The Perceptive Workbench: Towards Spontaneous and Natural Interaction in Semi-Immersive Virtual Environments

Bastian Leibe, Thad Starner, William Ribarsky, Zachary Wartell, David Krum, Brad Singletary, and Larry Hodges

GVU Center, Georgia Institute of Technology

Abstract

The Perceptive Workbench enables a spontaneous, natural, and unimpeded interface between the physical and virtual world. It is built on vision-based methods for interaction that remove the need for wired input devices and wired tracking. Objects are recognized and tracked when placed on the display surface. Through the use of multiple light sources, the object's 3D shape can be captured and inserted into the virtual interface. This ability permits spontaneity as either preloaded objects or those selected on the spot by the user can become physical icons. Integrated into the same vision-based interface is the ability to identify 3D hand position, pointing direction, and sweeping arm gestures. Such gestures can support selection, manipulation, and navigation tasks. In this paper the Perceptive Workbench is used for augmented reality gaming and terrain navigation applications, which demonstrate the utility and capability of the interface.

1. Introduction

Up to now, most of our interactions with computers have been through devices constrained by wires. Typically, the wires significantly limit the distance of movement and inhibit orientational freedom. In addition most interactions are indirect. One moves a device as an analogue for the action to be created in the display space. We envision an interface without these restrictions. It is untethered, accepts direct, natural gestures, and capable of spontaneously accepting as interactors any objects we choose.

For 3D interaction there is position and orientation tracking, but the sensors (on gloves or on devices that the hands hold) are still usually attached by wires. Devices that permit one to make what would seem to be more natural hand gestures, such as pinch gloves, are often found to perform less well and to be less preferred by users than simple handheld devices with buttons [8, 18]. This may be due to the need to wear a glove, to the fact that pinch gestures are not recognized all the time, and to the subtle changes to recognized hand gestures caused by the glove interface. Further, all devices, whether gloves or handheld devices, carry assumptions about the position of the user's hand and fingers with respect to the tracker. Of course, users's hands differ in size and shape, so that the assumed tracker position should be recalibrated for each user. This is hardly ever done. The result is that fine manipulations can be imprecise and the user often comes away with the feeling that the interaction is slightly off in an indeterminate way. If we can recognize gestures directly, we take into account the difference in hand sizes and shapes.

An additional problem is that any device held in the hand can sometimes be positioned awkwardly for gestures. We have found this even with a simple pointing device, such as a stick with a few buttons, that users usually prefer to other devices [18]. For example, the user may hold the button-stick like the hilt of a sword, making it awkward to point down with the stick for selection (using an imaginary beam that emanates from the end of the stick). If the user rather holds the stick like a pen, other pointing motions can be awkward. Also a user, unless fairly skilled, often has to pause to identify and select buttons on the stick. With accurately tracked hands most of this awkwardness disappears. We are adept at pointing in almost any direction and can quickly pinch fingers, for example, without looking at them.

Finally, it is often easy and natural to use physical objects as interactors (such as the physical icons in Ref. 25). However, presently these objects must be inserted in advance or prepared in a special way. One would like the system to accept objects that one chooses spontaneously for interaction.

In this paper we discuss methods for producing more natural interaction in a more natural environment. We have developed a vision-based, wireless interface that senses the placement and movement of real objects and that permits interaction via untethered manipulation. The objects are recognized by shape and their movements and orientation are tracked. Arm and hand gestures by users are also recognized and tracked. The untethered manipulation is not mediated by attached sensors, and this removes the extra layer of uncertainty, variability, and awkwardness. We have employed this more natural, more direct set of interaction modes on some applications, including a game and a terrain navigation system (Sec. 8). In this way we can look in detail at the affordances and limitations of the direct, wireless interface in action.

2. Related Work

While augmented desk projects have appeared in the literature over the years [1, 4, 9, 10, 11, 15, 17, 18, 25, 27, 31], the Perceptive Workbench is novel in its extensive ability to interact with the physical world. The

