M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 372
Appears in Applied Artificial Intelligence, Vol. 11, No. 4, June 1997

Perceptive Spaces for Performance and Entertainment: Untethered Interaction using Computer Vision and Audition

Christopher R. Wren Flavia Sparacino Ali J. Azarbayejani
Trevor J. Darrell Thad E. Starner Akira Kotani Chloe M. Chao
Michal Hlavac Kenneth B. Russell Alex P. Pentland

Perceptual Computing Section, The MIT Media Laboratory ; 20 Ames St., Cambridge, MA 02139 USA
{cwren,flavia,ali,trevor,thad,akira,cchao,hlavac, kbrussel,sandy}@media.mit.edu
http://vismod.www.media.mit.edu/groups/vismod/

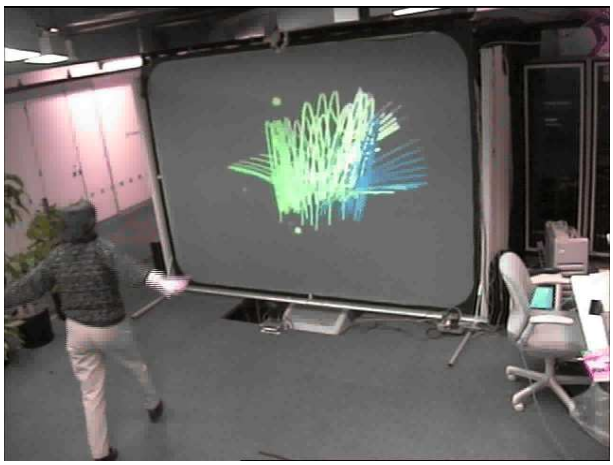


Figure 1: User dancing in a perceptive space and generating graphics.

Abstract

Bulky head-mounted displays, data gloves, and severely limited movement have become synonymous with virtual environments. This is unfortunate since virtual environments have such great potential in applications such as entertainment, animation by example, design interface, information browsing, and even expressive performance. In this paper we describe an approach to unencumbered, natural interfaces called Perceptive Spaces. The spaces are unencumbered because they utilize passive sensors that don't require special clothing and large format displays that don't isolate the user from their environment. The spaces are natural because the open environment facilitates active participation. Several applications illustrate the expressive power of this approach, as well as the challenges associated with designing these interfaces.

1 Introduction

We live in 3-D spaces, and our most important experiences are interactions with other people. We are used to moving around rooms, working at desktops, and spatially organizing our environment. We've spent a lifetime learning to competently communicate with other people. Part of this competence undoubtedly involves assumptions about the perceptual abilities of the audience. This is the nature of people.

It follows that a natural and comfortable interface may be designed by taking advantage of these competences and expectations. Instead of strapping on alien devices and weighing ourselves down with cables and sensors, we should build remote sensing and perceptual intelligences into the environment. Instead of trying to recreate a sense of place by strapping video-phones and position/orientation sensors to our heads, we should strive to make as much of the real environment as possible responsive to our actions.

Very few remote-sensing technologies live up to these goals; humans have evolved to primarily use vision and audition as their sources of perceptual information. We have therefore chosen to build vision and audition systems to obtain the necessary detail of information about the user. We have specifically avoided solutions that require invasive methods: like special clothing, unnatural environments, or even radio microphones.

This paper describes a collection of technology and experiments aimed at investigating this domain of interactive spaces. Section 2 describes some our solutions to the non-invasive interface problem. Section 3 discusses some of the design challenges involved in applying these solutions to specific application domains.

2 Unencumbered Interface Technology

While many advances have been made in creating interactive 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 encum-

ber the user with head-mounted displays, electro-magnetic or sonic position sensors, gloves, and/or body suits [2]. 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 they want 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 [11, 1, 19].

The ability to enter the virtual environment just by stepping into the 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 unencumbered interfaces, 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. This section describes the methods we use to create such systems.

