

Pointing in Intelligent Environments with the WorldCursor

Andrew Wilson¹ & Hubert Pham²

¹Microsoft Research, Redmond, WA

²Dept. EECS, Massachusetts Institute of Technology, Cambridge, MA

awilson@microsoft.com, hubert@mit.edu

Abstract: We introduce the WorldCursor, a pointing device and cursor designed for intelligent environments. The WorldCursor is analogous to the mouse and cursor used in traditional GUIs: the user may select and interact with a physical device by positioning the cursor on the device and clicking. We use the XWand (Wilson, 2003) as a physical pointing mechanism, and couple it with the WorldCursor device which projects a cursor on the physical environment. The WorldCursor improves upon the XWand by removing the need for external positioning technology such as video cameras. We detail the device and signal processing algorithms, and discuss its application in intelligent environments.

Keywords: Sensing, input devices, intelligent environments, remote control, gesture recognition

1 Introduction

Ubiquitous computing promises to blur the boundaries between traditional desktop computing and the everyday physical world. A popular vision of tomorrow's computing pushes computational abilities into everyday objects, each participating in a complex and powerful integrated intelligent environment. Tomorrow's home and office environments, for example, may include a variety of small and large networked displays and smart controllable devices. This migration away from the desktop and "into the walls" presents several challenges for user interface design.

For example, how does the user of tomorrow's intelligent environment select one of many devices? Today, this problem is most often addressed by maintaining a separate interface, such as an IR remote control, for each device. The intelligent environment presents an opportunity for a single intelligent user interface to control many such devices. One approach is to extend the IR remote so that it can control multiple devices. Many "universal" remotes, for example, allow the user to select which of several devices to control.

A second approach is to extend the notion of the cursor and pointing device that has been so successful in today's desktop computing

environments, to operate on the physical world. By adapting the pointer and cursor for the physical environment, we may hope to avoid overly complicated interface devices that require the user to devote attention to finding the right button rather than attending to the device under manipulation: people do not look at the mouse while using a WIMP interface. We also may exploit our natural tendency to look at, point at, and talk to whatever we wish to control (Brumitt, 2001).

In this paper we introduce the WorldCursor system, which combines a physical pointer and cursor designed to provide input to an intelligent environment.

2 The XWand

In (Wilson, 2003) we introduce the XWand, a hardware device and associated signal processing algorithms that controls multiple connected devices in a natural manner. For example, an XWand user may turn on a light in the room by pointing the wand at the light and pressing the button on the wand. Figure 1 shows the XWand prototype.

The XWand system determines which device the user is pointing at by combining the orientation and 3-d position of the wand with a 3-d model of the room and the devices within it. Orientation of the wand is determined from onboard sensors, while



Figure 1: The XWand.

wand position is determined with stereo computer vision. The 3-d model of the room and devices is entered into the system by pointing with the wand itself in a special training mode. With the orientation, position and model of the room, it is easy to determine object in the world model the wand is pointing at, if any. Audio feedback is provided to indicate to the user that the object is known to the system and can be controlled by the wand, but in general little feedback is necessary since the pointing is absolute in nature.

Users of the XWand are often impressed with the immediate and natural feel of absolute pointing. However, the pure geometry-based approach which enables absolute pointing also has a number of important drawbacks:

- Two or more cameras must be permanently mounted in the room. Besides the difficulty of installation, such cameras inevitably draw objections related to privacy.
- The cameras must be carefully calibrated to the room geometry upon installation, and recalibrated if they are moved.
- At least two cameras must have clear sight-lines to the wand at all times.
- The three dimensional position of each active device in the room must be known.
- Small errors in the orientation and position information translate to inaccuracy in pointing, possibly disrupting the interaction.

Note that many of the problems associated with the computer vision-based positioning system also hold for many other positioning technologies. For example, ultrasonic tracking suffers similar problems except perhaps for certain privacy concerns unique to camera-based systems (Priyantha, 2000, Randell, 2001).

Given these objections, we set about exploring alternatives to absolute pointing, with the goal of eliminating the three dimensional positioning system.