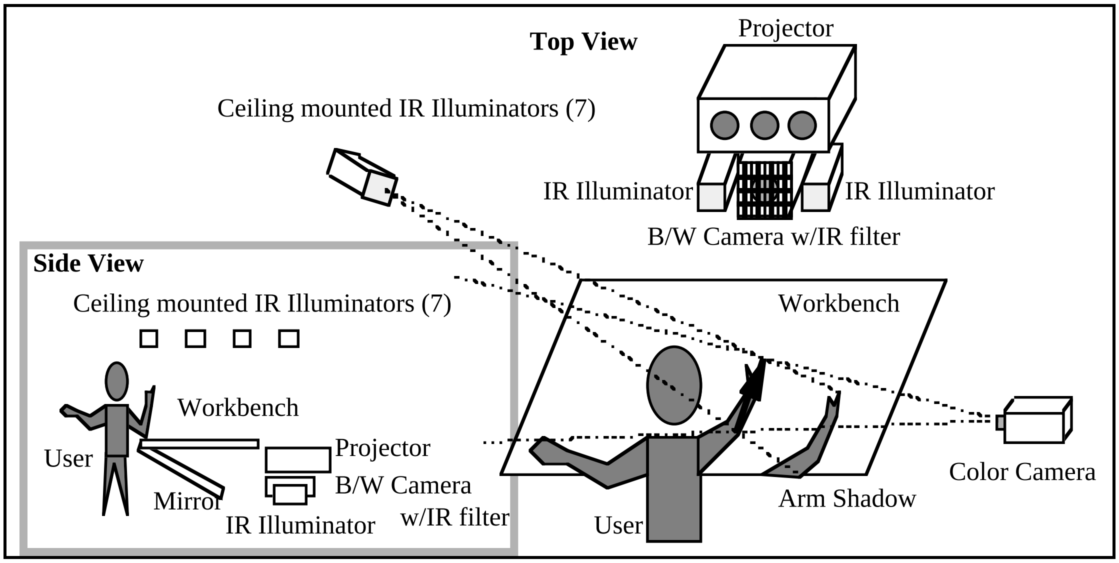

Fig. 1 Light and camera positions for the Perceptive Workbench. The top view shows how shadows are cast and the 3D arm position is tracked.

Perceptive Workbench can reconstruct 3D virtual representations of previously unseen real-world objects placed on the workbench's surface. In addition, the Perceptive Workbench identifies and tracks such objects as they are manipulated on the desk's surface and allows the user to interact with the augmented environment through 2D and 3D gestures. These gestures can be made on the plane of the desk's surface or in the 3D space above the desk. Taking its cue from the user's actions, the Perceptive Workbench switches between these modes automatically, and all interaction is done through computer vision, freeing the user from the wires of traditional sensing techniques. While the Perceptive Workbench is unique in its capabilities to our knowledge, it has a heritage of related work.

Many augmented desk and virtual reality designs use tethered props, tracked by electromechanical or ultrasonic means, to encourage interaction through manipulation and gesture [3, 4, 10, 17, 18, 23, 25, 27]. Fakespace sells the "Immersive Workbench", which normally uses tethered electromagnetic trackers and datagloves for interaction. Such designs tether the user to the desk and require the time-consuming ritual of donning and doffing the appropriate equipment. Fortunately, the computer vision community has taken up the task of tracking the user's hands and identifying gestures. While generalized vision systems track the body in room and desk-based scenarios for games, interactive art, and augmented environments [2, 32, 33], reconstruction of fine hand detail involves carefully calibrated systems and is computationally intensive [14]. Even so, complicated gestures such as those used in sign language [21, 28] or the manipulation of physical objects [19] can be recognized. The Perceptive Workbench uses computer vision techniques to maintain a wireless interface.

More directly related to the Perceptive Workbench, the "Metadesk" [25] identifies and tracks objects placed on the desk's display surface using a near-infrared computer vision recognizer mounted inside the desk. In fact, the vision system for Metadesk was designed and installed by the second author. Unfortunately, since not all objects reflect infrared light and infrared shadows are not used, objects often need infrared reflective "hot mirrors" placed in patterns on their bottom surfaces to aid tracking and identification. Similarly, Rekimoto and Matsushita's "Perceptual Surfaces" [15] employ 2D barcodes to identify objects held against the "HoloWall" and "HoloTable." The HoloWall can track the user's hands (or other body parts) near or pressed against its surface, but its potential recovery of the user's distance from the surface is relatively coarse compared to the 3D pointing gestures of the Perceptive Workbench. Davis and Bobick's SIDEshow [6] is similar to the Holowall except that it uses cast shadows in infrared for full-body 2D gesture recovery. Some augmented desks have cameras and projectors above the surface of the desk and are designed to augment the process of handling paper or interacting with models and widgets through the use of fiducials or barcodes [1, 9, 26, 31]. Krueger's VIDEODESK [10], an early desk-based system, used an overhead camera and a horizontal visible light table (for high contrast) to provide hand gesture input for interactions displayed on a monitor on the far side of the desk. However, none of these systems address the issues of introducing spontaneous 3D physical objects into the virtual environment in real-time and combining 3D deictic (pointing) gestures with object tracking and identification.

3. Hardware Setup

The display environment for the Perceptive Workbench is based on Fakespace’s immersive workbench,

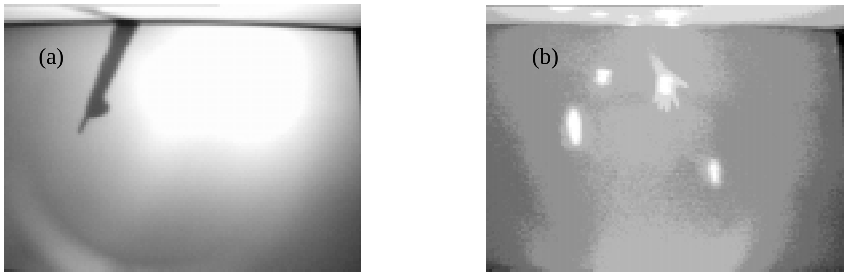

Fig. 2 Images seen by IR camera under the workbench display: (a) arm shadow from overhead lights; (b) reflections from underneath lights.

consisting of a wooden desk with a horizontal frosted glass surface, on which a stereoscopic image can be projected from behind the Workbench. However, the workbench environment is not specifically necessary and the direct gesture interface could be implemented in other large screen environments (e.g., the CAVE or a wall screen).

We placed a standard b/w surveillance camera under the projector that watched the desk surface from underneath. (See Fig. 1.) A filter placed before the camera lens makes it impervious to visible light and to images projected on the desk's surface. Two infrared illuminators placed next to the camera flood the surface of the desk with infrared light that is reflected toward the camera when objects are placed on the surface of the desk. A ring of seven similar light-sources is mounted on the ceiling surrounding the Workbench. Each is selectively switched by computer to make objects on the table cast distinct shadows on the desk's surface (Fig. 2a. A second camera, this one in color, is placed next to the desk to provide a side view of the user's arms for 3D information.

