Lessons Learned from the WILD Room, a Multisurface Interactive Environment

Michel Beaudouin-Lafon
LRI (Université Paris-Sud & CNRS) / INRIA
Bâtiment 650, Université Paris-Sud
91405 Orsay Cedex, France
mbl@lri.fr

RESUME

Créer la prochaine génération de systèmes interactifs nécessite de développer des plateformes expérimentales permettant d'explorer de nouvelles formes d'interaction dans un contexte réel. Cet article décrit WILD, un environnement haute performance destiné à explorer l'interaction multisurface et constitué d'un mur d'image ultra-haute résolution, d'une table multitactile, d'un système de suivi de mouvements et de divers dispositifs mobiles. L'article décrit l'approche de recherche intégrative suivie par le projet et les leçons tirées du point de vue du matériel, de la conception participative, des techniques d'interaction et de l'environnement logiciel.

MOTS CLES

Recherche intégrative, Interaction multi-surface, Mur d'image

ABSTRACT

Creating the next generation of interactive systems requires experimental platforms that let us explore novel forms of interaction in real settings. This article describes WILD, a high-performance environment for exploring multi-surface interaction that includes an ultra-high resolution wall display, a multitouch table, a motion tracking system and various mobile devices. The article describes the integrative research approach of the project and the lessons learned with respect to hardware, participatory design, interaction techniques and software engineering.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Miscellaneous

General Terms

Human Factors

Keywords

Integrative research, Multi-surface interaction, Wall display

IHM'11, October 24-27, 2011, Sophia Antipolis, France
Copyright © 2011 ACM 978-1-4503-0822-9/11/10 ...$10.00.

1. INTRODUCTION

The ecosystem of interactive devices and software that we use today is based on the seminal work conducted at Xerox PARC in the 1970s that led to the first graphical workstation, the Star [17]. A key success factor was the integrative nature of the work: it included the design, testing and assembling of both hardware and software, from individual components to a complete workstation, including the operating system and all of the application software.

I argue that a similar approach is required if we are to (re)invent the next generation of interactive computing. Although 'ubiquitous computing' [20] has been envisioned for many years as a way to move interaction beyond the desktop, many of its critical promises have not yet been fulfilled. For example, while users have access to a plethora of devices and services, these do not interoperate seamlessly but instead act as independent silos.

Moving beyond the desktop requires a complete rethinking of every aspect of today's interfaces because it challenges all the assumptions of traditional desktop computing: multiple devices, multiple users, mobility, etc. While HCI research has addressed these issues individually, through proofs of concept and point designs, I believe that a more holistic approach is also required [1].

For the past two years the In Situ lab has been developing an interactive environment called the WILD room, dedicated to small groups of users who need to interact with large amounts of complex data. (WILD stands for Wall-sized Interaction with Large Datasets.) The rest of this article describes the integrative research process that we have followed, from the initial design exploration with our target users to the choice of hardware, from the empirical studies and controlled experiments that we ran to explore new interaction techniques to the software architectures and applications that we developed for our users. In each section I emphasize the key challenges and lessons learned. Beyond the individual results however, my goal is to show the value of such integrative research.

2. INITIAL DESIGN EXPLORATION

From a user perspective, the key challenge of the WILD room is how to support seamless interaction in a rich environment featuring multiple display surfaces and interactive devices. In order to motivate and test our designs, we chose a complex and demanding application area: scientific discovery. Scientists, with their powerful instruments, massive computer simulations and global databases are at the forefront of computer technology. As *lead users* [19], they face data management problems long before most other users and are eager to explore new solutions such as the WILD room. Working with scientists can therefore give insights into how such technology may be used before it becomes mainstream.

Eight research laboratories from the Saclay campus expressed interest in WILD to help manage their massive quantities of data. We expected that their primary use would be to display very large images: telescope imagery for astrophysicists, particle detector data for physicists, huge molecules for chemists, biochemists and biologists, detailed brain imagery for neuroscientists, complex simulations of 3D phenomena for mechanical engineers and applied mathematicians. However, after running a needs assessment workshop in which each partner presented both their data and their processes for analyzing it, a richer picture emerged.

2.1 Strategies for Managing Complex Data

We identified four different strategies for the use of large surfaces in the process of scientific discovery. First, some scientists need to display data so massive that it exceeds the screen resolution of even a large wall display. For example, astrophysicists work with images as big as 400,000 pixels wide, while biochemists explore the details of the interactions between molecules, some with several hundred thousand atoms, at the atomic level. Thus, even a high-resolution wall display requires the ability to navigate full-size images.

Second, some scientists find large display surfaces ideal for comparing different data sets of the same phenomenon. For example, neuroscientists examine similarities and differences among brains. They want to examine several dozen or even hundreds of brain scans side by side, with the ability to rearrange them and manipulate them independently, facilitating identification of features difficult to detect in any other way.

Third, and most common, large display surfaces can help aggregate heterogeneous data about a specific phenomenon. For example, biochemists and biologists look simultaneously at molecular models, graphs showing the results of various experiments, existing publications, personal notes, web sites, on-line databases, and more. Large surfaces let them organize vast collections of information with minimal overlap and provide a physical space to organize group work: in some phases, each user works on a separate part of the room to find data to aggregate; in other phases they consolidate this data so that everyone can discuss it.

