

# Exploring Interaction with a Simulated Wrist-Worn Projection Display

Gábor Blaskó<sup>†</sup> Franz Coriand<sup>‡</sup> Steven Feiner<sup>†</sup>

<sup>†</sup>Department of Computer Science  
Columbia University  
New York, NY, 10027  
[gblasko, feiner]@cs.columbia.edu

<sup>‡</sup>Bauhaus University Weimar  
Bauhausstr. 11, 99423  
Weimar, Germany  
franz.coriland@medien.uni-weimar.de

## Abstract

One of the major limitations of portable computing devices is the small size of their built-in displays. Fortunately, extremely small projection systems are being developed that can be integrated into devices that are small enough to be body-worn, yet can project a large image onto surfaces in the environment. To explore how a user might interact with this near-horizon technology, we created a functional simulation of a wrist-worn projector. We then developed a set of interaction techniques that assume that the wrist-worn computer and projector are equipped with position and orientation sensors, in addition to a touch-sensitive built-in screen. To complement the techniques that rely on the spatial manipulation of the user's forearm and the device itself, we also describe the use of a cursorless watch user interface that minimizes the need for the user to look down at the device's built-in screen. Finally, we present a sample application that illustrates our interaction techniques.

## 1. Introduction

Ubiquitous portable devices, such as cell phones, portable digital assistants (PDAs), and watch computers, are useful for keeping in touch with friends and colleagues or managing personal information. However, their small displays make it difficult to present a large amount of information to a single user, or share information with a group. Furthermore, interaction with these small portable platforms is constrained by the input mechanisms they make available. The number of dedicated physical widgets (e.g., buttons, dials, and joysticks) with which they can be equipped is limited. In addition, graphical user interfaces that force the user to look at the built-in display (e.g., to navigate a cursor or manipulate on-screen widgets) are both time-consuming and distracting.

Extremely small projection displays [1,3] are being developed that could overcome the display limitations of current portable devices. They will make it possible to project moving images onto surfaces in environments in which no external display equipment is available. Recognizing this potential, we decided to explore the design



**Figure 1.** Use of simulated wrist-worn projection system to explore interaction possibilities: Panning, zooming, and interaction with content is possible with the use of orientation- and position-tracking technologies and virtual widgets implemented on the wrist-worn system's touch-sensitive surface.

of user interfaces for a wearable computing platform equipped with a tiny projection display. Our goal was to develop interaction techniques that could support ad-hoc data retrieval, content manipulation, and presentation, while minimizing the need to look away from the projected image.

After first describing related work, we present a functional simulation of a wearable projector, shown in Figure 1, which we developed to prototype interactions with this as yet unavailable technology. It uses a stationary commercial projector, position- and orientation-tracking systems, and a desktop computer. We then discuss simple content-manipulation techniques controlled by the orientation and position of the user's wrist, which we implemented with our simulator. Finally, we discuss how cursorless interaction methods [5] can keep users from having to look away from the projected display, and show how these methods can be applied to manipulate content and control a prototype stock market information browser application.

## 2. Related work

Pinhanez and colleagues [9] present a system in which a statically mounted projector's image is reflected off a robotic mirror assembly onto arbitrary surfaces in

the environment. They use vision-based hand and fingertip tracking to create a pointer-based graphical interface, in which the user can trigger a virtual button press event by resting the finger on the projected image for a few seconds. In contrast, our system does not use cameras for fingertip tracking, nor does it require the user to virtually touch any part of the projected image. With a body-mounted projector, if the throw distance of the projected image is farther than arm's length, then pressing virtual buttons on the projected image is not possible. Consequently, our interaction techniques are designed to use only the wearable device itself.

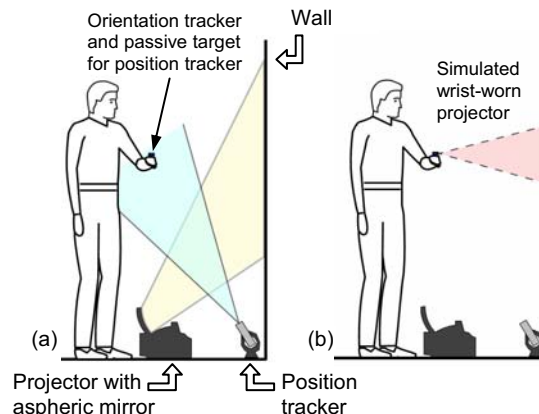
Beardsley and colleagues [4] explore the use of a tracked handheld 1 kg projector, outfitted with a camera that tracks fiducials on a wall. Unlike the cursorless interface we discuss, their prototype uses only cursor-based interaction with a projected user interface. A visual cursor is fixed at the center of the projector image plane, and as the projector itself is translated and rotated, the image is transformed to compensate for the motion. Thus, the image remains stationary on the wall, while the cursor moves with the projector.

