Focus Plus Context Screens: Combining Display Technology with Visualization Techniques

Patrick Baudisch, Nathaniel Good, and Paul Stewart
Information Sciences and Technologies Laboratory
Xerox Palo Alto Research Center
Palo Alto, CA 94304, U.S.A.
+1 (650) 812 4656
{baudisch, ngood, stewart}@parc.xerox.com

Abstract
Computer users working with large visual documents, such as large layouts, blueprints, or maps perform tasks that require them to simultaneously access overview information while working on details. To avoid the need for zooming, users currently have to choose between using a sufficiently large screen or applying appropriate visualization techniques. Currently available hi-res "wall-size" screens, however, are cost-intensive, space-intensive, or both. Visualization techniques allow the user to more efficiently use the given screen space, but in exchange they either require the user to switch between multiple views or they introduce distortion.

In this paper, we present a novel approach to simultaneously display focus and context information. *Focus plus context screens* consist of a hi-res display and a larger low-res display. Image content is displayed such that the scaling of the display content is preserved, while its resolution may vary according to which display region it is displayed in. Focus plus context screens are applicable to practically all tasks that currently use overviews or fisheye views, but unlike these visualization techniques, focus plus context screens provide a single, non-distorted view. We present a prototype that seamlessly integrates an LCD with a projection screen and demonstrate four applications that we have adapted so far.

Keywords
Display, focus plus context screen, mixed resolution, overview plus detail, fisheye view, video projector.

INTRODUCTION

Faster computers, inexpensive memory, and large storage have brought the ability to work with larger amounts of information to the computer user. While computational power and storage have increased rapidly over the past few years, the screen size and resolution available to consumers has not. This is an issue when users work with large visual objects, where overall structure is as important as detail information for getting the task accomplished.

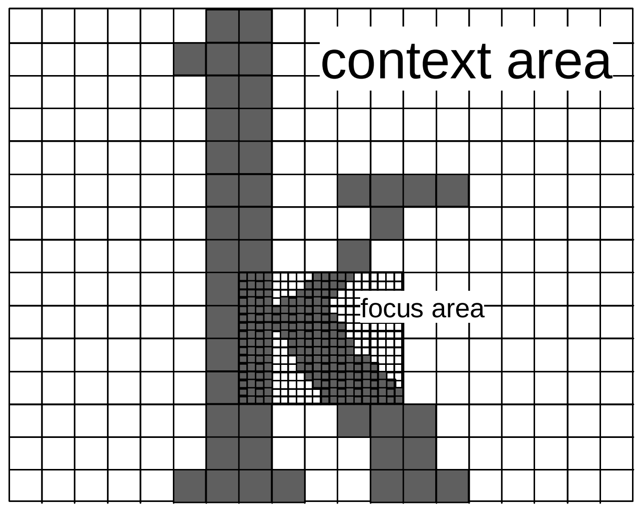

Figure 1: Focus plus context screens consist of low-res regions and hi-res regions. Image content displayed across regions preserves its scaling, while its resolution changes.

Designers working in the print industry, for example, have to make sure that the hardcopies of their layouts look perfect, whether they are viewed from a distance or close-up. Because print offers a much higher resolution than computer screens, examination of all possible facets of the layout via a computer screen involves a substantial amount of zooming. Similar needs for zooming occur when architects use a CAD program to edit blueprints, when radiologists analyze X-ray images on the screen, or when people examine a large city map on their computer screens. In all these cases, the display becomes the bottleneck of the computer systems.

When a user's display is not able to show the number of pixels required for displaying the entire content at the desired level of detail, the users can navigate the display to acquire the information sequentially. Additional navigation means additional user effort, which motivated researchers to explore other solutions to the problem. One approach involves replacing the current screen with a larger screen capable of displaying the required number of pixels. Another approach is to provide an appropriate visualization technique that allows fitting the required data into a smaller screen by reducing the space allocated for irrelevant information. In the following two sections, we will look at how existing display technologies and visualization techniques deal with these problems.

Related work in large hi-res displays

We know of no technology that allows production of one piece, high-quality displays of arbitrary size. Proposed techniques typically involve combining multiple smaller displays into a single large display of high pixel count.

One common solution is to connect two or more computer monitors to a single computer, as supported by current operating systems, such as Microsoft Windows. In this setup, all connected monitors form a single logical display. This display allows windows to be moved across display borders and large windows to span multiple displays. However, Grudin [8] observed that the visible gap between individual monitors discouraged users from having windows span multiple displays. His study suggests that users instead use additional monitors to separate windows belonging to different tasks.

In order for large displays to overcome this effect, a substantial amount of research has been invested in the creation of *seamlessly* tiled display systems [6]. Several solutions for avoiding the visible seams have been proposed, including the integration of up to four identical LCDs by butting them together into a single large LCD (a 9-megapixel display by IBM[1],), the construction of video walls by assembling back projection modules with small borders[2], as well as a series of research prototypes evolving around tiled hi-res projection displays [10]. Compound displays composed of multiple display units surrounding the user have been used in virtual reality, such as flight simulation [15], and in immersive environments, such as the CAVE [4]. These proposed solutions are still cost-intensive, space-intensive, or both, which has prevented these technologies from reaching the mass market.

Besides this work, which attempts to obtain large homogeneous displays, some research has been done in hybrid display systems. In the context of computer supported cooperative work, multiple displays have been combined loosely in order to provide users with personal space as well as shared workspace. I-LAND [23] gives users a shared workspace by providing a large projected area that users can interact with and use for collaboration. Jun Rekimoto's [19] augmented surfaces project allows notebooks to overlap the projection space and permits users to drag material between the notebook screen and the shared projection area. However, users of the system reported that the disproportionate scaling between the notebooks and the projection area was distracting.