One general approach is to place tags in the environment, but they have drawbacks as well. By design tags require installation on every active device. Active tags such as IR beacons, for example, require their own power, while passive tags such as RF ID tags only give proximity information over a limited range, and tags based on visual features rely on rather sophisticated onboard processing. See (Swindells, 2002, Olsen, 2001, Masui, 2000, Jojic, 2000) for related systems and tracking technologies.

3 WorldCursor Concept

The WorldCursor system uses the XWand device but does not rely on a geometric model of pointing that requires the three dimensional position of the wand, nor on tags placed in the environment, nor on any external sensing in general. Instead, a laser spot projected on the environment gives the user feedback as to where the system believes the user is pointing, much in the same way that the cursor icon in WIMP interfaces provides feedback to indicate where the user is pointing with the mouse.

The WorldCursor system consists of a small tele-operated motion platform upon which is mounted a laser pointer. This device is controlled via a wired connection to a host computer, which is also connected to the XWand RF base station. The WorldCursor platform is programmed to follow the motion of the XWand, such that when the user points the XWand to the left, for example, the WorldCursor moves a corresponding amount to the left in real time. The user attends to the projected laser spot (the cursor) in the environment. By moving the XWand the user is then able to place the cursor on any object in the room, as they would place the Windows cursor on an onscreen object with the mouse. Because we only use the orientation information from the XWand, and not the XWand's 3-d position, we remove the original XWand system's requirement of the external computer vision system.

Interacting with active devices in the intelligent environment proceeds much as in the original XWand system. For example, to turn a household lamp on or off, instead of pointing directly at the lamp, the user moves the laser spot onto the lamp and clicks the wand button. The system determines that the cursor is on the lamp by comparing the current cursor position with the recorded cursor position associated with the lamp, collected beforehand. Note that this closed feedback loop is not accomplished by simply mounting a laser pointer on the wand itself.

4 XWand Device

The XWand hardware device contains onboard sensors to support the computation of orientation information and gesture recognition. The following sensors are used to compute orientation:

- Analog Devices ADXL202 2-axis MEMS accelerometer. When motionless, this senses the acceleration due to gravity, and so can be used to sense the pitch and roll angle of the device.
- Honeywell HMC1023 3-axis magnetoresistive permalloy magnetometer. This senses the direction of the Earth's magnetic field in 3 dimensions, and can be used to compute the yaw angle of the device.

The values from the accelerometer and magnetometer are relayed to a host computer by radio link. These values are combined to find the absolute orientation of the device with respect to the room (see (Wilson, 2003) for details). This orientation is updated in real time at a rate of about 50Hz, and is accurate to a few degrees in each of yaw, pitch and roll axes.

5 WorldCursor Device

The WorldCursor motion platform is mounted on the ceiling near the center of the room (see Figure 2). It consists of two high speed miniature servos of the type used on radio-controlled model airplanes, one mounted for yaw and a second for pitch control. Both servos are controlled by a PIC microcontroller, which takes yaw and pitch commands from the host computer via RS-232. The servos are each capable of moving over nearly a 170 degree range, at a speed of 333 degrees per second.

Mounted on the servo assembly is a red laser similar to those used in laser pointers. By controlling the servos, the platform is able to steer the laser spot to most points in the room below the ceiling, provided there exists a sight line to that point. Effective resolution in steering the laser is about 0.25 degrees or about one half inch at a distance of 9 feet.

The motion of the laser dot is smooth and responsive. In the case when the laser must move from pointing to a point in front of the platform to a point behind the platform, there will be a discontinuity in movement as the pitch motor moves to the back and the yaw motor reflects about the vertical plane separating the front and rear hemispheres. Software API's on the host computer take yaw and pitch values as commands. The motors

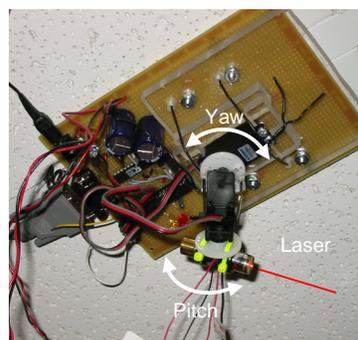


Figure 2: The WorldCursor device.

are moved as necessary to steer the laser. Finally, we mention that the ceiling may be reached with the addition of a second laser mounted opposite to the first.

