M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 350, Sept. 1995
Appeared: 7^{th} Int'l Forum on Frontier of Telecommunication Technology, Nov. 1995, Tokyo, Japan

Machine Understanding of Human Action

Alex Pentland
Perceptual Computing Section, The Media Laboratory
Massachusetts Institute of Technology, 20 Ames St., Cambridge, MA 02139
Electronic: sandy@media.mit.edu, http://www-white.media.mit.edu/vismod/demos/ive/

Abstract

The Perceptual Computing Section of the Media Laboratory is working to make computers that understand people, and can work with them in the manner of an attentive human-like assistant. To this end we have built a series of interactive office spaces that are used as real-time experimental testbeds. These spaces are instrumented with cameras and microphones, and perform audio-visual interpretation of human users. Real-time capabilities include 3-D tracking of head, hands, and feet; "holographic" audio; face recognition; and interpretation of face and hand gestures. People in the space can control programs, browse multimedia information, and experience shared virtual environments without wires or special goggles.

1 Introduction

A major research goal during the last thirty years has been to make computers intelligent, to work with us, and to be our helpers. Despite much effort, however, computers today remain difficult to use, and very much at the margins of human life.

I believe that this is principally because computers are deaf and blind. They experience the world around them through a slow serial line to a keyboard and mouse. Even when they handle signals, like sounds or images, they usually do so only as a transport device that knows nothing of the signals' content.

If you imagine raising a child in a closed box with only a telegraph connection to the outside world, you can quickly realize how difficult it is for computers to become intelligent and helpful. They exist in a world that is almost completely disconnected from ours, so how can they understand us?

To help us, a computer needs to share our perceptual environment. They need to be situated in the same world that we are...and the most important elements of that world are people. They need to know much more than just a text transcription of our words; they also need to know who we are, see our expressions and gestures, and hear the tone and emphasis of our voice.

Unfortunately, the scientific community has mostly avoided dealing with the problems of recognizing people, their actions, and their intentions in unencumbered, natural environments. It has been too easy to impose restrictions on people in order to make the problem easier. Consequently, human-computer interaction today requires people to poke at complicated input devices, use special computer languages, and to sit in front of (or wear!) special output devices. People can't interact with a computer anything like they do with people; it take years of training to use a computer well....just think of how difficult it was to learn to type!

2 An Experimental Testbed

The Perceptual Computing Section of the M.I.T. Media Laboratory was founded with the goal of making it possible for computers and people to engage in a full, natural dialogue. The first step toward making this dialogue possible is to make computers recognize who they are working with, and to be sensitive to people's gestures and expressions. To perform research on these problems we have created a series of experimental testbeds at the Media Laboratory.

These testbeds are performance spaces known as Interactive Video Environments (IVE) [4]. IVE spaces are instrumented with cameras and microphones, which allow the computer to see, hear, and interpret users' actions. People in an IVE can control programs, browse multimedia information, and experience shared virtual environments without keyboards, special sensors, or special goggles. Important near-term applications of this technology include practical interfaces to large-screen displays and virtual environments.

The IVE system was originally developed by Darrell, Maes, Blumberg and Pentland [4] to provide vision input for the ALIVE ("artificial life interactive video environment") system [8]. This system performed real-time person tracking but had no explicit model of the person and required a controlled background. In the succeeding years we have developed this system in to the more general, and more accurate, system described here.

While this technology began within the Media Lab, it has now grown into a collaborative experiment with laboratories around the world. There are currently five IVEs, three at two different locations in Boston, and one each at ATR in Japan, and BT labs in the UK. They are networked together to allow collaborative experiments. Additional installations are planned at research labs in Paris, New York, and Dallas.

The computer architecture employed is modular, allowing it to be configured in a variety of ways, as more or less interpretation capability is required. The basic element is an SGI Indy computer; these computers are connected by video, audio, ISDN, and ethernet networks. This allows each computer to independently access whatever portion of the audiovisual input it needs, to share control information among the other computers, and to connect with other installations over ISDN phone lines. Normally each computer is dedicated to perform only one type of interpretation. For instance, if video tracking, audio input, and gesture interpretation capabilities were desired, we would use three computers, one for each task.