Feiner proposed a hybrid display combining a semi-transparent head-mounted display with a conventional CRT monitor [5] (see [2] for more recent work on this track). This display used the monitor to show a selected portion of a larger X-Windows desktop, while the low-res head-mounted display gave an overview of the same desktop. In the overview, the image displayed by the goggles continued the monitor image virtually into the room. This solution, however, was limited by the lag of the head tracking apparatus that was required for aligning the goggles and the monitor. This lag caused the image content displayed across both displays to be temporarily disrupted whenever the user moved his or her head.

Related work in visualization techniques

Research in visualization techniques has resulted in methods for fitting more relevant data onto a given screen by reducing the space allocated for irrelevant information. Plaisant [17] and more recently Olston and Woodruff [16] provide overviews of the various types of visualization techniques in use.

The most prominent techniques for reducing navigation overhead are overview plus detail and fisheye views [7,3]. Overview plus detail visualizations [13] use two distinct views: one showing a close up and the other showing the entire document. While this technique helps users to orient themselves in large spaces, the drawback of this approach is that it requires users to visually switch back and forth between the two distinct views and to re-orient themselves within the view every time they do so.

By using non-linear scaling, focus plus context visualization techniques, such as fisheye views [7,3] and Document Lens [20] allow users to see one or more selected *focus* regions in full detail, while information in the periphery (the *context region*) is compressed to occupy less screen space. The benefit of fisheye views is that they keep adjacent information together, thereby avoiding the need for users to explicitly switch between multiple views. This provides for faster switching between the detail region and the periphery. The main drawback of fisheye views is the distortion that they introduce. This makes them inappropriate for content where proportions and distances matter. Photographic content, for example, easily becomes unrecognizable, which limits the applicability of fisheye views to such tasks as visual design.

FOCUS PLUS CONTEXT SCREENS

We propose *focus plus context screens* (f+c screens), as a new way of fitting a larger piece of large visual objects into a display in order to let users save zooming interactions. Focus plus context screens open a new field of re-

---

[1] http://www.research.ibm.com/resources/news/20010627_display.shtml

[2] http://www.panasonic.com/medical_industrial/11-16-00.asp

search, which—as emphasized by their name—is located in the intersection between display technology and visualization techniques.

Figure 1 shows the general concept. Focus plus context screens offer regions of high resolution and regions of low resolution[3]. Image contents preserve their scaling, even when their resolution varies. The geometry of the displayed content, i.e. the ratio between lengths in the image, is thereby preserved.

Figure 2: The high-res region in the center of this focus plus context screen provides users with detail information.

Figure 2 shows a photo of our prototype system displaying a map. The focus display is located at the same location as in Figure 1, but correct calibration of the display renders it invisible from a distance. However, the callout showing a close-up of the border region between the two resolutions unveils pixels of different sizes.

Focus plus context screens implement regions of different resolution by combining multiple display units of different resolution. Building a focus plus context screen therefore starts with the choice and configuration of display hardware. The rest of this paper is outlined as follows: In the following sections, we will describe the hardware and software requirements for a focus plus context screens, the methodologies used for combining them, as well as a concrete setup. Since software implementations can be application-specific, we will begin the second half of this paper with a presentation of the applications we built so far, followed by a description of the software implementation. A presentation of early results and a discussion of the achievements will then conclude the paper.

Requirements

To make sure that users perceive and use focus plus context screens as a single display and avoid the task-serration effect observed with two-headed displays, it is crucial to preserve image geometry across displays and minimize any gaps found on the display surface.

When we refer to geometry preservation, we mean that the lengths displayed are scaled representatives of the original image. When this is true, other attributes such as distances, proportions, and relative size will retain their fidelity in the system. Two-headed displays, as supported by MS Windows, for example, do not preserve geometry. Pixels located across display borders are logically adjacent, although on a setup with two monitors these two pixels are separated by the physical gap between the monitors.

In addition to preserving the image geometry the resulting image should be free of gaps. In display systems preserving image geometry, a visible gap between display units results in missing image content. While users are familiar with the fact that displays are finite and that clipping occurs at the display border, clipping inside the display space will typically be disrupting. Gaps within the display area should therefore be avoided.

When users change the angle from which they view the display, surfaces not located in the same physical plane can block portions of each other. This effect can also lead to perceived visible gaps and noticeable misalignment. This can be avoided if head tracking is used. However, this can result in the aforementioned lag in displaying the new images, which in turn distorts the geometry of the images. To avoid the described drawbacks, display units should generally be located in the same plane.

Combining multiple display units

Displays to be combined typically have a certain thickness and borders with certain widths and depths (depth denoting how far the border extends over the display plane). Figure 3 shows ways of combining two coplanar screens to form a single display.

Figure 3a shows a configuration that combines the two displays alongside each other. The benefit of this arrangement is that both display areas are in the same plane. The drawback of this arrangement is that the gap

---

[3] The term *resolution*, measured, for example, in pixels per inch, determines in how much detail image content can be displayed. Note that the term resolution is sometimes misused for communicating the number of pixels offered by a display (e.g. "a resolution of 1024x768 pixels"), which is *not* the meaning we refer to when using the term resolution.

between the displays is at least the sum of the border widths of the two displays. When the display borders are small, this solution works well.

Figure 3b shows a configuration where one display unit is located in front of the other. While this setup does not allow the two displays to be in the same plane, it minimizes the gap between the two displays units. The gap is now only determined by the border width of the front display.

