XWand: UI for Intelligent Spaces

Andrew Wilson **Steven Shafer**
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
awilson@microsoft.com stevensh@microsoft.com

ABSTRACT
The XWand is a novel wireless sensor package that enables styles of natural interaction with intelligent environments. For example, a user may point the wand at a device and control it using simple gestures. The XWand system leverages the intelligence of the environment to best determine the user's intention. We detail the hardware device, signal processing algorithms to recover position and orientation, gesture recognition techniques, a multimodal (wand and speech) computational architecture and a preliminary user study examining pointing performance under conditions of tracking availability and audio feedback.

Keywords
Sensing, gesture recognition, hardware devices, multimodal interfaces, intelligent environments

INTRODUCTION
Increasingly our environment is populated with a multitude of intelligent devices, each specialized in function. The modern living room, for example, typically features a television, amplifier, DVD player, lights, and so on. In the near future, we can look forward to these devices becoming more interconnected, more numerous and more specialized as part of an increasingly complex and powerful integrated intelligent environment. This presents a challenge in designing good user interfaces.

For example, today's living room coffee table is typically cluttered with multiple user interfaces in the form of IR remote controls, each covered with many buttons. Often each of these interfaces controls a single device, and requires the user to devote attention to finding the right button rather than attending to the device under control. Tomorrow's intelligent environment presents the opportunity to present a single intelligent user interface to control many such devices when they are networked. What will this interface look like?

CHI 2003, APRIL 5–10, 2003, FT. LAUDERDALE, FLORIDA, USA.

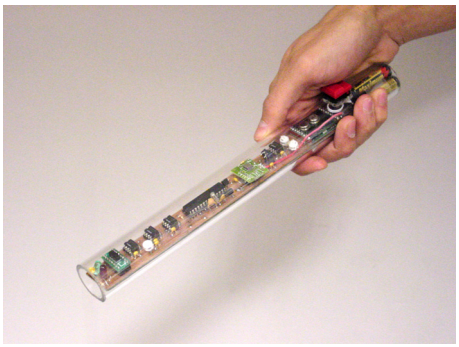

Figure 1: The XWand.

Here we present the XWand, a hardware device (Figure 1) and associated signal processing algorithms for an interface that may control multiple connected devices in a natural manner. The main idea is that the user should merely point at the device to be controlled, and use simple gestures or speech to control the device. The intelligent environment system interprets the user's manipulation of the wand to determine an appropriate action in context. The ultimate goal of such a natural interface is to provide an interface that is so simple that it requires no particular instruction or special knowledge to use, and instead relies on the intelligence of the environment to figure out what to do.

For example, the user may turn on a light in the room by pointing the wand at the light and pressing the button. Alternatively, the user may point the wand at the light and say "turn on". The user may then point the wand at the stereo amplifier and roll clockwise or counter-clockwise to turn the volume up or down.

Part of the motivation of the design is to exploit our natural tendency to look at, point at, and talk to whatever we wish to control [1]. We would also like to exploit the complementary nature of speech and gesture in our everyday interactions.

HARDWARE DEVICE
We have constructed an early hardware prototype of the XWand, a handheld device which embeds a variety of sensors which in combination support pointing and gesture recognition tasks. The XWand has the following features:

- 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.
- Murata ENC-03 1-axis piezoelectric gyroscope. This is an angular rate sensor, and is placed to sense motion about the vertical axis (yaw).
- BIM 418MHz FM transceiver (38kbps). The transceiver is used to send and receive digital information to a matching base station, which then communicates to a host PC via RS-232. Continual polling by the host yields a 50Hz frame rate.
- PIC 16F873 flash-programmable microcontroller running at 20MHz. The microcontroller reads each of the sensor values, formats data communication packets, decodes received packets, controls timing, power management, etc.
- Infra-red (IR) LED. Invisible to the naked eye, this LED can be seen by cameras equipped with an IR pass filter. This is used to support position tracking of the wand.
- Green and red visible LEDs. These can be used to display status information. Note that because the wand is equipped with a radio transceiver, these LEDs may be lit in response to commands received from the host PC.
- Pushbutton.
- 4 AAA batteries. Quiescent current when awake is approximately 52mA, less than 1mA while asleep.

This particular combination of sensors is similar to that found in [2, 3], but will be used to recover true 3-d orientation information. In the next two sections we describe how the output of the accelerometer and magnetometer may be combined to compute the full 3-d orientation of the wand with respect to the room, and how computer vision techniques may be used to find the 3-d position of the wand using the IR LED. The orientation and position of the wand may be used to compute what the user is pointing at with the wand, given a 3-d model of the room and its contents. This geometric approach contrasts with other related systems that rely on tags embedded in the environment [4, 5].