We also note the work of Raghunath and colleagues [10], who examine the symbiotic relationship that can exist between a public display and a personal portable device, such as a PDA or WatchPad [8], primarily addressing issues of privacy. Similarly, the prototype stock browser application that we discuss later also makes simultaneous use of the wearable device's personal built-in display and the projected publicly viewable display.

### 3. Overview of prototype implementation

Our goal was to provide the users of our testbed with the realistic look and feel of a wrist-worn computing system with a built-in projection display system. We were inspired by seeing working prototypes of the Symbol laser projection display (LPD) system, which produces a red monochromatic VGA-resolution image, equivalent in size to a desktop monitor, when projected from a distance of ~60–90cms. In darker environments, a ~76cm diagonal display can be projected from a 1.5m distance. According to its inventors, cost, not physical size, is the only limitation to creating an XGA resolution, full-color projector based on this technology [1].

Since light intensity decreases with the square of the projection distance, we assumed in designing our prototype that we would not project from farther than 2m away. We needed to create a projected image on a wall; however, we had to find an alternative to using a traditional front-projection system, which, at such a close distance, would have produced shadows of the user and their forearm on the wall. We decided to use an NEC WT600 projector, which uses aspheric mirrors, instead of lenses, to achieve extremely short throw distances [2]. Placing this projector on the floor, approximately 70cm from the wall, as shown in Figure 2, we are able to pro-



**Figure 2.** Wrist-worn projector testbed. (a) Projector on floor between user and wall, orientation tracker on wrist, and position tracker on floor targeting passive target on user's wrist. (b) Simulated wrist-worn projector.

ject a full color image about  $2\text{m} \times 1.5\text{m}$  in size, positioned approximately 74cm above the floor. We use only a very small portion of this large projection space to simulate the image of the virtual wrist-worn projector. Therefore, we are able to move the wrist-worn projector's virtual image anywhere within this large portion of the wall. Fortunately, since the real image is projected from below the user's forearm, visible shadows are cast only when the forearm is very close to the wall, and even then, these shadows are at head height, where there is no active content.

#### 3.1. Tracking system overview

We present the user with an image that appears to be projected from a wrist-worn computing device, with realistic keystone distortion that adds to the believability of the simulation, as shown in Figure 1. To accomplish this, the system must monitor the position and orientation of the user's wrist. Fortunately the benefit of tracking position and orientation in our testbed is twofold; in addition to using it to simulate the image from the virtual projector, we also use it to explore interaction techniques that rely on orientation and position sensing.

The wrist-worn computing platform that we envisioned during the design of the system was a conceptually enhanced version of the IBM WatchPad prototype wrist-worn computer [8]. Since the current WatchPad already has an orientation sensor built in, we thought that it would be realistic to assume that a future wrist-worn computer would also contain a position sensor (e.g., using some combination of GPS, ultrawideband, infrared, ultrasonics, cameras, gyroscopes, magnetometers, and accelerometers). For wrist position sensing in our current testbed, we chose to use an Origin Instruments DynaSight tracking system, which monitors the spatial movements of an untethered passive, retroreflective target using stereo infrared sensors.

We placed the position tracker on the floor, between the projector and the wall, as illustrated in Figure 2. The

position tracker was pointed upward at an angle, which allowed it to track from below the retroreflective target, which we attached to the wrist-worn system's wrist band.

We had originally planned to use the orientation sensor and touchscreen that are built into the WatchPad and create a wireless Bluetooth link to the PC controlling the simulated secondary projection display. Unfortunately, performance problems with the orientation sensor, Bluetooth link, and touchscreen convinced us to consider alternatives. For our wrist-worn system, we ended up using an InterSense InertiaCube2 orientation tracker, and a Synaptics capacitive touchpad overlaid with a plastic template equivalent in size to the frame of the WatchPad screen to offer the same passive tactile feedback.

Not using the WatchPad was a tradeoff that we tried to avoid. However, it became clear that we would not be able to track and transmit rapid hand and finger movements with the WatchPad's several-year-old architecture. Nevertheless, we believe that a wrist-worn computing device supporting all the capabilities that we simulate could be built in the near-term future.

### 3.2. Software system overview

To realistically simulate a small moving projector's image on the wall, we employ a relatively straightforward software pipeline. Using OpenGL, we create a virtual scene with a wall surface coincident with the real wall and a light source, whose position and orientation are determined by the tracking system. The graphical content to be displayed by the simulated projector is rendered into a texture buffer. Using projected textures [6,12], texture-mapping coordinates are generated, and the simulated projector's texture is mapped to the virtual wall in the scene. Finally, the frame buffer is rendered, using an orthographic projection from a viewpoint facing the virtual wall and output to the real projection system.