All vision processing is done on two SGI R10000 O2s (one for each camera), which communicate with a display client on an SGI Onyx RE2 via sockets. However, the vision algorithms could be run on one SGI with two digitizer boards or be implemented using semi-custom, inexpensive signal-processing hardware.

We use this setup for three different kinds of interaction which will be explained in more detail in the following sections: recognition and tracking of objects placed on the desk surface based on their contour, full 3D reconstruction of object shapes on the desk surface from shadows cast by the ceiling light-sources, and recognition and quantification of hand and arm gestures.

For display on the Perceptive Workbench, we use the Simple Virtual Environment Toolkit (SVE), a graphics and sound library developed by the Georgia Tech Virtual Environments Group. [8] SVE permits us to rapidly prototype applications used in this work. In addition we use the workbench version of VGIS, a global terrain visualization and navigation system [12, 13] as an application for interaction using hand and arm gestures. The workbench version of VGIS has stereoscopic rendering and an intuitive interface for navigation [29, 30]. Both systems are built on OpenGL and have both SGI and PC implementations.

4. Object Recognition and Tracking

As a basic building block for our interaction framework, we want to enable the user to manipulate the virtual environment by placing objects on the desk surface. The system should recognize these objects and track their positions and orientations while they are being moved over the table. Unlike systems that use color tags or bar codes to identify objects, the user should be free to pick a set of physical objects he/she wants to use. Thus our identification method can only rely on perceived features.

To achieve this goal we use an improved version of the technique described in [22]. The underside of the desk is illuminated by two near-infrared light-sources (Fig. 1). Every object close to the desk surface (including the user's hands) reflects this light and can be seen by the camera under the display surface (Figs. 1 and 2b). Using a combination of intensity thresholding and background subtraction, we extract interesting regions of the camera image and analyze them. The resulting blobs are classified as different object types based on a set of features, including area, eccentricity, perimeter, moments, and the contour shape.

As a consequence of our hardware setting, we have to deal with several problems. The foremost problem is that our two light-sources can only provide a very uneven lighting over the whole desk surface, bright in the middle, but getting weaker towards the borders. In addition, the light rays are not parallel, and the reflection on the mirror surface further exacerbates this effect. As a result, the perceived sizes and shapes of objects on the desk surface can vary depending on the position and orientation. Finally, when the user moves an object, the reflection from his/her hand can also add to the perceived shape. This makes it necessary to use an additional stage in the recognition process that matches recognized objects to objects known to be on the table and can filter out wrong classification of or even complete loss of information about an object for several frames.

In this work, we are using the object recognition and tracking capability mainly for "cursor objects". Our focus is fast and accurate position tracking, but the system may be trained on a set of different objects to be used as navigational tools or physical icons [25]. A

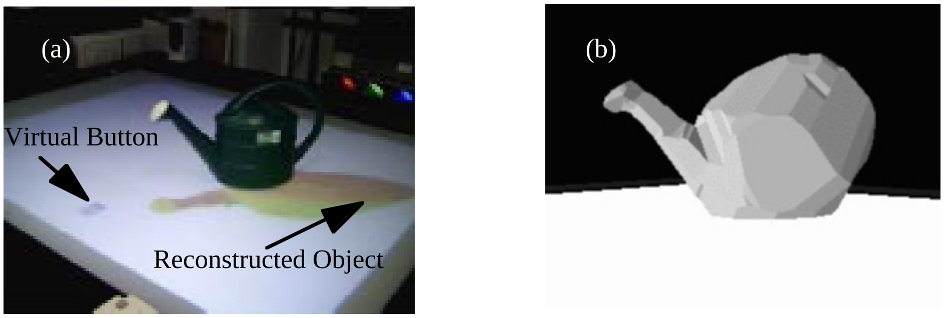

Fig. 3 (a) 3D reconstruction of an object placed on the workbench display; (b) resulting polygonal object.

future project will explore different modes of interaction based on this technology.

5. 3D Reconstruction

Several methods have been designed to reconstruct objects from silhouettes [20, 24] or dynamic shadows [5] using either a moving camera or light-source on a known trajectory or a turntable for the object [24]. Several systems have been developed for the reconstruction of relatively simple objects, including a commercial system Sphinx3D.

However, the necessity to move either the camera or the object imposes severe constraints on the working environment. To reconstruct an object with these methods, it is usually necessary to interrupt the interaction with it, take the object out of the user's environment, and place it into a specialized setting. Other approaches make use of multiple cameras from different view points to avoid this problem at the expense of more computational power to process and communicate the results. In this project, using only one camera and the infrared light sources, we analyze the shadows cast on the object from multiple directions. As the process is based on infrared light, it can be applied independent of the lighting conditions and without interfering with the user's natural interaction with the desk or the current visual display environment.

Our approach is fully automated and does not require any special hardware (like stereo cameras, laser range finders, etc.). On the contrary, the method is extremely cheap, both in hardware and in computational cost. In addition, there is no need for extensive calibration, which is usually necessary in other approaches to recover the exact position or orientation of the object in relation to the camera. We only need to know the approximate position of the light-sources (+/- 2 cm), and we need to adjust the camera to the size of the display surface, which must be done only once. Neither the camera and light-sources nor the object are moved during the reconstruction process. Thus recalibration is unnecessary. We have substituted all mechanical moving parts, which are often prone to wear and imprecision, by a series of light beams from known locations.