6 World Model

The WorldCursor points at a given object in the room by changing the pitch and yaw of its motors, and can only point at objects that have a sight line from the WorldCursor platform. It is therefore possible to uniquely associate a given object in the room with the yaw and pitch value used to point the WorldCursor at the object. The yaw and pitch values of each object of interest in the world are the basis for a convenient world model for the WorldCursor system based on spherical coordinates.

The spherical coordinate world model is easier to construct than the full three dimensional model of the original XWand system. For example, whereas in the three dimensional model the user had to either hold the wand over the object, or provide several pointing examples used to triangulate the position of the object, the WorldCursor system need only record the current yaw and pitch values of the device once the user has put the cursor on the object. One limitation of this approach is that the learned spherical coordinate world model is invalidated when the WorldCursor device is moved to a new location, and in general it may be a difficult representation for other external systems to utilize.

To determine if the user is pointing at a given object we may compute the distance in spherical coordinates between the current WorldCursor position and the position stored for that object. For coordinates (θ_c, ϕ_c) of the WorldCursor and coordinates (θ_o, ϕ_o) of the center of given object, the user is pointing at the object if

$\sqrt{(\theta_i - \theta_c)^2 + (\phi_i - \phi_c)^2} < r_i$, where radius r_i indicates the size of the object modeled as a circle in spherical coordinates.

Because the WorldCursor may be used to indicate a point in the environment with a high degree of precision, it may be used to input a set of vertices that form a polygonal model of an object. Once the model is entered by placing the cursor on each vertex in turn, the system may determine if the cursor is on the polygon by using standard point-in-polygon algorithms used in two dimensional graphics (Haines, 1994). This can be used to determine if the WorldCursor is on the active surface of a computer display, for example.

We may also like to know where the cursor is in the surface's local coordinate system. For a four vertex polygonal surface such as a computer display, this would be the cursor's screen coordinates. This transform from WorldCursor to screen coordinates allows the display to be seamlessly incorporated into the rest of the world model, as described later.

The projective transform (Zisserman, 2000) may be used to transform WorldCursor coordinates to screen coordinates (x, y) :

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & 1 \end{bmatrix} \begin{bmatrix} \theta_c \\ \phi_c \\ 1 \end{bmatrix}.$$

The parameters p_{ij} may be determined by solving a linear system of equations given the four corners of the display in both WorldCursor and screen coordinates. Here we make the assumption that the WorldCursor coordinate system is linear in the region of the display, a valid assumption for most displays.

7 Control

In this section we present various algorithms for controlling the WorldCursor with the XWand. The general scheme of the control mapping is that the WorldCursor should mimic the exact motion of the XWand, much in the same way that the cursor on a WIMP interface mimics the motion of the mouse. Where (θ_c, ϕ_c) is the yaw and pitch of the WorldCursor and (θ_w, ϕ_w) is the yaw and pitch of the XWand we have $(\theta_c, \phi_c) \approx (\theta_w, \phi_w)$.

7.1 Filtering

Before passing on the output of the XWand to the WorldCursor controller, we first filter the yaw and pitch values from the XWand to reduce the effects of

noise in the sensors and to ease placing the cursor precisely on a small target. We use two filters which average the last n samples (box filter). The first is a very slow filter which averages the last 2.5 seconds of sensor data. This filter tends to dampen most wand motion and allows the user to move the cursor relatively slowly for precise positioning. The second filter is much faster, averaging the last 0.3 seconds of sensor data. This filter is appropriate for fast wand movement, when the sensor noise is not as apparent, and responsiveness is desired.

The WorldCursor control algorithm switches between the slow and fast filters automatically. If the fast filter is in effect, and the average speed of cursor movement (in device polar coordinates) using the fast filter falls below some threshold τ_{slow} , the controller switches to the slow filter. If the slow filter is in effect and the same average speed goes above a threshold $\tau_{fast} \gg \tau_{slow}$ the controller switches to the fast filter. This mixture of filters achieves a balance of speed, responsiveness and a fine degree of control.

