The Effects of Dynamic Transparency on Targeting Performance

Carl Gutwin, Jeff Dyck, and Chris Fedak

Department of Computer Science, University of Saskatchewan
57 Campus Drive, Saskatoon, SK, S7N 5A9
carl.gutwin@usask.ca, jeff.dyck@usask.ca, chris.fedak@usask.ca

Abstract

Transparency can be used to increase the visibility of a user's workspace in situations where the space is obscured by floating windows and tool palettes. Dynamic transparency takes this approach further by making components more transparent when the user's cursor is far away. However, dynamic transparency may make palettes and floating windows more difficult to target. We carried out a study to test the effects of different types of dynamic transparency on targeting performance. We found that although targeting time does increase as targets become more transparent, the increases are small – often less than ten percent. Our study suggests reasonable maximum, minimum, and default transparency levels for designers of dynamic transparency schemes.

Key words: interface transparency, dynamic transparency, tool palettes, visual workspaces..

1 Introduction

Transparency has been proposed as a way to reduce the occlusion that is caused by floating windows such as toolbars and palettes (e.g. [2,3,5]). In many visual applications there can be several of these windows, each of which prevents the user from seeing some part of the underlying data (Figure 1). Adding transparency allows users to see the underlying data in all parts of the screen; however, in order to ensure that users can still see and distinguish items on the palettes, transparency levels for floating windows cannot be set very high.

To further improve the visibility of underlying images, some recent systems make use of multiple transparency levels for floating interface components. This technique adjusts transparency based on the position of the user's cursor on the screen, making use of the fact that not all palettes are in use at the same time. We call this technique *dynamic transparency*. For example, the online game EverQuest [9] (Figure 2) allows users to specify two levels of transparency for floating windows: a higher (more transparent) level that is used when the user's mouse cursor is outside the border of the palette, and a lower level (more opaque) used when the cursor is on the component.

Figure 1. Occlusion of a background image by tool palettes and floating windows in Photoshop.

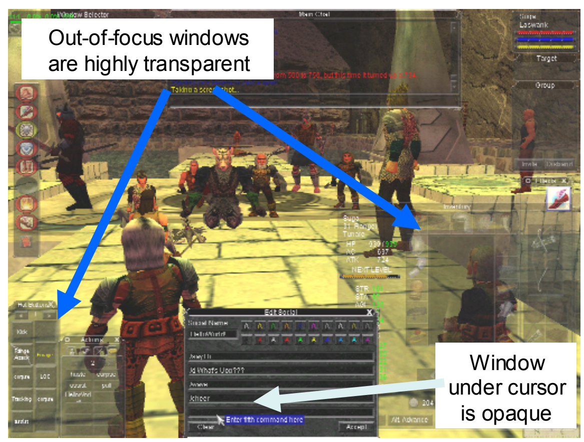


Figure 2. EverQuest interface showing multi-level transparency for floating windows.

Dynamic transparency thus allows for greater visibility of the underlying data when the user is not using the control, and greater visibility of the control when it is being used. However, increasing the transparency of floating windows may cause other problems for users, since the visibility of a palette is important for more than just the work that goes on inside the floating window. Visibility is also important for targeting the window and the tools inside it – that is, for seeing where the palette is on the screen, and for visual feedback as the user moves their cursor towards a particular tool on the component. If a more-transparent palette is harder to see, it may also be harder to select things on it quickly and accurately. Dynamic transparency was invented to improve performance in tasks that involve the underlying background image, but it is unclear what the resulting costs are to targeting performance.

To investigate these costs, we carried out a study of how different forms of dynamic transparency affect targeting speed and accuracy. We examined three factors in particular: the maximum alpha-channel transparency level that a floating window uses (i.e. the level used when the cursor is far away), the type of transition used to change transparency level as the cursor gets nearer, and the visual complexity in the background image underlying the palettes.

