Unencumbered Virtual Environments

Kenneth Russell **Thad Starner** **Alex Pentland**
Perceptual Computing Section, The Media Laboratory,
Massachusetts Institute of Technology
Room E15-383, 20 Ames Street, Cambridge MA 02139, USA
E-mail: kbrussel@media.mit.edu, thad@media.mit.edu, sandy@media.mit.edu

Abstract

Bulky head-mounted displays, data gloves, and severely limited movement have become synonymous with virtual environments. In this paper we describe an interactive virtual environment (IVE) that uses only a large field of view screen and passive sensors, in this case a video camera and microphone. Special clothing and modifications to the environment are not necessary (i.e. chroma-key backdrops are not required). Such a system has many advantages: ease of access when entering and exiting the environment, more active participation, and social context. These attributes are demonstrated in SURVIVE, a multi-player gesture-driven adaption of id Software's game Doom.

1 Introduction

While many advances have been made in rendering three dimensional worlds, techniques for human interaction with these worlds lag behind. In order to allow a user to navigate a three dimensional space, most commercial systems encumber the user with head-mounted displays, electromagnetic or sonic position sensors, gloves, and/or body suits [1]. While such systems can be extremely accurate, they limit the freedom of the user due to the tethers associated with the sensors and displays. Furthermore, the user must don or remove the equipment each time he wants to enter or exit the environment. Some systems avoid this problem by passively or actively "watching" the user. These systems often modify the environment with specially colored or illuminated backdrops, require the user to wear special clothes, or involve special equipment like range finders or active floor tiles [6; 7; 8]. In the tradition of previous IVE's [3; 9], we demonstrate a virtual environment, SURVIVE (Simulated Urban Recreational Violence Interactive Virtual Environment), which tracks the user in "three dimensions" without modifying the user or the physical environment and uses only off-the-shelf computers, a video camera, a microphone, and a video projector.

Figure 1 demonstrates the basic components of the sensing subsystem (IVE). Figure 2 shows a user in SURVIVE (Simulated Urban Recreational Violence Interactive Video Environment), and Figure 3 shows a sample screen of the game. The user interacts with the virtual environment in a room sized area (15'x17') whose only requirements are good, constant lighting and an unmoving background. A large projection screen (7'x10') allows the user to see the virtual environment, and a downward pointing video camera mounted on top of the projection screen allows the system to track the user. A microphone is mounted above the user's head in the center of the active area. Both input devices are monitored in real time by one or more Silicon Graphics computers.

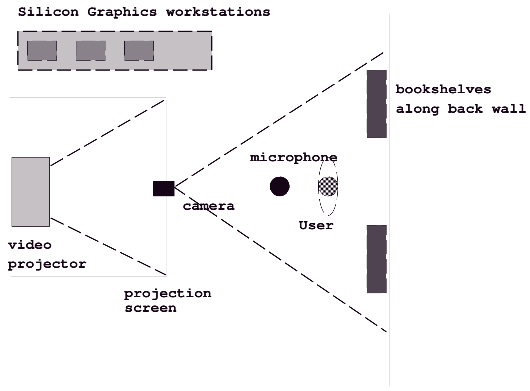

Figure 1: Interactive Video Environment hardware.

2 Vision Interface

At the heart of IVE is a passive computer vision tracking system. The necessary hardware can be bought off-the-shelf, and, depending on the desired accuracy, little initialization is necessary.

2.1 Figure Ground Segmentation

Figure ground segmentation, as used here, refers to comparing a current image to some initial image, ignoring shadows, and determining what has changed. In this case, the purpose is to locate a user that has walked into the view of the camera. Due to computational constraints, subsampled video is used (160x120). The main video board is a Galileo in a SGI Indigo 2 that automatically performs averaging when reducing the image

Figure 2: The user environment for SURVIVE.