2.1 The Interactive Space

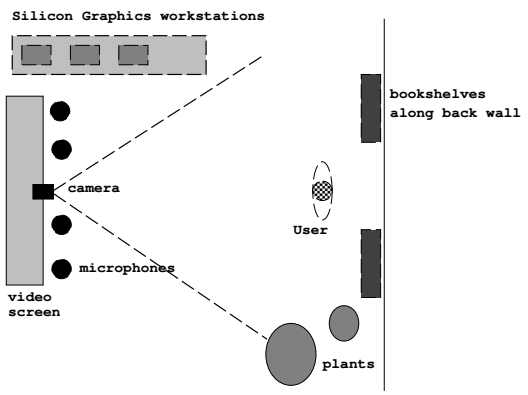


Figure 2: Interactive Virtual Environment hardware.

Figure 2 demonstrates the basic components of an Interactive Space that occupies an entire room. We also refer to this kind of space as an Interactive Virtual Environment (IVE). 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 wide-angle video camera mounted on top of the projection screen allows the system to track the user (see Section 2.2). A phased array microphone (see Section 2.4) is mounted above the display screen. A narrow-angle camera mounted on a pan-tilt head is also available for fine visual sensing. One or more Silicon Graphics computers are used to monitor the input devices in real-time.[19].

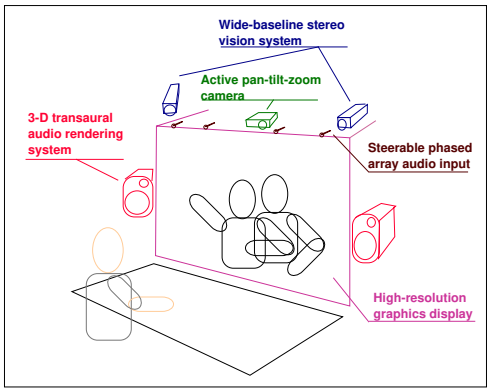


Figure 3: An Instrumented Desktop

Another kind of Interactive Space is the desktop. Our prototype desktop systems consist of a medium sized projection screen (4'x5') behind a small desk (2'x5'—See Figure 3). The space is instrumented with a wide-baseline stereo camera pair, an active camera, and a phased-array microphone. This configuration allows the user to view virtual environments while sitting and working at a desk. Gesture and manipulation occur in the workspace defined by the screen and desktop. This sort of interactive space is better suited for detailed work.

2.2 Vision-based Blob Tracking

Applications such as unencumbered virtual reality interfaces, performance spaces, and information browsers all have in common the need to track and interpret human action. The first step in this process is identifying and tracking key features of the user's body in a robust, real-time, and non-intrusive way. We have chosen computer vision as one tool capable of solving this problem across many situations and application domains.

We have developed a real-time system called Pfinder[21] ("person finder") that substantially solves the problem for arbitrarily complex but single-person, fixed-camera situations[1](see Figure 4a). The system has been tested on thousands of people in several installations around the world, and has been found to perform quite reliably.[21]

Pfinder is descended from a variety of interesting experiments in human-computer interface and computer mediated communication. Initial exploration into this space

[1] Use of existing image-to-image registration techniques [3, 14] allow Pfinder to function in the presence of camera rotation and zoom, but real-time performance cannot be achieved without special-purpose hardware.

Figure 4: Analysis of a user in the interactive space. Frame **(left)** is the video input (n.b. color image possibly shown here in greyscale for printing purposes), frame **(center)** shows the segmentation of the user into blobs, and frame **(right)** shows a 3-D model reconstructed from blob statistics alone (with contour shape ignored).

of applications was by Krueger [11], who showed that even 2-D binary vision processing of the human form can be used as an interesting interface. More recently the Mandala group [1], has commercialized and improved this technology by using analog chromakey video processing to isolate colored garments worn by users. In both cases, most of the focus is on improving the graphics interaction, with the visual input processing being at most a secondary concern. Pfinder goes well beyond these systems by providing a detailed level of analysis impossible with primitive binary vision.[21]

Pfinder is also related to body-tracking projects like Rehg and Kanade [17], Rohr [18], and Gavrila and Davis [9] that use kinematic models, or Pentland and Horowitz [16] and Metaxas and Terzopolous [15] who use dynamic models. Such approaches require relatively massive computational resources and are therefore not appropriate for human interface applications.