The main findings from the study suggest that dynamic transparency is a viable technique, even though there are definite costs to targeting. First, targeting time does increase with additional transparency, although these increases are relatively minor at transparency levels below 75% (where 100% transparent implies invisibility). Second, gradual transitions that increase visibility earlier in a targeting motion allow faster targeting, but again the differences are seen primarily at high transparency levels. Third, complex background images require a slightly lower transparency level in order to maintain good targeting performance, but they do not preclude the use of multiple transparency levels.

Our results suggest that although transparency levels and transition mechanisms must be carefully chosen, the idea of dynamic transparency is viable from a targeting perspective. In this paper we first briefly review the issues surrounding the use of transparency in user interfaces, and then report on the dynamic transparency study and its implications for the design of interactive systems.

2 Occlusion and Transparency in Interfaces

Several types of software applications – such as graphics editing, page layout, video editing, and games – have two clear characteristics: first, users must view and manipulate data in large 2D visual workspaces; and second, users have many controls and tools that can be selected for working with various parts of the data. Often, there is not enough room on the screen to display both the data and the controls without overlapping: as Harrison et al. state, "the small amount of display real estate available relative to the amount of data to be displayed presents a real challenge to user interface design." ([4], p. 317).

A standard approach in many systems has been to provide tools and controls as floating windows that sit above the background data. As can be seen in Figure 1, this approach presents problems since the floating windows occlude different parts of the image, and users are forced to move windows or scroll the workspace to bring certain parts of the data into view.

Transparency has been used in a variety of human-interface settings (e.g. [2,6,7]), and has also been suggested as a solution to the problem of occlusion (e.g. [1,4,5]). By making palettes and toolbars partially transparent, users can see both the workspace data and the tools floating above. Transparency changes the situation from one where the image and the controls are time-multiplexed to one where they are depth-multiplexed [4]. This approach allows users to divide their attention between the tasks that occur at two depths: at the level of the tool palettes, where users must choose and select individual tools, and at the workspace depth, where users must maintain awareness of the entire scene.

One main design issue to be considered in such interfaces is the level of transparency to use for the floating windows, since it is important to minimize interference between background and foreground layers [3,8]. Studies of simple word-naming and colour-naming tasks [4] found considerable interference at transparency levels from completely invisible (i.e. 100%) to about 75% transparent. The closest research to our work, however, is Harrison's [5] study of transparent tool palettes, in which participants were asked to select tools from palettes with different (but static) transparency levels, over a number of different image and wire-frame backgrounds. The study found reasonably stable performance up to 75% transparency, but that 90% transparency was unusable for realistic work.

These studies suggest static transparency levels that are appropriate for both targeting a palette and working inside the component. In our experiment, we extend this work to consider the situation of multiple transparency levels where the maximum transparency is intended to be considerably higher than a static value would be. In the following sections, we describe a targeting experiment that we carried out to investigate how different factors in the design of a dynamic transparency scheme would affect targeting time, errors, and user preference.

3 Study Methods

In the following sections we provide details about the study participants, the apparatus and tasks used, the experimental factors, and the study design.

3.1 Participants

Ten people (7 male, 3 female) were recruited from the computer science department of a local university, and were given course credit for participating in the study. All participants were frequent users of mouse-and-windows based systems (at least 20 hours per week). Most of the participants had at least some experience with transparency in user interfaces, either in computer games (e.g. Unreal Tournament, EverQuest) or in applications (e.g. Photoshop, WinXP).

3.2 Apparatus and targeting tasks

The experiment was conducted on a PII Windows XP PC running a custom-built Java application. The display was a 21" monitor set to 1280x1024 resolution. Three targeting tasks were used: a one-dimensional selection task (Figure 3), a multidirectional point-select task described in ISO 9241-9 [10] (Figure 4), and a palette selection task (Figure 5).