Figure 3: Screen image.

size. The system is initialized by taking 40 images of the static background scene and calculating the mean and variance of the red, green, blue, and luminance values at each pixel location ($\mu_c(x,y)$ and $\sigma_c(x,y)$ respectively, where c represents a color or the luminance). At each iteration thereafter, a pixel is classified as part of the foreground if it meets one of the following conditions:

$$\frac{|l(x,y) - \mu_l(x,y)|}{\sigma_l} > T_l$$

$$\frac{|r(x,y)n(x,y) - \mu_r(x,y)|}{\sigma_r} > T_c$$

$$\frac{|g(x,y)n(x,y) - \mu_g(x,y)|}{\sigma_g} > T_c$$

$$\frac{|b(x,y)n(x,y) - \mu_b(x,y)|}{\sigma_b} > T_c$$

where T_c is the chrominance threshold, T_l is the luminance, and $n(x,y)$ is defined as

$$n(x,y) = \begin{cases} \frac{\mu_l(x,y)}{l(x,y)} & \text{if } l(x,y) < \mu_l(x,y) \\ 1 & \text{otherwise} \end{cases}$$

To a first approximation, an object in shadow does not change chrominance. The last three conditions take this into account. The r, g, and b values are normalized by the mean luminance when the current luminance is less than the stored mean. Thus, the user's shadow should not affect these tests. However, if the current luminance is greater than the stored mean, then an object with a higher reflectivity has moved in front of the camera. In this case, the color values are not normalized and the test is triggered.

Since ignoring the user's shadow is a desired effect, T_l is generally set large. However, in some situations, the user may be wearing colors similar to those of the background. In these cases, the luminance test may catch foreground cases that the chrominance checks ignore. An adaptive algorithm is used to balance the luminance versus chrominance thresholds based on the changes of the resultant foreground "silhouette" over time.

Checking each pixel each iteration is computationally expensive. Therefore, the silhouette is grown from a seed point in the image. Initially, the seed is found quickly by scanning an image for a pixel that exceeds the above thresholds. After the user is found, the seed is set to the centroid from the last frame. At each iteration, once a pixel has been determined to exceed one of the thresholds, its eight nearest neighbors are tested. Each pixel that is tested is considered part of the foreground, but the silhouette can only grow from pixels that exceed one of the thresholds. This produces an inherent morphological dilation of the silhouette that helps avoid lighting and noise aberrations. Growth stops when no new pixels meet the requirements. This algorithm runs at around 20 frames/sec on a 150Mhz R4400 Indigo 2.

2.2 Locating Extremities

After background subtraction has produced a silhouette of the user, the perimeter of the silhouette is traversed to discover appropriately sized extremities which may correspond to the user's head, hands, and feet. Starting at the bottom of the silhouette beneath the centroid, the direction of the contour is determined at each point while traversing the contour clockwise. If the direction of the contour 16 to 20 steps from the current position is opposite that of the current direction, the centroid of the points in this window is then labelled an extremity. Note that this procedure will cause the extremities to be labelled redundantly. This is addressed by simple clustering and averaging of the values.

Next, the list of extremities (converted to world coordinates) are labelled as the head, hands, and feet. As a bootstrap, the head is assumed to be in the top part of the silhouette within 6" of the x position of the centroid, the feet are assumed to be the leftmost and rightmost points in the bottom third of the silhouette, and the hands are assumed to be the leftmost and rightmost points approximately two thirds from the top of the silhouette. Extremities found in these locations are labelled appropriately. Often, the hands or feet are not distinct from the rest of the silhouette. For example, in Figure 4, the user's relaxed hand (left) merges into the rest of his body. In such cases, the bootstrapping position is assumed. However, once an appropriate extremity is located, temporal continuity is used to track

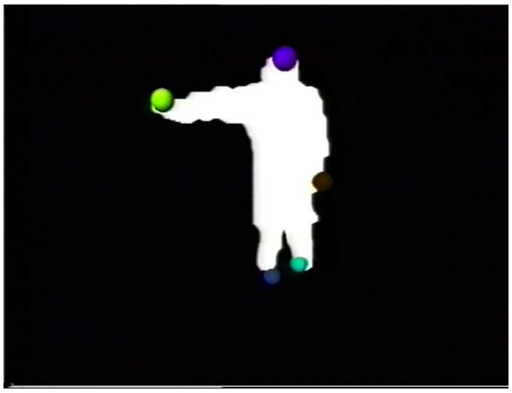

Figure 4: Labelling the head, hands, and feet from a silhouette.