The following sections will briefly describe each of the modules used to find, track, and interpret human actions. The last section will illustrate the use of these modules in various applications. In order of description, the modules are:

- Pfinder, a real-time program that tracks the user in 3-D, and recognizes a basic set of hand gestures and body postures,
- The face processor, an interactive-time program that recognizes people by their face,
- The expression processor, an interactive-time program that uses motion-energy templates to classify people's facial expressions,
- Vision-directed audio, a real-time system that uses the head position information provided by Pfinder to steer a phase-array microphone and to free-field 3-D sound synthesis,
- Action recognition, two real-time systems that examine user's movements in order to determine what they are doing.

For additional detail the reader should examine the numerous papers and technical reports available at our web site, http://www-white.media.mit.edu/vismod

3 Pfinder

Pfinder ("person finder") is a real-time system for tracking and interpretation of people. It runs at 10hz on a standard SGI Indy computer, and has performed reliably on thousands of people in many different physical locations. The system uses a multi-class statistical model of color and shape to segment a person from a background scene, and then to find and track people's head and hands in a wide range of viewing conditions. Pfinder produces a real-time representation of a user useful for applications such as wireless interfaces, video databases, and low-bandwidth coding, without cumbersome wires or attached sensors.

Pfinder employs a maximum likelyhood (ML) approach by using simple $2\frac{1}{2}$-D models to perform detection and tracking of the human body. It incorporates *a priori* knowledge about people primarily to bootstrap itself and to recover from errors.

In order to present a concise description of Pfinder, we will only describe its representations and operation in the "steady state" case, where it is tracking a person moving around an office environment, neglecting the difficult problems of learning a person's description. For additional detail see Wren, Darrell, Azarbayejani, and Pentland [15].

3.1 Modeling The Person

Pfinder models the human as a connected set of blobs. Each blob has a spatial (x, y) and color (Y, U, V) Gaussian distribution. We define $\mathbf{m}_k$ to be the mean (x, y, Y, U, V) of blob k, and $\mathbf{K}_k$ to be the covariance of that blob's distribution. Because of their different semantics, the spatial and color distributions are assumed to be independent. That is, $\mathbf{K}_k$ is block-diagonal, with uncoupled spatial and spectral components.

Each blob also has associated with it a *support map* that indicates which image pixels are members of a particular blob. An aggregate support map $s(x, y)$ over all the blob models is also a useful data structure. Since the individual support maps indicate which image pixels are members of that particular blob, the aggregate support map represents the segmentation of the image into spatial/color classes.

Each blob can also have a detailed representation of its shape and appearance, modeled as differences from the underlying blob statistics. The ability to efficiently compute compact representations of people's appearance is useful for low-bandwidth telecommunications, such as our demonstration of a shared virtual envionments at SIGGRAPH '95 [5].

The statistics of each blob are recursively updated to combine information contained in the most recent measurements with knowledge contained in the current class statistics and the priors. Because the detailed dynamics of each blob are unknown, we use approximate models derived from experience with a wide range of users. For instance, blobs that are near the center of mass have substantial inertia, whereas blobs toward the extremities can move much faster.

3.2 Modeling The Scene

We assume that the majority of the time Pfinder will be processing a scene that consists of a relatively static situation such as an office, and a single moving person. Consequently, it is appropriate to use different types of model for the scene and for the person.

We model the scene surrounding the human as a texture surface; each point on the texture surface is associated with a mean color value and a distribution about that mean. Color is expressed in the YUV space. The color distribution of each pixel is modeled with the Gaussian described by a full covariance matrix. Thus, for instance, a fluttering white curtain in front of a black wall will have a color covariance that is very elongated in the luminance direction, but narrow in the chrominance directions.

In each frame visible pixels have their statistics recursively updated using a simple adaptive filter. This allows us to compensate for changes in lighting and even for object movement. For instance, if a person moves a book it causes the texture map to change in both the locations where the book was, and where it now is. By tracking the person we can know that these areas, although changed, are still part of the texture model and thus update their statistics to the new value. The updating process is done recursively, and even large changes in illumination can be substantially compensated within two or three seconds.