- *One-dimensional selection*. In this task (see Figure 3), two rectangular targets are placed at the left and right sides of the screen. The next target is always marked with a purple cross. Participants click on alternate targets in succession for a total of 24 selections. This task was chosen to represent situations where target locations are very well known to the users.
- *Multidimensional point-select*. In this task (see Figure 4), 24 circular targets are arranged in a ring, and the participant clicks on each target in succession, where the next target is always directly across the ring. The next target to be clicked was marked with a purple cross. In this task, target locations are also fairly well known, but for each target there are several others in the neighborhood which can act as distractors.
- *Palette object selection*. In this task (see Figure 5, four tool palettes were placed at the four corners of the screen. Each palette contained six targets of different types (capital letters, small letters, numbers, and symbols). Participants clicked a central circle to see the next target, and then found and selected that target from the palettes. Participants repeated this process until all 24 targets were selected. This task involves targeting both the palettes and then the specific items, and also requires remembering which palette contains which items.

3.3 Transparency Levels

Participants carried out the targeting tasks at four different alpha-channel transparency levels (Figure 6): 100% (invisible), 90%, 75%, and 0% (opaque). These levels were chosen due to their use in previous transparency studies of transparent interfaces (e.g. [5]). Note that the 0% condition is equivalent to the traditional situation of static opaque floating windows; this condition was used as a baseline.

3.4 Background visual complexity

The motivation behind dynamic transparency is to improve visibility of the data behind the palettes and floating windows. However, this data presents a visual background of varying complexity for the targeting task. To assess the effects of background complexity on transparent targets, we used two types of background in the study: a blank background (e.g. Figures 3-5), and a complex visual background shown in Figure 7.

3.5 Transition mechanisms

Dynamic transparency requires a mechanism for changing between the different levels of transparency set up for a floating window. Our study compared two mechanism, one sharp and one gradual.

- *Sharp transition*. This mechanism used a simple threshold at 50 pixels from the target border (see Figure 8). Outside the threshold, the maximum transparency level was used; inside the threshold, the target was opaque. This mechanism is similar to that used in EverQuest.
- *Gradual transition*. This mechanism took effect when the cursor was between 300 and 50 pixels from the target border. Transparency varied linearly from the maximum level to opaque.

3.6 Procedure

Participants went through three separate sessions for the three different tasks (one-dimensional, multidirection, and palette targets), and for each session, there was a similar procedure.

First, participants were randomly assigned to one of four orders. Participants were then introduced to the experiment and to the dynamic transparent targets, and were then given a series of practice trials with all conditions. Participants then completed 24 targeting tasks in each of the 16 study conditions. Participants were instructed to click on the targets as quickly and as accurately as possible. Rests were allowed between conditions and between sessions. After all conditions for a session were complete, participants were asked two sets of questions: first, which level of transparency they would accept as the maximum for both the blank and complex backgrounds, and second, which transition type they felt was fastest, which they felt was most accurate, and which they preferred overall.

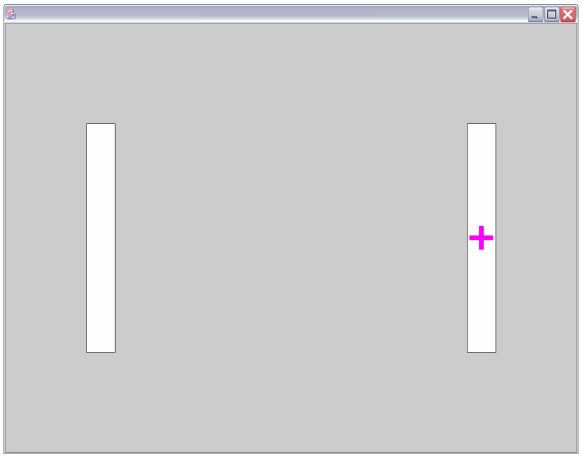


Figure 3: One-dimensional targeting task setup with opaque targets (purple cross indicates next target).

