Pre-Screen Projection: From Concept to Testing of a New Interaction Technique

Deborah Hix [1,2]
[1] Dept. of Computer Science
Virginia Tech
Blacksburg VA 24061 USA
hix@vt.edu

James N. Templeman
[2] Naval Research Laboratory
Washington DC 20375 USA
templeman@itd.nrl.navy.mil

R.J.K. Jacob
[3] Dept. of Electrical Engineering
& Computer Science
Tufts University
Medford MA 02155 USA
jacob@cs.tufts.edu

ABSTRACT

Pre-screen projection is a new interaction technique that allows a user to pan and zoom integrally through a scene simply by moving his or her head relative to the screen. The underlying concept is based on real-world visual perception, namely, the fact that a person's view changes as the head moves. Pre-screen projection tracks a user's head in three dimensions and alters the display on the screen relative to head position, giving a natural perspective effect in response to a user's head movements. Specifically, projection of a virtual scene is calculated as if the scene were *in front of* the screen. As a result, the visible scene displayed on the physical screen expands (zooms) dramatically as a user moves nearer. This is analogous to the real world, where the nearer an object is, the more rapidly it visually expands as a person moves toward it. Further, with pre-screen projection a user can navigate (pan and zoom) around a scene *integrally*, as one unified activity, rather than performing panning and zooming as separate tasks. This paper describes the technique, the real-world metaphor on which it is conceptually based, issues involved in iterative development of the technique, and our approach to its empirical evaluation in a realistic application testbed.

KEYWORDS: Interaction techniques, empirical studies, pre-screen projection, egocentric projection, formative evaluation, user tasks, input devices and strategies, interaction styles, input/output devices, Polhemus tracker, visualization, metaphors, user interface component.

INTRODUCTION

Pre-screen projection [16] is a new interaction technique for integrally panning and zooming through a scene. On the surface, developing a new interaction technique such as pre-screen projection sounds as if it should be easy. However, once we devised the initial concept — specifically, panning and zooming a display relative to the user's head position (described later) — we found that designing and implementing it in the context of meaningful tasks in a realistic setting required a myriad of unanticipated design decisions, which had unobvious consequences. Because the design space for an interaction technique like this is so large, many of these decisions are best made based on very simple, almost continuous *formative evaluation* [9] — early and frequent empirical testing with users to improve a user interface design. We discuss this general design and evaluation process for inventing, implementing, and testing new interaction techniques, using examples of some of the more interesting specific design issues we encountered in developing pre-screen projection.

A key aspect of our work is that we want to go beyond simply creating new technology and assuming it, merely by its novelty, is inherently better for a user than existing technology. Instead, we continually prototype and evaluate with users to assess the effects of our new technology on human task performance. As the HCI community moves beyond WIMP (window, icon, menu, pointer) interfaces, the evolutionary cycle — conceptualization, implementation, and evaluation — we describe becomes essential for producing interaction techniques that are effective and efficient, not merely new and different.

RESEARCH IN INTERACTION TECHNIQUES

An *interaction technique* is a way of using a physical input/output device to perform a generic task in a human-computer dialogue [5]. It abstracts a class of interactive tasks, for example, selecting an object on a screen by point-and-click with a mouse, with the focus on generic tasks. The pop-up menu is an interaction technique. It is not a new device, but rather a way of using a mouse and graphic display to perform selection from a list.

In many applications, a user performs information retrieval and planning tasks that involve large quantities of spatially arrayed data, for example, presented on a map in a military command and control system. Typically, all available or even desired information cannot be legibly displayed on a single screen, so a user must zoom and pan. Zooming and panning are "overhead" tasks that bring the needed information into view; the user then has "real" tasks to perform using the presented information. A goal of our research is to minimize the mental and physical effort required for such overhead tasks, freeing the user to perform the real tasks with minimal distractions. By studying new means of communication to facilitate human-computer interaction, we develop devices and techniques to support these exchanges. Our research paradigm is to invent new interaction techniques, implement them in hardware and software, and then study them empirically to determine whether they improve human performance.

Interaction techniques provide a useful research focus because they are specific enough to be studied, yet generic enough to have practical applicability to a variety of applications. Interaction technique research also bridges a gap between computer science and psychology, especially human factors. But research into interaction techniques *per se* is relatively rare, and research in this area that combines technical innovation with empirical evaluation is unfortunately even rarer.

RELATED WORK

Other researchers have explored the coupling of a dynamic scene to head tracking [4, 17]. In the geometry for this "fish tank" projection, the scene is calculated either behind or within a few centimeters in front of the screen, creating a realistic illusion of 3D scenes. The visual effect of this placement appears very different to a user than pre-screen projection, which exaggerates perspective effects to provide a dramatic way to pan and zoom. Other work on progressive disclosure, a key feature of pre-screen projection, has been done by [6, 8].