7.2 Absolute vs. Relative Pointing

The most basic control mapping is to compute WorldCursor yaw and pitch (θ_c, ϕ_c) from wand yaw and pitch (θ_w, ϕ_w) as

$$\begin{aligned} \theta_c &= \theta_{c0} + \theta_w - \theta_{w0} \\ \phi_c &= \phi_{c0} + \phi_w - \phi_{w0} \end{aligned}$$

where (θ_{c0}, ϕ_{c0}) and (θ_{w0}, ϕ_{w0}) are offset angles for the WorldCursor and XWand, respectively. These offsets may be set to align the origins of the XWand and WorldCursor in a one time calibration procedure.

Unless the WorldCursor platform and the XWand are very close to one another, it will be impossible to choose offsets (θ_{c0}, ϕ_{c0}) and (θ_{w0}, ϕ_{w0}) such that the XWand points directly at the WorldCursor laser spot throughout the range of pointing in the room. It will be possible to achieve approximate correspondence for a limited range of angles such as one wall of a room, but as soon as the WorldCursor is brought onto the opposite facing wall, for example, the correspondence will be far off.

It is not clear if users require even approximate correspondence for successful use of the device. Users' experience with mice suggests that absolute correspondence is not necessary. Much in the same way that users easily adapt to the relative movement of mouse and cursor, users of the WorldCursor may be able to adapt to the lack of absolute pointing,

subject to the limitation that the laser spot is in their field of view. However, we have developed a number of mechanisms by which the WorldCursor and XWand are brought into alignment to partially restore absolute pointing. Figure 3 illustrates the absolute and relative pointing modes of the XWand and WorldCursor systems.

7.3 Clutching

If the user wishes to re-establish absolute pointing, the user may “clutch” the WorldCursor. This is an operation similar to the movement of picking up the mouse, moving it in air, and putting the mouse down on the desk again, without the cursor moving. To clutch, the user clicks the XWand button. At that point the laser dot stops moving with the wand. The user may then reorient the wand, possibly lining it up so that it points directly at the laser spot. When ready to resume WorldCursor control, the user presses the button once more and the laser spot resumes moving. At this moment new offset values (θ_{c0}, ϕ_{c0}) and (θ_{w0}, ϕ_{w0}) are collected.

Note that the user may clutch to align the XWand with the WorldCursor, but may also clutch to simply put the laser spot in their field of view, or to establish a particular desired range of operation for the XWand. For example, by clutching the user may set the offsets so that the XWand may be held comfortably at the side of the body while the WorldCursor appears on a surface in front of the user.

7.4 Exploiting Room Geometry

If the geometry of the room is known, including the position of each wall, the WorldCursor device, and the XWand itself, the WorldCursor may be controlled such that the wand always points directly at the laser spot without clutching. For a polygonal model of the room, this control algorithm proceeds by casting a ray from the wand to determine which polygon (wall) the ray intersects, and the three dimensional point of intersection on that polygon. Using simple trigonometry, this 3-d ‘wall point’ may then be related back to the position of the WorldCursor device to compute an updated value of (θ_c, ϕ_c) . Figure 4 illustrates the geometry.

By relying on the three dimensional position of the wand, we may seem to be re-introducing the very same problem that we would like to remove in developing the WorldCursor, namely the XWand’s reliance on external 3-d positioning technology. In practice, however, it is sufficient to know the room

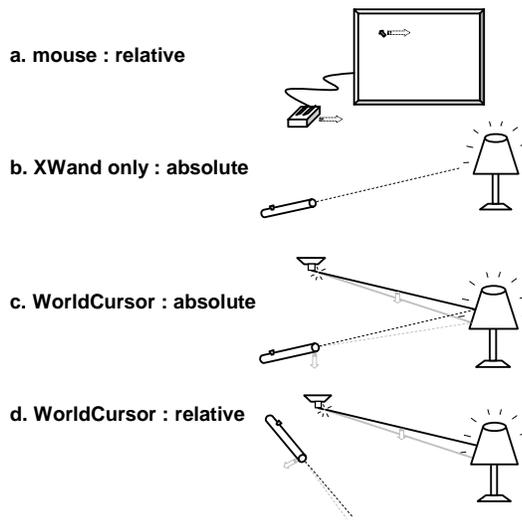


Figure 3: Mapping modes for mouse, XWand and WorldCursor.

geometry only approximately and still achieve a useable alignment that requires no clutching. For example, it will typically suffice to fix the assumed wand position in the middle of a typical office space, even though this may not coincide with the actual wand position. Again, we rely on the fact that users tend to be tolerant to constant offsets in alignment similar to that found in using relative pointing mechanisms such as mice.