ORIENTATION

Each of the 3 orthogonal axes of the magnetometer senses the degree to which it lies along the direction of the Earth's magnetic field. This is not enough information however to compute a full 3-d rotation. For example, if you rotate the magnetometer about magnetic north, none of the sensor readings change.

Similarly, if you slowly rotate the accelerometer about the direction of gravity, neither of the accelerometer readings change. However, we can combine the magnetometer and accelerometer outputs to find the full 3-d orientation of the wand. The main idea is to take the accelerometer outputs as pitch and roll, and then use the output of the magnetometer to compute yaw. The calculation of yaw from the magnetometer takes into account the pitch and roll information as follows. First take range-normalized (in [-1, 1]) accelerometer values as the pitch and roll, and form the 3-vector $\mathbf{m}$ from the similarly range-normalized magnetometer values. The pitch and roll then corrects the output of the magnetometer:

$$\mathbf{m}_c = \mathbf{R}_{a_1,a_2,0}\mathbf{m}$$

where $\mathbf{R}_{\theta_x,\theta_y,\theta_z}$ is the Euler angle rotation matrix about x, y and z axes. Let $\mathbf{N}$ be the output of the magnetometer when the device is held flat, lying along the y axis, (yaw, pitch, roll) = (0, 0, 0). Project onto the ground plane and normalize:

$$\mathbf{m}_p = \begin{bmatrix}1 & 1 & 0\end{bmatrix}^{\mathrm{T}}\mathbf{m}, \mathbf{N}_p = \begin{bmatrix}1 & 1 & 0\end{bmatrix}^{\mathrm{T}}\mathbf{N}$$

$$\mathbf{m}_{np} = \frac{\mathbf{m}_p}{\left\|\mathbf{m}_p\right\|}, \mathbf{N}_{np} = \frac{\mathbf{N}_p}{\left\|\mathbf{N}_p\right\|}$$

Yaw is then computed as the angle between

$$\text{yaw} = \text{sign}(\mathbf{m}_{np} \times \mathbf{N}_{np}) \cos^{-1}(\mathbf{m}_{np}^{\mathrm{T}}\mathbf{N}_{np})$$

The range of the magnetometer is computed online by twirling the wand for a minute or so. The range of the accelerometer is found statically.

There are a number of caveats to this approach. First, the accelerometers only give true pitch and roll information when the device is motionless. This problem can be avoided by relying on the orientation information only when the device is motionless, as determined by no change in the magnetometer and accelerometer outputs. Secondly, magnetic north can distort unpredictably in indoor environments and in close proximity to large metal objects. In practice, we have found that for typical indoor office environments, magnetic north does not always agree with magnetic north found outdoors, but typically will be fairly constant throughout a typical office or living room. Note that if the directions of magnetic north and gravity are co-linear, the above calculations for yaw will fail.

POSITION

A number of techniques may be used to recover the 3-d position of the device. Acoustic-based tracking techniques are popular in similar applications [6, 7]. Presently we use a computer vision technique which is capable of computing the 3-d position of the wand to an accuracy of an inch or two. While research in computer vision has focused on the difficult problem of object tracking (see [8] for an example of recognizing pointing

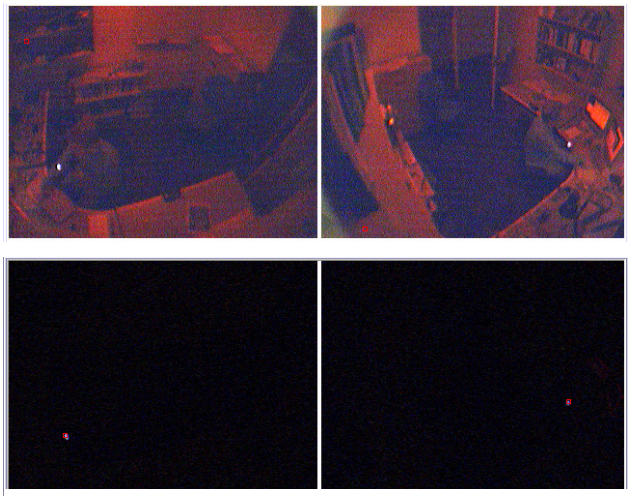

Figure 2: Computer-vision based positioning system. 2 cameras near the ceiling on either side of the room are equipped with wide angle lenses and IR pass filters. Top: Unprocessed input from cameras (with IR pass filter), looking on an office scene, with IR LED (bright dot). Bottom: processed images show only the IR LED.

from computer vision alone), we have the advantage of being able to put a marker on the device. The position system works by finding the 2-d position of a flashing IR LED on the device from two different video cameras trained on the room. These 2 2-d observations are then combined to find the 3-d position by triangulation.