Because this process is very fluid, it is difficult for the scientists to come to the room with all the data that they need. As the session progresses they need to bring in new data and documents from their laptops or through the Web. Thus, aggregating heterogeneous data on walls and tables also requires the ability to import data from laptops brought by participants. At the end of the session, they must be able to create a snapshot of what they have collected to continue working in their lab or the next time they use the room.

The fourth recurring pattern we identified was to bring remote colleagues into a session. The scientists often mentioned the need to discuss research with distant colleagues, whether at their desks or in a similar room, and viewed this as an essential aspect of the process of scientific discovery.

2.2 Key Findings

The work with scientists as lead users gave us important insights into how WILD could be used effectively. It must support *navigation* of large datasets, *comparison* among instances of a similar phenomenon, *aggregation* of heterogeneous content about a single phenomenon, and *communication* with remote participants, either at their desk or at a similar platform. These insights drove much of the research and development described in the following sections.

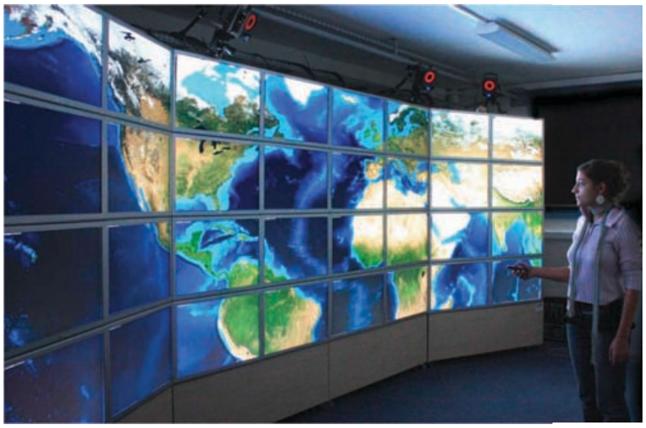

Figure 1: The WILD wall display.

3. THE WILD ROOM: HARDWARE

The hardware of the WILD room consists of a large, ultra-high resolution wall display powered by a computer cluster, a motion-tracking system, a multitouch table and a variety of hand-held devices. The combination and simultaneous availability of such interactive surfaces lead naturally to the concept of multi-surface interaction [2]: the ability to seamlessly combine several devices in a single interaction as in, e.g., Rekimoto's Pick-and-drop [16].

The main challenge for selecting the hardware was to create a platform that addressed our users' needs. We decided to use off-the-shelf components, both for budget reasons and to ensure higher reliability, while offering a variety of interaction surfaces to test as diverse a set of scenarios as possible. By combining a large vertical surface (the wall display), a medium-size horizontal surface (the table), and a variety of portable devices (iPodTouch, iPad, laptops), WILD encompasses a wide range of usage scenarios. The potential limitations come from having a single wall and a single table and not supporting remote collaboration.

3.1 Ultra-high Resolution Wall Display

The goal of the wall display is to provide a display surface that can accommodate a very large quantity of easily accessible information. The wall is made of 32 off-the-shelf 30-inch displays with a resolution of 2560x1600 pixels each, organized in an 8x4 matrix (Fig. 1). The physical display size is 5.5m x 1.8m and the overall resolution 20480 x 6400 pixels (about 131 million pixels).

Four wheeled carts support eight displays each. This makes it possible to create non-flat walls, such as the three-panel configuration in Fig. 1. When the wall is flat, the top and bottom rows can be oriented towards the user. Each display can also be adjusted individually so that the whole wall can approximate a curve. While this flexibility allows us to test various layouts, reconfiguring the wall takes several hours to a day, including the recalibration of the motion tracking system.

One downside of using off-the-shelf displays is the size of their bezels, which creates the so-called "French-door" effect. While some displays now exist that have bezels a few millimeters wide, they do not have sufficient resolution. We therefore traded off reso-

Table 1: Resolution and density of 3 wall displays

	Projection	Thin bezel	30” display
Display res.	1024 x 768	1366 x 768	2560 x 1600
# displays	6 x 3	5 x 3	8 x 4
Physical size	5.2m x 2m	5.1 x 1.7m	5.5m x 1.8m
Resolution	6144 x 2304	6830 x 2304	20480 x 6400
Pixel density	30 dpi	34 dpi	102 dpi
# pixels	14.1 Mpixels	15.7 Mpixels	131 Mpixels

lution for bezel size, with the expectation that high-resolution thin-bezel displays would eventually become available.

Table 1 compares the size and resolution of three walls of similar size: the projection-based wall at the University of Toronto DGP, a wall made of thin-bezel LCD panels (NEC X461UN), and the WILD wall, made of 30” displays. The ultra-high resolution of WILD makes a huge difference, which can only be experienced *in situ*. It creates unique affordances for navigating and interacting with complex content: unlike lower-resolution walls where individual pixels are visible from a distance, WILD keeps revealing detail as the user approaches it. This encourages users to move forward for more detail and backward for an overview.

This affordance in turn reveals a limitation of our design: the displays are not touch-sensitive. Multitouch solutions compatible with ultra-high resolution displays were not available when we built the room, and current solutions entail even wider bezels. The tradeoff was to use the motion-tracking system (see below) to emulate touch interaction. This was further motivated by the fact that remote interaction was absolutely necessary to interact with the wall at a distance, whereas providing only touch capabilities would restrict the range of possible interactions.