An obvious limitation for this approach is that we are, at the same time, confined to only a fixed number of different views from which to reconstruct the object. The turntable approach allows to take an arbitrary number of images from different view points. However, Sullivan's work [24] and our experience with our system have shown that even for quite complex objects usually 7 to 9 different views are enough to get a reasonable 3D model of the object. Thus, to obtain the different views, we mounted a ring of 7 infrared light sources in the ceiling, each one of which is switched independently by computer control. The system detects when a new object is placed on the desk surface, and the user can initiate the reconstruction by touching a virtual button rendered on the screen (Fig. 3a). (This action is detected by the camera.) After only one second, all shadow images are taken. After another second, the reconstruction is complete (Fig. 3b), and the newly reconstructed object is part of the virtual world.

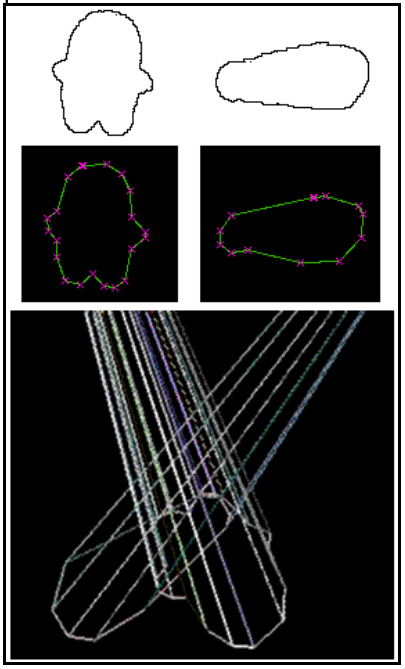

Fig. 4 Steps of 3D object reconstruction including extracting of contour shapes from shadows and multiple view cones (bottom).

The speed of the reconstruction process is mainly limited by the switching time of the light sources. Whenever a new light-source is activated, the image processing system has to wait for several frames until it can be sure to get a valid image. The camera under the desk records the sequence of shadows an object on the table casts when illuminated by the different lights. Fig. 4 shows a model reconstructed from a series of contour shadows, where the contour shadows extracted by using different IR sources. By approximating each shadow as a polygon (not necessarily convex) [16], we create a set of polyhedral "view cones", extending from the light source to the polygons. Intersecting these cones creates a polyhedron that roughly contains the object. Fig. 3b shows the polygons resulting from the previous shadows and a visualization of the intersection of polyhedral cones.

6. Deictic Gesture Recognition and Tracking

Hand gestures for interaction with a virtual environment can be roughly classified into symbolic (iconic, metaphoric, and beat) and deictic (pointing) gestures. Symbolic gestures carry an abstract meaning that may still be recognizable in iconic form in the associated hand movement. Without the necessary cultural context, however, they may be arbitrary. Examples for symbolic gestures include most conversational gestures in everyday use, and whole gesture languages, for example, American Sign Language. Previous work by Starner [21] has shown that a large set of symbolic gestures can be distinguished and recognized from live video images using hidden Markov models (HMMs).

Deictic gestures, on the other hand, are characterized by a strong dependency on location and orientation of the performing hand. Their meaning is determined by the position at which a finger is pointing, or by the angle of rotation of some part of the hand. This information acts not only as a symbol for the gesture's interpretation, but also as a measure of how much the corresponding action should be executed or to which object it should be applied.

For navigation and object manipulation in a virtual environment, many gestures are likely to have a deictic component. It is usually not enough to recognize that an object should be rotated, but we will also need to know the desired amount of rotation. For object selection or translation, we want to specify the object or location of our choice just by pointing at it. For these cases, gesture recognition methods that only take the hand shape and trajectory into account will not be sufficient. We need to recover 3D information about the user's hand and arm in relation to his/her body.

In the past, this information has largely been obtained largely by using wired gloves or suits, or magnetic trackers [3]. Such methods provide sufficiently accurate results but rely on wires and have to be tethered to the user's body, or to specific interaction devices. These wires are cumbersome at best. They restrict the user's freedom of movement and tend to get entangled with objects in the user's environment. Our goal is to develop a purely vision-based architecture that facilitates wireless 3D interaction.

With vision-based 3D tracking techniques, the first issue is to determine which information in the camera image is relevant, i.e. which regions represent the user's hand or arm. This task is made even more difficult by variation in user clothing or skin color and by background activity. Although typically only one head is tracked and only one user interacts with the environment at a given time using traditional methods of interaction, the physical dimensions of large semi-immersive environments such as the workbench invite people to watch and participate.

In a virtual workbench, there are few places where a camera can be put to provide reliable hand position information. One camera can be set up next to the table without overly restricting the available space for users, but if a similar second camera were to be used at this location, either multi-user experience or accuracy would be compromised. We have addressed this problem by employing our shadow-based architecture (as described in the hardware section). The user stands in front of the workbench and extends an arm over the surface. One of the IR light-sources mounted on the ceiling to the left of, and slightly behind the user, shines its light on the desk surface, from where it can be seen by the IR camera under the projector--see Fig. 1). When the user moves his/her arm over the desk, it casts a shadow on the desk surface (see Fig. 2a). From this shadow, and from the known light-source position, we can calculate a plane in which the user's arm must lie.