Each of two Firewire video cameras is equipped with an IR pass filter. On the device, an IR LED is flashed for 3ms duty cycle at 15Hz, while the video cameras are set to acquire images at 30Hz. Thus the IR LED is present in one image and absent in the next. The pixel values of each successive image are then subtracted to obtain an image which reveals only the IR LED (Figure 2). This bright spot is then located by finding the maximum pixel value in the image. This process is applied to the images (320 by 240 pixels) acquired from both cameras, and takes less than one third of the CPU time of a 1GHz Pentium III processor.

We use standard computer vision techniques to find the 3-d position of an object from 2 2-d observations (see [9]). The system requires the position, focal length, lens distortion parameters and other parameters of each camera, computed by camera calibration procedures that are well known in the computer vision literature. This process involves choosing a coordinate system for the room. We choose a natural origin and coordinate axes, such as the corner of the room. Note that the calibration of (yaw, pitch, roll) = (0, 0, 0) must be consistent with the choice of spatial coordinate systems.

POINTING AT TARGETS

With the position and orientation of the wand determined, we now consider the task of determining if the wand is pointing at a known target. Each object of interest is

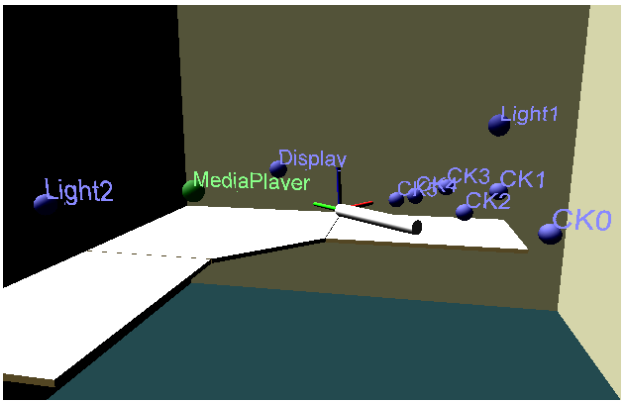

Figure 3: A 3-d graphics visualization of the wand world model with several trained targets in an office space. The wand (foreground) is shown as a white cylinder and coordinate axes.

modeled as a 3-d Gaussian blob with mean $\boldsymbol{\mu}_i$ and covariance Σ_i. Multivariate Gaussians are probability distributions that are easy to learn from data, and can coarsely represent an object of a given size and orientation.

A simple technique is to evaluate the Gaussian distribution at the point that is same distance away as the wand is from the target, but lying along the ray cast by the wand. The likelihood of pointing at target i is then

$$l_i = g\left(\mathbf{x} + \left\|\boldsymbol{\mu}_i - \mathbf{x}\right\| \mathbf{w}, \Sigma_i\right)$$

where $\mathbf{x}$ is the position of the wand, $\mathbf{w}$ is the ray along the wand, and $g(\boldsymbol{\mu}, \Sigma)$ is the multivariate Gaussian probability distribution. If the wand is not in motion, the target for which l_i is greatest above some minimum threshold is taken as the current selected target.

TRAINING TARGETS

It is easiest to use the wand to enter the position and shape of each target into the model of the room. A world model of an office space with several trained targets is shown in Figure 3. This begins by the user entering a target training mode, and specifying which target is to be trained.

A simple method to train target location is to hold the wand at the target's location. We collect a series of 3-d wand position $\mathbf{x}_i$ observations and take the mean $\boldsymbol{\mu}$ and covariance of those observations Σ as the center and shape of the object. This method has the drawback the object being trained must be in the line of sight of both cameras or the tracking will fail. This can be a problem when training a set of objects arrayed along a wall.