3.3 The Analysis Loop

Given a person model and a scene model, Pfinder then acquires a new image, interprets it, and updates the scene and person models. To accomplish this there are several steps:

1. First, we predict the appearance of the user in the new image using the current state of our model. This is accomplished using a set of Kalman filters with simple Newtonian dynamics that operate on each blob's spatial statistics.
2. Next, for each image pixel we must measure the likelihood that it is a member of each of the blob models and the scene model. Self-shadowing and cast shadows are a particular difficulty in measuring this likelihood; we address this by normalizing the hue and saturation by the overall brightness.
3. Resolve these pixel-by-pixel likelihoods into a support map, indicating for each pixel whether it is part of one of the blobs or of the background scene. Spatial priors and connectivity constraints are used to accomplish this resolution.

Figure 1: Analysis of a user in the ALIVE environment. The frame on the left is the video input (n.b. color image shown here in black and white for printing purposes), the center frame shows the support map $s(x, y)$ which segments the user into blobs, and the frame on the right showing a 3-D model reconstructed from blob statistics alone (with contour shape ignored).

Figure 2: The user's hand and face blobs have very similar color statistics. These two blobs end up very close to each other in this frame.

4. Update the statistical models for each blob and for the background scene; also update the dynamic models of the blobs.

Figures 1 and 2 illustrate Pfinder's functioning. At the left is the original video frame (shown in black and white rather than the original color). In the middle is the resulting segmentation into head, hands, feet, shirt, and pants. At the right are one-standard-deviation ellipses illustrating the statistical blob descriptions formed for the head, hands, feet, shirt, and pants. Note that despite having the hands either in front of the face or the body a correct description is still obtained. For additional detail see reference [15].

3.4 Interface

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. Before the tracking information can be useful for some applications it is necessary to process the data through a few interpretive layers, for instance, to detect dynamic hand gestures or to interpret the hand-eye pointing geometry in 3-D by use of a camera model.

Pfinder defaultly provides some such interpretation. For instance, Pfinder detects and classifies static hand and body poses. When Pfinder is given a geometric model for the camera [1], it also backprojects the 2-D image information to produce 3-D position estimates using the assumption that the user is standing on a planar floor.

A wide range of data is exported through the Pfinder interface:

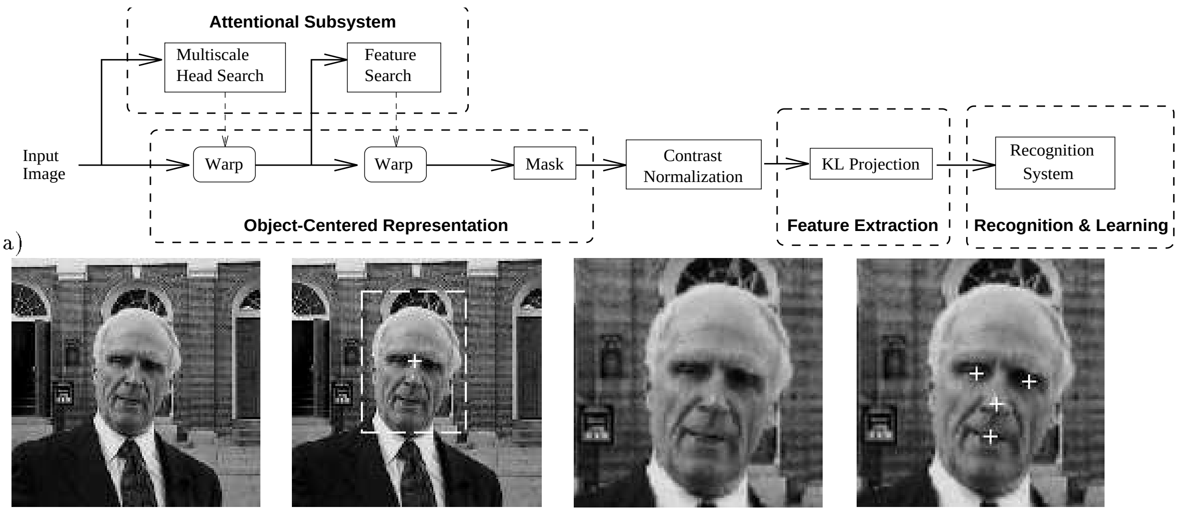
- a polygon representation of the support map,
- blob model statistics,
- semantically-labeled feature positions (e.g. head, right hand, etc.),
- static body postures and gestures (e.g. standing, sitting, pointing, etc.)

