects likes and dislikes of the different visual effects, and those results are presented second.

5.1 Empirical results

The analysis of the experimental data has focussed on the percent of correct positions per total possible correct positions for each of the 40 tasks. Results were analysed using one-way ANOVA's using the SYSTAT software package. The analysis showed there was no significant difference between subjects across all tasks ($F(8, 351) = 0.5, p = 0.84$), and there was no significant learning effect across the tasks ($F(39, 320) = 1.3, p = 0.125$). As was expected, there was a significant effect for the number of memory items ($F(4, 355) = 10.4, p < 0.001$) [2, 8]. As previously mentioned, the tasks ranged from remembering the position of four to eight graphical objects. Table 1 shows the number of memory items versus percent of correct answers across the four visual effects.

Table 2 shows the visual effect versus percent of correct answers as measured as a mean across the different number of memory items. There was a significant effect for the effect type ($F(3, 356) = 51, p < 0.001$), except between the colour and new telltale cues. Table 3 shows the significance values when the percent of correct answers is compared between each of the effect types. This shows the new telltale

Between effect type	F value	p value
no effect – new telltale	111.7	< 0.001
no effect – wiggle	39.0	< 0.001
no effect – colour	115.7	< 0.001
new telltale – wiggle	18.7	< 0.001
new telltale – colour	0.024	= 0.887
wiggle – colour	20.2	< 0.001

Table 3. Significance values between two effect types

Effect type	Mean score	SD
colour	5.8	0.8
new telltale	4.8	0.5
wiggle	3.8	1.4
no effect	2.6	0.5

Table 4. Effect types versus mean attitude score

and colour visual effects are superior to the wiggle visual effect and no visual effect, and that when one of the three visual effects was present, the subject's ability to remember the previous position increased.

5.2 Survey results

Table 4 shows the effect type versus the mean of the attitude score for each of the visual effects. When asked "Overall reaction to the task of remembering the objects' positions", subjects rated the task with a neutral response ($mean = 4.1, SD = 0.5$). Subjects significantly rated ($F(1, 38) = 14.4, p < 0.001$) the task while using the colour visual effect more positively ($mean = 5.8, SD = 0.8$) than the overall experiment task of remembering positions. This indicates the subject's preferred visual effect was the colour coding.

The survey showed that subjects significantly rated the colour ($F(1, 38) = 54.5, p < 0.001$), new telltale ($F(1, 38) = 27.8, p < 0.001$), and wiggle ($F(1, 38) = 8.3, p < 0.001$) visual effects more positively than the no feedback task. This indicates users would prefer to perform these tasks with some form of visual feedback.

Subjects had a negative response ($mean = 2.6, SD = 0.5$) to the task with no visual cues and significantly rated ($F(1, 38) = 12.1, p < 0.001$) it more negatively than the overall experiment task of remembering positions. We believe this shows that users, once shown visual feedback for this form of task, find having no visual feedback impedes their effectiveness.

6 Conclusions

The major conclusion drawn from this experiment is the use of visual feedback allows users to remember the previous position of graphical objects after an alignment operation. The results may be applied to solve the problem of: “While an operation is being considered (but before it is committed) we wanted to show both what would happen if the operation is committed and what would happen if it were cancelled.”

Two strategies were investigated in this paper. One strategy was to visually show the previous position of the graphical through the use of telltales. The second strategy was to encode the initial position of the graphical objects with a unique visual effect, colour or canned animation. As previously mentioned, we wished to improved on the previous telltale method by making the visual cues smaller and less obtrusive. Both the colour coding and new telltale visual cues proved to be equally effective. The canned animations, or wiggles, proved to be more effective than no visual feedback, but it was shown to be less effective than the colour and telltale effects.

A number of significant results were observed. These are as follows

1. The subject’s ability to remember the previous position was increased with the use of one of the three visual effects.
2. The new telltale and colour visual effects are superior to the wiggle and no visual effects.
3. There was a significant effect for the number of memory items the subject was required to remember.
4. The subject’s preferred visual feedback was the colour coding.
5. Users would prefer to perform these memory tasks with some form of visual feedback.
6. Once shown visual feedback for this form of task, users find having no visual feedback impedes their effectiveness.

In the future, we wish to look further into incorporating coding strategies (as in the colour visual effect) with the positional strategies (as in the telltale visual effects) into a more powerful cueing mechanism. We will investigate different wiggle effects; one idea is to try and use the canned animations as a positional strategy. We wish to further investigate the effect of using animation to smooth the state changing. We believe this helps reduce the cognitive load for the users.

References

- [1] R. Baecker and I. Small. Animation at the interface. In B. Laurel, editor, *The Art of Human-Computer Interface Design*. Addison-Wesley Publishing Co., Reading, MA, 1990.
- [2] S. K. Card, T. P. Moran, and A. Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, 1983.
- [3] S. K. Card, G. G. Robertson, and J. D. Mackinlay. The information visualizer, an information workspace. In *Proceedings of ACM CHI’91 Conference on Human Factors in Computing Systems*, Information Visualization, pages 181–188. 1991.
- [4] B.-W. Chang and D. Ungar. Animation: From cartoons to the user interface. In *Proceedings of the ACM SIGGRAPH Symposium on User Interface Software and Technology*, Animation/Visualization, pages 45–55. 1993.
- [5] S. Chatty. Defining the dynamic behaviour of animated interfaces. In *Engineering for Human-Computer Interaction. IFIP TC2/WG2.7 Working Conference*, volume A-18, pages 95–111, Ellivuori, Finland, 10-14 Aug. 1992.
- [6] S. Chatty and M. Beaudouin-Lafon. Integrating animation with interfaces. In *Proceedings of ACM CHI’92 Conference on Human Factors in Computing Systems – Posters and Short Talks*, Posters: Designing the Ins & Outs, page 70. 1992.
- [7] B. Laurel. *Computers as Theatre*. Addison-Wesley Publishing Co., Reading, MA, 1991.
- [8] G. Miller. The magical number seven plus or minus two. *Psychological Review*, 63:81–97, 1956.
- [9] B. Shneiderman. Direct manipulation: A step beyond programming languages. *IEEE Computer*, pages 57–69, Aug. 1983.
- [10] H. Shulman. Semantic confusion errors in short-term memory. *Journal of Verbal Learning and Verbal Behavior*, 11:221–227, 1972.
- [11] T. Tessman and G. Bell. *Butterfly*. Silicon Graphics Inc.
- [12] B. Thomas, P. Calder, and V. Demczuk. Experiments with animating direct manipulation in a drawing editor. In *ACSC’98 - The 21st Australasian Computer Science Conference*, page to appear, Perth, Australia, Feb. 1998.
- [13] B. H. Thomas and P. R. Calder. Animating direct manipulation interfaces. In *Proceedings of UIST’95: ACM Symposium on User Interface Software and Technology*, pages 3–12, Pittsburgh, Nov. 1995. ACM SIGGRAPH and ACM SIGCHI.
- [14] B. H. Thomas and P. R. Calder. Animating widgets in the interviews toolkit. In *Proc. The 1995 East-West International Conference on Human-Computer Interaction*,., pages 26–44, Moscow, Russia, Jul 1995.
- [15] B. H. Thomas and P. R. Calder. Animating indirect manipulation in direct-manipulation editors. In *The Sixth Australian Conference on Computer-Human Interaction*, pages 184–188, Hamilton, New Zealand, Nov. 1996. Ergonomics Society of Australia CHISIG.
- [16] C. Wickens. *Engineering Psychology and Human Performance*. Harper Collins, 1992.