A second method to specify location is to use the wand to point at the target from various positions throughout the room, and compute the mean and covariance for the target from the intersection of these rays. Minimally two such observations are required. In particular, if the wand is at position $\mathbf{x}_i$ and pointing along the ray $\mathbf{w}_i$ for the i th

pointing observation, we find $\boldsymbol{\mu}$ by solving the linear system of equations

$$\mathbf{x}_i + s_i \mathbf{w}_i = \boldsymbol{\mu}$$

via least squares, where the distances s_i are unknown. The covariance of the target can be computed by a minimum covariance Σ_0 added to the spread of the differences between calculated target location $\boldsymbol{\mu}$ and its multiple estimates $\mathbf{x}_i + s_i \mathbf{w}_i$:

$$\Sigma = \Sigma_0 + (\mathbf{x}_i + s_i \mathbf{w}_i - \boldsymbol{\mu})(\mathbf{x}_i + s_i \mathbf{w}_i - \boldsymbol{\mu})^T$$

In these target training methods, the shape of a target may be modified by adding any number of pointing observations along the body of the target.

POINTING ACCURACY

The accuracy in pointing with the wand depends on the accuracy of tracking and orientation information. Orientation calculations are subject to errors due to local magnetic field distortions, misalignment of the magnetometer in relation to the accelerometer, and imprecise calibration. Ultimately, we are concerned with angular error in pointing, as this determines performance in target selection and what size targets can be selected in an indoor environment. Here we wish to characterize the performance of the device, and so neglect errors due to users' imprecision in pointing.

To characterize the angular error we may compare the wand direction vector $\mathbf{w}$ against the direction from the wand position to the pointing target: $\boldsymbol{\mu} - \mathbf{x}$. We can avoid errors due to calibration of the tracking system to ground truth world coordinates if we estimate $\boldsymbol{\mu}$ from a series of wand observations as described in the previous section.

In this procedure, a laser pointer is taped to the wand. Several pointing observations are then collected during which the user is careful to line up the laser spot on the same precise location on a given target. This eliminates any error due to the user's pointing abilities. The average angular error may then be computed from the set of pointing observations as the average difference between wand direction and direction from wand position to target position (the true target direction). In one experiment, observations taken over a 6' volume yielded an average angular error of less than three degrees. Without reference to ground truth position, this estimate of error is a measure of the internal accuracy and repeatability of the wand pointing and target training routines.

GESTURE RECOGNITION

As described above, the orientation and position of the wand may be found by a combination of sensors and signal processing techniques. This allows a pointing algorithm that is based on a geometric model of the room and the objects of interest. A target may then be selected by the user by pointing at the target and holding the wand motionless, at which point the orientation calculation described above is precise. This pause in movement may also be detected and used to indicate the user's intention to point at a given target rather than pass over it for another.

We are also interested in allowing the user to control devices associated with the target by gesturing with the wand. Here we describe a simple approach based on one or more of the accelerometer, gyro, position and orientation values. See [3] for a related application of gesture recognition with a device similar to the XWand.

We exploit very simple gestures and gesture recognizers based on the instantaneous values of the sensors and their derivatives, while relying on the context of the interaction to map the gesture appropriately. For short and simple gestures a recognition strategy is to look for simple trends or peaks in one or more of the sensor values. For example, pitching the wand up may be detected by simply thresholding the output of the accelerometer corresponding to pitch. Clearly such an approach will admit many false positives if run in isolation. However, in a real system the gesture will be performed in the context of an ongoing interaction, during which it will be clear when a simple pitch up indicates the intent to control a device in a particular way. For example, the system may only act on the gesture recognition results if the user is also pointing at an object, and furthermore only if the gesture applies to that particular object. In this way simple gesture recognizers coupled with strong context models may be more robust overall than a system relying on very specific gesture models that are prone to failure due to individual differences that are not captured during training. The present system uses this strategy, and further reduces the risk of false positives by requiring the user to press and hold down the wand button while gesturing.

Requiring the user to press the button while gesturing allows the system to easily determine when a gesture begins. In the present system the start of the gesture indicates a natural origin from which to detect trends in sensor values. Continuing the up motion example, "up" in the context of pointing at an object on the floor means pitching up from a pitched down position. The gesture recognition process records the sensor readings at the time the user presses the button and uses them as an origin for subsequent sensor readings. In the context of gesturing while pointing at an object, this process sets up a local coordinate system around the object, so that "up", "down", "left" and "right" are relative to where the object appears to the user.