Simultaneously, the second camera to the right of the table (Fig. 1) records a side view of the desk surface and the user's arm. It detects where the arm enters the image and the position of the fingertip. From this information, it extrapolates two lines in 3D space, on which the observed real-world points must lie. By intersecting these lines with the shadow plane, we get the coordinates of two 3D points, one on the upper arm, and one on the fingertip. This gives us the user's hand position, and the direction in which he/she is pointing. As shown in Fig. 5, this information can be used to project a hand position icon and a selection ray in the workbench display.

We must first recover arm direction and fingertip position from both the camera and the shadow image. Since the user is standing in front of the desk and user's arm is connected to the user's body, the arm's shadow should always touch the image border. Thus our algorithm exploits intensity thresholding and background subtraction to discover regions of change in the image and searches for areas where these touch the front border of the desk surface (which corresponds to the top border of the shadow image or the left border of the camera image). It then takes the middle of the touching area as an approximation for the origin of the arm (Fig. 2a). For simplicity we will call this point the "shoulder", although in most cases it is not. Tracing the contour of the shadow, the algorithm searches for the point that is farthest away from the shoulder and takes it as the

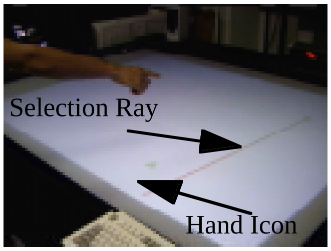

Fig. 5 Pointing gesture with hand icon and selection ray.

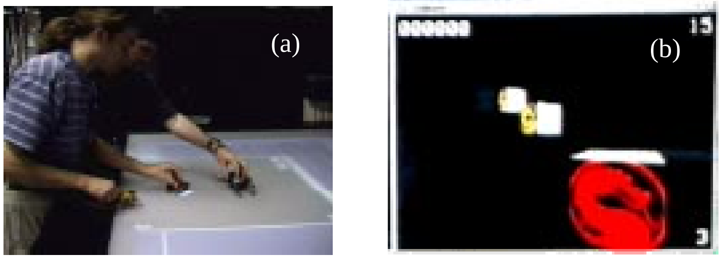

Fig. 6 (a) Game masters controlling monster positions; (b) monsters moving in the 3D space as a result of actions in Fig. 6a.

fingertip. The line from the shoulder to the fingertip reveals the 2D direction of the arm.

In our experiments, the point thus obtained was coincident with the pointing fingertip in all but a few pathological cases (such as the fingertip pointing straight down at a right angle to the arm). The method does not depend on a pointing gesture, but also works for most other hand shapes, including but not restricted to, a flat horizontally or vertically held hand and a fist. These shapes may be distinguished by analyzing a small section of the side camera image and may be used to trigger specific gesture modes in the future.

The computed arm direction is correct as long as the user’s arm is not overly bent. In such cases, the algorithm still connected shoulder and fingertip, resulting in a direction somewhere between the direction of the arm and the one given by the hand. Although the absolute resulting pointing position did not match the position towards which the finger was pointing, it still managed to capture the trend of movement very well. Surprisingly, the techniques is sensitive enough such that the user can stand at the desk with his/her arm extended over the surface and direct the pointer simply by moving his/her index finger, without arm movement.

Limitations

The architecture used poses several limitations. The primary problem with the shadow approach is finding a position for the light-source that can give a good shadow of the user’s arm for a large set of possible positions, while avoiding, at the same time, also capturing a shadow from the user’s body. Since the area visible to the IR camera is coincident with the desk surface, there are necessarily regions where the shadow is not visible in, touches, or falls outside of the borders. Our solution to this problem is to switch to a different light-source whenever such a situation is detected, the choice of the new light-source depending on where the shadows touched the border. By choosing overlapping regions for all light-sources, we can keep the number of light-source switches to a necessary minimum. In practice, 4 light-sources were enough to cover the relevant area of the desk surface.

A bigger problem is caused by the location of the side camera. If the user extends both of his/her arms over the desk surface, or if more than one user tries to interact with the environment at the same time, the images of these multiple limbs can overlap and be merged to a single blob. As a consequence, our approach will fail to reliably detect the hand positions and orientations in these cases. A more sophisticated approach using previous position and movement information could yield more reliable results, but we chose, at this first stage, to accept this restriction and concentrate on high frame rate support for one-handed interaction. This may not be a serious limitation for a single user for certain tasks; a recent study shows that for tasks normally requiring two hands in a real environment, users have no preference for one versus two hands in a virtual environment [18].

7. Performance Analysis

Both object and gesture tracking perform at a stable 12-18 frames per second. Frame rate depends on the number of objects on the table and the size of the shadows, respectively. Both techniques are able to follow fast motions and complicated trajectories. Latency is currently 0.25-0.33 seconds but has improved since last testing (the acceptable threshold is considered to be at around 0.1s). Surprisingly, this level of latency seems adequate for most pointing gestures. Since the user is provided with continuous feedback about his hand and pointing position and most navigation controls are relative rather than absolute, the user adapts his behavior readily to the system. With object tracking, the physical object itself can provide the user with adequate tactile feedback as the system catches up to the user's manipulations. In general, since the user is moving objects across a very large desk surface, the lag is noticeable but rarely troublesome in the current applications.