The 32 displays are run by a cluster of 16 computers and a front-end interconnected by a 10Gb/s network. Each computer (Apple Mac Pro with 2 x 3.2GHz quadricore processors) has 10Gb RAM (16Gb for the front-end) and 2Tb of local storage. After testing several configurations, we found that a good compromise between performance and cost was to fit each computer with two high-end graphics cards (NVidia 8800GT), each driving a single display. However, we discovered later that the drivers do not support full-screen mode on both displays in this configuration, which complicates programs and affects performance. So a better choice may be to use a single card per computer and either 32 computers with one display per card or 16 computers with two displays per card.

3.2 Multitouch Table

The IntuiFace multitouch table is a medium-size horizontal interactive surface (Fig. 2, left). The resolution of the display is 1920x1080 for a physical size of 100cm x 56cm, i.e. a pixel density of 49dpi (half that of the wall). The table uses FTIR [5] to track up to 32 simultaneous contact points. It also features an RFID tag reader, which makes it possible to tag physical objects and use them as input. For example, bringing a tagged physical document next to the table can display its PDF version.

We observed that users tended to ignore the table during the design sessions. Obviously, the wall dominates the room due to its large size, but two other factors are also at play. First, the low pixel density of the table makes it less attractive than the wall for displaying detailed information, especially since touch-based interaction forces users to be close to the table when using it.

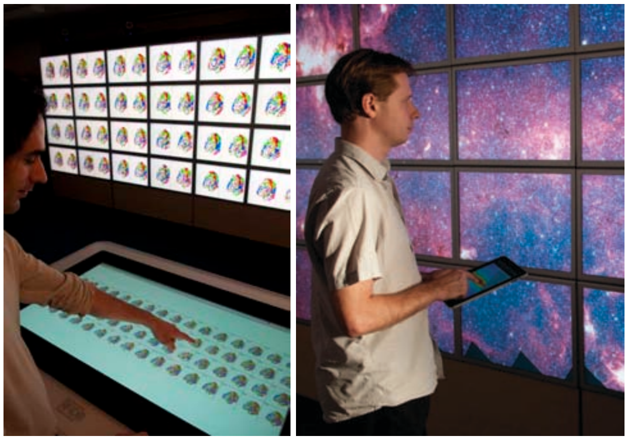

Figure 2: Multitouch table (left) and iPad (right).

Second, users must stand rather than sit around the table because of the projector and camera underneath. Standing in the room during long sessions is tiring, and a flat-screen table would be a better solution. We plan to add such a table to the WILD room in the near future. Having a second table will also support collaborative interaction across two tables, which should better distribute interaction between the wall and the tables.

3.3 Motion-Tracking System

Interacting in the WILD room requires the ability to track devices and people in a fairly large space with high accuracy and minimal lag. We wanted to avoid solutions that required tethering and settled for a camera-based solution.

We use a VICON motion tracking system with 10 infrared cameras that track passive retroreflective markers. The real-time position of the tracked objects is accessible by program over the network. Some cameras are visible above the wall (Fig. 1, 7) and markers are visible on the plastic brain (Fig. 3) and glove (Fig. 5).

Once properly calibrated and under controlled lighting conditions, the system exhibits sub-millimeter accuracy, short lag and good reliability. However, it is not really usable under daylight conditions and it is quite sensitive to spurious reflections caused by shiny objects such as jewelry (or bald heads). Calibration can also be cumbersome; laying out the markers properly to avoid occlusion and ensuring proper recognition is a process of trial-and-error. Overall, we found that the VICON system was adequate for an experimental platform but would not be usable in a real setting.

3.4 Mobile Devices

The WILD room features several mobile devices: PDAs (iPod-Touch), tablets (iPad, see Fig. 2, right), 3D input devices (Logitech Gyromouse) and tangible objects such as sticks, gloves or caps that are tracked by the motion tracking system. The motion tracking system can also track PDAs and tablets, as this is much more accurate than the sensors embedded in these devices. Users can also bring their own laptops and connect them to the wireless network.

We wanted to experiment with various devices and make it easy to add a new device, replace one device with another and create new devices by assembling existing ones. Since the variety of protocols (USB, Bluetooth, WiFi) and operating constraints make it difficult to make such changes transparent to the application software, we had to develop a dedicated input server (see section 6).

3.5 Key Findings

The WILD room is unique in that it combines an ultra-high resolution wall display and multitouch table with rich interaction capabilities. While larger walls exist that use 30" displays, such as the HIPerSpace[1] wall at UCSD and the HIPerWall[2] at UC Irvine, their interactive capabilities are extremely limited. However, it is clear that for this technology to become mainstream, new technology such as thin-bezel multitouch displays, flat-screen high-resolution tables and better motion tracking systems are required.

Using off-the-shelf components allowed us to create a functional platform in a short time, with minimal trade-offs and at reasonable cost (about 230k€). Scaled-down versions can be built without sacrificing too much performance or flexibility. For example, a platform with a 3x4 or 3x5 wall display, a fixed support structure, a mid-level motion tracking system and a smaller cluster where each computer drives more screens would cost less than 50k€.

4. EMPIRICAL STUDIES WITH USERS

Once the platform was functional, we engaged with a subset of the partner laboratories to conduct participatory design sessions and empirical studies. The goal was to explore interaction in the WILD room and to determine whether generic techniques, such as drag-and-drop in a desktop environment, would emerge.

4.1 Using Personal Devices for Interaction

W. Mackay ran the first workshop with neuroscientists. The author had created a prototype application called *WebWall* that displayed a grid of 2D brain scans and supported simple operations such as exchanging two images. This is a case where the bezels helped organize data because of the grid-like nature of the display.