Early work on interaction techniques is found in the Interaction Techniques Notebook section of the *ACM Transactions on Graphics*. Examples of other interaction technique work include see-through tools [2], a collection of 3D interaction techniques [15], and a collection of eye movement-based techniques [10]. However, much of this work emphasizes invention of new techniques over evaluation, especially evaluation of human performance. Some work that discusses evaluation includes fish tank virtual reality [17], the alphaslider [1], the Toolglass [12], marking menus [13], and eye tracking [18].

THE CONCEPT UNDERLYING PRE-SCREEN PROJECTION

Real-World Metaphor

The real-world metaphor upon which pre-screen projection is based is the fact that our view of the world changes as we move our head. We are completely familiar with how our view changes as we move about, and we control our view by positioning our head. For example, as we move nearer to objects, they appear larger, and appear smaller as we move away. Pre-screen projection uses a three-dimensional tracking device to monitor head movement and alters the view presented to the user (in a specific way, which we discuss later) based on head position. It is a new technique for integrally panning and zooming through a display, and for progressive disclosure and hiding of information under user control.

Using pre-screen projection, a user wears a light-weight helmet or headband with a three-dimensional Polhemus tracker mounted on the front, as shown in Figure 1. As the user moves from side to side, the display smoothly pans over the world view. As the user moves closer to or further from the screen, the display smoothly zooms in and out, respectively. (Figure 2 shows a sketch of how the scene on the screen changes as the user moves.) Head tracking is used to control the screen view by making the viewpoint used to calculate a virtual scene correspond to a user's physical viewpoint.

Just as with any zoom or pan technique, pre-screen projection allows a user to selectively reveal different portions of the whole world view on the screen. For example, working with a system that updates a geographic display as new information arrives, a user can use pre-screen projection to zoom in when a new situation occurs, to evaluate details, and to zoom out to see an overall view of the situation.

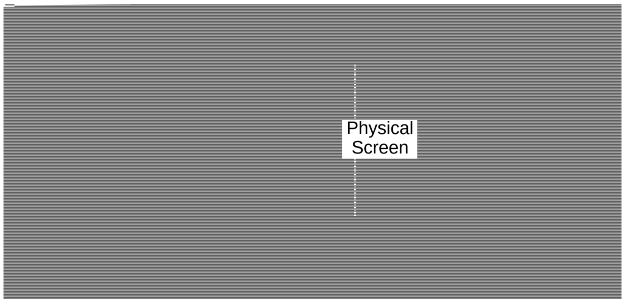

FIGURE 1. Pre-screen projection concept.

Note that the scene actually is displayed on the physical screen, but its dynamic perspective from the user's viewpoint reacts to the user's head movement as if the scene were in front of the screen.

Physical geometry is such that as we move toward a scene, objects in the scene enlarge in our vision; objects closer to us enlarge more rapidly than those in the background. To obtain a dramatic zoom effect in the virtual world presented by the computer, the scene (e.g., a map being panned and zoomed) must be treated as if it were close to the user. We therefore calculate the geometry of pre-screen projection as if the scene were located *in front of* the physical screen (i.e., nearer the user than the physical screen), as Figures 1 and 2 illustrate. This causes the scene to enlarge more rapidly than the screen as a user moves toward it and therefore produces a dramatic zoom. A user, unaware of the pre-screen geometry, sees the visible scene displayed on the physical screen. The screen changes continuously (not discretely like some pan or zoom techniques) as the user moves.

In the real world, a person's view naturally pans and zooms simultaneously as the head moves in all three dimensions simultaneously. However, in many current user interfaces that incorporate panning and zooming, a user performs panning or zooming tasks by moving a mouse or trackball, first to perform one task and then the other, often using two different interaction techniques. This forces a user to perform separately what naturally are two integral tasks [11]. A user typically does not think of zooming or panning as separable tasks, but thinks rather of unified operations like "focus in on that area over there". A strong feature of pre-screen projection is that a user can navigate (pan and zoom) concurrently. We call this operation *integral pan and zoom*, to distinguish it from traditional two step approaches.

Further, because panning and zooming are closely coupled tasks in a user's mind, it is more natural for a user to make a gesture that performs the overall operation, using an integral 3D input device, rather than having to reach for or hold a specific device (e.g., mouse or trackball). Because a user of pre-screen projection does not have to perform a deliberate physical action, such as reaching for a mouse, to acquire the control device (the Polhemus 3D tracker), another feature is that integral panning and zooming is enabled at all times.