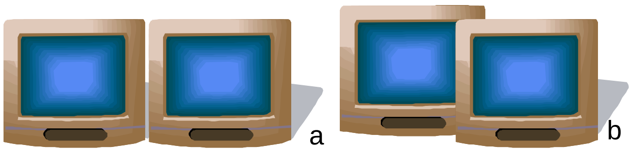

Figure 3: Combining two coplanar display units, such that they are in the same plane (a) or such that the perceived gap is minimal (b).

The potential of the in-front setup shown in Figure 3b is less obvious than of the alongside setup. While the in-front setup looks awkward for two *monitors*, it becomes useful for other types of displays. To minimize the gap and the depth distance, three dimensions have to be minimized, i.e. thickness of the display in front, border width of the display in front, and border depth of the display behind.

For two coplanar displays there exist implementations that fulfill these requirements. This combination is depicted in Figure 4. The shown setup combines a flat screen monitor in the center with a customized projection surface surrounding it. The basic idea behind this setup is that the border of the flat screen monitor is covered with the projection surface, which in turn is thin enough to keep the two display planes very close. Our f+c screen prototype is implemented based on this design, but in order to make the prototype more space-efficient, we used an LCD screen as the focus display.

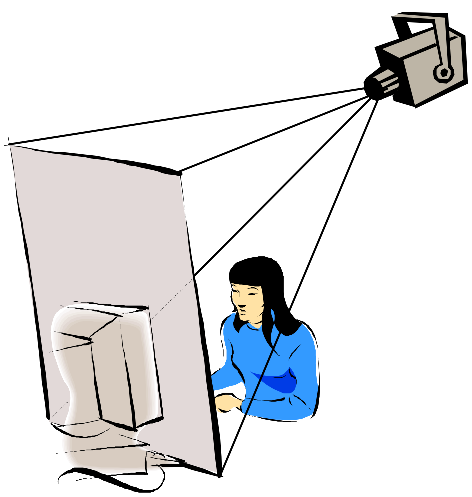

Figure 4: F+c prototype combining a monitor having a flat surface with a projection system

The pixels displayed by the video projector and the LCD are of substantially different sizes, which allows this combination to achieve high-resolution in the center, while simultaneously offering a large screen surface (see Figure 2).

Setting up an f+c screen installation

The setup shown in Figure 4 requires only moderate modification of a regular office workplace and can be built with comparably inexpensive off-the-shelf components. Figure 5 shows how this is accomplished.

Prior to the modification, the office shown contains a regular PC workplace with a Windows PC and an SGI 1600SW flat panel monitor (Figure 5a). The flat panel stays in place and becomes the focus display of our focus plus context installation. To bring the display planes closer together, the flat panel's protruding front cover is removed, making the LCD completely flat.

Next, the customized projection screen, which consists mainly a 3x4 foot (90x120cm) piece of white foam core, is added to our setup (Figure 5b). A hole large enough to accommodate the entire flat panel display allows the flat panel to be embedded within the projection screen. The surfaces of the flat panel display and the projection screen are aligned in the same plane. The installation shown uses an antique golden frame to hold the canvas, which not only allows the projection screen to stand on the desktop while leaning against the wall, but also gives the installation a stylish look. To cover the gap between the two display areas, a paper mask of appropriate size is used to extend the projection surface across the borders of the focus display, thereby creating a seamless display area (Figure 5d).

The projector (initially a portable Sony VPL-XC50, later a NEC MT 1035) fits conveniently in the space behind the user (Figure 5c), which makes this installation very space-efficient. The 8½ feet (approx. 2½ meters) wide office provides enough projection throw to make the projection fill the entire 12 square foot projection screen. Generally, projectors have to be positioned above the user's head to keep the user from casting a shadow on the projection screen (see also Figure 4). One way of accomplishing that is by mounting the projector on the ceiling. In the shown setup, the projector is placed on a shelf on the opposite side of the office (Figure 5c). To avoid keystoning, the projection surface is tilted slightly.

In the configuration described so far, the projector not only projects on the projection screen, but also on the flat panel monitor. While this overlap is key to achieving the desired zero-gap integration of the two display areas, it results in a double image and reflections on the flat panel display. This effect is avoided by placing a black object over the respective part of the projection. For this purpose, a simple program that creates a resizable window is used. After moving the window to the desired position, all window decorations can be removed by hitting the window's *freeze* button, which leaves the window entirely black. This completes the display installation setup.

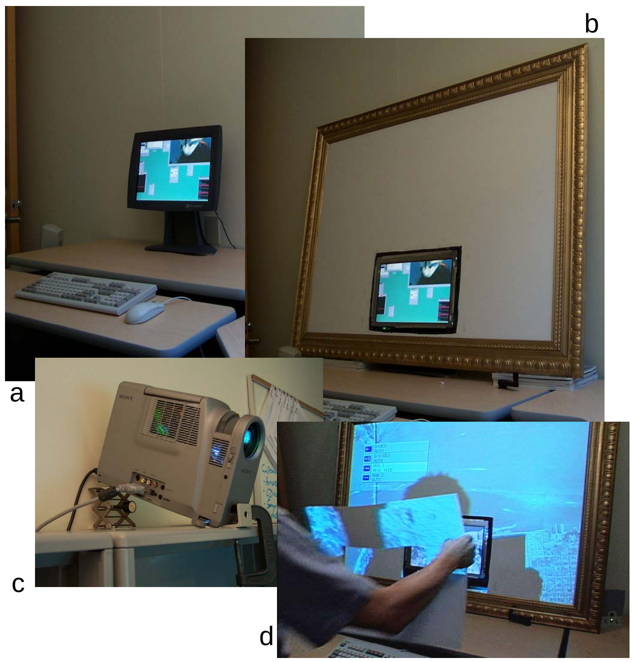

Figure 5: (a) The workplace prior to the modification. (b) The foam core projection screen is placed around the flat panel display. (c) A projector is positioned at the opposite side of the office. (d) A paper mask is added to cover the frame of the flat panel display.