The scientists wanted to interact both with individual images and with all of them in parallel. In particular, the actual brain images that they use are 3D models, and they need to control the orientation of all the brains simultaneously in order to see them all from the same angle. During the workshop, one scientist spontaneously came up with the idea of using a physical model of the brain to control the orientation of the images (Fig. 3), similar to Hinckley's props [6]. A follow-on idea used a stick to designate a part of the brain and to highlight the corresponding structure in all the images.

The idea of using physical props or handheld devices to manipulate data on the wall or table came up in other workshops. For example, astrophysicists used a handheld tablet (Fig. 2, right) to navigate a large image and control display parameters.

While the use of such personal devices may be due to the fact that the wall is not touch-sensitive, our observations show that there are other reasons as well. First, users move around in the room and are often away from the wall and the table: Carrying one's interface allows remote interaction in such situations. Second, using a personal interface avoids cluttering the shared display space and occluding parts that others may be using. Third, a personal device can carry a personalized interface as well as private data, supporting both customization and collaborative work.

4.2 A Real-World Experiment

We conducted an empirical study to test whether WILD could be useful for real-world tasks. Since developing an application that our lead users could use for their real work was not yet possible, we decided to conduct a first experiment with ourselves as users.

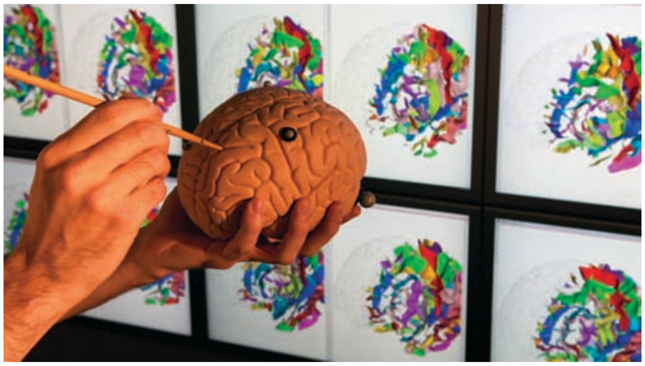

Figure 3: Interacting using a physical prop.

W. Mackay and the author were co-chairing a sub-committee of the CHI 2010 conference. This involved assigning about 140 articles to the 13 members of the subcommittee. We both had experience with this process and had carried it out using paper printouts in a large room, or on-line using a Web-based interface, or some combination of both. The task typically takes two hours for the initial assignment for a conference of this size.

We modified the *WebWall* software introduced above to display the summary pages of the submitted papers. We used an iPodTouch as a laser pointer to select pages; buttons on the screen of the iPodTouch allowed us to pick a page, drop it in an empty slot, or exchange it with another page. We configured *WebWall* so that the top row displayed a page for each committee member, including his or her name, keywords and conflicts of interest. To the right of the wall was the "pile" of papers: each time a page was picked up from the pile, the next paper would be displayed, until the pile was empty. Assigning a paper simply consisted of moving it to the column under the corresponding committee member.

Rather than assigning the papers one by one, we quickly found that it was more efficient to drag several papers from the pile and place them in the available space in the middle of the wall, assigning them from there. We also realized that stacking the papers from the bottom of the screen resulted in a visual histogram of the load of each committee member, which proved very useful in balancing the numbers and types of contributions.

We took about one hour to make the complete assignment, or about half the time it took, in our experience, with a traditional process. This was also a much more satisfying experience, because we had a clear overview of the state of the assignments. We have several ideas for improving *WebWall* for this task. For example, it would be helpful to display conflicts between a paper and a committee member by checking author names and affiliations, or by marking the conflict by hand. Similarly, it would be helpful to mark tentative assignments that the user is willing to change.

Overall, this experiment demonstrated that WILD can improve real-world tasks and support new ways of working. A next step will be to conduct similar real-world tasks with scientists, once the applications are robust enough for real-world use of the platform.

4.3 Key Findings

We were pleasantly surprised by our users' level of engagement and willingness to explore novel interaction techniques unlike any

[1] `http://vis.ucsd.edu/projects/hiperspace`

[2] `http://hiperwall.calit2.uci.edu/`

they had ever used before on a computer, but still natural with respect to their work. The use of participatory design clearly increased our understanding of the users' needs while enabling us to benefit form their innovations.

We also found that personal devices are often more appealing than direct interaction with the wall. In a collaborative context, users seem to feel more comfortable using a personal device to interact with shared data. Also, the size of the room encourages both moving around and interacting at a distance. This approach looks very promising for creating a general interaction model for such environments. In particular, these studies support the use of instrumental interaction in a multi-surface environment [8].

5. INTERACTION TECHNIQUES

The user studies described above showed that interaction in the WILD room is characterized by users standing, walking around and working collaboratively as well as individually. They need to interact with display surfaces from a distance as well as from up close, transfer information from one surface to the next, organize the content of the displays and interact with individual applications. Such use is quite different from that of a single user in front of a personal computer with a mouse and keyboard. The basic interaction techniques used in today's interfaces must therefore be completely redesigned and re-evaluated for these new environments.

Many researchers, e.g. [6, 16, 18], have studied interaction techniques for wall displays, multitouch tables, tangible interfaces and multiple surfaces. However the unique features of WILD, in particular the ultra-high resolution of the wall, require further work. We began by studying two essential tasks: reaching a target and navigating content with pan-and-zoom while standing away from the wall.