7.5 Inferring 3-D Position

If the geometry of the room is known, we may infer the 3-d position of the XWand from a number of clutching operations. If the user is clutching so that the XWand points at the laser spot, each clutching operation provides information that can be related mathematically to the 3-d position of the XWand.

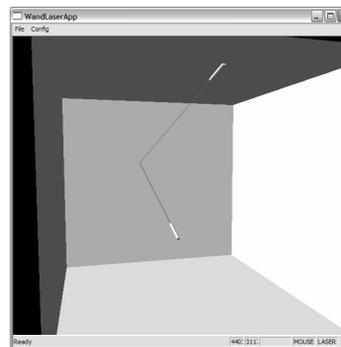


Figure 4: Software system illustrating XWand (lower white cylinder) and WorldCursor device (upper white cylinder), where room geometry is known.

After a few such clutching operations, it is possible to compute the 3-d position of the XWand.

For each clutching operation we have an associated wall point \mathbf{p}_i and vector \mathbf{w}_i that is the ray pointing along the wand. These values are collected at the end of the clutching operation, after the user has realigned the wand with the laser spot and pressed the wand button to resume WorldCursor control. Assuming the wand position \mathbf{x} remains constant over a number of successive clutching operations, we have

$$\mathbf{x} + s_i \mathbf{w}_i = \mathbf{p}_i.$$

Wand position \mathbf{x} may be found by solving this linear system of equations via least squares. A minimum of two clutching operations is required, but for robustness it may be desirable to collect several, particularly if some of the wand rays \mathbf{w}_i are similar, in which case the solution will be sensitive to small errors in \mathbf{w}_i . In practice, an implementation may dynamically choose the number of clutching operations to use based on how different they are to each other.

Once the estimate of the wand position has been updated, the control algorithm described in Section 7.4 may be used. So long as the actual position of the XWand does not change dramatically, no more clutching operations will be necessary to maintain XWand-WorldCursor correspondence. This online calibration requires no more clutching operations than the system which does not exploit the approximate room geometry, and in the long run requires fewer clutching operations if the user does not move about the room often. The user would only be aware of the algorithm in that after a while no more clutching operations would be required to keep the cursor and the wand in alignment as the move from one side of the room to another.

Finally, we suggest that if the positions of the wand and the WorldCursor devices are known, it may be possible to similarly solve for unknown wall points \mathbf{p}_i . In this way the 3-d position of objects and devices in the room may be learned, much as in the original XWand system. With sufficiently many clutching operations it may also be possible to learn the entire room geometry, including a polygonal model of the walls that is presently specified manually. Even in this scenario, no more clutching operations would be necessary than in the case of not using a geometric model.

These explorations in using the information in a clutching operation assume that the user is clutching to re-establish wand and cursor alignment. Without

making some assumptions about the movement of the user in the room, it is difficult to determine whether the user is following this convention. This is a drawback of the approach of exploiting room geometry. We note, however, that most users are likely to use such a device from one a small set of static locations, much as the TV remote control is likely to be used from either the couch or the easy chair. Prior knowledge of these locations may be useful in guiding our geometry-based control algorithms.

8 Applications

8.1 Home Automation

The WorldCursor system is capable of performing all the home automation-related tasks of the original XWand system, including turning on and off lights via X10, selecting and manipulating the Media Player with gestures to control track and volume, and finally selecting and controlling a cursor on a Windows display.

The WorldCursor improves on the XWand system by giving the user much more precision in selecting devices. In the original XWand system, for example, it is difficult without audio feedback to select one of two devices that are located 18 inches apart from 12 feet away. This is due not only to the limited precision of the XWand system, but also users' limited precision in pointing, which can be easily demonstrated by having users turn on a laser pointer *after* pointing it at a given target. This uncertainty can be addressed by placing a laser pointer (always on) on the XWand itself, but only up to the precision of the XWand signal processing algorithms.