In the following sections, we describe the interaction techniques that we have designed and prototyped, discuss our experiences, and present a sample application, all implemented with the testbed system described above.

## 4. Orientation-tracker-based interactions

We cannot assume that most users of our testbed will

have previously tried a wearable projection display. Therefore, the first application that we developed was designed to be simple to understand and to provide a good feel for what a wrist-worn projector could make possible.

### 4.1. Orientation-controlled panning

Our initial application models the expected optical behavior of a wrist-worn projector, augmented with user-controlled panning of passive content (a web page). By moving their wrist in front of the wall, a user can move the entire projected image around—as the wrist approaches or recedes from the wall, the wrist-worn projector's simulated image appropriately decreases or increases in size.

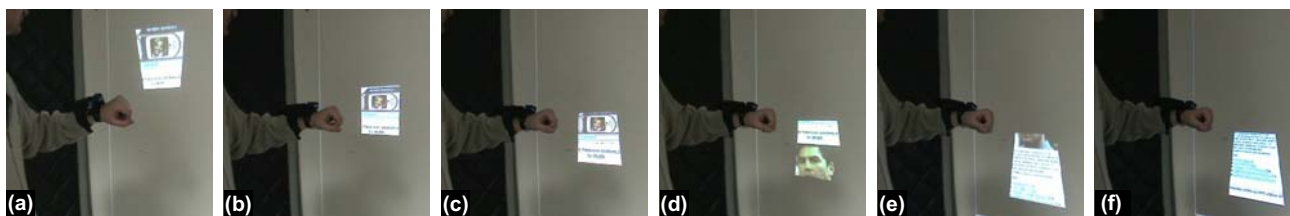
A number of researchers have explored the use of orientation sensors built into a hand-held display to pan information as the display is tilted up/down or left/right [7,11]. Similarly, we wanted our users to be able to pan projected content vertically or horizontally by turning their wrist and forearm.

While the axis of projection is off-perpendicular, relative to the wall, by no more than 15°, the projected content remains still, as illustrated in Figure 3(a–c). Increasing the tilt between 15–30° off-perpendicular, the panning velocity of the content linearly increases with angle, as can be seen in Figure 3(d–f), which demonstrates vertical panning. If the user returns their wrist to a “neutral” position while panning, the content stops moving. If the wrist is tilted beyond 30°, the panning velocity is clamped to its maximum value, to keep the content from moving so quickly that it becomes unintelligible.

### 4.2. Experiences and observations

By being able to simulate user interaction with a technology to which we currently do not have access, we can explore issues that will be influential in the design of user interfaces for future wrist-worn computers with projection displays. We informally observed how users execute orientation-based panning interactions with this user interface, and how the biomechanical limitations of the user's arm influence usability.

The wrist-worn device can be easily rotated about the axis defined by the horizontal forearm (pitch), controlling vertical panning. In contrast, rotation of the wrist-worn device about an axis perpendicular to its display,



**Figure 3:** (a–c) Simple simulation of static content being projected from virtual projector on user's wrist. (d–f) If tilt angle of wrist exceeds 15°, the content that is projected begins to pan in the appropriate direction (in this case, vertically). A downwards tilt pans towards the bottom of the page. Horizontal panning is accomplished similarly. (Outline of real projector's full imaging area is shown for comparison.)

and perpendicular to the horizontal forearm itself (yaw) can easily be accomplished by rotating the entire forearm at the elbow joint or the arm at the shoulder, or by turning at the waist. Since horizontal and vertical rotations of the wrist-worn device are controlled by different muscles and joints, we recommend the use of an interface that handles panning in these two axes independently—disallowing diagonal panning—when using device-orientation-based input with such a wrist-worn device.

Holding one's forearm motionless for more than a few seconds, to stabilize the projected image, is challenging, since the muscles may become strained and tense, resulting in user discomfort. Users respond with slight limb motions, resulting in the vibration or wobbling of the projected image. Unfortunately, slight angular motions of the hand, wrist or forearm that one might think are insignificant, can result in considerable motion of the projected image. Anyone who has tried to keep the beam of a laser pointer stable on a distant screen during a presentation is familiar with this problem.

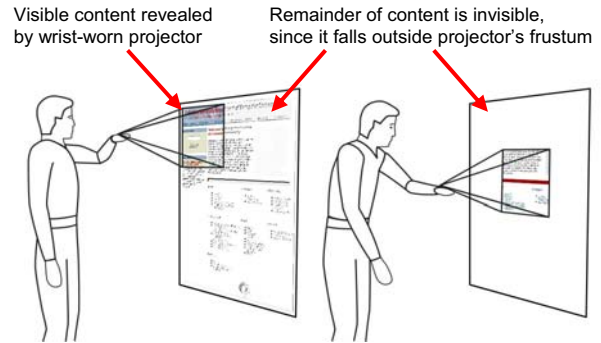
Since the wrist-worn components of our testbed are extremely lightweight (e.g., comparable to a WatchPad), we believe that the user's experience with them is comparable to what they would experience with an actual wrist-worn watch computer with a built-in projection display. We note that the noise of our tracking systems is a component of the lower amplitude, high frequency image vibration that we observed. However, the higher amplitude, lower frequency image wobbling we observed appears to be predominantly the result of the instability of the user's forearm. We plan to investigate this issue further.