Another benefit is the "lightweight" nature of integral pan and zoom obtained using head tracking and rapid, continuous display update. Moving the head toward the screen momentarily to read a portion of the display and then moving away becomes a rapid and natural operation, with less physical and cognitive effort than more conventional zoom or pan techniques. As a result, zooming and panning can now be used in situations where "heavier" techniques would be inappropriate (e.g., zooming in to read a text legend and then immediately zooming back out). Use of a "heavier" zoom technique could be too disruptive for this task. We exploited this property through user-controllable progressive disclosure of information, described later.

Visual Perception

Pre-screen projection derives from an understanding of J.J. Gibson's [7] ecological approach to visual perception. Gibson suggests that the visual system is an organ that allows people to acquire useful information about their environment. A person's view of the world changes every time they move their head, and each view discloses new information about both the environment and the user's place in it. People perceive relationships between their movements and the views they see. A knowledge of these relationships is the basis for visual perception. For example, as mentioned earlier, we expect physical objects to appear larger as we move toward them. We use this knowledge to interact with the real world. A dynamic view is the norm in the real world, unlike the static views typically presented on a computer screen. Pre-screen projection carries many dynamic properties of the visual world into the world of computer-generated imagery, because of the real-world metaphor we have adopted. Although it might seem odd to invoke three-dimensional viewing techniques just to shift and scale a flat picture on the screen, by doing so we gain a natural technique for performing head-based integral panning and zooming.

ITERATIVE DEVELOPMENT OF PRE-SCREEN PROJECTION

Having developed the concept of pre-screen projection, we then began the challenge of developing the interaction technique itself, which includes design, implementation, continuous iterative evaluation, and refinement. As we proceeded with the design, we discovered an enormous design space of possible attributes. Each attribute could have several (and in some cases, many) values, so the combinatorics and interactions of the numerous design decisions quickly became very large. Thus, early design decisions had to be made based on rapid prototyping and quick, informal formative evaluation cycles, rather than on comprehensive experimental evaluation. Below we describe this process for two attributes — scaling and fade-in/out — for which the choice of values was very large and final decisions were counterintuitive or otherwise unexpected.

Scaling

Scaling — the geometry by which objects in a scene appear larger or smaller as a user zooms (in pre-screen projection, as a function of head movement perpendicular to the screen) — was one of our biggest challenges. Three types of objects in the scene could potentially be designed to scale: the map, icons, and text.

In all design iterations, the map scaled, because the map provides the predominant spatial context and expected visual perspective. Our testbed is a Naval command and control application (briefly described later) in which icons represent military objects such as ships and planes. In our earliest designs, icons did not scale because scaling icons on a map might imply that their size has meaning. In our application, the size of icons conveys no meaning, although it could in other applications. But in formative evaluation we observed that users found it difficult to zoom in on non-scaling icons because their sense of perspective expansion was reduced. So in our final design, icons are scaled.

We chose not to scale text, to maximize the amount of readable text presented to the user at any time. If text is scaled, when the user's head is close to the viewing plane the text can get too large to read, and, conversely, as the user's head moves further away, the text decreases in size and becomes too small to read. We wondered if our design in which the map and icons scale but text does not would be confusing or unnatural to users, but we found in formative evaluation that users were comfortable with this design.

In early designs, we used a scaling factor of real-world linear perspective, 1/z (where z is the distance from the user's head to the virtual scene), for the map. When we applied this geometry to make icons scale, they were almost invisible until the user's head was quite close to the viewing plane. That is, a ship in accurate relative scaled size to Europe became a tiny dot when the user had a reasonable portion of Europe in the scene using pre-screen projection. So we redesigned again, by making icons disproportionally large relative to the map.

Yet another issue surfaced with the 1/z scaling: a user could not back far enough away (i.e., zoom out) to get the entire world view onto the screen. So we tried various computations for amplifying the scaling as a user moves further away until we found one that worked well for scaling both the map and the icons. We used linear perspective within a 1 foot distance beyond the virtual scene, and beyond that, amplified the scaling by an additional linear factor. In the final design an entire map of Europe and Africa appears when the user is at a distance of about 3 feet from the virtual scene.