5.1 High-Precision Distant Pointing

The first study (Nancel et al. [11]) examined the problem of pointing on a wall-sized display. Reaching a target on the WILD wall requires combining high precision with a large amplitude: the maximal theoretical Fitts Index of Difficulty is $ID_{max} = log_2(1 + 20480/1) = 14.3$ bits compared to about 10 bits on a typical desktop display. Moreover, it must be possible to point while standing away from the display, not only because the wall does not have touch capability, but also because users move in the room and need to access content from a distance.

A number of techniques have been developed for distant pointing. The simplest, laser pointing [13], is not practical because hand tremor is amplified by the distance to the screen, reducing precision. A more recent technique [18] combines two modes: a coarse mode similar to laser pointing and a precise mode where hand movements are interpreted in a relative way, as with a mouse, to precisely control the cursor when close to the target. The smallest target size tested with this technique was 1.6 cm, or 64 pixels on the WILD wall. At the typical distance of 1 to 2 meters from the wall, this is still a fairly large target, about twice that of a typical icon. The goal was to improve on this technique.

The first experiment compared the standard absolute technique, laser pointing, with two relative techniques based on a gyroscopic mouse (a mouse that senses angular movements in mid-air): one used a linear gain, the other a dynamic gain based on the device's angular velocity, similar to the mouse acceleration on a desktop. Laser pointing was the worst technique ($\mu = 3.9s$) and the gyroscopic mouse with dynamic gain was the best ($\mu = 3.1s$). The limit of the latter (in terms of reasonable pointing time and error rate) was about 7mm (28 pixels) at a distance of 2m from the display, which makes it adequate for many tasks. A problem however is that the gyroscopic mouse requires clutching. Since the mouse is held in mid-air, the user must use a button, which makes the technique less intuitive than a traditional mouse.

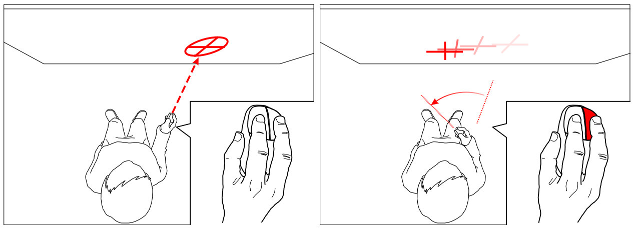

Figure 4: Dual-mode pointing: absolute (left), relative (right).

The second experiment compared the best technique from Exp.1 with a combination of laser pointing for approaching the target and a relative technique for precision pointing. Three relative techniques were tested: 2D motion (similar to [18]), angular motion using a gyroscopic mouse, and touch input on a handheld trackpad. All three dual-mode techniques outperformed the gyromouse from Exp.1 for the most precise tasks (4mm target). Overall the combination of laser pointing and angular relative pointing (Fig. 4) had marginally better performance and was preferred by users.

While these techniques are better than the state of the art, the smallest target size (4mm = 16 pixels) is still quite large when compared with the resolution of the screen. Also, the use of two buttons, one for mode-switching and the other for clicking, is cumbersome. Finally, dual-mode techniques create a recalibration problem: When in relative mode, the user moves the pointing device so that when switching back to the absolute mode (laser pointing), the cursor jumps far away from the original position, requiring the user to reacquire it. Further work should address these limitations to create an efficient and intuitive pointing technique for the wall.

5.2 Mid-air Pan and Zoom

The second study (Nancel et al. [12]) addressed the problem of navigating through very large images using panning and zooming techniques. Pan-and-zoom has been thoroughly studied on desktop interfaces, however the only previous work applicable to the WILD room is Cyclostar [10], which uses elliptical movements to control panning and zooming in a single gesture.

Rather than trying to develop a single technique, Nancel et al. [12] explored the design space for pan-and-zoom techniques that can be used while standing in front of a wall display. Each technique is a unique combination of values from the following three factors:

- *Unimanual vs. bimanual interaction*: In unimanual conditions, the dominant hand controls both panning and zooming whereas in bimanual conditions, the dominant hand controls panning and the non-dominant hand controls zooming;
- *Linear vs. circular gestures for zooming*: Linear gestures are more intuitive but may require clutching whereas circular gestures can be performed continuously;
- *Guidance provided for zooming*: 1D (mouse wheel for linear gestures, iPod-like scroll wheel for circular ones), 2D (touch surface of a handheld device), or free-hand.

Fig. 5 illustrates a *bimanual*, *linear*, *1D* technique in this space: the non-dominant hand controls zooming using a mouse wheel while the dominant hand controls panning. In the *unimanual* version of this technique, the mouse is held in the dominant hand; one button is used for panning and the wheel is used for zooming. For *circular* gestures, the mouse wheel is replaced by an iPod-like scroll wheel. In the *2D* version of this technique, the mouse is replaced by a handheld device with a touchscreen on which the user performs the linear or circular zooming gestures. Finally, in the *free-hand* version of this technique, the user's hand gestures are recognized according to their trajectory and orientation, leading to an interaction similar to that seen in movies such as "Minority Report".

The twelve techniques that were tested can be classified into four groups with respect to overall performance [12]. The fastest techniques ($\mu = 8.1 - 8.4s$) were the bimanual techniques with 1D or 2D guidance and linear motion (as in Fig. 5). The next group of four techniques ($\mu = 9.2 - 9.5s$) included unimanual versions of the top performers, which is interesting in cases where bimanual techniques cannot be used. The other six techniques ($\mu > 11.3s$) performed poorly. It was especially clear that while the free-hand gestures seemed appealing, they were less efficient (except for bimanual linear) with higher levels of fatigue and poorer accuracy.