Pfinder is perhaps most closely related to the work of Bichsel [6] and Baumberg and Hogg [5]. The limitation of these systems is that they do not analyze the person's shape or internal features, but only the silhouette of the person. Pfinder goes beyond these systems by also building a blob-based model of the person's clothing, head, hands, and feet. These blob regions are then tracked in real-time using only a standard Silicon Graphics Indy computer. This allows Pfinder to recognize even complex hand/arm gestures, and to classify body pose (see Figure 4b)[21].

Pfinder uses a stochastic approach to detection and tracking of the human body using simple $2\frac{1}{2}$-D models. It incorporates *a priori* knowledge about people primarily to bootstrap itself and to recover from errors. This approach allows Pfinder to robustly track the body in real-time, as required by the constraints of human interface.[21]

We find RMS errors in pfinder's tracking on the order of a few pixels, as shown in Table 1. Here, the term

test	hand	arm
translation (X,Y)	0.7 pixels (0.2% rel)	2.1 pixels (0.8% rel)
rotation (Θ)	4.8 degrees (5.2% rel)	3.0 degrees (3.1% rel)

Table 1: Pfinder Estimation Performance

"hand" refers to the region from approximately the wrist to the fingers. An "arm" extends from the elbow to the fingers. For the translation tests, the user moves through the environment while holding onto a straight guide. Relative error is the ratio of the RMS error to the total path length.

For the rotation error test, the user moves an appendage through several cycles of approximately 90 degree rotation. There is no guide in this test, so neither the path of the rotation, nor even its absolute extent, can be used to directly measure error. We settle for measuring the noise in the data. The RMS distance to a low-pass filtered version of the data provides this measure.

Pfinder provides a modular interface to client applications. Several clients can be serviced in parallel, and clients can attach and detach without affecting the underlying vision routines. Pfinder performs some detection and classification of simple static hand and body poses. If Pfinder is given a camera model, it also back-projects the 2-D image information to produce 3-D position estimates using the assumption that a planar user is standing perpendicular to a planar floor (see Figure 4c)[21].

2.3 Stereo Vision

The monocular-Pfinder approach to vision generates the $2\frac{1}{2}$-D user model discussed above. That model is suffi-