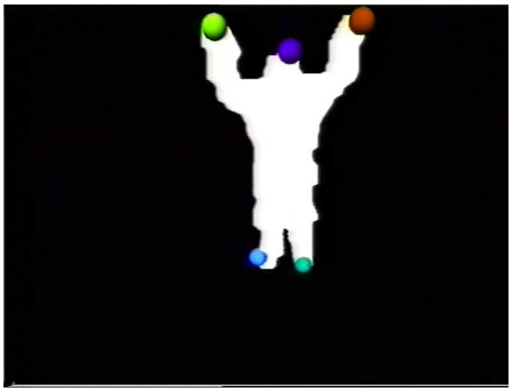

Figure 5: Using tracking to assist labelling.

it through various motions (Figure 5). "Certainty" values are assigned to reflect how confident the algorithm is in its labelling. These values are at a minimum when the defaults are used and at a maximum when a distinct extremity has been tracked for the past 5 frames.

2.3 Calibration and User Modeling

In order to determine the 3D position of the user, a calibration procedure associates 3D points with their respective 2D points in the camera image. After determining the absolute position of physically marked points in the viewing area, a least squares approximation technique determines a camera transform matrix. This matrix is then used to translate observed positions into their 3D values.

This calibration matrix is used explicitly when evaluating the height and width of the user when he enters the viewing area. Since it is assumed that the user will be touching the floor, the inverse camera transform can be used to translate his image's height and width to a real height and width. This then can be used to initialize the gesture recognition code.

2.4 Gesture Recognition

In SURVIVE, the user steps forward, backward, left, and right to slide in those respective directions in the game without turning. He raises both arms to open doors and points left or right to turn. The speed with which a player turns in the game is proportional to the size of the arm as seen by the camera. This roughly corresponds to the direction in which the hand is pointed. For example, if the hand is pointed ahead and slightly to the left, the camera sees a foreshortened arm, and the view in the game turns slowly to the left. Note, however, this algorithm may present a problem with different sized users (say, a child as compared to an adult). In order to address this problem, the height information determined when the user enters the environment is used to approximate the length of the user's arm.

In order to give the user a better sense of presence in the SURVIVE environment, a toy gun is used. Because the gun is large, two-handed, and held against the inner arm, turning with the gun looks like a pointing gesture to the camera. Thus, the vision interface did not need modifications. However, users adapt to turning and the game play more quickly when using the gun than when they use the pointing gestures. In addition, the simple tactile feedback of holding the gun seems to add to the experience.

3 Audio Interface

At first, any loud noise was used to trigger the weapon in SURVIVE. No interface was provided to change weapons in the game. However, this primitive had an interesting side effect: by-standers would often "join in" the game by yelling "BANG!" to trigger the participant's weapon. As the audio interface advanced, this ability was retained.

All audio processing is performed on an SGI Indy or Indigo. The initial interface simply evaluated the power of the incoming audio to determine if the threshold for triggering the weapon was crossed. With the addition of the electronic toy gun mentioned earlier, a more advanced audio interface was created. The gun produced different electronic noises depending on the button pressed by the player. Sensing these different noises and appropriately switching the weapons in Doom required noise modeling since the IVE space ajoins a public area and several machines are maintained in the near vicinity

A piezoelectric microphone with a hemispherical sensing area was mounted onto the ceiling in the middle of the space. During SURVIVE initialization, the background noise is sampled and stored. Sampling is performed at 32Khz. The user provides samples of each sound to be associated with a particular weapon in Doom. During play, template matching on the spectrogram is performed whenever a loud sound is sensed. Template matching is only used to determine if the weapon should be changed. Whichever gun is currently selected is fired whenever a loud noise is heard in order to keep the audience participation attributes discussed earlier. Since the templates are actually shorter than the duration of the sounds from the gun, several matches may occur for each sound. The results are averaged, and the appropriate weapon is selected and fired (if a change of weapon is detected). This method allows the software to run in real time without causing too much load on the CPU.