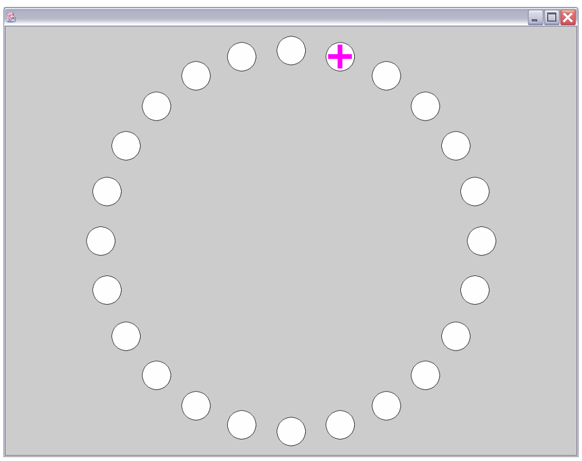


Figure 4: Multidirectional point-select task with opaque targets (purple cross indicates next target).

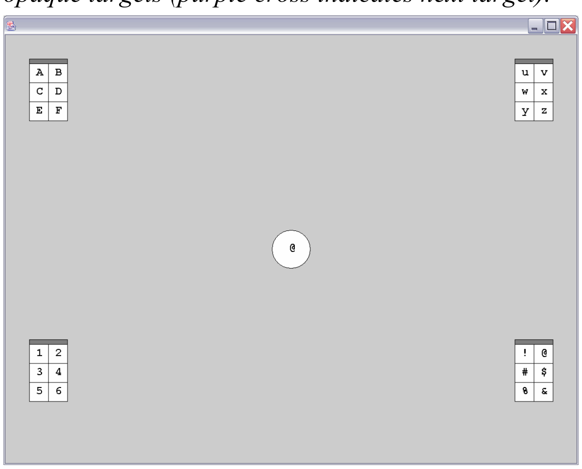


Figure 5: Palette object selection task with opaque targets. The circle at centre showed the next target that the user was to find and select (here the '@' in the lower right palette).

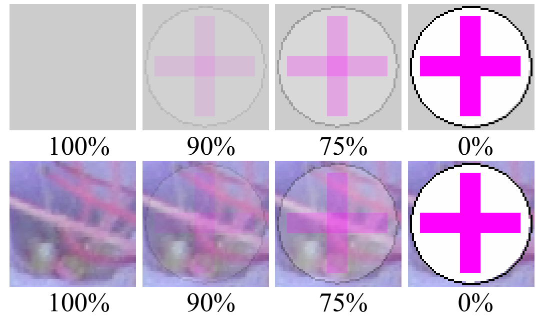


Figure 6. Transparency levels used in the study, shown on blank (above) and complex (below) backgrounds. (Images must be viewed in colour to see transparency levels accurately).

Figure 7. Complex background with multidirectional selection task at 75% maximum transparency (the mouse cursor is over the current target).

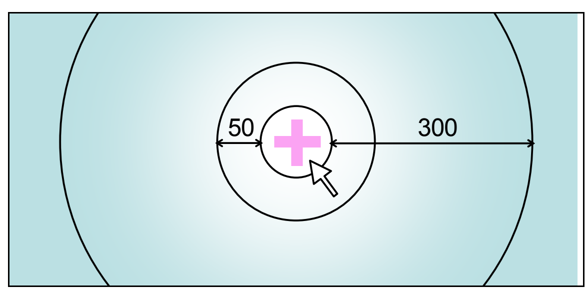


Figure 8. Distances from central target at which transition mechanisms take effect. Gradual begins at 300 pixels; sharp begins at 50 pixels.

3.7 Study design

The study used a 3 x 4 x 2 x 2 within-participants factorial design. Order was balanced using a Latin square method: each level of each factor occurred in every position in the sequence equally (e.g. all levels were first in the sequence an equal number of times). The factors were:

- Task type: one-dimensional, multi-direction, and palette targets. (Note that since differences between tasks are expected, this factor was not considered in the analysis);
- Maximum transparency level (i.e. the level when the cursor is far from the target): 100% (invisible), 90%, 75%, and 0% (opaque);
- Transition mechanism: gradual or sharp;
- Background visual complexity: blank or complex.