Preliminary observations of our simulator system in use lead us to believe that small body-worn or handheld projection systems would greatly benefit from the use of image stabilization techniques similar to those used in cameras and binoculars to counter the effects of hand tremor.

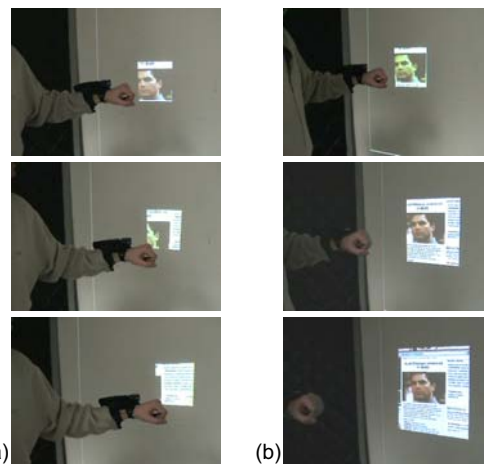
## 5. Spotlight technique: Spatial position-based interactions

Because of its small size and weight, a laser projection display can not only be easily and swiftly rotated, but can also be easily repositioned and moved in front of the surface onto which it is projecting. If a small portable computing device were equipped with a position sensor, as well as an orientation sensor, additional interaction techniques would be possible.

We were inspired by the pose-aware flashlight described by Teller and colleagues [13], in which a conventional externally tracked projector was used to register textual and geometrical metadata directly onto environmental surfaces. This work focused on context-aware overlay of imagery, and did not address interaction with the system. We decided to prototype and explore a similar interaction mechanism with our testbed, which we



**Figure 4:** Schematic illustration of spotlight technique; Section of content revealed is determined by spatial position of wrist-worn projector. Content appears to be statically fixed to the wall surface, with projector acting as a revealing “light source.”



**Figure 5:** Simulation of spotlight technique; Effects of spatial wrist movement: (a) parallel to the wall, and (b) perpendicular to the wall.

call the *spotlight technique*, using it for simple content manipulation. We wanted to find out how usability is influenced if the projector is small, lightweight, and wrist-worn, and explore interactions that are not practical with conventional projector systems.

The spotlight technique, illustrated schematically in Figure 4, is based on a simple metaphor: the movement of a small focused light source in front of a wall covered with graphical content (e.g., successively illuminating portions of a poster on the wall with a flashlight in a darkened room). With the spotlight technique, the projected content is panned horizontally and vertically in the opposite direction of wrist movement to give the impression that the projection is a moving window onto a static wall-sized graphical image.

The user is able to see a subsection of the content at its most detailed resolution and is able to pan around it by physically moving the projector itself in a plane parallel to the projection surface, as shown in Figure 5(a). By moving the virtual projector along the axis perpendicular to the wall surface forward or backward, the window of visible content shown is varied in size, simi-

lar to how shining light onto an image from farther away would expose a larger portion of the image (Figure 5b).

### 5.1. Experiences and observations

We have found this technique to be very efficient and quick, especially in applications in which the user needs to get a quick overview of the content, and then focus on a section of it. For example, when a user is scanning a map, to get a better sense of where a certain address is located, it is often useful to see the surrounding neighborhood. By swiftly moving her wrist away from the wall, and possibly stepping back, thereby increasing the size of the “revealing window”, she can view more of the content.

Even if the resolution of the image on the wall at this greater distance is less detailed (as it would be with a real wrist-worn projector), the user can quickly get an overview. Then, once the portion of interest is found, the user can move their wrist closer to the wall, to concentrate on the appropriate section, and see that section in the highest resolution possible. During this interaction, the full content (e.g., the entire map) appears to be statically fixed to the wall surface.

Here, we note an important distinction between a real wrist-worn projector and our simulated wrist-worn projector: in a real wrist-worn projector’s image, the number of pixels remains constant, while the size of each pixel varies with the distance between the projector and the wall, getting smaller as the projector moves closer, resulting in a smaller, *brighter*, and *higher-resolution* image of a smaller portion of the content. In contrast, in our simulated wrist-worn projector’s image, the number of pixels varies with the distance between the user’s wrist and the wall, getting smaller as the simulated projector moves closer, while the size of each pixel remains constant, resulting in a smaller, but *constant-brightness* and *constant-resolution*, image of a smaller portion of the content.