This resulting setup offers a 3x4 feet (90x120cm) display area with a seamlessly integrated hi-res region. The focus display offers 1600x1024 fine pixels, while the context region provides 1024x768 (minus the removed overlap region) coarse pixels. In this particular installation, the resolution in the focus region is 5.1 times bigger than pixels on the focus display, so that each context pixel corresponds to about 26 focus pixels. A large hi-res display with the same surface and the resolution of the focus region throughout the whole display area would have around 20.5 mega pixels.

APPLICATIONS AND TASKS ON F+C SCREENS

Provided with this display, the next step is to adapt applications to it. Before we can understand what applications benefit from a focus plus context screen, we have to understand what the strengths and limitations of this novel display type are. We will analyze f+c screens in comparison to other visualization techniques designed to minimize the need for zooming interactions. Figure 6 shows how f+c screens relate to the visualization techniques mentioned earlier.

Focus plus context screens combine the advantages of overview plus detail *and* fisheyes views. Specifically, f+c screens provide users with the physical continuity of the fisheye *and* the non-distortedness of overview plus detail. The price is that users lose the possibility to zoom and pan the focus independently of the context. The focus region and the context region have become one and the zoom factor between them (e.g. the 1:5.1 ratio in the case of our installation) is now fixed. Besides that, there is the obvious need for additional space for the projection surface and the projector. This limits the applicability of f+c screens to stationary setups and excludes, for example, portable computers and palmtops.

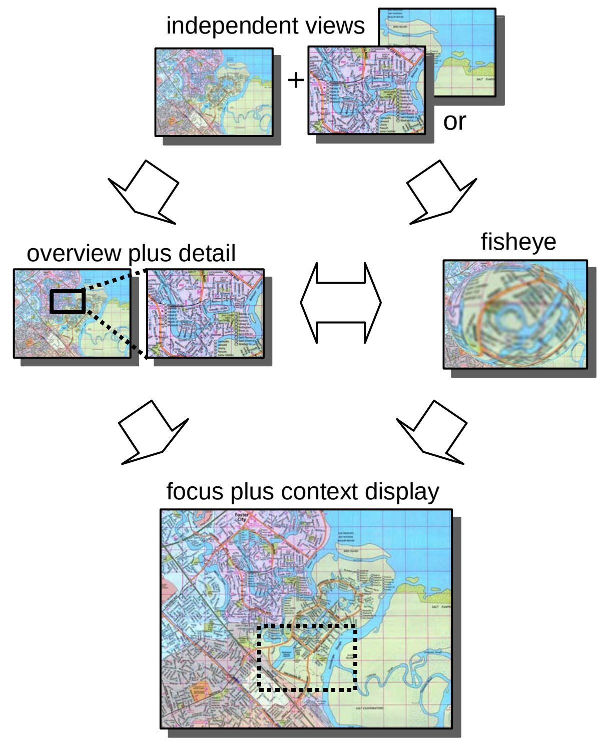

Figure 6: F+c screens compared to other visualization techniques designed for saving zooming interactions.

In order to exploit the benefits of f+c screens, we should apply them to tasks where users switch between the focus region and the context region frequently (because this is where overview plus detail requires users to carry out additional visual navigation) and where the absence of distortion is crucial (because this is where fisheyes do not work properly). How often users switch between focus and context depends on the task. This will be discussed later with the applications we actually implemented. Sensitivity to distortion varies from task to task as well. If the task requires users to compare the lengths of distances or the sizes of surfaces across image regions, distortion makes this task difficult. Additionally, images of real-world objects become difficult to recognize when distorted.

Many types of content fit this model, including representations of cities (street maps and satellite photos), buildings (architectural blueprints, renderings, CAD), integrated circuits and other technical material (drawings or photos), desktops (real and GUI desktops), biological