With 10 participants and 24 targeting tasks per condition, there were 11520 tasks recorded in total. The study system collected completion times and error information for each target. In addition, answers to summary questions were recorded on a questionnaire.

4 Results

We first discuss main effects of our three study factors, and then consider interactions between the factors, and results of the preference survey. Charts of primary results are shown in Figures 9 and 10.

4.1 Main effects of transparency, transition type, and background complexity

For completion time data, significant main effects were found for almost all of the main factors tested in the study, for each task type (see Table 1). The one exception was the effect of transition type in the one-dimensional selection task. For error data, no effects were found, and since error rates were low (less than one error per 24 trials on average) we will not consider errors further here.

The main effects on completion time imply that:

- increasing transparency negatively affects targeting speed;
- complex backgrounds reduce performance;
- a gradual transition from transparent to opaque allows faster targeting than a sharp transition, except in the case where target location is very well known.

The main effects of transparency and transition type were weaker in the 1D selection task, suggesting that when target locations are very well known, people may require less visual feedback about the target – although complex background images do appear to compromise this memory-based strategy.

The findings in general are in line with our overall expectations; however, there were several significant interactions that shed more light on how performance is affected differently by different combinations of transparency, transition, and background.

One-dimensional selection			
Factor	*df*	*F*	*p*
transparency level	(3,27)	4.48	<0.05
transition type	(1,9)	1.43	=0.261
background complexity	(1,9)	55.36	<0.001
Multi-directional selection			
Factor	*df*	*F*	*p*
transparency level	(3,27)	33.0	<0.001
transition type	(1,9)	66.92	<0.001
background complexity	(1,9)	50.08	<0.001
Palette-target selection			
Factor	*df*	*F*	*p*
transparency level	(3,27)	52.35	<0.001
transition type	(1,9)	17.09	<0.005
background complexity	(1,9)	39.79	<0.001

Table 1. ANOVA results for main effects of transparency level, transition type, and background complexity for all task types.

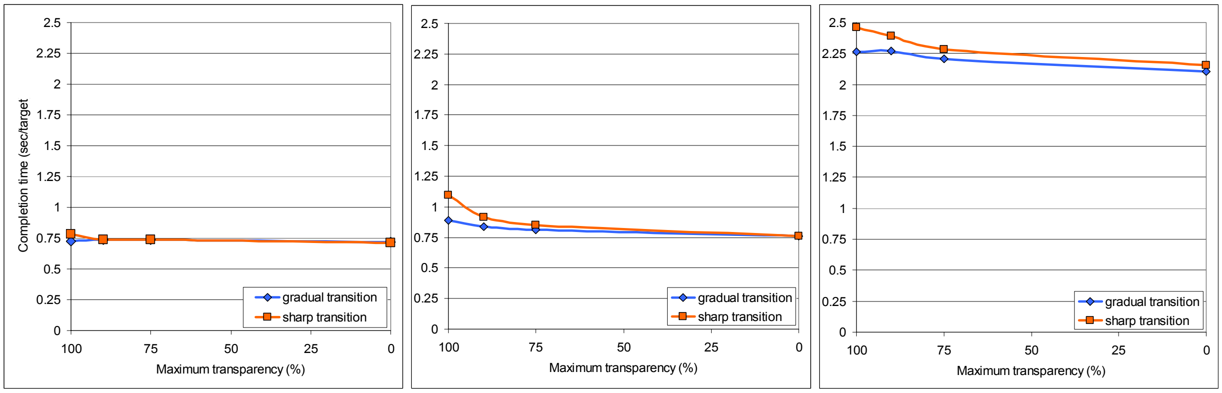
4.2 Interactions with transparency level

We were particularly interested in interactions between transparency and the other two factors, since we are interested in finding usable transparency levels and appropriate mechanisms for use with dynamic transparency schemes.

First, there was a clear interaction between transition type and transparency level (Table 2). As can be seen from Figure 9, targeting performance with the sharp transition is increasingly worse than with the gradual transition at high levels of transparency. At 100% transparency, the differences amount to time increases of 7%, 23%, and 8% (compared to the unchanging baseline condition) for the three tasks.