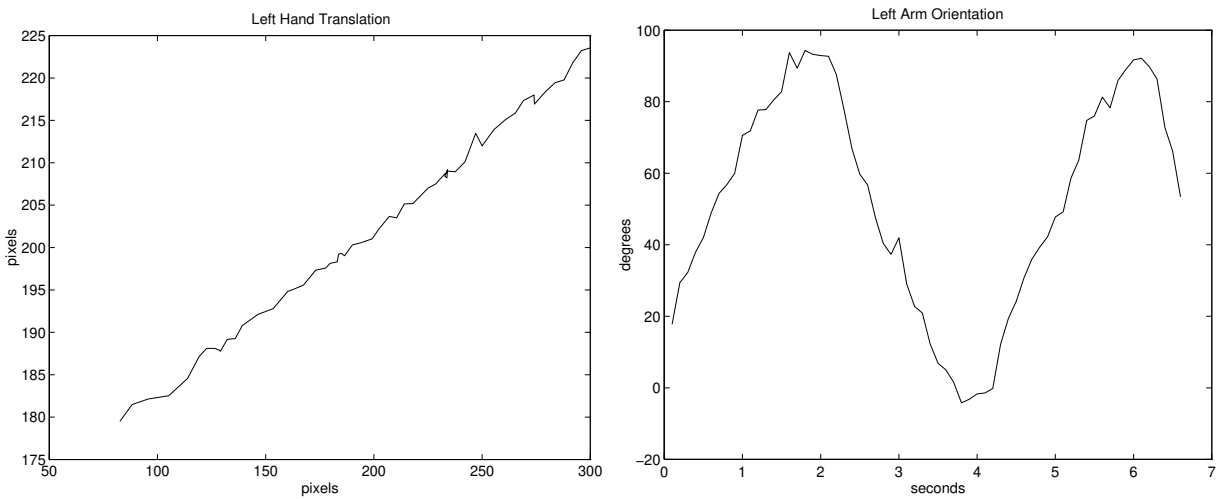


Figure 5: **(left)** shows data from hand tracking while the hand was slid along a straight guide. **(right)** shows a similar experiment for rotation

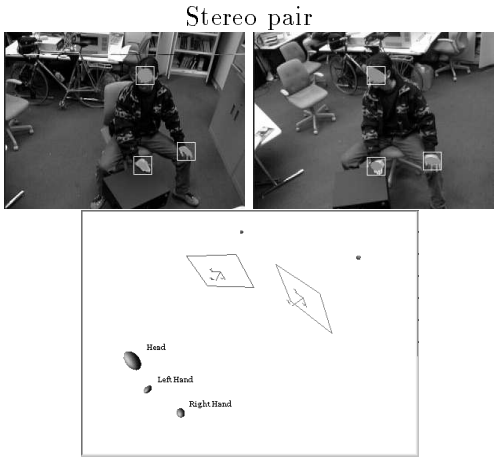


Figure 6: Real-time estimation of position, orientation, and shape of moving human head and hands.

cient for many interactive tasks. However, some tasks do require more exact knowledge of body-part positions.

Our success at 2-D tracking motivated our investigation into recovering useful 3-D geometry from such qualitative, yet reliable, feature finders. We began by addressing the basic mathematical problem of estimating 3-D geometry from blob correspondences in displaced cameras. The relevant unknown 3-D geometry includes the shapes and motion of 3-D objects, and optionally the relative orientation of the cameras and the internal camera geometries. The observations consist of the corresponding 2-D blobs, which can in general be derived from any signal-based similarity metric.[4]

We use this mathematical machinery to reconstruct 3-D hand/head shape and motion in real-time (about 10 to 15 frames per second) on a pair of SGI Indy workstations without any special-purpose hardware. In tests similar to those used with pfinder (see Section 2.2), we find RMS errors on the order of a few centimeters or degrees, as shown in Table 2. The translation errors are larger than the corresponding translation errors in the 2-D case because estimation along the Z axis is a mathematically ill-conditioned problem.

This stereo information is used by client applications much the same way the 2-D tracking is used: either as direct input to an interface application, or as input to a gesture recognition layer.[4]

2.4 Visually Guided Input Devices

Robust knowledge of body part position and body pose enables more than just gross gesture recognition. It provides boot-strapping information for other methods to determine more detailed information about the user. Electronically steer-able phased array microphones can use the

test	hand
translation (X,Y,Z)	2.55 cm (1.8% rel)
rotation (Θ,Φ,Ψ)	1.98 degrees (2.2% rel)

Table 2: Stereo Estimation Performance

head position information to reject environmental noise. This provides the signal-to-noise gain necessary for remote microphones to be useful for speech recognition techniques [7]. Active cameras can also take advantage of up-to-date information about body part position to make fine distinctions about facial expression, identity, or hand posture.[8]

3 Perceptive Spaces

Unencumbered interface technologies do not, by themselves, constitute an interface. A mapping must exist between the input technology and the system to be manipulated. This mapping must be carefully chosen, because it defines the metaphor that the user is forced use when they interact with the system. The desired level of abstraction, tolerance to interface accuracy and lag, even the prior expectations of the user must be taken into account when designing this mapping.

This section describes several systems that have been built in our lab, each with a distinct interface/system mapping. The focus will be on these interface mappings: how they work with the interface technology, and also how they affect the interactive experience.

3.1 SURVIVE

Figure 7: The user environment for SURVIVE.