5.3 Key Findings

When designing an integrated environment, it is important to note that the best technique is not necessarily the most efficient. For pan-and-zoom, the most efficient techniques are both two-handed techniques, which may not be practical for tasks in which the user carries an object or the second hand is needed for other interactions. Similarly, for pointing, recalibrating or using extra buttons to switch modes incurs costs that may not be worth the benefits. It is thus essential to conduct experiments that consider the context of use, rather than simply proclaiming "which technique is best" [9].

6. SOFTWARE FOR THE WILD ROOM

Developing applications for the WILD room is a challenge, as it requires supporting graphical rendering and interaction across multiple surfaces in a distributed environment: The wall is run by a computer cluster, the VICON tracking system and the multitouch table each have their own computer, mobile devices such as the iPodTouch and iPad are themselves full-fledged computers, and computers brought by users must also be supported.

Chromium[3], Equalizer[4] and CGLX[5] support high-performance cluster-based rendering and can drive wall displays, but we found them difficult to use. They also require major rewriting of applications that are not based on OpenGL. A more lightweight solution is needed that can also reuse existing applications. SAGE[6] is closer to our needs, but its interaction support is poor and it only displays the output of existing applications using pixel scaling, which does not take advantage of the ultra-high resolution of WILD.

We also looked into ubiquitous computing software for distributed interaction but found only experimental systems that were either unavailable, not well documented, or not adapted to the platform. The closest was iROS [7], developed for Stanford's iRoom.

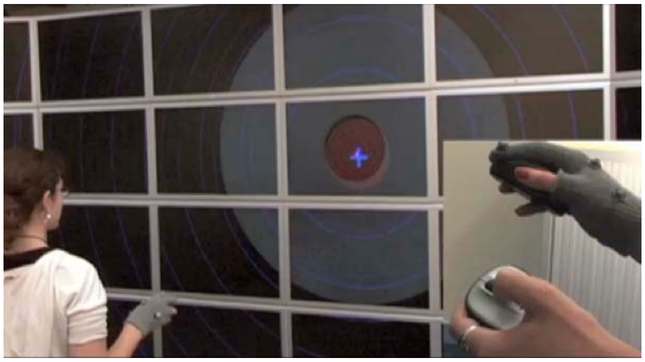

Figure 5: Bimanual pan-and-zoom using linear gestures.

6.1 Early Experiments

The original goal was to reuse existing software, but we quickly realized that we must develop our own. We began with two simple applications to support participatory design sessions with users (and give demos): J. Eagan developed *PhotoTreillis*, a slideshow application for displaying full-screen images and the author created *WebWall*, a grid of windows each displaying a web page.

In both cases, a display client is replicated on each computer in the cluster. A master program running on the front-end controls these clients as well as user interaction. An iPodTouch tracked by the VICON is used as a laser pointer to select objects on the wall. The iPodTouch runs OSCremote[7], an iOS app that lets users create interfaces with buttons, sliders and 2D touch areas and then attach to them the OSC[8] messages that are sent to the master program.

Based on these experiments, we created two middleware frameworks to develop applications for WILD: *jBricks* and *Substance*. They follow the architecture outlined above, except that input is mediated by a separate component, the WILD Input Server.

6.2 jBricks

Pietriga et al's *jBricks* [15] combines the ZVTM zoomable interface toolkit [14] with the WILD Input Server. ZVTM supports vector-based graphics organized into a scene graph. Cameras are nodes in the scene graph that specify what to display. Modifying camera settings, e.g. for zooming, generates smooth animations. ZVTM was ported to the wall by running replicas on each cluster node and controlling the camera of each replica in a coordinated way through a master application. With this approach, scenes with up to 200,000 objects can be displayed at interactive rates.

S. Huot created the WILD Input Server using ICon [3], a tool that manages interaction between input devices and application through *input configurations* that can be created using a visual editor. ICon was extended to support OSC input from devices such as the iPodTouch and OSC output to applications. It proved extremely valuable in prototyping sessions as it makes it possible to dynamically change the devices and configurations while the application is running. Fig. 6 shows a configuration for one of the pan-and-zoom techniques described earlier. It combines touch input from an iPodTouch with location input from the VICON motion tracker. When the VICON is not available, e.g. when developing in one's office,

[3] http://chromium.sourceforge.net/

[4] http://www.equalizergraphics.com/

[5] http://vis.ucsd.edu/~cglx/

[6] http://www.sagecommons.org/

[7] http://www.nr37.nl/OSCRemote/

[8] OSC (http://opensoundcontrol.org/) is a lightweight network protocol used in music and multimedia. Client libraries exist in a variety of languages.

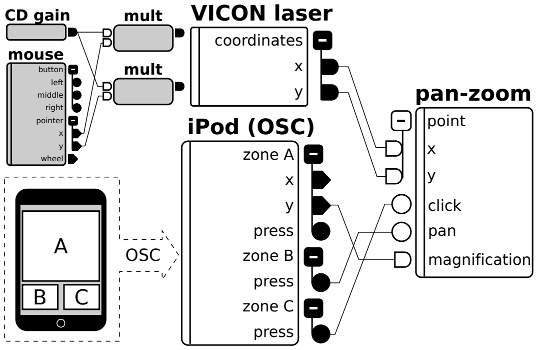

Figure 6: Sample configuration of the WILD Input Server.

the developer can use the grey modules instead of the VICON one to simulate the tracked location with a mouse.