The spotlight technique allows for interaction at a close distance to the wall, which is advantageous, because the image of the projection display is not only brighter, but of better quality at close range (in a real wrist-worn projector), since the same number of pixels are concentrated in a smaller area. Furthermore, no matter how imprecisely the orientation of the wrist-worn projector is changed due to hand or wrist movements, if the content that is projected is moved appropriately to counter the wrist-worn projector’s physical movement, the image on the projected surface appears to be static. Keeping the appearance of the content static, by counter-moving the image based on sensor data, allows the user to not worry about keeping their hand still while they are examining a specific section of content in the small window. (We note here that our use of a fixed projector models the stability of a perfect position tracker.)

## 6. Multiwidget-based interactions

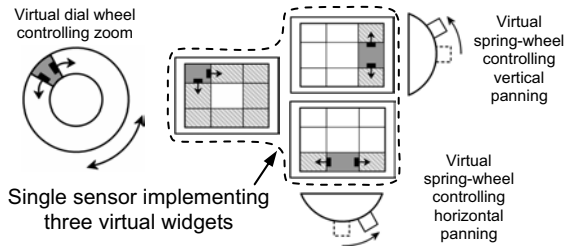
The two interaction techniques discussed above are intended for environments in which there is a wall surface that is large enough on which to project (for orientation-based panning) and in front of which the user’s wrist can move (for position-based panning). However, these techniques may be inappropriate to use in a collaborative setting. For example, if multiple people are interacting, and one or more are projecting content on the wall, then moving one’s limbs around, changing body posture, and continually varying the spatial position of the visual content in front of the other participants may be problematic.

Therefore, we believe that an alternative method for panning and zooming should be offered that involves discreet interaction with the input devices built into the wrist-worn device, instead of these more demonstrative spatial manipulation techniques. We chose to employ the *multiwidget* [5]. The multiwidget allows a single touch sensor (e.g., a touchscreen or a section of a capacitive touchpad) to virtually implement a multitude of directly accessible cursorless manipulator widgets, with the assistance of passive haptic feedback.

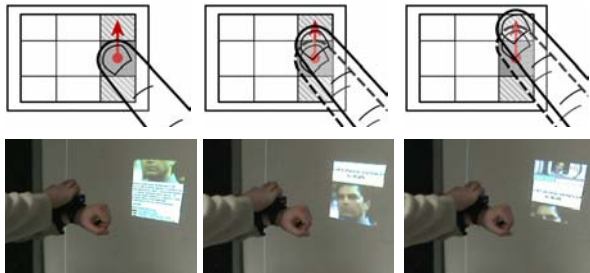
The key benefit of the multiwidget is that during the manipulation of parameters it does not require the user to look at either the device or its display, since passive haptics guide the fingers, no on-screen widgets are used, and no visual cursor needs to be navigated. Since the widgets are implemented in software, their layout can be arbitrarily reconfigured instantaneously without hardware modification. By monitoring which of the eight corners or edge sections the user’s finger contacts first on a rectangular sensor, and depending on the directionality (clockwise or counterclockwise) of the initial finger motion, up to eight parameter adjustment widgets can be implemented with a single small rectangular sensor.

We use two virtual spring-wheel widgets (accessible from the right edge and the bottom edge sections) to pan the image content, and a virtual dial wheel widget (accessible from the top left corner segment of the sensor) for zooming, as illustrated in Figure 6. The spring-wheel widgets are manipulated by dragging the finger up/down along the right edge or left/right along the bottom edge of the sensor. Figure 7 illustrates the upward dragging of the finger, and the resulting vertical panning of content. The dial wheel, which controls zoom in incremental steps, is operated by dragging the finger along the frame of the sensor in a circular motion, starting from the upper left corner, as shown in Figure 8. As long as the finger remains on the sensor, the direction of the finger motion may be reversed at any time to zoom in or out.

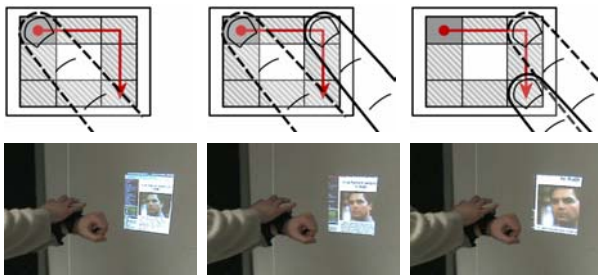
These virtual widgets provide quick panning and zooming capabilities without navigating an on-screen cursor, manipulating on-screen widgets, or spatially moving the user’s wrist and forearm. Furthermore, during interaction, as illustrated in Figure 1, the hand executing the finger motions on the touch-sensitive surface



**Figure 6:** Multiwidget used to pan and zoom; single touch sensor implements three virtual widgets with overlapping active sections (overlaid on the touchpad, but shown separately here for clarity).