The simplest mapping is, of course, the direct one: map interface device features directly (one-to-one) into the control space of some application. Usually a small amount of filtering will be required, and possibly it's desirable to use non-linear mappings, but otherwise interface outputs feed directly into application inputs.

SURVIVE (Simulated Urban Recreational Violence Interactive Virtual Environment) is an entertainment application that uses a direct mapping. SURVIVE allows the user to interact with a 3D game environment using the IVE space. Figure 7 shows a user in SURVIVE. The gestural interpretation provided by the vision system (Section 2.2) is mapped into the game controls for the popular id Software game Doom.

The user holds a large (two-handed) toy gun, and moves around the IVE stage. Position on the stage is fed into Doom's directional velocity controls. The hand position features are used to drive Doom's rotational velocity control. The results of a matched-filter on and audio input stream provide control over weapon changes and firing. This direct mapping, given the application, may be called "user-as-joystick".[19]

Although simplistic, this mapping has some very important features: low lag, intuitive control strategy, and a control abstraction well suited to the task. The mapping requires little post-processing of the interface features, so it adds very little lag to the interface. Since many games have velocity-control interfaces, people adapt quickly to the control strategy because it meshes with their expectations about the game.

Finally, it's insightful to contrast the SURVIVE interface with the standard keyboard Doom interface. The task in Doom is navigating through a virtual environment. This is usually accomplished by pressing keys on a keyboard. Changing the direction of travel is as easy as picking up one finger and pressing down another. Split-second decisions become split-second actions. The SURVIVE interface is much less forgiving. Movement of the virtual body is linked to the movement the real body. A change of virtual direction actually requires a movement in that direction, maybe several feet of movement. This leads to a much more engrossing, visceral experience of the game.

Interestingly, even when people use the keyboard interface, they tend to move their heads, and sometime their whole body, while playing the game. SURVIVE capitalizes on this natural link between visual and visceral experience to create a more immersive, if more physically demanding, experience.

3.2 Visually-Animated Characters

A literal mapping is one that treats the tracking features as exactly what they are: evidence about the physical configuration of the user in the real world. In this context the tracking information becomes useful for understanding simple pointing gestures. With quite a bit more work, systems can use this information to estimate a more complete picture of the user's configuration.

Figure 8: A synthetic character taking direction from a human user who is being tracked in 3-D with stereo vision

Complex 3-D characters can be built up and rendered using high-speed graphics rendering hardware, but they tend to lack natural coordinated movement because animators have to move joint angles individually. This problem is often solved using "motion-capture" systems in which a user is instrumented with accurate sensors to measure the locations and angles of joints whose dynamic trajectories are used to animate corresponding locations and angles of joints on the character (see Figure 8).

In a perceptual space instrumented with multiple cameras, the same procedure can be done passively with vision systems. We have implemented a system in which the stereo system described in Section 2.3, is combined with a literal mapping between user configuration and corresponding parts of an animated character.

The system allows the user to animate the 3-D head and hand movements of a virtual puppet by executing the corresponding motions in the perceptual space. The features from the vision system drive the endpoints of a kinematic engine inside the puppet.

3.3 NetSpace

A gesture-based interface mapping interposes a layer of pattern recognition between the input features and the application control. When an application has a discrete control space, this mapping allows patterns in feature space, better known as gestures, to be mapped to the discrete inputs. The set of patterns form a gesture-language that the user must learn. It is worth noting that this kind of rigid gesture-language tends to be sensitive to failures in tracking, classification, and user training. Systems that employ this kind of mapping must have very flexible, and quick, mechanisms for resolving misunderstandings. See Sections 3.4 and 3.5, for interesting answers to this problem. Netspace is an example of an application that uses a gesture-based mapping.

NetSpace is an immersive, interactive web browser that makes use of people's strength at remembering the surrounding 3D spatial layout. For instance, everyone can easily remember where most of the hundreds of objects in their house are located. In comparison to our spatial memory, our ability to remember other sorts of information is greatly impoverished. NetSpace capitalizes on this ability by mapping the contents of URLs into a 3D graphical world projected on the large IVE screen. This gives the user a sense the URLs existing in a surrounding 3D environment.