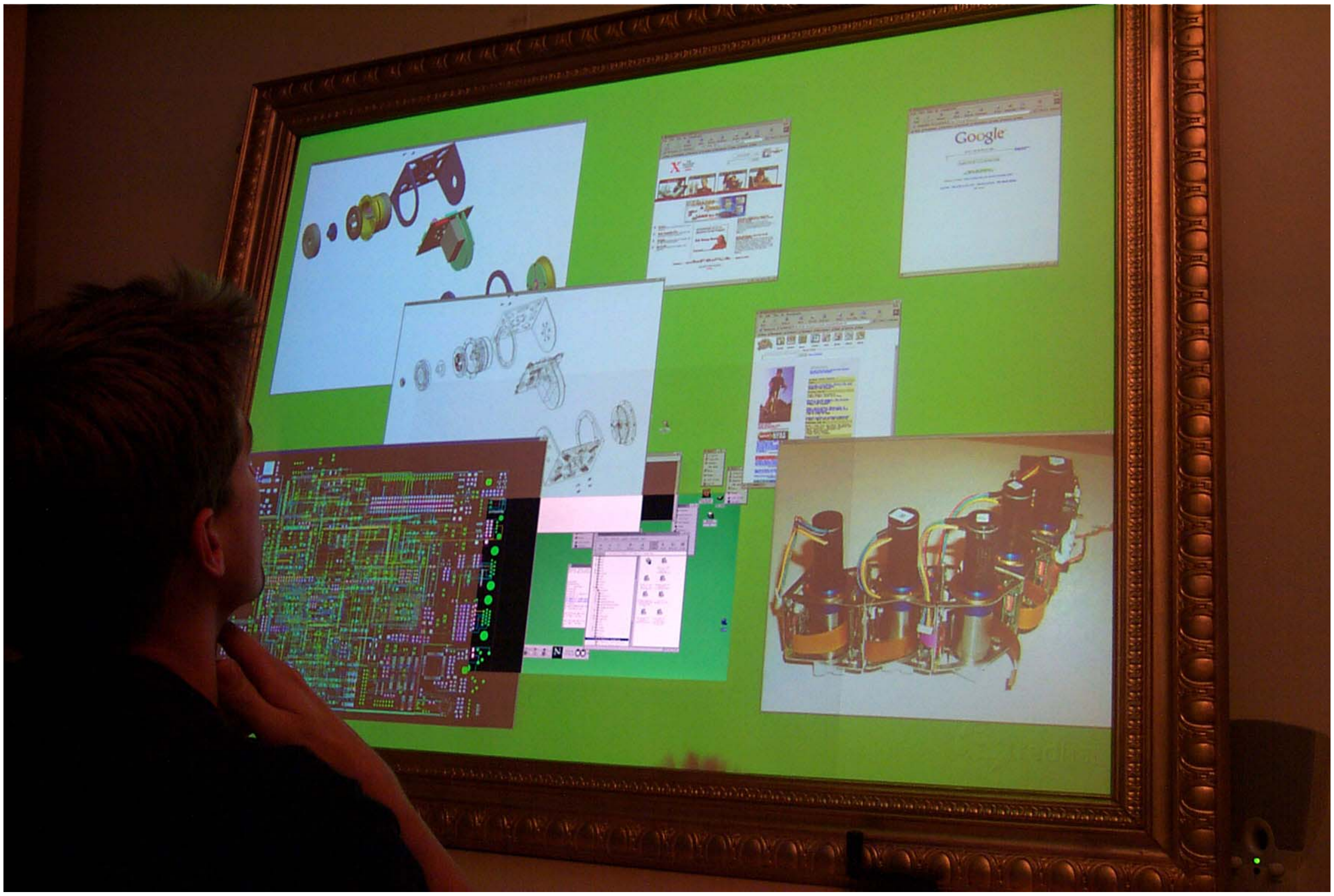

Figure 7: Working with large images and drawings on an f+c screen under Linux.

data (human skin, X-ray images, anatomy, microscopy), star and weather maps, large layouts, designs, and pieces of art (posters, paintings, including zoomable documents [1]).

Note that all these objects can come in very different representations. They can be shown in two dimensions (drawings) or three (renderings), they can be captured optically (photographs) or modeled using a computer (renderings), they can be encoded in a pixel-based (bitmap) or in a vector-based format (drawing), and they can be static (photos) or dynamic (videos, animations). The system displaying them can allow users to browse them in two (drawing program) or three dimensions (VRML editor); they may allow editing (image editor) or not (image viewer).

Applications we implemented

We found the following concrete applications most interesting and therefore implemented f+c screen solutions for them.

Scenario 1 (Editing print products) As mentioned in the introduction, print typically has a much higher resolution than computer displays. To obtain a print product that looks perfect from a distance as well as with a magnifying glass in hand, editors have to work with the print product in many different zoom levels. On an f+c screen (Figure 7), editors of print products can work on details while constantly being aware of each region's context and how detail modifications affect the overall impression. We implemented this scenario on the Linux operating system, using the available image editing and layout applications. As a side effect, the system provides users with large physical space for spreading out information, which Henderson and Card [9] found to be important.

Scenario 2 (Finding the shortest path on a map) When browsing large images, users zoom in and out to inspect objects of different scales, to compare distant objects, and to gain orientation. If the user's task is to find the shortest path from a residential address in one city to a residential address in another city, users need to zoom in to be able to read street addresses, recognize one-way streets, etc. They also need to zoom out to get an overview of the highways connecting the two cities. On an f+c screen (Figure 2), this navigation is simplified because users can constantly see a large region of the map, while simultaneously having access to street-level information in the focus display. Since f+c screens preserve geometry, comparison of distances is straightforward, even across display borders.

Scenario 3 (Videoconferencing and teleteaching) There are many situations where a video presentation simultaneously involves objects of incompatible scales. In our demo scenario shown in Figure 8, a person describes a small robot module she is holding. While it would be difficult to convey the speaker's gestures as well as a detailed view of the robot using the limited resolution of a single TV camera, f+c screens allow all this information to be displayed. On an f+c screen, viewers can simultaneously see the speaker and a detailed view of the object as well as gestures connecting them. An overview plus detail solution involving a separate "document" camera for the object would cause the relation between the speaker's gestures and the presented object to be lost. As a side effect, the large screen of our f+c installation allows the presenter and the objects to be seen at their actual size, which helps the viewer understand the scale of the presented content.

Figure 8: A videoconference partner displayed on an f+c screen. The higher resolution in the focus region allows communicating relevant details.