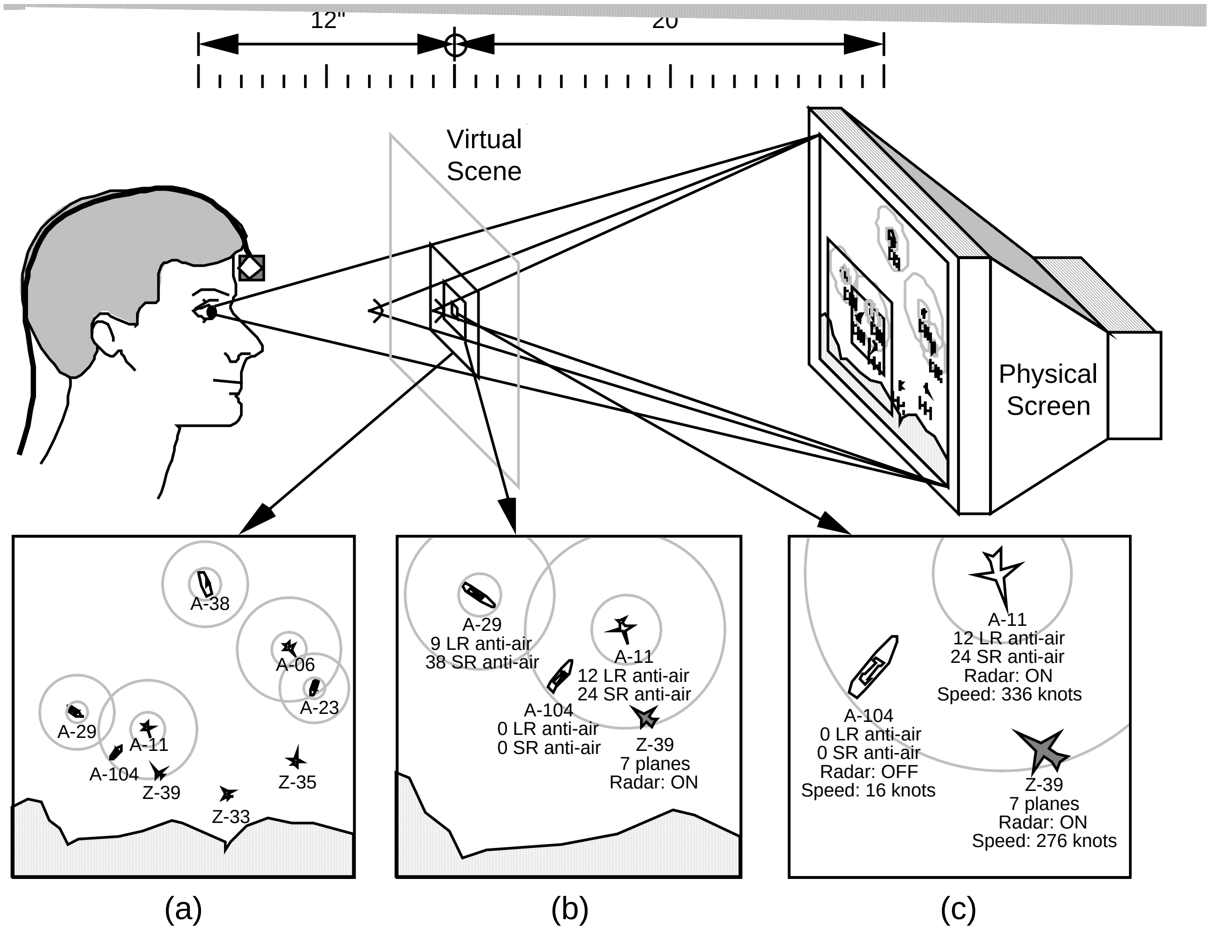

FIGURE 2. Example of scaling (of map and icons) and fade-in (of text).

Figure 2(a), the first level of fade-in/out, depicts the scene (seen by the user on the physical screen) when the user's head is 12 inches from the virtual scene. At this level, only the military ID is shown for each icon. Figure 2(b), the second level of fade-in/out, shows the screen as the user moves in closer to look at a specific engagement. Additional lines of text appear for each icon. Transition among displays occurs continuously and smoothly, not discretely. In Figure 2(c), the third level of fade-in/out, the user is even closer, to examine further details. (For illustrative purposes here, formats have been altered; in our testbed design, icons take up a smaller portion of the screen, and there are more lines of text.)

Fade-in/out

Fade-in and -out — the gradual appearance and disappearance of objects in a scene — were somewhat easier to design than scaling, but we still made numerous iterations with users before we settled on a design. As with scaling, the same three types of objects — the map, icons, and text — could potentially be designed to fade-in/out.

In all design iterations, the map and icons scaled but did not fade in/out, because of their role in maintaining spatial context. However, text associated with each icon in a scene fades in and out at three different *levels of disclosure*, as a user moves toward and away from the screen. In the first level of fade-in, an icon identifier appears (a military ID), and in each of the next two levels, additional icon-specific information about weapons, speed, fuel, and so on, appears. The first level of fade-in/out gradually occurs between 13 and 12 inches from the virtual scene; the second level gradually fades in/out between 4 and 3 inches from the virtual scene, and the third level between 2 and 1 inches. A calculated perspective for the virtual scene at 20 inches in front of the physical screen works well for these levels of disclosure.