NetSpace was conceived as a natural extension to Hyperplex [20], our first experiment using IVE as an immersive browser for movies. To navigate this virtual 3D environment, users stand in front of the screen and use voice and hand gestures to explore (Figure 9). Pointing to a link will highlight the corresponding text and either advancing towards to IVE screen or saying "there" will load the new URL page. The user can scroll up and down a page by pointing up and down with either arm. When a new page is loaded, the virtual camera of the 3D graphics world will automatically move to a new position in space that constitutes an ideal viewpoint for the current page.

The URLs are displayed so as to form a landscape of text and images through which the user can "fly". When the user wants to see previously loaded pages they open up their arms in flying mode and visit the web landscape by moving their body left/right, closer to the screen, or by tilting their arms to tilt the virtual camera.

The browser currently supports standard HTML with pictures and MPEG movies. Future extensions include stereo browsing, with the use of Crystal Eyes glasses, and exploring a variety of web landscape architectures.

(a)

(b)

Figure 9: **(a)** User browsing the web in NetSpace **(b)** NetSpace landscape with some of the authors' web pages

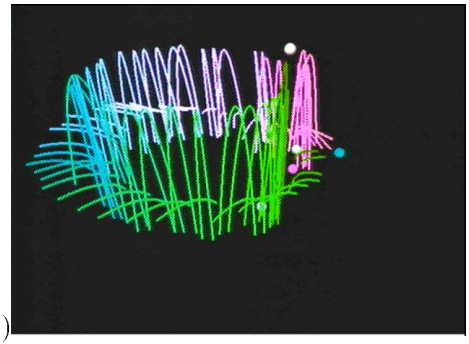
(a)
(b) 

Figure 10: **(a)** User dancing with her colored shadow in DanceSpace **(b)** Dancing shadow generated by the user in DanceSpace

3.4 DanceSpace

Closely related to the gesture-based interface mapping discussed in Section 3.3, the conductor-style interface mapping of DanceSpace also uses a form of predefined gesture language. The important difference lies in the design of that language. The gesture-language of NetSpace is very rigid. Specific gesture sequences generate specific reactions, and conversely, failures in the tracking and classification of the user's actions can result in inappropriate actions by the system. The conductor mapping results in a much more fluid interface. The user can certainly try to explore the control space, learn it, and use it as a rigid language, but the system is designed to produce constructive, interesting results when this doesn't happen. The interactions the user has with this system are arguably more interesting when the user doesn't know the details of the mapping.

DanceSpace is an interactive performance space where both professional and non-professional dancers can generate music and graphics through their body movements (See Figure 10).

The music begins with a richly-textured melodic base tune which plays in the background for the duration of the performance. As the dancer enters the space, a number of virtual musical instruments are invisibly attached to their body. The dancer then uses their body movements to magically generate an improvisational theme above the background track.

The dancer has a cello in their right hand, vibes on their left hand, and bells and drums attached to their feet. The dancer's head works as the volume knob, bringing down the sound as they move closer to the ground. The distance from the dancer's hands to the ground is mapped to the pitch of the note played by the musical instruments attached to the hands. Therefore a higher note will be played when the hands are above the performer's head and a lower note when they are near their waist. Both hands' musical instruments are played in a continuous mode (i.e., to get from a lower to a higher note the performer will have to play all the intermediate notes). The bells and the drums are on the contrary "one shot" musical instruments. When the performer raises their feet more than 15 inches

off the ground then either of the bells/drums are triggered, according to which foot is raised.

The music that is generated varies widely among different users of the interactive space. Nevertheless all the music shares the same pleasant rhythm established by the underlying, ambient tune, and a style the ranges from "pentatonic" to "fusion" or "space" music.

As the dancer moves, their body leaves a multicolored trail across the large wall screen that comprises one side of the performance space.