Scenario 4 (Simulation games) Games that immerse the user in virtual worlds have a single focus of attention. The position of the user's persona in the virtual world determines which objects or game characters are visible, accessible, and potentially dangerous. At the same time, these games often require the user to make decisions that require knowledge of the world around them. In sports games, users have to be aware of the position of other players on the field; in real-time strategy games, users continuously make decisions based on the opponent's activities on a large battlefield. Similar focus plus context effects occur in 3D simulation games, such as the first person shooter Unreal Tournament (http://www.unreal-tournament.com), shown in Figure 9. Users can pick up or shoot objects only when they are in the crosshair section in the middle of the screen. The crosshair never moves, so instead of moving their eyes to objects of interest, users continuously pan objects of interest into the crosshair region. This model causes the user to continuously fixate on the screen center. The f+c screen provides a high-resolution picture in the region surrounding the crosshair, while providing a much larger peripheral image in the context area. The fact that the context area is low-resolution does not affect the user's experience, because human vision in the peripheral regions is also limited to low resolution [18].

For all four scenarios, there are systems that employ overview plus detail techniques. However, since these tasks require users to switch between views frequently, we expect f+c screens to be able to boost user's performance. Also, in all four scenarios, geometry preservation is important, which renders distorting techniques such as fisheye views inappropriate.

Figure 9: F+c screens allow players of 3D games (Unreal Tournament) to perceive their surrounding through peripheral vision.

How we implemented these applications

The Gnome Desktop, running on a *Virtual Network Computing* (VNC) XWindows display (http://www.uk.research.att.com/vnc) is shown in Figure 10. For this setup, a VNC server runs on a remote Linux machine to create a 5228x3921 pixel frame buffer, which is the resolution that the display would offer if it were all hi-res. The two physical displays of the f+c screen are connected to a dual-headed Windows PC. This PC runs two instances of VNC viewer that transfer the content from the VNC server over the network. The context display uses the "scale" option (a feature currently only available in the Windows version of VNCviewer) to scale the frame buffer down by 5.1 (= 97/19 ratio), representing the size ratio between focus and context pixels. Since VNC scales by averaging pixel colors (filtering), the image information is preserved as well as is possible. The resulting 1024x768 pixel-sized window holds the entire virtual desktop and is fully interactive. Dragging this VNC window into the projector display increases the size of its pixels, which compensates for the scaling process. The focus display uses a VNC viewer as well, but without the scaling. The virtual frame buffer is bigger than the physical display, so the VNC viewer provides scrollbars to pan around the image. Using the scrollbars, the focus display is panned until focus and context images line up. Both VNC viewers are now switched to full screen mode, so that the Linux desktop fills the display.

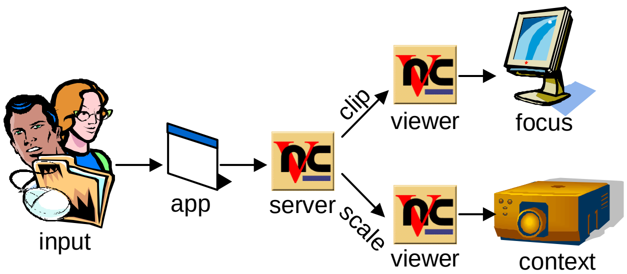

Figure 10: Running Linux on F+c screens via VNC

This setup is fully functional, i.e. it allows running arbitrary XWindows applications. Windows can be dragged around seamlessly across display regions. We successfully ran several applications including Star Office (office productivity application), gimp (image processor), Netscape (web browser), and several Linux tools. Due to several optimization techniques, such as selective refresh, VNC updates window content at a reasonable speed and small windows can be moved in real time. Redrawing a full screen window, however, can require up to a couple of seconds. We are working on improving this by creating a single-machine version. We have also started experimenting with implementations based on a single graphics card that could run both displays. Nonetheless, the fact that the virtual frame buffer is about ten times bigger than a normal PC display limits the achievable speed when applications are stretched to fill most of the screen.

In situations involving panning a full-screen image, a full-resolution bitmap would involve an unwieldy amount of memory. An efficient solution might involve creating different views of the content suited to the individual resolutions and area coverage for each display. If this is supported, each view can be generated directly by a separate application. The navigation on each of the views has to be coordinated in order to preserve geometry of content displayed across display borders. We will use the term *coupled views* to refer to this type of setup. The three f+c implementations shown in Figure 2, 8, and 9 are based on coupled views, which allows them to run at the same speed as they would on a normal PC display.

There are several different ways of obtaining coupled views, e.g. by using applications that allow a single document to be viewed in multiple windows, such as Adobe Photoshop or the Microsoft Office programs. The f+c scenario shown in Figure 2 is based on two image viewers running on different networked machines. Figure 11 shows how this works. The image viewer (ACDsee, http://www.acdsystems.com) uses a "nearest neighbor" approach to zoom images, which would introduce undesired image artifacts. Additionally, the source image for the context display would be larger than necessary. To obtain high-quality output, source images are scaled offline using Adobe Photoshop. A full-resolution focus version and a scaled-down context version of the image are saved to disk. An instance of ACDsee is run on each networked PC; one of them drives the focus display, while the other one drives the context screen. The images are aligned manually and the viewers are switched to full-screen mode.

To allow users to pan within the images, the views must now be coupled. To preserve the image's geometry, both images have to be panned in parallel, but at different speeds. To achieve this, the input from the user is transmitted to both viewer instances and is scaled corresponding to the scaling factor of the bitmaps. In our installation, for example, a 1-pixel pan in the context display accompanies a 5 to 6 pixel pan in the focus image. To accomplish forking and scaling of input events, we wrote the software tool mouseFork. MouseFork receives mouse events from the mouse/trackball device, duplicates them, scales them, and sends them across the net to each display application. MouseFork also replaces the hand tool provided by the individual image viewers with a navigation method that is more convenient for navigating in large images. Panning across large distances with a hand tool requires users to grab the plane (mouse down), pan it (mouse move), release it (mouse up), and to go back (mouse move), to avoid running into the limits of the client screen. MouseFork converts the single stream of mouse move events generated by the trackball into a series of such hand tool operations, so that the user can operate the system by simply rolling the trackball.