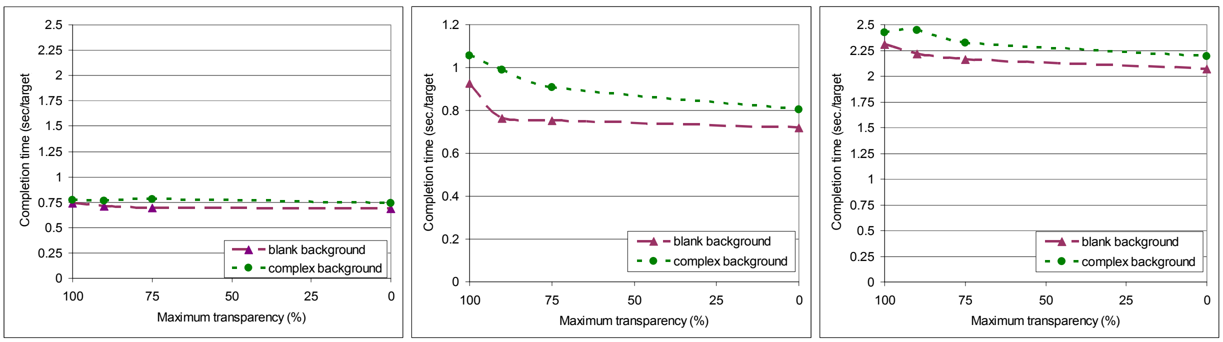
One-dimensional selection			
Interaction	*df*	*F*	*p*
transparency x transition	(3,27)	4.48	<0.05
transparency x background	(3,27)	3.32	<0.05
Multi-directional selection			
Interaction	*df*	*F*	*p*
transparency x transition	(3,27)	17.80	<0.001
transparency x background	(3,27)	8.87	<0.001
Palette-target selection			
Interaction	*df*	*F*	*p*
transparency x transition	(3,27)	6.80	<0.005
transparency x background	(3,27)	0.38	=0.76

Table 2. ANOVA results for interactions between transparency, transition, and background.



(a) One-dimensional selection *(b) Multi-directional selection* *(c) Palette target selection*

Figure 9. Mean completion time per target for all three tasks at all transparency levels, by transition type.



(a) One-dimensional selection *(b) Multi-directional selection* *(c) Palette target selection*

Figure 10. Mean completion time per target for all three tasks at all transparency levels, by background type.

Second, there was also an interaction between background complexity and transparency, although different stories are told by the different tasks. Times were marginally higher for complex backgrounds at lower levels of transparency in the one-dimensional task, which goes against our expectations (however, this effect is small). The multi-directional task showed an interaction that was more in line with expectations; as can be seen in Figure 10b, the curve for the complex background does not drop nearly as fast with decreasing transparency (a similar shape exists for the palette-targets task, but the effect is not significant). This suggests that interference from the background becomes a serious problem as targets approach the threshold of visibility for the current background, at least in tasks where the target locations are not easily memorized.

4.3 Participant Preferences

At the end of the one-dimensional and multi-directional tasks, we asked participants two kinds of preference questions: the first set asked which transition type the participant preferred, and the second asked about the level of transparency that the participant felt was the maximum usable level. A summary of these results are shown in Figures 11 and 12. For transition type, participants stated a clear and consistent preference for the gradual transition. For maximum usable transparency, there was a distinct difference between the complex and blank background: strong majorities stated that 90% transparency was the maximum for the blank background, but that 75% was the maximum for a complex background. This result echoes the quantitative interaction of transparency level with background type as discussed earlier.