The graphics is generated by drawing two bezier curves to abstractly represent the dancer's body. The first curve is drawn through coordinates representing their left foot, head, and right foot. The second curve is drawn through coordinates representing their left hand, center of their body, and right hand. Small 3-D spheres are also drawn to map onto hands, feet, head and center of the body of the performer, both for a reference for the dancer and to accentuate the stylized representation of the body on the screen. The multicolored trail is intended to represent the dancer's shadow that follows them around during the performance. The shadow has a variable memory of the number of trails left by the dancer's body. Hence if the shadow has a long memory of trails (more than thirty) the dancer can paint more complex abstract figures on the screen.

The choreography of the piece can then vary according to which one of the elements of the interactive space the choreographer decides to privilege. In one case the dancer might concentrate on generating the desired musical effect; in another case or in another moment of the performance, the dancer may want to concentrate on the graphics - i.e. painting with the body - or finally the dancer might just focus on the dance itself and let DanceSpace generate the accompanying graphics and music.

The philosophy underlying DanceSpace is inspired by Merce Cunningham's approach to dance and choreography [10]. The idea is that dance and movement should be designed independently of music and that music should be subordinate to movement and may be composed later for a piece as a musical score is done for film. When concentrating on music, more than dance, DanceSpace can be thought of as a "hyperinstrument"[12]. Hyperinstruments are musical instruments primarily invented for non-musical-educated people who nevertheless wish to express themselves through music. The computer that drives the instruments adds the basic layer of "musical knowledge" needed to generate a musical piece. Moreover we have thought of DanceSpace as a tool for a dancer/mime to act as a street performer who has a number of musical instruments attached to their body. The advantage of DanceSpace over the latter is that the user is unencumbered and wireless and can be more expressive in other media as well (its own body or computer graphics). The disadvantage is that DanceSpace is mainly a music improvising system and it is therefore difficult to use it to reproduce well known musical tunes.

Future improvements to DanceSpace include having a number of different background tunes and instruments available for the dancer to use within the same performance. Another important addition will also allow the user to adjust the music's rhythm to their rhythm of movement. We would also like the color of the dancer's graphical shadow to match an expressive or emotional pattern in the dance and become an active element in the choreography of the piece.

We see DanceSpace as a possible installation for indoor public places, as for example airports, where people usually spend long hours waiting, or interactive museums and galleries. DanceSpace could also become part of a performance space, allowing a dancer to play with their own shadow and generate customized music for every performance.

3.5 ALIVE

Figure 11: Chris Wren playing with Bruce Blumberg's virtual dog in the ALIVE space

The last of the gesture-language mappings is the most abstract. Again, it's related to the other gesture-languages discussed above, and the primary distinction lies in a subtle, but important, difference in the design of the interface. Best called "gesture in context" this mapping attempts to create an interface that is intuitive given the context. Ideally, the mapping is aligned so that failures in tracking or classification are transparent to the user. Clever mapping design can thus greatly reduce the need for sensor systems to perform flawlessly by playing off the expectations and socialization of the user. Because of that trait, this was the first system to be implemented in our lab, in the form of the Artificial Life Interactive Virtual Environment (ALIVE).

ALIVE combines autonomous agents with an interactive space. The user experiences the agents (including

hamster-like creatures, a puppet, and a well-mannered dog—Figure 11) through a "magic-mirror" idiom. The interactive space mirrors the real space on the other side of the projection display, and augments that reflected reality with the graphical representation of the agents and their world (including a water dish, partitions, and even a fire hydrant).