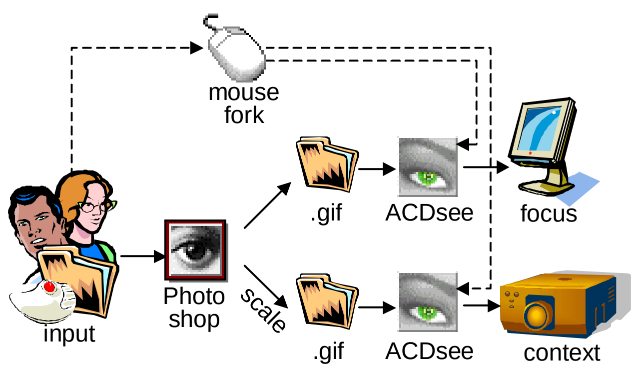

Figure 11: Running an image view at full speed

We used this viewer setup to explore several large images such as a 13,800x12,000 pixel satellite image of San Francisco, a 15,000x15,000 pixel map of London, a 10,000x 30,000 pixel Mandelbrot images, as well as a series of technical drawings, renderings, and circuit layouts. The viewer runs smoothly and there is no perceivable lag between the two display units.

The teleconferencing scenario shown in Figure 8 is still in an experimental stage. It uses forked views as well, which makes it similar to the image viewer setup, but since the user has no means to navigate the shown image, there is no transmission and forking of navigation events. Synchronization between views is achieved by using two piggybacked cameras to perform capture, i.e. the coupling between the two views (panning only, so far) is done mechanically. Note that it is also possible to implement the entire teleconferencing/teleteaching setup entirely with analog technology, i.e. by connecting one analog camera to an analog focus monitor and another one to the analog input of the projector.

The last of our four scenarios, the 3D game shown in Figure 9 is implemented using coupled views, but in this case, the coupling is supported by the game itself. The setup is done in three steps. First, the game allows players to automatically follow another player in a networked game using "spectator mode" and to look through this player's virtual eyes using the command "behindview 0". Using this feature, the view shown on the context display can be synchronized with the view on the focus display. Second, views are calibrated so that their centers (marked with a crosshair) are aligned. This is done by running the game on the projector machine in window mode (instead of fullscreen mode), which allows the window to be moved around until its crosshairs meets the crosshair of the focus display. Third, in order to calibrate scaling, the computer running the context display is given a wider view, by setting its "field of view" variable to a larger value. This completes the setup. While it is possible to run this game on two machines, we used a three-machine setup (one machine for running the game and two "spectators" to generate the views) to better synchronize views. The necessity for this adaptation did not emerge from network lag, but from the fact that Unreal sends update events to spectators only when the player's movements have exceeded a certain tolerance. Using two spectators applies the same lag to both displays and thereby gets them synchronized. The game runs at full speed and is fully playable.

EARLY RESULTS

Our prototype is currently set up in the personal office of one of the authors at Xerox PARC, which allows us to continuously experiment with the system, to demo it frequently, and to let other researchers try it out. Over the past six weeks, we demonstrated the system to about seventy of our colleagues; at least fifteen of them tried it out themselves. We let them experiment with the Linux setup and several applications on top of it (including Star Office, the Gimp, Netscape, etc.), as well as with the image viewer (allowing them to browse a satellite image of San Francisco, a London map, and a fractal). Some of our colleagues also tried out our adaptation of Unreal Tournament. Listed below are some of the impressions of several of our colleagues who used the f+c screen.

The Linux implementation was the first one that we had up and running. It received a lot of feedback and inspired many great suggestions, including the three setups that we later implemented using coupled views. Its support for working with large documents was widely appreciated. Despite the fact that our f+c screen actually displays fewer physical pixels than the two-headed SGI LCD setups that some of our colleagues use, the display was generally perceived as providing "lots of space". The large screen was judged especially useful in combination with the high panning speed of the image viewer, which received a great response especially with the San Francisco satellite image. We adapted the 3D game only very recently, but the few people who tried it described the additional resolution in the focus region as beneficial.

Practically all our testers immediately reflected on how an f+c screen would affect their daily work at PARC. Their feedback varied based on the documents and tasks their daily work involves. The most enthusiastic feedback came from people in our media group (MARS) and in the Research in Experimental Design group (RED). Both groups work with large visual objects, such as posters, design sketches, collections of photographs, etc. Hardware designers also appreciated the display's capability of showing large construction drawings. Two of our colleagues expressed interest in using an f+c screen for managing large websites or network plans. On the other hand, the display generated only limited interest among those users who primarily work with text, especially in program code editing tasks. This feedback is not surprising, when taking into account that regular-sized text becomes unreadable when moved to the context region. Users working with text with an emphasis on layout, such as the people in the media group, however, judged f+c screens as a desirable enhancement of their work environment.

FUTURE WORK AND CONCLUSIONS

In the future, we plan to work in three major directions. We are currently setting up an experiment comparing f+c screens with overview plus detail views with respect to a chip design task. We also plan to experiment with f+c screens in multi-user applications, such as walls or tables with multiple embedded focus displays. Furthermore, we planning improvement to the applicability of f+c screens to text-based applications by applying selected visualization techniques, such as Thumbnails [24].