MULTIMODAL INTERPRETATION

The complementary nature of speech and gesture is well established. It has been shown that when naturally gesturing during speech, people will convey different sorts of information than is conveyed by the speech [10]. In

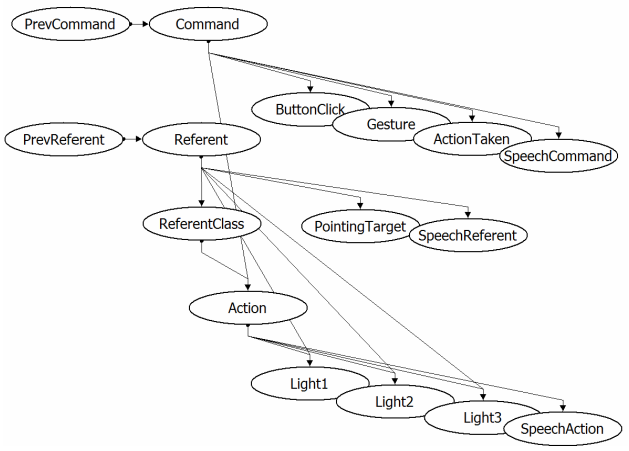

Figure 4: Dynamic Bayes network used in combining wand (PointingTarget, Gesture, ButtonClick), speech input (SpeechReferent, SpeechCommand, SpeechAction), and world state (Light1, Light2, Light3) to determine the next action (Action) as a combination of command (Command) and referent (Referent) and past beliefs (PrevCommand, PrevReferent).

more designed settings such as interactive systems, it may also be easier for the user to convey some types of information with either speech or gesture or a combination of both. For example, if the user has selected the stereo amplifier, it may be possible to say "up volume" a number of times until the desired volume is reached, but it is likely to be more convenient and more precise to give the user a volume knob to turn. When using the wand, this can be accomplished by pointing at the stereo and rolling the wand clockwise.

We have developed a simple framework for combining the outputs of various modalities such as pointing targets, wand gestures, and speech, to arrive at a unified interpretation that instructs the system on an appropriate course of action. This framework decomposes the desired action (e.g., "turn up the volume on the amplifier") into a command ("turn up the volume") and referent ("amplifier") pair. Presently, the referent may be determined from the wand pointing target or speech recognition events, while the command may be specified by wand gesture, a button press event, or a speech recognition event. With this command/referent representation, it is possible to effect the same action in multiple ways. For example, all the following actions on the part of the user will result in a light turning on:

- Say "turn on the desk lamp"
- Point at the lamp and say "turn on"
- Point at the lamp and perform the "turn on" gesture
- Say "desk lamp" and perform the "turn on" gesture
- Point somewhere closer to the desk lamp than the floor lamp and say "lamp" and perform the "turn on" gesture
- Point at the lamp and click the button

where the last example relies on the fact that the default behavior when the lamp is off and the button is clicked is to turn the lamp on.

The speech recognition system is based on a simple command and control (CFG) style grammar, with predetermined utterances for the various objects and simple command phrases that apply to the objects. The user is required to wear a wireless lapel microphone to use the speech recognition system. We would like to incorporate a microphone into the wand in a future hardware design.

By unifying the results of pointing detection and speech recognition, the overall speech recognition is more robust. For example, a spurious recognition result "volume up" while pointing at the light is ignored. Our overall motivation in working with speech is to show that with strong contextualizing cues provided by devices such as the wand, speech recognition may be made more robust [11]. We note that while speech clearly has enough expressive power to make the wand unnecessary, relying on speech alone can be difficult in practice. In many environments speech recognition may be too unreliable to use exclusively, particularly with an open microphone. Secondly, by exploiting pointing with the wand, we avoid the problem of determining the object of the user's speech, as well as whether the speech is intentionally directed to the system (no push to talk signal is necessary).

BAYES NETWORK

Multimodal integration is accomplished by a dynamic Bayes network [12] which encodes the various ways that sensor outputs may be combined to find the referent, command, and action. This network is illustrated in Figure 4. When the wand is pointing at the light, the PointingTarget variable in the Bayes net is set to Light1, for example. This causes the Action node to assign equal probability to the "TurnOnLight" and "TurnOffLight" variable settings, since these are the only admissible actions on lights. When the user then says "turn on", the speech node is set to "TurnOn" and the distribution over the Action node collapses to "TurnOnLight". The system then takes the appropriate action to turn on the light.