Figure 2 shows a simple example of how scaling and fading in pre-screen projection allow successive disclosure (or hiding) of details as a user moves toward (or away from) the screen.

Figure 3 shows what a display would look like if all information were displayed at once. Pre-screen projection allows a user to control this kind of clutter, by selectively presenting details as the user moves nearer to and further from the screen. Although no one would purposefully design a display to look like this, in a dynamic system this muddle could unpredictably occur.

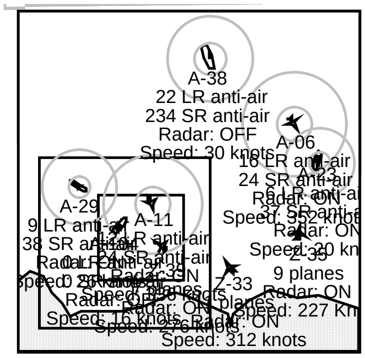

FIGURE 3. Cluttered display.

We frankly chose distances for the levels of progressive disclosure based on trial and error, but formative evaluation showed that those just described work well. However, we were surprised at the relatively small quantity of text (about 8 to 10 lines per icon) that can be displayed before the text lines for different icons overlap substantially as a user gets close in. We had expected that, because icons move visually further apart as a user gets closer to the viewing plane, there would be space for a rather large amount of information (in our case textual, but it could have been graphical) to fade in.

We found that even something as simple as spacing between lines of text must be carefully designed and evaluated. Since text does not scale but the distance between lines of text does, in early designs of fade-in/out, text was too densely packed even after a user zoomed far in. We realized that line spacing must be computed in screen units rather than world map coordinates, even though placement of a block of text is tied to a fixed location on the map. We could have anticipated this during design sessions, but with the large number of design decisions to consider, it is all too easy to overlook or neglect such details. Implementation and user comments quickly make such oversights glaringly obvious.

Also in early stages of design, text at one level faded out as different text from the next level replaced it. We tried this because we wanted to display as much information as possible, which meant re-using screen space. However, users found it hard to determine which level they were in as they looked for specific information. We changed the design to fade in only additional text, beneath (rather than on top of the location of) text displayed at prior levels. We found for our testbed, as mentioned above, that a total of 8 to 10 lines (we use 1 line at the first level and 4 additional lines at both the second and third levels) was the maximum amount that could be displayed and still be readable. This design could vary for different applications.

Counterintuitive Results in Final Design

It is interesting to note that our final design for scaling and fade-in/out of text is exactly the opposite of how scaling and fading naturally work in the real world, as shown below in Table 1. Namely, as a user moves toward and away from text in the real world, that text appears to get larger and smaller (scales) but does not disappear altogether (fade). Based on empirical observations during formative evaluation of pre-screen projection, as discussed above, we found it works best when text fades but does not scale. *This kind of unexpected design decision could only have been corroborated by empirical observations with users.*

	In the real world		In our final design	
Icons/	Text	Icons/	Text Map	Map
Scaling	Y	Y	N	Y
Fade-in/out	N	N	Y	N

TABLE 1. Design decisions for scaling and fade-in/out.

Design Attributes for Pre-Screen Projection

While our previous discussion revolved around the attributes of scaling and fade-in/out, there are numerous other factors in the design space for pre-screen projection. Very few interaction techniques are reported in the literature with an attribute list; some notable exceptions are [2, 3, 14]. Some attributes and a few of their possible values are as follows:

- *Scaling* (e.g., type, what is scaled, interaction with fade-in/out)
- *Fade-in/out* (e.g., type, what fades in/out, how many levels, where on screen)
- *Context "world view" presentation* (e.g., miniature map, size, location, information content)
- *Interactions among attributes* (e.g., scaling & fading)
- *Freeze/unfreeze of dynamic screen image* (e.g., how performed by user, relative vs. absolute)

This latter attribute, freeze/unfreeze, is an interesting one for which we are still exploring possibilities. Obviously, with the constantly changing scene afforded by pre-screen projection, a user may, for various reasons, want to momentarily halt movement of the scene. We have incorporated a freeze feature through which, simply by pressing a button, a user can decouple head movement from scene movement. The user can continue to view the static scene as it appears when frozen, while moving the head as desired. When the user wishes to unfreeze the scene, re-engaging the coupling between head movement and scene changes, the user again presses a button. We have designed two choices: an absolute and a relative unfreeze.