4 System Architecture

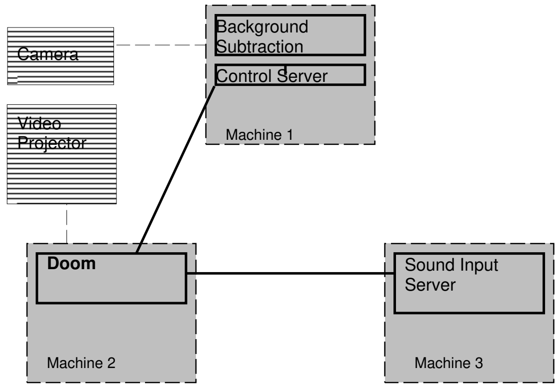

Figure 6: SURVIVE architecture.

The architecture for SURVIVE is diagramed in Figure 6. The figure ground segmentation module finds the user's contour and discovers regions of extremity-size (hands, head, feet) curvature. Both the contour and the regions of curvature are passed to the control server, which then determines which regions correspond with which parts of the body. The control server also performs gesture recognition and translates its results to Doom interface controls. The sound input server determines if the user has made a loud noise and distinguishes between gun sound effects.

Note that these modules are divided among three computers. Sockets and signal interrupts are used to allow the processes to remain asynchronous. This separation of the modules across machines is necessary to avoid lag in both the low level vision and sound processes. In this case, three smaller machines provide much more power than one high-end machine, which would suffer from continual paging as it tried to accommodate real-time sound, real-time video, and interactive graphics. However, this problem brings an interesting point to the fore. The processing power necessary to add features to a virtual environment of this scope can often be added by simply increasing the number of machines that are used. While this may seem obvious, it is not always so simple in some environments. For example, suppose more sophisticated vision routines are desired for identifying the user. One way to achieve this is to have the figure ground segmentation module send an appropriate pixmap of the user to another machine which would perform the recognition. However, this would slow down each machine as well as the network. A potential solution is to have both processes running in the same address space. Unfortunately, the background subtraction needs to be run at a high frame rate to prevent lag.

Alternatively, the NTSC video image could be daisy-chained to the recognition machine. Then only the few bytes needed to describe a bounding box and a recognition result are necessary to transmit between the machines. In it's current implementation, IVE uses both this "analog video bus" and a typical data bus to communicate information to its component processes. ALIVE, another environment based on the IVE technology, uses such an implementation to composite the user with the graphics. In the future, SURVIVE may also use this method to allow compositing the users' images with the computer graphics in a multiple player game.

5 Discussion

Doom allows players to compete against each other across the Internet. This attribute was used last fall to link two SURVIVE installations, one in Cambridge, MA and the other at BT (British Telecom) Laboratories in Woodbridge, England. The BT installation used two machines, but the audio process was simplified so that it did not affect game play. In addition, the BT installation used a professional speaker installation to envelop the user in the game's sounds. The effect was truly addictive, as demonstrated by BT employees' using their tea break to play SURVIVE.

SURVIVE demonstrates both the positive and negative aspects of using vision for game interfaces. First, the ability to enter the virtual environment just by stepping into sensing area is very important. The users do not have to spend time "suiting up," cleaning the apparatus, or untangling wires. Furthermore, social context is often important when using a virtual environment, whether it be for game playing or designing aircraft. In a head mounted display and glove environment, it is very difficult for a bystander to participate in the environment or offer advice on how to use the environment. With IVE, not only can the user see and hear a bystander, the bystander can easily take the user's place for a few seconds to illustrate functionality or refine the work that the original user was creating.

Another advantage of SURVIVE is that it does not involve specialty equipment. Even though game play would be slow and a large screen is desirable, an entry level SGI Indy can be used to run vision, graphics, and sound. While it may be a few years before such a system can be economically produced for a "set-top box," the availability of the equipment from a major vendor reduces the cost of acquiring, maintaining, and upgrading the apparatus in the future.