Bayes networks have a number of advantages that make them appropriate to this task. First, it is easy to break apart and treat separately dependencies that otherwise would be embedded in a very large table over all the variables of interest. Secondly, Bayes networks are adept at handling probabilistic (noisy) inputs. Although this remains future work, it is possible to train the dependencies in the network so that, for example, the system learns that Target3 is the desk lamp when the user points at Target3 and utters the phrase "desk lamp". Lastly, as the example above detailing the change in the Action distribution illustrates, the network represents

ambiguity and incomplete information that may be used appropriately by the system. For example, if the user doesn't point at the light, the system might ask which light is meant after hearing the utterance "light". Similarly if there is exactly one thing to be done with a light, such as toggling on or off, the system will appropriately reflect that in the distribution over the action node after the user points at the light and will require no further clarification.

The dynamic Bayes network also performs temporal integration. The PrevCommand and PrevReferent nodes hold the distribution over the Command and Referent over the previous moment in time. These heavily influence the distribution over the same variables in the current time step, such that the network tends to hold a memory of the current command and referent which decays over time, and it is thus unnecessary to specify the command and referent at exactly the same moment in time. This propagation occurs four times a second.

The Bayes network also has the ability to incorporate device state in its interpretation. For example, Light1 holds the state (on or off) of Light1. The associated distribution over this variable and its parents, Action and Referent, are configured so that the only admissible action with Light1 when it is on is to turn it off, and likewise when it is off the only action available is to turn it on.

DEVICE CONTROL

We have assembled a demonstration of the wand used to control a variety of devices in a living room-like scenario. The user may control the following with the wand:

- X10 lighting: Multiple lights in the room may be turned on and off by pointing and clicking, or uttering the phrases "turn on" and "turn off". The lights may be dimmed or brightened by gesturing down and up.
- Windows Media Player: Pointing and clicking starts the media player playing or pauses it (Figure 5). Rolling left or right changes the volume, gesturing up and down moves the previous and next tracks in the play list. "Volume up", "volume down", "next" and "previous" utterances are mapped appropriately.
- Cursor control: Pointing and clicking at the computer display gives control of the cursor to the wand, with the wand button taking the function of the left mouse button. See [13] for considerations in designing GUIs used with pointing devices away from the desktop.
- Color Kinetics lights: Pointing at these special computer controlled arrays of red, green, and blue lights brightens them over time. Rolling left and right changes the red, green and blue combination sent to the selected light, changing the light's color. When the user points away, the color gradually decays.

For the demonstration system, audio feedback is provided when the selected target changes. This audio feedback assures the user that the object pointed to has been selected, and is currently the same for all objects. In addition, the green LED on the wand is lit by commands from the host when the wand is pointing at any object known to the system.

Figure 5: Controlling the Media Player, with X10 controlled lights and video camera shown.

The wand demonstration system was shown to several hundred people in a technical conference and trade show setting. The overall feedback was overwhelmingly positive, many people inquiring when the device would be commercially available, and many referring to it as the "magic wand", Harry Potter's wand, the über-universal remote, and so on.

People that tried the wand needed a few hints on pointing, such as that it is necessary to hold the wand motionless for a moment and listen for audio feedback to verify the target selection. Everyone found the gestures easy to learn once they were demonstrated. People often assumed that special sensors were embedded in the various objects in the room to sense the wand pointing. When the full system was explained to them, many of these people asked if the system would "know" if one of the objects were moved.

Experience in showing the device has highlighted an important advantage the XWand has over standard button-laden remote controls: users maintain their visual attention on the device under control, not the controlling device as is typically required using a remote.

USER STUDY: POINTING PERFORMANCE

Motivation

One common concern regarding the XWand system is that it presently requires two calibrated video cameras. Besides requiring installation and calibration of video cameras, the vision system has the drawback that in order for the wand to be tracked, at least two cameras must be able to see the IR LED. For the wand to work well throughout the room, more than two cameras may be required to ensure that at least two cameras can see the IR LED. Clearly, the acceptance of the XWand or a related device is limited by the limitations imposed by the installation and calibration of the cameras. Note that many other positioning technologies have similar infrastructure requirements.

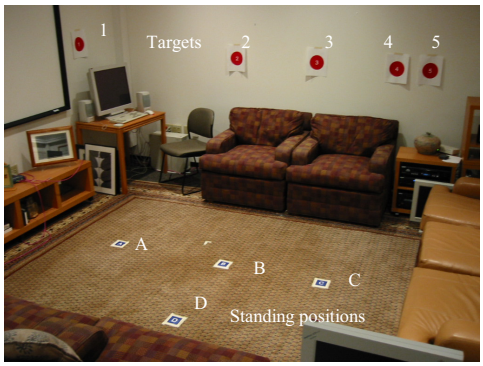

Figure 6. Physical layout of targets and standing positions used in the experiment. Users are instructed to stand at one of the 4 positions and point and click at one of 5 targets.