In absolute unfreeze, what a user sees changes to reflect the user's head position when the unfreeze is performed, resulting in a slow drifting of the scene on the screen to the new perspective. An absolute unfreeze adjusts the virtual scene to that which would have been seen at the user's head position at the time of the unfreeze if dynamic perspective had never been frozen. This maintains the scale and placement of the virtual scene in front of the screen.

In relative unfreeze, what a user sees does not change to reflect the new head position; to accommodate this, the virtual scene is computationally adjusted (scaled and shifted) to maintain its appearance to the user at the new head position. The scene on the screen resumes from where it was when the user performed the freeze, regardless of where the user's head is when the unfreeze is performed. Currently, a user can choose between either of these types of unfreeze, and further evaluation will give us more indications of which is preferable in which situations.

DEVELOPING TASKS, APPLICATION, AND EMPIRICAL STUDIES

As we developed pre-screen projection and incorporated it into a testbed application, we found the major components and relationships in the process of developing an interaction technique to be as shown in Figure 4. Interaction techniques research often stops after the technique has been implemented and tested to make sure that it works. The remaining three components are ignored. We have described the first component, iterative development of an *interaction technique*. However, our work proceeds within the context of this overall development process for interaction techniques, because we want to evaluate effects of new interaction techniques on human performance, not simply see if they operate correctly. It is thus necessary to define *tasks* for a user to perform with the new technique, to set those tasks in an *application* that can be used in an empirical study, and to design the *empirical study* itself. This approach, once again, led us to complexities we had not expected.

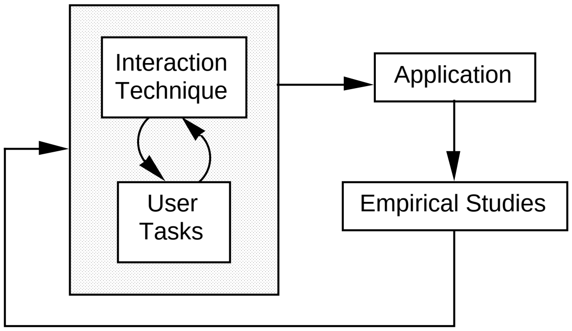

FIGURE 4. Relationship among major components in interaction technique development process.

User Tasks and Application

Panning and zooming are inherently spatial information-seeking activities. A user can, for example, casually browse or actively search for specific information. A user can seek different kinds of information. For example, high-level results of a library search might be displayed as a scatterplot through which a user navigates by panning and zooming to find details about a particular document in the results set. Panning and zooming are pervasive and fundamental activities in many applications, including scientific or information visualization, spatial or geographic presentation, and virtual reality. The way in which a user performs panning and zooming tasks shapes the relationship between that user and a computer environment. To evaluate pre-screen projection, we needed to create user tasks that would capture the essential nature of panning and zooming.

We co-evolved user tasks along with the interaction technique, to produce the best tasks not for the sake of the application, but for evaluating pre-screen projection in a meaningful situation. In fact, we had numerous design iterations of both the technique and the tasks used to evaluate it. We chose Naval command and control (C^2) systems as a rich domain relevant to Naval applications. C^2 systems support the planning, coordination, and execution of military missions. We incorporated pre-screen projection into a C^2-like testbed running on a Silicon Graphics Iris 4D/210VGX workstation, and created task scenarios for evaluation of the technique. The purpose of this testbed is to serve as a context for evaluating interaction techniques, not to develop the best C^2 system.

An interaction technique and goals for its evaluation should drive task development. The task, in turn, influences details of the instantiation of the interaction technique (e.g., in the task of allocating weapons, designing what textual information to display and how to display it for a ship icon). Tasks used for evaluation of interaction techniques are often extremely simplistic (for example, in Toolglass evaluation [12], the experimental task was a simple colorized version of connect-the-dots). This is obviously much easier, but it may be a potentially risky reduction of more complex tasks. Nevertheless, any evaluation is better than none.

We found that important criteria for designing tasks to evaluate pre-screen projection included:

- must have a strong spatial and navigational component (to exploit integral pan and zoom)
- must involve more information than can be presented on a static screen (to evaluate fading and scaling)
- must be simple, quick, and easy to learn (because participants are generally non-military types)
- must be as deterministic as possible with a minimum of variability caused by individual differences in user strategy

To meet these criteria, we developed a defensive engagement situation, in which friendly, enemy, unknown, and neutral military units (ships and planes) are displayed on a map of the Mediterranean. We created several scenarios in which multiple enemy, unknown, and neutral units appeared during the scenario, moving along a course to intersect friendly units. Figure 2 gives a flavor of how the screen looks and changes. Circles around each friendly unit (denoted by an A) indicate the range of its missiles. We used realistic (unclassified) military data when feasible in the scenarios.