**Figure 7:** Dragging upward from right edge section controls the vertical panning of projected content, with a virtual spring wheel. Length of dragged vector determines speed of panning.



**Figure 8:** A circular clockwise dragging motion on a virtual dial wheel, starting from the upper left corner, zooms the projected content.

helps stabilize the forearm that has the small projector mounted on it. This helps to avoid the awkward jittering or wobbling of the projected image that can result from unintentional forearm movements.

## 7. Stock information browser application

To provide a compelling, illustrative example for the utility of a projective display with an interactive wearable computer, we chose the scenario of visualizing stock market data. Our rationale is as follows. When dealing with a particular stock, or a comparison of multiple stocks, a large quantity of numerical data needs to be displayed in context, all at the same time, and this requires screen space. It is also very useful to simultaneously show graphs of trends and price levels to assist in quick decision making, which also requires screen space. An on-the-go businessperson could greatly benefit from a portable device equipped with a small projection dis-

play that can instantaneously present a large amount of visual data anywhere.

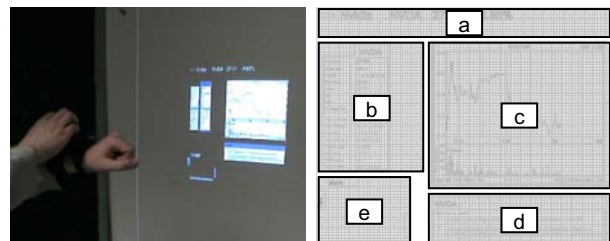
In the scenarios mentioned earlier, the content that is shown does not lend itself to much interaction and may be created before it is presented. A map can be drawn or downloaded, a photo can be taken, a document can be written on another system in advance, all prior to presentation. Once this content is created, the only interaction that may be needed is view-altering—such as panning and zooming—rather than content-altering. In contrast, when visualizing stock market data, the system should offer the user efficient input mechanisms to modify the content being displayed. In a feature-rich interactive system, there is a much stronger emphasis on the input mechanisms by which user interaction can be facilitated.

We believe that there is a significant advantage to building applications with the toolset provided by our cursorless watch computing interface framework. For the interactions discussed below, there is no need to spend time navigating an on-screen cursor and manipulating on-screen widgets with it; hence, faster interactions are possible.

### 7.1. System description

Our goal was not to create a commercial-grade stock market data browser system with a vast number of features, but rather to take a realistic scenario and create a testbed application with which we can demonstrate the capabilities of the wrist-worn projection simulator and our user interface framework developed for wrist-worn computer interaction. We chose a set of realistic tasks that a businessperson might wish to accomplish to access, manipulate, browse, or quickly glance at information.

The application design makes simultaneous use of both displays. The projected display is divided into five sections, as pictured in Figure 9. These include (a) the title section showing company name, ticker symbol and current stock price, (b) a table section for detailed price and volume data, (c) a section for graphs of stock performance, and (d) a section to display company-related news headlines. The lower left corner section (e) of the projected display can be used to present market overview information (e.g., industrial indices), or to show a



**Figure 9:** Projected content of prototype stock browser application: (a) title section, (b) stock price and volume data, (c) chart of stock performance, (d) company news headlines, and (e) screen portion showing market summary data or showing clone of the built-in display of the wrist-worn device. (Portions of screen content Copyright 2005 Yahoo! Inc.)

clone of the wrist-worn device's built-in primary screen.

The wrist-worn device's built-in display could possibly be used to display private information that the user does not wish to share on the projected screen. For example, the built-in display may present the amount and monetary value of the stock the user owns in the company for which the projector is presenting information, or the total value of the user's stock portfolio. Currently, the application design we implemented only calls for the use of the small display during menu control, to display lists of menu elements.

In the final version of our simulator system, we used a capacitive touch sensor, and not the WatchPad, which is equipped with a touchscreen. Therefore, we used the lower left section of the projected image to display what would be shown on the wrist-worn device's screen. As we discuss later, since experienced users may interact with the device with gestural shortcuts that may be executed eyes-free, neither the wrist-worn device's built-in display screen, nor its clone on the projected screen, need be visible during interaction.

In our simulator, all the application control functions for the stock information browser (menu navigation, parameter adjustment, and text entry) are handled by the wrist-worn touch sensor's  $\sim 2.5\text{cm} \times 2\text{cm}$  sensing area. No other input device, other than this sensor, is used to facilitate cursorless interaction with the application to execute the tasks described below.

## 7.2. Interactions with the application

In this section, we briefly present a subset of the features of our prototype application and describe the input techniques it uses. First, we explain the two methods provided to select the company for which data is displayed.