But is precise tracking necessary for users to make use of the wand? Perhaps users will still be able to select targets if the tracking is only approximate, or if the virtual position of the wand is fixed at a known location that is central to the room, or fixed at one of a few important locations, such as the user's favorite chair. Here we describe a preliminary user study in which we test the pointing performance of wand users when the tracking system is disabled, with the wand placed at a fixed position in the room. We also test how audio feedback may play a role in aiding the performance in this task. The hypothesis is that with appropriate audio feedback users may approach levels of pointing performance achieved when tracking is enabled. See [14] for related studies studying basic pointing performance using other devices such as laser pointers.

SUBJECTS

Ten male subjects were selected from around the research lab to participate in this study. Though a few of them had seen the wand before the study, none had used the wand.

Experimental design

A two factor within-subject design was employed. Independent variables were the use of tracking (tracking/no tracking), the use of audio feedback (audio feedback/no audio feedback), and the position of the subject in the room. Dependent variables were accuracy in pointing, time to completion of a pointing task, and responses to a post-task questionnaire.

In the 'tracking' condition, the computer vision system was used along with orientation from the wand to sense target selection by pointing. In the 'no tracking' condition, the position information was ignored and the position of the wand was fixed at 50 inches directly above a specially marked position over the floor. In the audio feedback condition, the user heard a special sound when the target selection changes, and then only when the wand is held motionless for a brief period (less than 1s). This sound was the same for all targets. Also in the audio feedback condition, a success or failure sound was played when the correct or incorrect target was selected.

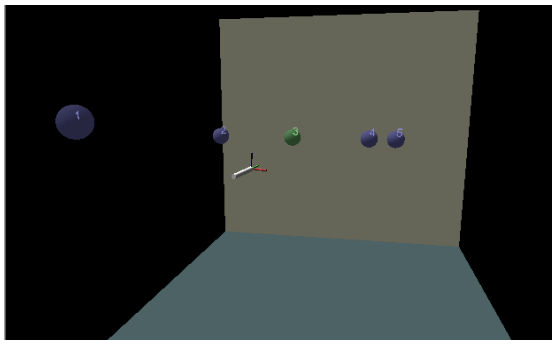

Figure 7: Graphical representation of the wand and targets used in the experiment, with the wand fixed directly above position B in the 'no-tracking' condition.

Procedure

Five round red printed targets each with a clearly visible number were hung along one side of a typical living room environment. These were spaced such that target 1 was spaced 58 inches to the left of target 2. Targets 2, 3, and 4 were spaced at 3 foot intervals along the wall, and finally target 5 was spaced 1 foot to the right of target 4. Target 1 was hung at a height of 60 inches, while the remaining targets were hung at 50 inches.

Four blue square labels were taped to the floor, marked 'A', 'B', 'C' and 'D'. Positions A, B, and C were positioned 6.5 feet from the wall with the targets, and spaced at 3 foot intervals. Position D was placed 3 feet directly behind position B. Figure 6 and Figure 7 illustrate the setup. In the no-tracking condition, the position of the wand was fixed over position B.

The within subjects design consisted of 4 blocks of 40 trials, each with the following conditions: (1) no audio feedback/tracking, (2) audio feedback/tracking (3) no audio feedback/no tracking (4) audio feedback/tracking. Before each block subjects were instructed in how the audio feedback works (if present) and how the pointing target was determined, including detailed instructions in the no-tracking case. Each trial began with a verbal instruction from the experimenter consisting of a letter and number combination. The subject then stood before the position on the floor corresponding to the letter and then pointed and clicked at the numbered target.

Results

Subjects showed no significant change in pointing accuracy or time to complete the task between the first half and second half of each block of 40 trials. Subjects improved pointing accuracy when audio feedback was added in both the tracking and no tracking conditions (Table 1). In the no tracking conditions this improvement was statistically significant (t test $p = 0.024$), suggesting that when tracking was disabled, subjects found the audio feedback useful in maintaining pointing performance.

However, subjects took more time to complete the task in the same condition ($p < 0.01$) (Table 2).