Even so, we expect simple improvements in the socket communication between the vision and rendering code and in the vision code itself to improve latency significantly. In addition, due to their architecture, the R10000-based SGI O2's are known to have a less direct video digitizing path than their R5000 counterparts. Thus, by switching to less expensive machines we expect to improve our latency figures. For the terrain navigation task below, rendering speed provides a limiting factor. However, render lag may be compensated by employing predictive Kalman filters that will also add to the stability of the tracking system.

To calculate the error from the 3D reconstruction process requires choosing known 3D models, performing

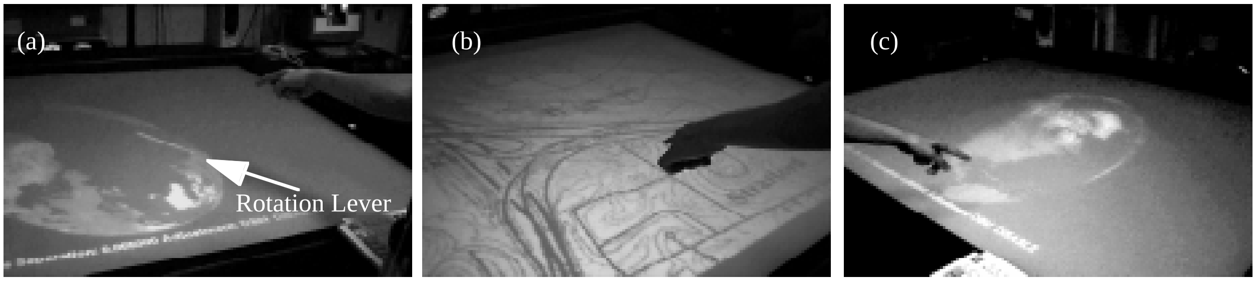

Fig. 7 Terrain navigation using deictic gestures: (a) rotation (about an axis perpendicular to and through the end of the rotation lever); (b) zooming in; (c) panning. The selection ray is too dim to see in this view (see Fig. 5.)

the reconstruction process, aligning the reconstructed model and the ideal model, and calculating an error measure. For simplicity, a cone and pyramid were chosen. The centers of the bounding boxes of the ideal and reconstructed models were set to the same point in space. To measure error, each vertex on the reconstructed model was compared to the point made by the intersection of the ideal surface and the line made by the center of the bounding box and the reconstructed vertex. As an additional measure of error, the same process was performed using the calculated centers of mass for the ideal and reconstructed models. The resulting mean square error averaged 4.0% of the average chord length from the center to the ideal vertex (~0.015m), with the bounding box method averaging 3.3% (~0.012m) and the center of mass method averaging 4.8% (~0.018m). While improvements may be made by precisely calibrating the camera and lighting system, by adding more light sources, and by obtaining a silhouette from the side camera (to eliminate ambiguity about the top of the surface), the system meets its goal of providing virtual presences for physical objects in a quick and timely manner that encourages spontaneous interactions.

8. Putting It to Use: Spontaneous Gesture Interfaces

The perceptive workbench interface can switch automatically between gesture recognition and object recognition, tracking, and reconstruction. When the user moves her hand above the display surface, the hand and arm are tracked as described in Sec. 6. A cursor appears at the projected hand position on the display surface and a ray emanates along the projected arm axis. These can be used in selection or manipulation, as in Fig. 5. When the user places an object on the surface, the cameras recognize this and identify and track the object. A virtual button also appears on the display (indicated by the arrow in Fig. 3a). Through shadow tracking, the system determines when the hand overlaps the button, selecting it. This action causes the system to capture the 3D object shape, as described in Sec. 5.

This set provides the elements of a perceptual interface, operating without wires and without restrictions as to objects employed. For example, we have constructed a simple application where objects placed on the desk are selected, reconstructed, and then placed in a “template” set, displayed as slowly rotating objects on the left border of the workbench display. These objects could act as new physical icons that are attached by the user to selection or manipulation modes. Or the shapes themselves could be used in model-building or other applications.

An Augmented Reality Game

We have created a more elaborate collaborative interface using the Perceptive Workbench. This involves the workbench communicating with a person in a separate space wearing an augmented reality headset. All interaction is via image-based gesture tracking without attached sensors. The game is patterned after a martial arts fighting game. The user in the augmented reality headset is the player and one or more people interacting with the workbench are the game masters. The workbench display surface acts as a top-down view of the player’s space. The game masters place different objects on the surface, which appear to the player as distinct monsters at different vertical levels in his space. The game masters move the objects around the display surface, towards and away from the player; this motion is replicated by the monsters, which move in their individual planes. Fig. 6a shows the game masters moving objects and Fig. 6b displays the moving monsters in the virtual space.

The player wears a “see-through” Sony Glasstron equipped with two cameras. Fiducials or natural features in the player’s space are pre-input for head tracking. Currently the player is assumed to stay in relatively the same place, his head rotation is recovered, and graphics (such as the monsters) are rendered that register with the physical world. The cameras look down at the player’s hands and capture hand gestures using a toolkit developed by one of the authors [21]. Based on hidden Markov models, these tools can recognize 40 American Sign Language signs in real-time with over 97% accuracy. In the game implementation, a simple template matching method is sufficient for recognizing a small set of martial arts gestures. To effect attacks on the monsters, the user must accompany the gestures with a Kung Fu yell (“heee-YAH”). There is a different gesture for each type of foe. Foes that are not fended off can enter the player’s personal space and injure him. Enough injuries will cause the player’s defeat.