The user's goals (tasks) were to acquire and maintain an awareness of the evolving situation, and to allocate missiles from friendly units to encroaching enemy planes. Specifically, a user monitors the scene looking for threats. When one is observed, the user determines the number of planes in that threat (by zooming in on textual details that progressively fade-in as described earlier) and the number of missiles the friendly unit has available to fire. The user then uses a slider to enter the number of missiles to shoot and, based on simple "rules of engagement" (see below), then fires those missiles. This series of tasks is time-critical (to detect and eliminate enemy planes as soon as possible), and therefore lends itself well to quantitative metrics such as time to perform tasks and accuracy of task performance. Entering the number of missiles and issuing a fire command are unrelated to panning and zooming, but allow us to determine task closure to measure human performance.

We formulated our "rules of engagement" by which a user knows how to allocate missiles, to preclude, as much as possible, interference of individual user strategies for performing tasks, as well as the need to understand complicated military rules. For example, one rule that was overly simplified from a real military situation is that each allocated missile will bring down one aircraft with a probability of 100%.

Empirical Studies

Empirical evaluation is a key component in developing a new interaction technique. Best guesses about design are substantiated or refuted by many tight, short cycles of formative evaluation. We have already performed numerous cycles of formative evaluation — some as short as five minutes (these were the really bad designs!), others lasting nearly an hour. Evolution of designs for scaling and fading of the map, icon, and text, described previously, as well as many other design details, evolved from numerous rounds of formative evaluation.

Many of our formative evaluation results were discussed when we explained the design of scaling and fade-in/out. There were other counterintuitive and surprising results. We expected a slight illusion of three-dimensionality from the perspectively correct visual transformations, but it turned out not to be there. Despite this, users claim that pre-screen projection provides a natural way to navigate a scene. Users are comfortable with the technique and generally can use it effectively within about five minutes. Some users reported that after using pre-screen projection awhile, they found the static screen of a standard system to be "dead".

One interesting experience occurred with a user wearing bifocal glasses. As the user moved her head in close enough to see details, she also had to tip her head backward to use the lower portion of her bifocals. Obviously, this caused the scene on the screen to shift completely away from where the user was trying to look! We are still considering solutions to this difficulty.

An ergonomic issue to be explored is that of fatigue due to constrained movement over long periods of use of pre-screen projection. Clearly, forcing users to hold their head or upper body at a particular position in order to maintain the desired scene could cause serious problems, even being harmful to users over long periods of time. The freeze/unfreeze feature mentioned previously is one mechanism that we will further investigate as a possibility for overcoming this potential ergonomic problem.

Although this paper discussed our formative evaluation, summative evaluation is also important. *Summative evaluation* [9] is empirical evaluation in which several designs are compared, with primarily quantitative results. In our research, the purpose of summative evaluation is to perform comparative studies of user performance with different interaction techniques for the same tasks. A next step in our research is to perform summative evaluation, comparing user performance and satisfaction using pre-screen projection to other pan and zoom techniques, using the defensive engagement scenarios. These scenarios can be used to evaluate any pan and zoom technique, particularly those involving progressive disclosure of information.

CONCLUSIONS

We learned several important lessons during design, implementation, and evaluation of pre-screen projection. Those of most general interest include:

- When incorporating interaction techniques into an application, we found deliberate violation of user interface design guidelines was sometimes necessary to construct an effective evaluation situation.

For example, we wanted to determine how users use pre-screen projection for navigation when monitoring a situation, so we did not include any audible notification of a new enemy, even if it was off the screen when it appeared. We wanted to see how a user maintains vigilance and develops situational awareness with pre-screen projection. Further, we allowed a user to fire an incorrect number of missiles (more or less than needed), because we wanted to make sure a user can access detailed information (about numbers of missiles, for example).

- We struggled constantly to keep the application testbed simple, finding it all too easy to develop an unnecessarily elaborate testbed design (i.e., C^2 system) and to lose sight of evaluating the interaction technique.

The current version of the testbed is so simplistic that most tasks performed by a user could be automated. If all tasks were automated, there obviously would be no need for an interaction technique! We had to keep it simple because real C^2 systems take months of training, which is obviously inappropriate for our needs.

- As with any experiment, user performance variability among participants needs to be minimized and/or controlled as much as possible, and experimental tasks need to be deterministic.

This is particularly important for measuring human performance with a novel interaction technique. This constraint limits both the tasks a user can perform during evaluation and also may limit design of the interaction technique itself. Attribute values of the interaction technique must be controllable by the experimenter.

- We often got wildly unexpected results when our continually evolving designs for pre-screen projection were implemented.