Informal observations of subjects indicate that in the no-tracking/audio case, subjects often exploited a few simple strategies to improve performance. In trying to select target 5 for example, subjects would often go further to the right than they needed to insure that their point would not be confused with 4 (similarly with target 1). To select target 4, the same subjects would often aim past 5, wait for audio feedback for the selection of 5, then move slowly over to 4 until they received audio feedback. This behavior was not as frequent in picking the other targets. In all conditions, over half of the total errors committed were in attempting to select target 4, while target 5 proved almost as easy to select as target 1. Audio feedback was not as helpful in the tracking case and informal observations indicate that subjects did not alter their pointing strategy over their natural behavior while tracking was enabled.

A follow up survey indicates that subjects felt that the no-tracking case "required too much thinking" but that audio feedback was "helpful in selecting the correct target."

This user study suggests that users are most comfortable with a system that incorporates good tracking, but a lack of precise tracking may be compensated for by concise audio feedback, or judicious spacing of the targets, or both. This has implications in the design of larger intelligent environments where users are likely to roam among areas with varying degrees of tracking resolution and intelligence, and yet are likely to expect a high degree of functionality everywhere.

Table 1. Mean and standard deviation of percent correct target selection (on one trial).

	Tracking	No tracking
No Feedback	87.2 ± 33.5	80.6 ± 39.6
Feedback	90.0 ± 30.0	86.9 ± 33.8

Table 2. Mean and standard deviation of time to task completion (seconds).

	Tracking	No tracking
No Feedback	5.23 ± 1.59	5.89 ± 1.69
Feedback	5.73 ± 2.36	6.90 ± 3.85

CONCLUSION

We have introduced a novel user interface device that is designed to address the need for an easy to use interface for intelligent environments. The XWand relies on our natural tendency to point at and speak to objects that we wish to control, and leverages the intelligence of the environment to determine what the user means.

Future work includes improvements in the hardware design such as the addition of a microphone for off-board speech recognition, device miniaturization, and investigations into alternatives to camera-based tracking. The multimodal interpretation process may be enhanced by the ability to do certain kinds of learning, including associating a spoken name to the object pointed to by the wand. As demonstrated by the user study, appropriate feedback to the user requires careful consideration. Finally, real world deployment requires an infrastructure to support a series of networked, controllable devices.

ACKNOWLEDGMENTS

We thank Daniel Wilson and Mike Sinclair for support in assembling the hardware prototype.

REFERENCES

1. Brummitt, B., and JJ Cadiz. *"Let There Be Light": Examining Interfaces for Homes of the Future*. in *Human-Computer Interaction: INTERACT '01*. 2001.
2. Johnson, M., A. Wilson, C. Kline, B. Blumberg and A. Bobick. *Using a Plush Toy to Direct Synthetic Characters*. in *Proceedings CHI*. 1999.
3. Marrin, T. *Possibilities for the Digital Baton as a General-Purpose Gestural Interface*. in *Proceedings CHI*. 1997. Atlanta.
4. Masui, T., and I. Silo. *Real-World Graphical User Interfaces*. in *HUC*. 2000.
5. Swindells, C., K. Inkpen, J. Dill, and M. Tory. *That one there! Gesture for on-demand device identity*. in *Proceedings UIST*. 2002. Paris.
6. Priyantha, N.B., A. Chakraborty, H. Balakrishnan. *The Cricket Location-Support System*. in *Proceedings 6th ACM MOBICOM*. 2000. Boston, MA.
7. Randell, C., and H. Muller. *Low Cost Indoor Positioning System*. in *Proceedings Ubicomp*. 2001. Atlanta, Georgia: Springer-Verlag.
8. Jojic, N., B. Brummitt, B. Meyers, S. Harris, and T. Huang. *Estimation of Pointing Parameters in Dense Disparity Maps*. in *IEEE Intl. Conf. on Automatic Face and Gesture Recognition*. 2000. Grenoble.
9. Horn, B.K.P., *Robot Vision*. 1986, Cambridge, MA: MIT Press.
10. MacNeil, D., *Hand and Mind*. 1992: University of Chicago Press.
11. Oviatt, S.L., *Taming Speech Recognition Errors Within a Multimodal Interface.* Communications of the ACM, 2000. **43**(9): p. 45-51.
12. Pearl, J., *Probabilistic Reasoning in Intelligent Systems*. 1988, San Mateo, CA: Morgan Kaufmann.
13. Olsen, D.R.J., T. Nielsen. *Laser Pointer Interaction*. in *Proceedings CHI*. 2001. Seattle.
14. Myers, B.A., R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long. *Interacting At a Distance: Measuring the Performance of Laser Pointers and Other Devices*. in *Proceedings CHI*. 2002. Minneapolis, Minnesota.