The system has been used by faculty and graduate students in the GVU lab. They have found the experience compelling and balanced. Since it's difficult for the game master to keep pace with the player, two game masters are better (Fig. 6a). This is straightforward to do using the Perceptive Workbench interface. The player has a unique experience, seeing and hearing foes overlaid in a physical room and then being able to respond with sharp gestures and yells. For a fuller description of this application, see [22].

3D Terrain Navigation

We have developed a global terrain navigation system on the virtual workbench. This permits one to fly seamlessly between global, regional, and local terrain. One can fly continuously from outer space to terrain or buildings with features at 1 foot or better resolution [29]. The navigation is both compelling and loaded with detail since the user sees terrain and building as properly displayed stereoscopic images [30]. We have developed a complete interface for third person navigation through this space [29]. In third person navigation, one interacts with the terrain as if it were an extended relief map laid out below one on a curved surface. Thus the main actions are zoom in or out, pan, and rotation. However, the user still sees the terrain and objects on it in full 3D perspective and, since she is head-tracked, can even move her head to look at objects from different angles. Previously interaction has been by using button sticks with 6 DoF electromagnetic trackers attached. Thus both the button sticks and the trackers have attached wires.

This is the type of interface that we would like to free from the burden of constrained movement and awkward gestures discussed in the Introduction. To do this we employ the deictic gestures of the Perceptive Workbench, as described in Sec. 6. Direction of navigation is chosen by pointing and can be changed continuously (Fig. 7b). Moving the hand towards the display increases speed towards the earth and moving it away increases speed away from the earth. Panning is accomplished by lateral gestures in the direction to be panned (Fig. 7c). Rotation is accomplished by making a rotating gesture with the arm (Fig. 7a). At present these three modes are chosen by keys on a keyboard attached to the workbench. In the future we expect to use gestures entirely (e.g., pointing will indicate zooming).

Although there are currently some problems with latency and accuracy (both of which will be diminished in the future), a user can successfully employ gestures for navigation. In addition the set of gestures are quite natural to use. Further, we find that the vision system can distinguish hand articulation and orientation quite well. Thus we will be able to attach interactions to hand movements (even without arm movements).

9. Future Work and Conclusions

Several improvements can be made to the Perceptive Workbench. Higher resolution reconstruction and improved recognition for small objects can be achieved via an active pan/tilt/zoom camera mounted underneath the desk. The color side camera can be used to improve 3D reconstruction and construct texture maps for the digitized object. The reconstruction code can be modified to handle holes in objects. The latency of the gesture/rendering loop can be improved through code refinement and the application of Kalman filters. When given a difficult object, recognition from the reflections from the light source underneath can be successively improved by using cast shadows from the different light sources above or the 3D reconstructed model directly. Hidden Markov models can be employed to recognize symbolic hand gestures for controlling the interface. Finally, as hinted by the multiple game masters in the gaming application, several users may be supported through careful, active allocation of resources.

In conclusion, the Perceptive Workbench uses a vision-based system to enable a rich set of interactions, including hand and arm gestures, object recognition and tracking, and 3D reconstruction of objects placed on its surface. These elements are combined seamlessly into the same interface and can be used in diverse applications. In addition, the sensing system is relatively inexpensive, retailing ~$1000 for the cameras and lighting equipment in the addition to the cost of a computer with one or two video digitizers, depending on the functions desired. As seen from the the multiplayer gaming and terrain navigation applications, the Perceptive Workbench provides an untethered, spontaneous, and natural interface that encourages the inclusion of physical objects in the virtual environment.

Acknowledgments

This work is supported in part by a contract from the Army Research Lab, an NSF grant, and an ONR AASERT grant. We thank Brygg Ullmer for sparking an interest in augmented desks, Jun Rekimoto for a long ago brainstorming session, and Jim Davis for help in purchasing IR illuminators. In addition we thank Paul Rosin and Geoff West for their line segmentation code.

References

1. Arai, T. and K. Machii and S. Kuzunuki. Retrieving Electronic Documents with Real-World Objects on InteractiveDesk. UIST ‘95, pp. 37-38 (1995).
2. Bobick, A., S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schutte, and A. Wilson. The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment. MIT Media Lab Technical Report (1996).
3. Bolt R. and E. Herranz. Two-handed gesture in multi-modal natural dialog. UIST ‘92, pp. 7-14 (1992).
4. Coquillart, S. and G. Wesche. The Virtual Palette and the Virtual Remote Control Panel: A Device and an Interaction Paradigm for the Responsive Workbench.

IEEE Virtual Reality '99 Conference (VR'99), Houston, March 13-17, 1999.