### 7.2.1. Company selection using menu navigation

It is realistic to assume that of the thousands of companies listed on a stock exchange, a user has a short list of favorite companies. Suppose that this short list is hierarchically organized according to the market sector by which companies are categorized. To quickly navigate among members of this preset list of companies and bring up their stock information, we utilized our earlier work on menu navigation [5]. Novice users may navigate with simple steps and receive visual feedback along the way, by looking at the watch's display or its clone on the projected screen. At each node in the hierarchy, users may iterate and select from a list of menu elements (in this case, market sectors and ticker symbols).

The key advantage of this menu navigation system is that the hierarchy may also be traversed, and a ticker symbol may be selected, by executing a gestural shortcut. By executing finger stroke gestures with the finger on the sensor, the user may bring up information about a desired company instantaneously. Thanks to the tactile guidance offered by the sensor frame, these shortcuts may be executed eyes-free. Since no on-screen cursor, widgets or menus need to be displayed when these

shortcuts are used, the information shown publicly with the projection system will change instantaneously, like a slideshow, and present new content.

### 7.2.2. Company selection using text entry

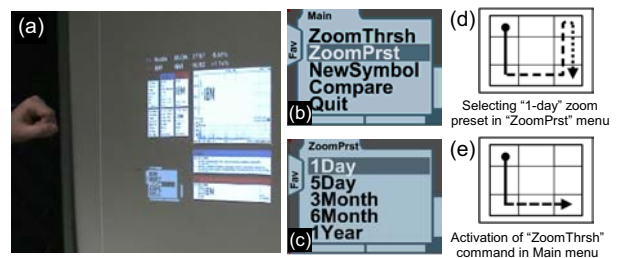
Listed companies are identified on the stock exchange by their ticker symbol, which is an abbreviated name for each company of up to four characters. To allow the user to enter arbitrary ticker symbols or company names, and look up their stock information, we implemented the EdgeWrite [14] text entry system in our wrist-worn user interface framework. EdgeWrite places similar requirements on the input device—it requires a sensor with four corners—as our menu control and parameter manipulation input techniques. Users can also use manual text entry when they want to compare the stock performance of two companies (Figure 10a) and wish to specify the second company by its ticker symbol.

### 7.2.3. Changing chart with shortcut gestures

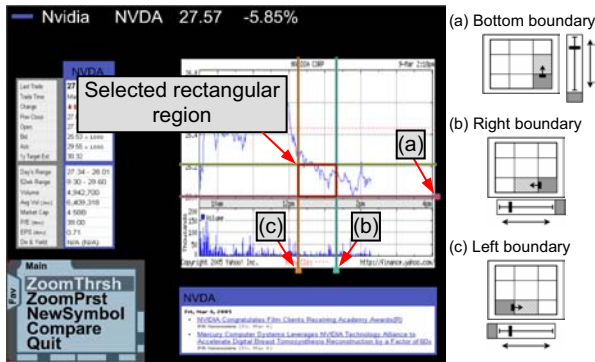
The user is able to rapidly change the time range of the graph showing stock performance by using quickly executable gestural shortcuts. This input method demonstrates how one can combine a command (e.g., zoom by a preset value) and its argument (e.g., select from among preset values) by concatenating stroke segments into a single continuous finger gesture. For example, the gesture illustrated in Figure 10(d) is composed of three sub-strokes. The first (continuous line), single-length sub-stroke is used to specify that the command to activate is on the "Main" menu tab. The next sub-stroke section (dashed line), encodes with its length—two "edges"—that it is the second "ZoomPrst" (zoom by a preset value) command that should be executed (Figure 10b). Finally, after a change in the direction of the dragging motion, a one-length short stroke (dotted line) encodes the data component; the first preset value in the list specifies that the chart should be scaled to chart the stock performance for a single day (Figure 10c).

### 7.2.4. Zooming using rectangular box selection

In our prototype application, we implemented a new cursorless method of region-based zooming. This method is complementary to, and may be combined with, the multiwidget-based panning and zooming method described in Section 6. By allowing users to specify a rectangular selection box and zooming into it,



**Figure 10:** (a) Projected content during comparison of two companies. (b-c) Sample menu screens (d-e) Sample gestural shortcuts using concatenated multistrokes [5].



**Figure 11:** Rectangular region (with preset aspect ratio) is selected for zooming by using three sliders of a multiwidget. (a) Bottom boundary. (b-c) The same sensor section (bottom edge) is utilized by two virtual slider widgets for right and left boundary. (Portions of screen content Copyright 2005 Yahoo! Inc.)

we can demonstrate how one is able to specify a rectangular region without the use of a two dimensional cursor.