We have exploited this precision in one installation where to control the Media Player, the user puts the WorldCursor on one of several paper icons (play, pause, next track, previous track) hung on the wall. The user 'pressed' the Media Player button by pushing the wand button. Menus and other traditional GUI metaphors may also be used. A drawback is that these GUI elements are always present in the environment when they are not needed, and they are not animated. We note, however, that in combination with the IBM Everywhere Displays (Pinhanez, 2001), the WorldCursor could be used to 'right-click' an object in the environment to trigger the Everywhere Displays unit to display a context sensitive menu next to the physical object, with the WorldCursor used to select a menu item.

8.2 Display Surfaces

In the original XWand system, the user may control a cursor on a Windows console by first pointing the wand at the display, and entering a cursor control mode by clicking the wand button. Exiting the cursor control mode is accomplished by clicking on a special button presented in the Windows interface during cursor control mode.

With the precision afforded by the WorldCursor, it is possible to improve upon this interaction by seamlessly integrating the display surface in the world model, without needing to enter a special cursor control mode. For example, once the four corners of the display are specified with the WorldCursor, the projective transform calculations described in Section 6 may be used to determine if the cursor is on the display. If the cursor is on the display, the laser is turned off, and the prospective projection equations are used to move the Windows cursor to the exact spot on the display where the laser would be if it had not been turned off. Once the user moves the cursor off the display, the Windows cursor is hidden and the laser is turned back on. Because of the nature of the WorldCursor geometric model, the registration between the two coordinate systems can be quite precise. Figure 5 illustrates the interaction.

With networked WorldCursor client processes it will be possible to extend this functionality across multiple displays simultaneously, moving from display to display seamlessly.

The drawback of the WorldCursor display integration is that the control of a small display from across the room can be quite difficult, since the size of the ‘mousing surface’ is proportional to the angle subtended by the display. In the XWand system, this problem is avoided in the special cursor control mode by using a constant scaled range of angles for control that is independent of where the user is located in the room. One approach to address this limitation is to nonlinearly warp the coordinate system in the neighborhood of the display.

8.3 Learning Geometric Models

The algorithms for learning geometry from clutching operations presented in Section 7.3 raise interesting possibilities for inferring geometry over the normal use of the device, without relying on a special training mode or a lengthy upfront training session. This has interesting implications for ubiquitous computing applications, many of which rely on geometric models to develop location based services, and to reason about user co-presence and proximity to devices. One significant barrier to deploying such

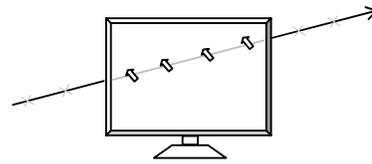


Figure 5: Integrating the WorldCursor seamlessly with the Windows cursor.

systems will be the construction of the necessary geometric models, especially in the home.

Besides using the WorldCursor to ‘point out’ devices to the system, it may also be used by the system to ‘point out’ devices to the user. For example, if the intelligent environment knew where the user’s lost keys were, the WorldCursor system might direct the user’s attention to their location in response to the user’s query.

8.4 Non-Veridical Behavior

Thus far we have considered WorldCursor control and applications which strive to replicate the exact motions of the user’s manipulations of the wand. Here we suggest that there may be interesting possibilities in considering how this one to one mapping may be violated.

For example, in (Myers, 2002) users’ ability to finely control a standard laser pointer in an object selection task is studied. In part, performing object selection and other GUI related tasks with a laser pointer is difficult because it is surprisingly difficult to hold a laser pointer still. In Section 7.1 we introduce a switching filter algorithm designed to significantly dampen the motion of the cursor when the user is moving the wand slowly. The WorldCursor can thus do for laser pointer-based interaction what image stabilization has done for home videos, by eliminating the jitter and noise associated with standard laser pointers. This damping filter is especially useful in working with display surfaces, where often GUI elements are small and densely packed.

In an environment where there are many selectable objects, it may be advantages to explore a ‘Snap To’ feature for the WorldCursor. If the user guides the cursor near an active device, a simple spring model may be used to bring the cursor precisely on the target. Not only may this ease target selection, it is also a way to indicate to the user which objects are active and selectable.