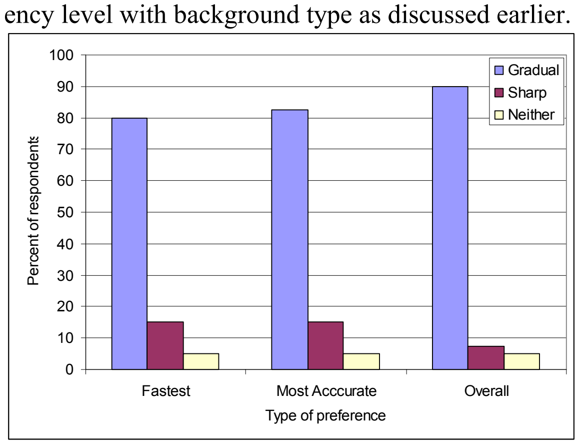


Figure 11. Preference for transition types (percent of respondents), in terms of subjective speed (which was fastest), subjective accuracy, and overall preference.

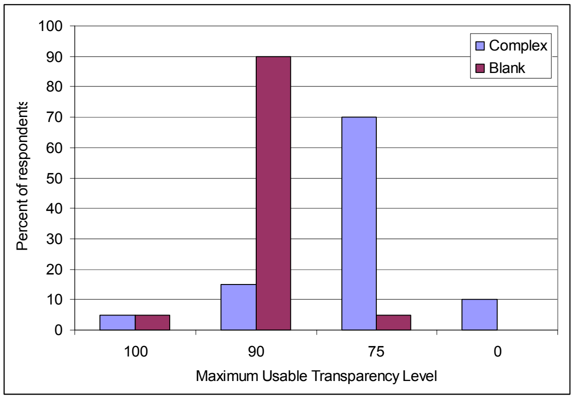


Figure 12. Preference for maximum usable transparency level (percent of respondents) for complex and blank backgrounds.

5 Discussion

The main conclusion that can be drawn from our findings is that although dynamic transparency does reduce targeting performance, the reductions are not so large that they would make the technique unviable for interactive systems. Figure 13 shows summary curves that compare performance at different transparency levels to that with opaque targets; there are many potential implementations of dynamic transparency with quite high transparency levels that would result in less than a 10% performance reduction. Since dynamic transparency trades off targeting performance for underlying-task performance, there may therefore be many situations where dynamic transparency results in overall performance gains for the user.

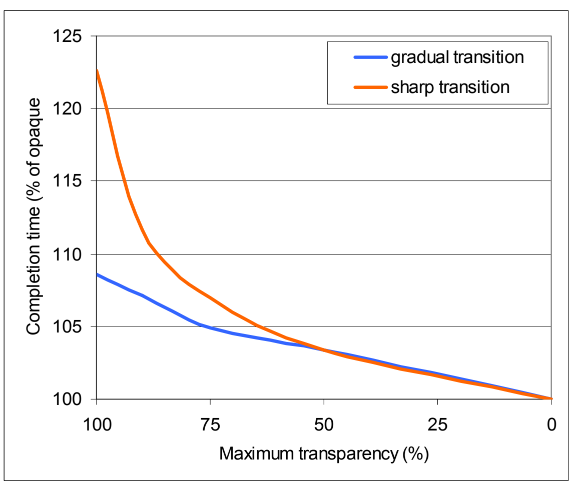


Figure 13. Completion time (mean of all tasks) as a percentage of the CT for opaque targets, by transition type.

Given the overall viability of the technique, the next issue is to determine how to set up the scheme for best results. We were interested in three questions: what are the upper and lower transparency limits for floating windows, and what is a reasonable default transparency value.

Lower limit. In situations where occlusion is a problem, it seems reasonable to impose an lower limit of 50% transparency for tool palettes; this level causes only small performance losses for targeting, and allows either transition type to be used. These results are in agreement with earlier work [5] suggesting that there is little difference between 50% and 0% transparency.