While large hi-res displays match all the usability characteristics of f+c screens plus offering high resolution throughout the entire screen surface, f+c screens are more feasible. They are more than an order of magnitude less expensive than a comparable 20-megapixel display based on tiled projections. Equally important, f+c screens require substantially less space, which allows them to be set up in normal offices settings. Today, two-headed systems are in common use, but as dropping prices currently bring projectors to the mass market [14], focus plus context screens offer an alternative for users working with visual content, such as designers or architects.

In this paper, we presented a new means for supporting users working with large visual content. By combining visualization techniques with a new type of display setup, f+c screens achieve characteristics expected to outperform existing visualization techniques. While overview plus detail visualizations require users to switch between multiple views, focus plus context screens allow users to *simultaneously* keep track of context information via peripheral vision. Since f+c screens do not distort display content, they are applicable to situations where fisheye views cannot be used.

ACKNOWLEDGMENTS

We'd like to thank the members of RED, especially Mark Chow and Rich Gold, as well as Dale MacDonald who provided us with the wonderful golden frame. We'd also like to acknowledge Brian Tramontana for his efforts in setting up the videoconferencing scenario, Jeff Breidenbach for his Unreal hack, Bill Janssen for the initial idea of using VNC, Mark Stefik and Nola Mae McBain for their valuable comments, and Mark Yim and Dave Duff for providing the modular robot components and the renderings that we used in our demo scenarios.

REFERENCES

1. Bederson, B. B., and Hollan, J.D. Pad++: A zooming graphical interface for exploring alternate interface physics. *Proceedings of UIST' 94*, pp. 17-26. New York: ACM.
2. Butz, A., Hollerer, T., Feiner, S., MacIntyre, B., and Beshers, C. Enveloping users and computers in a collaborative 3D augmented reality, *International Workshop on Augmented Reality 1999*, San Francisco California, October 20-21, 1999.
3. Carpendale, S. A. *Framework for Elastic Presentation Space*, Ph.D. thesis, March 1999, School of Computing Science at Simon Fraser University.
4. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.V., and Hart, J.C. The CAVE: audio visual experience automatic virtual environment. *Commun. ACM* 35(6): 64–72, June 1992.
5. Feiner, S. and Shamash, A. Hybrid user interfaces: breeding virtually bigger interfaces for physically smaller computers. *Proc. of UIST' 91*. pp. 9–17, 1991.
6. Funkhouser, T., and Li, K. On the wall large displays. *IEEE Computer Graphics & Applications*, 20(4), July/August 2000.
7. Furnas, G. Generalized fisheye views. *Proceedings CHI 1986,* pp. 16–23, 1986. New York, ACM, 1986.
8. Grudin, J. Partitioning Digital Worlds: Focal and peripheral awareness in multiple monitor use. *Proceedings CHI 2001*, pp. 458-465.
9. Henderson, D. A., and Card, S. Rooms: The use of multiple virtual workspaces to reduce space of contention in a window-based graphical user interface. *ACM Transactions on Graphics*, 5(3):211-243, 1986.
10. Hereld, M., Judson, I., Stevens, R., Introduction to Building Projection-based Tiled Display Systems *IEEE Computer Graphics & Applications* Vol. 20, No. 4, July/August 2000
11. Hinckley, K., Czerwinski, M., and Sinclair, M. Interaction and modeling techniques for desktop two-handed input. *Proceedings of UIST '98*, pp. 49–58, 1998.
12. Holmquist, L. E. Focus + context visualization with flip zooming and the zoom browser. In *CHI '97 Extended Abstracts* (Atlanta GA, Mar 1997) ACM Press, 263-264
13. Hornbaek, K., Frokjaer, E. Reading of electronic documents: the usability of linear, fisheye, and overview+detail interfaces. *Proceedings of CHI '01*, pp. 293-300, 2001.
14. Lieberman, D. Projectors posed as focal point of networked home. *EE Times* 03/22/01. Available online at http://www.eetimes.com/story/OEG20010321S0055
15. Menendez R. G., and Bernard J.E. Flight simulation in synthetic environments. *Proc. 19th Digital Avionics System Conference*, pp. 2A5/1-6 vol.1, 2000.
16. Olston C., and Woodruff. A. Getting Portals to Behave. *Proc. Information Visualization. 2000*, Salt Lake City, Oct. 2000, pp. 15-25.
17. Plaisant, C., Carr, D., Shneiderman, B., Imagebrowsers: Taxonomy and design guidelines, *IEEE Software*, 12,2 (march 1995), 21-32
18. Rayner K., Pollatsek A. Eye movement and scene perception. *Canadian Journal of Psychology*, 46:3:342-376, 1992.
19. Rekimoto, J. and Saitoh, M. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proceeding of CHI '99*, pp. 378–385, 1999. New York, ACM, 1999.
20. Robertson, G., and Mackinlay, J. The Document Lens. *Proceedings of UIST '93*. 1993, pp. 101 – 108.
21. Shneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Third edition. Reading MA: Addison-Wesley, 1998.
22. Stone, M.C., Fishkin, K., and Bier, E.A. The movable filter as a user interface tool. *Proceedings of CHI '94*, 1994, pp. 306 – 312.
23. Streitz, N.A. Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P, Seitz, P., and Steinmetz, R. I-LAND: An interactive landscape for creativity and innovation, In *Proceedings of CHI '00*, pp. 120–127, New York, ACM, 1999.
24. Woodruff, A., Faulring, A., Rosenholtz, R. Morrsion, J., and Pirolli, P. Using thumbnails to search the Web *Proceedings of CHI '01*, 2001, pp. 198 – 205.