A detriment of the system is that gesture driven interfaces tend to have slower control than equivalent keyboard interfaces. For example, multiplayer SURVIVE is only interesting when both users control their characters by gesture. A keyboard player has faster control due to the fundamental physics of a keystroke versus and arm or body motion. Furthermore, the minimum video processing lag possible is 1/30 of a second. A solution may be to use predictive look-ahead techniques (for example, Kalman filtering) to help reduce the perceptual lag.

Gesture and voice interfaces, especially when used together, lend a greater sens of presence than keyboard or joystick interfaces. For example, we've had cases of normally soft-spoken people screaming at the virtual enemies in SURVIVE to fire the weapon. Furthermore, gesture interfaces are much richer than normal game interfaces. A wealth of information is carried in the player's body language, facial gestures, tone of voice, etc. In the future, we expect to use these attributes to provide an even more compelling interface.

These observations can be used by the entertainment industry. For example, an IVE interface may be better suited for "VR-cades" than the current equipment, since standard virtual reality equipment may cause uneasiness due to hygiene and comfort concerns. In addition, the IVE interface does not inherently limit the types of movement that can be detected due to wired interfaces (for example, participation a fight game with current virtual reality equipment might be hazardous). IVE also greatly reduces the chance of users hurting themselves through the interface.

With better social context, the IVE interface may assist in creating a more collaborative atmosphere. Since gesture interfaces generally imply more advanced hardware and slower response time, the role of an IVE participant in multi-player scenarios should be that of sophisticated "power user." For example, in role playing scenarios, the IVE player may be a "games master." Another good role is that of "starbase" in the game "Netrek." In general, the IVE interface demonstrates a new, low risk medium for the entertainment industry.

6 Conclusion

We have demonstrated a compelling virtual environment that does not intrude on the body of the user in any way and does not require specialty hardware. The advantages of such a system include ease of entry/exit, social context, maintenance, hygiene, greater safety, expandability, and a richer interface. In the future, not only will the entertainment industry benefit from this technology, but business and home users may use such techniques to communicate with their computers in general.

Acknowledgments

Thanks to Pattie Maes for her role in starting construction of such interactive environments here at the Media Laboratory. Many thanks to Bruce Blumberg and Michael P. Johnson for lending code and expertise to what started as "fun hack." Special thanks go to id Software for making Doom accessible and available. Finally, thanks go to the rest of the Vision and Modeling group and particularly the denizens of E15-368 for putting up with late night sounds of machine guns, rocket launchers, and the like while the system was being created.

Doom is a trademark of id Software Inc.

References

[1] Aukstakalnis, S. &Blatner, D. *Silicon Mirage*. Peachpit Press, 1992.

[2] Blumberg, B., et. al, in Proc. Conference on Simulation and Adaptive Behavior (SAB-94), Brighton, U.K., 1994

[3] Darrell T., Maes P., Blumberg B., and Pentland A. A Novel Environment for Situated Vision and Behavior, *IEEE Workshop on Visual Behaviors, CVPR conference* Seattle, June 1994

[4] Friedmann M., Starner T., and Pentland A. Device Synchronization using an Optimal Linear Filter, *ACM Conference on Interactive 3D Graphics*, 1992.

[5] Kalman, R. E. & Bucy, R. S. New results in linear filtering and prediction theory. In *Transaction ASME (Journal of basic engineering)*, 83D, 95-108. (1961).

[6] Krueger M.W., Artificial Reality II, Addison Wesley, 1990.

[7] Krueger M.W., Small Planet, SIGGRAPH-93 Visual Proceedings, Machine Culture, ACM SIGGRAPH, pp. 141, 1993.

[8] Vincent V.J., Mandala: Virtual Village, SIGGRAPH-93 Visual Proceedings, Tomorrow's Realities, ACM SIGGRAPH 1993, pp. 207, 1993.

[9] Maes, P., et al., Alive: An Artificial LIfe Interactive Video Environment, SIGGRAPH Visual Proceedings, Orlando, 1993.