Upper limit. Choices for an upper transparency limit are more variable and will depend on the amount of targeting required for the task, the complexity of the background, the transition type, and the user's tolerance for transparency. For sharp transitions, about 80% transparency appears to be a reasonable upper bound (with about an eight percent reduction in performance); for gradual transitions, however, it may be possible to go higher, even as far as 100% transparency. Full transparency – that is, completely invisible tool palettes – represents the optimal situation for visibility of the underlying image. Targeting performance in this extreme was surprisingly good with the gradual transition; this means that for only an eight percent reduction in targeting performance, users could enjoy a much clearer and much less obstructed view of the images and objects that are the focus of their work. There are caveats to the use of full transparency, however: invisible targets were not a particularly popular choice with the study participants, and we have also observed that some people have particular difficulty with this level. Full transparency should likely not be the first transparency level shown to users in a dynamic transparency implementation, and should not be the default value.

Default level. Transparency levels around 75% appear to be a reasonable middle ground. This would provide considerable visibility of the background image, but only reduce targeting performance by 5-8%. However, since this level was involved in the interaction between background and transparency (see Figure 10b), any default value must be adjusted to ensure basic visibility over the background image. For sparse backgrounds, the level should be increased; for full or complex backgrounds, the level will have to be reduced.

Finally, our recommendations must be considered in light of the fact that target locations in our study were well-known. If users do not set up their workspace in a consistent fashion, with tool palettes regularly placed in consistent locations, then they will be introducing a visual search task into the targeting task. It is possible that high levels of transparency will interact poorly with visual search. However, there are many situations where the locations of floating windows is well known – for example, experts generally set up their workspaces in very regular and consistent ways – and in these situations we believe that our results will have wide applicability.

6 Conclusions and Future Work

Dynamic transparency has been proposed as a way to increase the visibility of background images and objects when floating windows are not in use. However, this is accomplished by decreasing the visibility of the floating windows. Since users still need to be able to target these interface components, we carried out a study to examine the effects of dynamic transparency on targeting. We found that although targeting time increases with increasing dynamic transparency, there are many situations where this increase is less than 10% over opaque targets. We conclude that dynamic transparency is a viable technique in settings where the visibility of the background data is important.

Our main goal for future work is to test dynamic transparency in a realistic application situation, to explore the actual performance gains in situations where both targeting tasks and workspace tasks are involved. In addition, we plan to implement other mechanisms for managing the transition from transparency to opacity. For example, adding a dead-reckoning component to the gradual transition mechanism could allow us to avoid showing palettes that are nearby, but clearly not in the targeting path.

Acknowledgments

This research is supported by the Natural Sciences and Engineering Research Council of Canada.

References

[1] Matt Belge, Ishantha Lokuge, and David Rivers. Back to the Future: A Graphical Layering System Inspired by Transparent Paper. *Adjunct Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 1993, p.129-130.

[2] Eric A. Bier, Maureen C. Stone, Ken Fishkin, William Buxton, and Thomas Baudel. A Taxonomy of See-Through Tools. *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, 1994, p.358-364.

[3] Donald A. Cox, Jasdeep S. Chugh, Carl Gutwin, and Saul Greenberg. The Usability of Transparent Overview Layers. *Companion Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1998, p.301-302

[4] Beverly L. Harrison, Hiroshi Ishii, Kim J. Vicente, and William A. S. Buxton. Transparent Layered User Interfaces: An Evaluation of a Display Design to Enhance Focused and Divided Attention. *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1995, p.317-324.

[5] Beverly L. Harrison, Gordon Kurtenbach, and Kim J. Vicente. An Experimental Evaluation of Transparent User Interface Tools and Information Content. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 1995, p.81-90.

[6] Hiroshi Ishii and Minoru Kobayashi. ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact. *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, 1992, p.525-532.

[7] Axel Kramer. Translucent Patches: Dissolving Windows. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 1994, p.121-130.

[8] Henry Lieberman. Powers of Ten Thousand: Navigating in Large Information Spaces. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 1994, p.15-16.

[9] Sony Entertainment. *EverQuest*. Main web site: http://everquest.station.sony.com/

[10] ISO ISO/TC 159/SC4/WG3 N147: *Ergonomic requirements for office work with visual display terminals – Part 9 – Requirements for non-keyboard input devices*, International Organization for Standardization, May 25, 1998.