Nancel et al. used *jBricks* to develop the previously described experiments [11, 12]. E. Pietriga and R. Primet also used it to display extremely large images, including a 400,000 pixel-wide image of the Milky Way galaxy (Fig. 2, right) and a 27 Gpixel image of Paris. They are currently creating an application that will allow astrophysicists to display and navigate large images in the native FITS format used in their field. This requires real-time filtering and image transformations to turn the high-dynamic signal into a useful display, and involves mapping the coordinates of objects identified in the image into entries in astronomical object databases.

6.3 Substance

Gjerlufsen et al's *Substance* [4] is designed to explore multi-surface interaction. A *Substance* application is a collection of *environments* running as separate processes. Each environment manages a tree of objects and can share subtrees with other environments. Sharing can be done through mounting, where the shared subtree is accessed through remote procedure calls, or through replication, where the shared subtree is copied and the multiple copies kept in sync. Functionality can be added to the nodes of the tree at run-time. Together with the dynamic discovery of environments running on the local network, this provides the basic support to create flexible distributed applications. *Substance* uses the WILD Input Server to support distributed interaction instruments [8]. Instruments are typically separate environments and can be easily added and removed at run-time and reused across applications.

We used *Substance* to create two applications for participatory design sessions. C. Klokmose and J. Eagan developed *SubstanceCanvas* (Fig. 7), which implements a canvas that spans the wall, the table and any other surface that users want to add to it, such as part of a laptop screen. Various types of content can be displayed on this canvas and freely manipulated using instruments such as a handheld iPodTouch to move or scale content. The system is extensible by creating new content providers. Default content providers include the display of images, PDF documents and live windows from laptops running Mac OS X. An email content provider automatically displays attachments sent to an email address that was created for the room. This is the simplest way to add content to the canvas: just send an email to the WILD room. Another provider maps RFID tags to documents, making it easy to add the PDF version of a tagged paper document to the canvas.

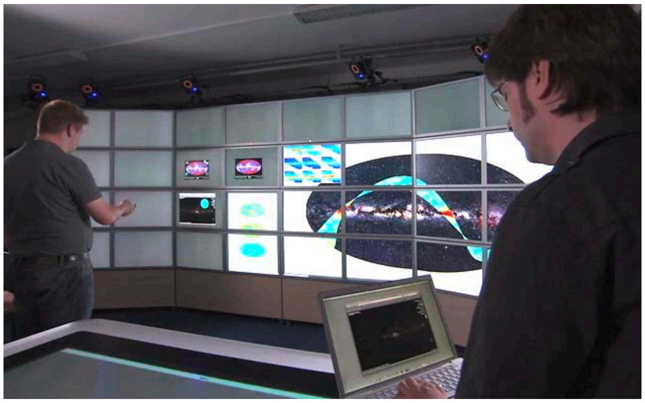

Figure 7: Using *SubstanceCanvas*.

C. Pillias created *SubstanceGrise* (“GreyMatter”), which displays an array of 64 brain scans that users can freely organize from the table (Fig. 2, left) and orient in 3D using the brain prop (Fig. 3). It demonstrates the ability of the framework to integrate existing applications. In this case the BrainVisa/Anatomist[9] 3D viewer is wrapped as a *Substance* environment through a set of scripts. The sharing capabilities of *Substance* then make it easy to share, e.g., the camera that is used to control the orientation of the brain scans.

6.4 Key Findings

Creating our own software for WILD not only provided additional flexibility, it also encouraged us to explore novel approaches. Developing two frameworks helped us better understand the design space. For example, while *jBricks* follows a more centralized model where the application is run on the front-end and the framework handles distribution in the background, *Substance* requires the developer to handle distribution explicitly. Ultimately, it is not yet clear whether these two approaches should converge or not.

Programming the WILD platform was daunting at first, but was greatly simplified by two insights. First, we discovered that we could use a naïve approach when displaying data on the wall: running replicas of the application and making copies of the data sets on each node of the cluster. Each replica knows which part it is supposed to display. A master version of the application runs on the front-end and coordinates the replicas. This worked surprisingly well, even without explicitly synchronizing the replicas. For example, C. Pillias created *WILDMol*, a wall-size version of the PyMol[10] molecular visualization system, simply by using its scripting capability: When the user manipulates the molecule, e.g. to rotate it, the commands are sent as OSC messages to the master application, which forwards them to each replica. *WILDMol* can display and rotate full-wall 3D molecules in real-time without visible tearing.

The second insight was to leverage the OSC protocol to manage all input as well as most inter-application communication. Because of the simplicity, flexibility and availability of OSC and its integration into ICon, we can manage new devices and test new techniques very quickly, including during design sessions with users.

7. CONCLUSION AND FUTURE WORK

The WILD project followed an integrative research process: We engaged with target users early on to understand their needs and created a flexible hardware platform to support them. We pur-

[9] `http://brainvisa.info`

[10] `http://www.pymol.org`

sued a participatory design approach to explore solutions while also conducting basic research on interaction techniques and creating a software platform to develop applications. Beyond the individual results in each of these areas, the point of this article is to emphasize the benefits of such an integrative approach. Each piece of the project both relies on progress in other areas and fosters it. Each advance in one area is validated by its use in another. For example, using position-tracked handheld devices came out of participatory design, and was made possible by the choice of hardware; experimental studies validated its use for specific tasks, and it was implemented as multi-surface instruments in the *Substance* middleware.