We found that even our best guesses, based on expertise, experience, and good design guidelines, did not always work as expected. As discussed, fade-in/out of text was a surprise. The combinatorics of alternative design details (e.g., color, font, spacing, icon shape) were huge. Nonetheless, the devil really is in the details. For example, in an early design, all text was the same color (white), and we found that when icons were close enough for their associated text to overlap, it was not possible to tell which text went with which icon. By tinting the associated text when a specific friendly (blue) and a specific enemy (red) icon were selected, users could then tell which text went with which icon. This small change made a previously impossible task do-able.

- We tried very hard to avoid having issues, constraints, and difficulties with the implementation of pre-screen projection affect design decisions for the technique.

There were a few times when this was unavoidable. For example, using high-resolution data for the map caused (not unexpectedly) the screen to update too slowly and to jitter, so we simplified the map outline. Design of pre-screen projection has led us through many of the same issues involved in designing virtual reality applications.

SUMMARY

Our development of *pre-screen projection* has shown that an interaction technique based on a real-world metaphor can provide a natural, useful means of human-computer interaction. But developing new interaction techniques is difficult, because their design space can be large, implementation complicated, and evaluation time-consuming and resource-intensive. The evolutionary cycle — conceptualization, implementation, and evaluation — we have presented is nevertheless essential for producing interaction techniques that are effective and efficient, not merely new and different.

ACKNOWLEDGMENTS

Kapil Dandekar, Ankush Gosain, Steven Kerschner, and Jon Yi were super-implementers for the testbed. Linda Sibert and Scott Powell contributed many excellent design ideas. Dr. Rex Hartson provided invaluable comments on an early draft of this paper. This work is sponsored by the Decision Support Technology block (RL2C) within the ONR Exploratory Development Program, and is managed by Dr. Elizabeth Wald.

REFERENCES

1. Ahlberg, C., and B. Shneiderman. (1994). The Alphaslider: A Compact and Rapid Selector. *Proc. CHI'94 Conf.*, 365-371.

2. Bier, E.A., M.C. Stone, K. Fishkin, W. Buxton, and T. Baudel. (1994). A Taxonomy of See-Through Tools. *Proc. CHI'94 Conf.,* 358-364.

3. Bleser, T.W., and J.L. Sibert. (1990). Toto: A Tool for Selecting Interaction Techniques. *Proc. UIST'90 Symp.*, 135-142.

4. Deering, M. (1992). High Resolution Virtual Reality. *Computer Graphics*, 26(2), 195-202.

5. Foley, J.D., A. van Dam, S.K. Feiner, and J.F. Hughes. (1990). *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA.

6. Furnas, G.W. (1986). Generalized Fisheye Views. *Proc. CHI'86 Conf.*, 16-23.

7. Gibson, J.J. (1986). *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, Hillsdale, NJ.

8. Herot, C.F., R. Carling, M. Friedell, and D. Kramlich. (1980). A Prototype Spatial Data Management System. *Computer Graphics,* 14(3), 63-70.

9. Hix, D., and H.R. Hartson. (1993). *Developing User Interfaces: Ensuring Usability through Product & Process*. John Wiley & Sons, Inc., New York, NY.

10. Jacob, R.J.K. (1991). The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At Is What You Get, *ACM Trans. Info. Sys.*, 9(3), 152-169.

11. Jacob, R.J.K., and L.E. Sibert. (1992). The Perceptual Structure of Multidimensional Input Device Selection. *Proc. CHI'92 Conf.,* 211-218.

12. Kabbash, P., W. Buxton, and A. Sellen. (1994). Two-Handed Input in a Compound Task. *Proc. CHI'94 Conf.*, 417-423.

13. Kurtenbach, G., A. Sellen, and W. Buxton. (1993). An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction,* 8, 1-23.

14. Mackinlay, J.D., S.K. Card, and G.G. Robertson. (1990). A Semantic Analysis of the Design Space of Input Devices. *Human-Computer Interaction,* 5, 145-190.

15. Snibbe, S., K.P. Herndon, D.C. Robbins, D.B. Conner, and A. van Dam. (1992). Using Deformations to Explore 3D Widget Design. *Computer Graphics,* 26(2), 351-352.

16. Templeman, J.N. (1993). Pre-screen Projection of Virtual Scenes. *Proc. Fall'93 Virtual Reality Conf.*, 61-70.

17. Ware, C., K. Arthur, and K.S. Booth. (1993). Fish Tank Virtual Reality. *Proc. InterCHI'93 Conf.*, 37-41.

18. Ware, C., and H.T. Mikaelian. (1987). An Evaluation of an Eye Tracker as a Device for Computer Input. *Proc. CHI+GI'87 Conf.*, 183-188.