The "magic-mirror" paradigm is attractive because it provides a set of domain constraints which are restrictive enough to allow simple vision routines to succeed, but is sufficiently unencumbered that is can be used by real people without training or a special apparatus.[13]

One agent the user can interact with in ALIVE is a puppet that tries to act like a small child. The user can interact with the agent using certain hand gestures, which are interpreted in the context of the particular situation. For example, when the user points away and thereby sends the puppet away, the puppet will go to a different place depending on where the user is standing. If the user waves or comes towards the puppet after it has been sent away, this gesture is interpreted to mean that the user no longer wants the puppet to go away, and so the puppet will smile and return to the user. In this manner, the gestures employed by the user can have rich meaning which varies on the previous history, the agents internal needs and the current situation. [13]

4 Conclusion

The preceding examples illustrate successful interfaces built for a wide range of application domains from animation to artistic expression to information browsing. They all differ in the mappings they employ between sensed features, and application control. However, they all have in common the use of remote sensing technology coupled with perceptual intelligence built into the environment. The common idea is the realization that state-of-the-art vision and audition systems are capable of providing enough information to drive interactive systems, and that they provide that information in a non-invasive way that is compatible with social, natural, and creative interaction.

By adding intelligence to the surrounding space to make it responsive to the user, Perceptive Spaces offer new venues for art and entertainment. They provide solutions to man-machine interface design problems that have historically been difficult or impractical to address with traditional technologies. We believe that the notion of a perceptual space will become central to future entertainment installations, and that this technology has the potential to enhance human expressive abilities.

5 References

[1] ACM. *Mandala: Virtual Village*, ACM SIGGraph, Computer Graphics Visual Proceedings, 1993.

[2] S. Aukstakalnis and D. Blatner. *Silicon Mirage*. Peachpit Press, 1992.

[3] A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

[4] Ali Azarbayejani and Alex Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. In *Proceedings of 13th ICPR*, Vienna, Austria, August 1996. IEEE Computer Society Press.

[5] A. Baumberg and D. Hogg. An efficient method for contour tracking using active shape models. In *Proceeding of the Workshop on Motion of Nonrigid and Articulated Objects*. IEEE Computer Society, 1994.

[6] Martin Bichsel. Segmenting simply connected moving objects in a static scene. *Pattern Analysis and Machine Intelligence*, 16(11):1138–1142, Nov 1994.

[7] Michael A. Casey, William G. Gardner, and Sumit Basu. Vision steered beam-forming and transaural rendering for the artificial life interactive video environment (alive). In *Proceedings of the 99th Convention of the Aud. Eng. Soc.* AES, 1995.

[8] T. Darrell, B. Moghaddam, and A. Pentland. Active face tracking and pose estimation in an interactive room. In *CVPR96*. IEEE Computer Society, 1996.

[9] D. M. Gavrila and L. S. Davis. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *International Workshop on Automatic Face- and Gesture-Recognition*. IEEE Computer Society, 1995. Zurich.

[10] James Klosty. *Merce Cunningham: dancing in space and time*. Saturday Review Press, 1975.

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

[12] Tod Machover. *HyperInstruments: A Composer's Approach to the Evolution of Intelligent Musical Instruments*, pages 67–76. Miller Freeman, 1992.

[13] Pattie Maes, Bruce Blumberg, Trevor Darrell, and Alex Pentland. The alive system: Full-body interaction with animated autonomous agents. *ACM Multimedia Systems*, 5:105–112, 1997.

[14] S. Mann and R. W. Picard. Video orbits: characterizing the coordinate transformation between two images using the projective group. *IEEE T. Image Proc.*, 1997. To appear.

[15] D. Metaxas and D. Terzopoulos. Shape and non-rigid motion estimation through physics-based synthesis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15:580–591, 1993.

[16] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):730–742, July 1991.

[17] J.M. Rehg and T. Kanade. Visual tracking of high dof articulated structures: An application to human hand tracking. In *European Conference on Computer Vision*, pages B:35–46, 1994.

[18] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94–115, Jan 1994.

[19] Kenneth Russell, Thad Starner, and Alex Pentland. Unencumbered virtual environments. In *IJCAI-95 Workshop on Entertainment and AI/Alife*, 1995.

[20] Flavia Sparacino, Christopher Wren, Alex Pentland, and Glorianna Davenport. Hyperplex: a world of 3d interactive digital movies. In *IJCAI-95 Workshop on Entertainment and AI/Alife*, 1995.

[21] Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.