WILD is still a work in progress. We continue to develop the areas outlined in this paper with the goal of making the room available to outside users. Based on input from our users, we will also support remote collaboration both with individuals and with other large-scale display environments. Specifically, the DIGISCOPE[11] project will create a network of nine high-end interactive visualization rooms (including WILD) interconnected by a high-end videoconferencing system. In addition to scientific discovery, DIGISCOPE will address product lifecycle management, business intelligence, crisis management and education, dramatically expanding the scope of the integrative approach that was used in WILD.

WILD demonstrates that the benefits of an integrative approach greatly outweigh the costs. Critical to the success was combining a diverse set of skills to address the various challenges, the choice of lead users, and generous hardware to make simple solutions work. Just as integrative projects in other areas of computer science have changed the face of computing, HCI should do the same to shift paradigms and create the next generation of interactive systems.

8. ACKNOWLEDGMENTS

WILD[12] is the result of extensive collaboration among the members of the In Situ lab. I especially want to thank Emmanuel Pietriga, Clément Pillias, Romain Primet, Stéphane Huot, Olivier Chapuis, Wendy Mackay, James Eagan, Clemens Klokmose, Tony Gjerlufsen, Mathieu Nancel and Julie Wagner for their tireless work on the project. I also thank our partner laboratories, in particular IAS (astrophysics), LAL (particle physcics), IGM (biology) and Neurospin (neuroscience). WILD is supported by a Région Île-de-France / Digiteo grant and by Université Paris-Sud, INRIA, CNRS, ANR and INRIA-MSR.

9. REFERENCES

[1] M. Beaudouin-Lafon. Designing interactions, not interfaces. In *Proc. Advanced Visual Interfaces, AVI'04*, pages 15–22. ACM, 2004.

[2] J. Coutaz, C. Lachenal, and S. Dupuy-Chessa. Ontology for multi-surface interaction. In *Proc. Human-Computer Inteaction, INTERACT'03*, pages 447–454. IOS Press, 2003.

[3] P. Dragicevic and J.-D. Fekete. Input device selection and interaction configuration with ICON. In *People and Computers XV - Interaction without frontiers, IHM-HCI'01*, pages 543–558. Springer Verlag, 2001.

[4] T. Gjerlufsen, C. Klokmose, J. Eagan, C. Pillias, and M. Beaudouin-Lafon. Shared Substance: Developing flexible multi-surface applications. In *Proc. Human Factors in Computing Systems, CHI'11*, pages 3383–3392. ACM, 2011.

[5] J. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. User Interface Software and Technology, UIST'05*, pages 115–118. ACM, 2005.

[6] K. Hinckley, R. Pausch, J. Goble, and N. Kassell. Passive real-world interface props for neuro-surgical visualization. In *Proc. Human Factors in Computing Systems, CHI'94*, pages 452–458. ACM, 1994.

[7] F. A. Johanson, B. and T. Winograd. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1(2):67–74, Jan 2002.

[8] C. Klokmose and M. Beaudouin-Lafon. VIGO: Instrumental interaction in multi-surface environments. In *Proc. Human Factors in Computing Systems, CHI'09*, pages 869–878. ACM, 2009.

[9] W. Mackay. Which technique works when? floating palettes, marking menus and toolglasses support different task strategies. In *Proc. Advanced Visual Interfaces, AVI'02*, pages 203–208. ACM, 2002.

[10] S. Malacria, E. Lecolinet, and Y. Guiard. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the Cyclostar approach. In *Proc. Human Factors in Computing Systems, CHI'10*, pages 2615–2624. ACM, 2010.

[11] M. Nancel, E. Pietriga, and M. Beaudouin-Lafon. Precision pointing for ultra-high-resolution wall displays. Technical Report 7624, INRIA, 2011. http://hal.inria.fr/inria-00595046.

[12] M. Nancel, J. Wagner, E. Pietriga, O. Chapuis, and W. Mackay. Mid-air pan-and-zoom on wall-sized displays. In *Proc. Human Factors in Computing Systems, CHI'11*, pages 177–186. ACM, 2011.

[13] D. Olsen Jr. and T. Nielsen. Laser pointer interaction. In *Proc. Human Factors in Computing Systems, CHI'01*, pages 17–22. ACM, 2001.

[14] E. Pietriga. A toolkit for addressing HCI issues in visual language environments. In *Proc. Visual Languages and Human-Centric Computing, VL/HCC'05*, pages 145–152. IEEE, 2005.

[15] E. Pietriga, S. Huot, M. Nancel, and R. Primet. Rapid development of user interfaces on cluster-driven wall displays with jBricks. In *Proc. Engineering Interactive Computing Systems, EICS'11*, pages 185–190. ACM, 2011.

[16] J. Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proc. User Interface Software and Technology, UIST'97*, pages 31–39. ACM, 1997.

[17] D. Smith, C. Irby, R. Kimball, and E. Harslem. The Star user interface : an overview. In *Proc. National Computer Conference, AFIPS*, pages 515–528. ACM, 1982.

[18] D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. User Interface Software and Technology, UIST'05*, pages 33–42. ACM, 2005.

[19] E. von Hippel. Lead users: a source of novel product concepts. *Management Sc.*, 32(7):791–805, 1986.

[20] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.

[11] `http://digiscope.fr`

[12] `http://insitu.lri.fr/Projects/WILD`