5. Daum, D. and G. Dudek. On 3-D Surface Reconstruction Using Shape from Shadows. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, 1998.
6. Davis, J.W. and A. F. Bobick. SIDEshow: A Silhouette-based Interactive Dual-screen Environment. *MIT Media Lab Tech Report No. 457*
7. Kessler, G.D., L.F. Hodges, and N. Walker. Evaluation of the CyberGlove as a Whole-Hand Input Device. *ACM Tran. on Computer-Human Interactions*, 2(4), pp. 263-283 (1995).
8. Kessler, D., R. Kooper, and L. Hodges. The Simple Virtual Environment Libary: User´s Guide Version 2.0. Graphics, Visualization, and Usability Center, Georgia Institute of Technology, 1997.
9. Kobayashi, M. and H. Koike. EnhancedDesk: integrating paper documents and digital documents. *Proceedings of 3rd Asia Pacific Computer Human Interaction*, pp. 57-62 (1998).
10. Krueger, M. *Artificial Reality II*. Addison-Wesley, 1991.
11. Krueger, W., C.-A. Bohn, B. Froehlich, H. Schueth, W. Strauss, G. Wesche. The Responsive Workbench: A Virtual Work Environment. *IEEE Computer*, vol. 28. No. 7. July 1995, pp. 42-48.
12. Lindstrom, Peter, David Koller, William Ribarsky, Larry Hodges, Nick Faust, and Gregory Turner. Real-Time, Continuous Level of Detail Rendering of Height Fields. Report GIT-GVU-96-02, *SIGGRAPH 96*, pp. 109-118 (1996)
13. Lindstrom, Peter, David Koller, William Ribarsky, Larry Hodges, and Nick Faust. An Integrated Global GIS and Visual Simulation System. Georgia Tech Report GVU-97-07 (1997).
14. Rehg, J.M and T. Kanade. DigitEyes: Vision-Based Human Hand-Tracking. *School of Computer Science Technical Report CMU-CS-93-220,* Carnegie Mellon University, December 1993.
15. Rekimoto, J., N. Matsushita. Perceptual Surfaces: Towards a Human and Object Sensitive Interactive Display. *Workshop on Perceptual User Interfaces (PUI '97)*, 1997.
16. Rosin, P.L. and G.A.W. West. Non-parametric segmentation of curves into various representations. *IEEE PAMI'95*, 17(12) pp. 1140-1153 (1995).
17. Schmalstieg, D., L. M. Encarnacao, Z. Szalavar. Using Transparent Props For Interaction With The Virtual Table. *Symposium on Interactive 3D Graphics (I3DG'99)*, Atlanta, 1999.
18. Seay, A.F., D. Krum, W. Ribarsky, and L. Hodges. Multimodal Interaction Techniques for the Virtual Workbench. Accepted for publication, CHI 99.
19. Sharma R. and J. Molineros. Computer vision based augmented reality for guiding manual assembly. *Presence*, 6(3) (1997)..
20. Srivastava, S.K. and N. Ahuja. An Algorithm for Generating Octrees from Object Silhouettes in Perspective Views. *IEEE Computer Vision, Graphics and Image Processing*, 49(1), pp. 68-84 (1990).
21. Starner, T., J. Weaver, A. Pentland. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *IEEE PAMI* , 20(12), pp. 1371-1375 (1998).
22. Starner, T., B. Leibe, B. Singletary, and J. Pair. MIND-WARPING: Towards Creating a Compelling Collaborative Augmented Reality Gaming Interface through Wearable Computers and Multi-modal Input and Output. Submitted to *International Conference on Intelligent User Interfaces (IUI'2000)*, (2000).
23. Sturman, D. Whole-hand input. Ph.D. Thesis, MIT Media Lab (1992).
24. Sullivan, S. and J. Ponce. Automatic Model Construction, Pose Estimation, and Object Recognition from Photographs Using Triangular Splines. *IEEE PAMI* , 20(10), pp. 1091-1097 (1998).
25. Ullmer, B. and H. Ishii. The metaDESK: Models and Prototypes for Tangible User Interfaces. *Proceedings of UIST'97*, October 14-17, 1997.
26. Underkoffler, J. and H. Ishii. Illuminating Light: An Optical Design Tool with a Luminous-Tangible Interface. *Proceedings of CHI '98*, April 18-23, 1998.
27. van de Pol, Rogier, William Ribarsky, Larry Hodges, and Frits Post. Interaction in Semi-Immersive Large Display Environments. Report GIT-GVU-98-30, Virtual Environments '99, pp. 157-168 (Springer, Wien, 1999).
28. Vogler C. and D. Metaxas. ASL Recognition based on a coupling between HMMs and 3D Motion Analysis. *Sixth International Conference on Computer Vision*, pp. 363-369 (1998).
29. Wartell, Zachary, William Ribarsky, and Larry F.Hodges. Third Person Navigation of Whole-Planet Terrain in a Head-tracked Stereoscopic Environment. Report GIT-GVU-98-31, *IEEE Virtual Reality 99*, pp. 141-149 (1999).
30. Wartell, Zachary, Larry Hodges, and William Ribarsky. Distortion in Head-Tracked Stereoscopic Displays Due to False Eye Separation. Report GIT-GVU-99-01, *SIGGRAPH 99*,.pp. 351-358 (1999).
31. Wellner P. Interacting with paper on the digital desk. *Comm. of the ACM*, 36(7), pp. 86-89 (1993)
32. Wren C., F. Sparacino, A. Azarbayejani, T. Darrell, T. Starner, A. Kotani, C. Chao, M. Hlavac, K. Russell, and A. Pentland. Perceptive Spaces for Performance and Entertainment: Untethered Interaction Using Computer Vision and Audition. *Applied Artificial Intelligence* , 11(4), pp. 267-284 (1995).
33. Wren C., A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE PAMI*, 19(7), pp. 780-785 (1997).