Lastly, in the case when the XWand is used with gestures to control the currently selected device, we may use the WorldCursor to ‘play back’ the gesture

as a way to teach the user the available gestures for that device.

9 User Experience

The WorldCursor has been demonstrated to several hundred people, some of which have used the device to turn lights on and off, manipulate the cursor on the display, and control the Media Player. All users found the use of the WorldCursor immediately understandable, including the clutching operation. Many are captivated by the fluid and responsive motion of the laser dot. However, it is interesting to note that many people do not find its implementation apparent. If the WorldCursor is clutched so that the wand and laser spot is aligned, many external observers at first conclude that a laser pointer is mounted on the wand, and that some external sensing mechanism is employed. Those that recognize that the laser pointer is on the ceiling often ask why the laser is not instead mounted on the wand itself.

As with the mouse, users of the WorldCursor attend the cursor and almost never look at the wand itself. In some cases users have difficulty finding the laser spot, particularly in the case when the cursor is 'parked' at a location in the environment out of the field of view, as in the beginning of a clutching operation. One solution to this problem is to bring the cursor to a 'home' position after a period of inactivity, or to animate the laser spot to draw the user's attention when the wand is first picked up.

While a user study evaluating the differences between relative and absolute pointing using the WorldCursor remains future work, our informal observation is that if the offset while using relative motion is not too great, users are adept at selecting objects. It may be the case that inexperienced users find absolute pointing easier to grasp at first, however.

10 Conclusion

We have introduced the WorldCursor system, a novel user interface to intelligent environments that addresses the need to select and manipulate physical devices without using the mouse and keyboard.

The WorldCursor improves upon the XWand system by not requiring an external position sensing technology, and by enabling the user to point with a high degree of precision. The WorldCursor is inspired mainly by the mouse and cursor of today's WIMP interfaces. We believe this strategy of patterning tomorrow's user interface for the intelligent environment after today's GUI interfaces

has some merit, in that we may draw upon interaction paradigms that users already know, and thus ultimately ease the adoption of tomorrow's ubiquitous computing platforms.

References

- Brumitt, B., and JJ Cadiz (2001), "Let There Be Light": Examining Interfaces for Homes of the Future, in *Human-Computer Interaction: INTERACT '01*(Ed, Hirose, M.), pp. 375-382.
- Haines, E. (1994) In *Graphics Gems IV*(Ed, Heckbert, P.) Academic Press, pp. 24-46.
- Jojic, N., B. Brumitt, B. Meyers, S. Harris, and T. Huang (2000), Estimation of Pointing Parameters in Dense Disparity Maps, in *IEEE Intl. Conf. on Automatic Face and Gesture Recognition* Grenoble.
- Masui, T., and I. Silo (2000), Real-World Graphical User Interfaces, in *HUC*, pp. 72-84.
- Myers, B. A., R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long (2002), Interacting At a Distance: Measuring the Performance of Laser Pointers and Other Devices, in *Proceedings CHI* Minneapolis, Minnesota.
- Olsen, D. R. J., T. Nielsen (2001), Laser Pointer Interaction, in *Proceedings CHI* Seattle, pp. 17-22.
- Pinhanez, C. (2001), The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces, in *Proc. of Ubiquitous Computing 2001* Atlanta, Georgia.
- Priyantha, N. B., A. Chakraborty, H. Balakrishnan (2000), The Cricket Location-Support System, in *Proceedings 6th ACM MOBICOM* Boston, MA.
- Randell, C., and H. Muller (2001), Low Cost Indoor Positioning System, in *Proceedings Ubicomp*(Ed, Abowd, G. D.) Springer-Verlag, Atlanta, Georgia, pp. 42-48.
- Swindells, C., K. Inkpen, J. Dill, and M. Tory (2002), That one there! Gesture for on-demand device identity, in *Proceedings UIST* Paris.
- Wilson, A. (2003), XWand: UI for Intelligent Environments, in *CHI 2003*.
- Zisserman, H. R. a. A. (2000) *Multiple View Geometry in Computer Vision*, Cambridge University Press.