By locking the aspect ratio of the selection box to the aspect ratio of the chart section of the image (Figure 9c), only three parameters need to be specified; if the left, right and bottom edge boundaries are manipulated manually, then the top of the selection box may be automatically calculated. To control these three parameters we use three directional slider widgets, as illustrated in Figure 11. Note that the bottom edge section of the sensor is the active section used during the control of the left/right movement of both the right boundary (Figure 11b) and the left boundary (Figure 11c). After the user initially contacts either the right or the left bottom corner section of the sensor and moves out of that corner section into the bottom edge section, the choice is implicitly made, as to whether the right or left boundary of the rectangular region is manipulated.

Once the rectangular region is specified, we either navigate to the "ZoomThresh" (zoom based on thresholds) menu entry in the "Main" menu tab (Figure 10a), and activate it, or we execute the shortcut gesture illustrated in Figure 10(e), which combines tab selection, command selection, and command execution in a single gesture.

## 8. Conclusions and future work

We presented a testbed for the exploration of interactions with a simulated small wearable projection display. Interactions based on spatial position and orientation of the wrist and forearm were discussed. Based on our preliminary observations of users of our testbed, we described usability issues influencing the future design of wrist-worn platform integrating projection displays. We plan to investigate these issues further (e.g., the ability of users to keep their forearm and the projected image motionless) and to perform user studies that validate our observations. We also plan to explore scenarios that range beyond a single standing user, to address multiple

collaborating users who are seated, as well as standing. We expect that contention for potential projection surfaces may well be an issue in these circumstances.

We also demonstrated the utility of a cursorless user interface, utilizing a wrist-worn touch sensor, which allows for users to interact with projected content without the need to look at the wrist-worn device, or to visually navigate an on-screen cursor and manipulate on-screen widgets with it.

Finally, we note that although we described our experiences with a simulated wrist-worn display, since our simulator system utilizes a small orientation sensor and an untethered retroreflective target for position tracking, we are not limited to simulating wrist-worn displays alone. It is possible to try out other placement strategies (e.g., on the torso, shoulder, head, or hips, or embedded in handheld devices) for small projection systems and further explore other possibilities for interaction.

## 9. Acknowledgements

This research is funded in part by NSF Grant 0099184, ONR Contract N00014-04-1-0005, and gifts from IBM, Microsoft, and Synaptics.

## References

- [1] Symbol Laser Projection Display  
<http://www.symbol.com/products/oem/lpd.html>
- [2] NEC WT-600 Projector: Mirror projection method  
<http://www.nec-pj.com/products/wt/index.html>
- [3] Fraunhofer-Gesellschaft press release  
<http://www.fraunhofer.de/fhg/press/pi/2004/11/Mediendienst112004Thema6.jsp>
- [4] Beardsley, P., Van Baar, J., Raskar, R., and Forlines, C., "Interaction using a handheld projector," *IEEE Computer Graphics and Applications*, vol. 25, no. 1, pp. 39-43, 2005.
- [5] Blaskó, G. and Feiner, S., An Interaction System for Watch Computers Using Tactile Guidance and Bidirectional Segmented Strokes, *In Proc. of ISWC '04*, 2004, pp. 120-123.
- [6] Projective Texture Mapping  
<http://developer.nvidia.com/attach/6549>
- [7] Harrison, B.L., Fishkin, K.P., Gujar, A., Mochon, C., and Want, R., Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces, *In Proc. of CHI '98*, 1998, pp. 17-24.
- [8] Narayanaswami, C., Kamijoh, N., Raghunath, M., Inoue, T., Cipolla, T., Sanford, J., Schlig, E., Venkiteswaran, S., Guniguntala, D., Kulkarni, V., and Yamazaki, K., "IBM's Linux watch, the challenge of miniaturization," *IEEE Computer*, vol. 35, no. 1, pp. 33-41, 2002.
- [9] Pinhanez, C., The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces, *In Proc. of Ubicomp 2001*, 2001, pp. 315-331.
- [10] Raghunath, M., Narayanaswami, C., and Pinhanez, C., Fostering a Symbiotic Handheld Environment, *IEEE Computer*, 36(9), 2003
- [11] Rekimoto, J., Tilting operations for small screen interfaces, *In Proc. of UIST '96*, 1996, pp. 167-168.
- [12] Segal, M., Korobkin, C., van Widenfelt, R., Foran, J., and Haerberli, P., Fast shadows and lighting effects using texture mapping, *In Proc. of Siggraph '92*, 1992, pp. 249-252.
- [13] Teller, S., Chen, J., and Balakrishnan, H., "Pervasive pose-aware applications and infrastructure," *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 14-18, 2003.
- [14] Wobbrock, J.O., Myers, B.A., and Kembel, J.A., EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion, *In Proc. of UIST '03*, 2003, pp. 61-70.