4 Face Recognition

Once we know roughly where the person's head is, we can attempt to recognize their face. As with Pfinder, we also employ a maximum likelyhood (ML) approach to this problem. We have accomplished this by developing a method for determining the probability distribution function for face images within a low-dimensional eigenspace. Knowlege of this distribution then allows us to precisely locate faces and face features, and to compare faces along meaningful dimensions. The following gives a brief description of this system; for additional detail see Moghaddam and Pentland [9].

4.1 Face and Feature Detection

The standard detection paradigm in image processing is that of normalized correlation or template matching. However this approach is only optimal in the simplistic case of a *deterministic* signal embedded in white Gaussian noise. When we begin to consider a target *class* detection problem — *e.g,* finding a generic human face or a human hand in a scene — we must incorporate the underlying probability distribution of the object of interest. Subspace or eigenspace methods, such as the KLT and PCA, are particularly well-suited to such a task since they provide a compact and *parametric* description of the object's appearance and also automatically identify the essential components of the underlying statistical variability.



(a) (b) (c) (d) (e)

Figure 3: (a) The face processing system, (b) original image, (c) position and scale estimate, (d) normalized head image, (e) position of facial features.

In particular, the eigenspace formulation leads to a powerful alternative to standard detection techniques such as template matching or normalized correlation. The reconstruction error (or residual) of the KLT expansion is an effective indicator of a match. The residual error is easily computed using the projection coefficients and the original signal energy. This detection strategy is equivalent to matching with a linear combination of *eigentemplates* and allows for a greater range of distortions in the input signal (including lighting, and moderate rotation and scale). Some of the low-order eigentemplates for a human face are shown in Figure 4(a). In a statistical signal detection framework, the use of eigentemplates has been shown to be orders of magnitude better than standard matched filtering [9].

Using this approach we have reformulated the target detection problem from the point of view of a ML estimation problem. In particular, given the visual field, estimate the position (and scale) of the subimage which is most representative of a specific target class Ω. Computationally this is achieved by sliding an m-by-n observation window throughout the image and at each location computing the *likelihood* that the given observation $\mathbf{x}$ is an instance of the target class Ω — *i.e,* $P(\mathbf{x}|\mathbf{\Omega})$. After this probability map is computed, we can select the location corresponding to the highest likelihood as our ML estimate of the target location.

4.2 The Face Processor

We have used this ML-based face finder as the basic building block of an automatic face recognition system. The function of the face finder is to very precisely locate the face, since Pfinder produces only low-resolution estimates of head location. The block diagram of the face finder system is shown in Figure 3 which consists of a two-stage object detection and alignment stage, a contrast normalization stage, a feature extraction stage, followed by recognition (and, optionally, facial coding.) Figure 3(b)-(e) illustrates the operation of the detection and alignment stage on a natural image containing a human face.

The first step in this process is illustrated in Figure 3(c) where the MAP estimate of the position and scale of the face are indicated by the cross-hairs and bounding box. Once these regions have been identified, the estimated scale and position are used to normalize for translation and scale, yielding a standard "head-in-the-box" format image (Figure 3(d)). A second feature detection stage operates at this fixed scale to estimate the position of 4 facial features: the left and right eyes, the tip of the nose and the center of the mouth (Figure 3(e)). Once the facial features have been detected, the face image is warped to align the geometry and shape of the face with that of a canonical model. Then the facial region is extracted (by applying a fixed mask) and subsequently normalized for contrast. This geometrically aligned and normalized image (shown in Figure 5(a)) is then projected onto the set of eigenfaces shown in Figure 4(a).

The projection coefficients obtained by comparison of the normalized face and the eigenfaces form a feature vector which accurately describes the appearance of the face. This feature vector can therefore be used for facial recognition, as well as for facial image coding. Figure 4(b) shows a typical result when using the eigenface feature vector for face recognition. The image in the upper left is the one to be recognized and the remainder are the most similar faces in the database (ranked by facial similarity, left to right, top to bottom). The top three matches in this case are images of the same person taken a month apart and at different scales. The recognition accuracy of this system (defined as the percent correct rank-one matches) is 99% [9].

Figure 5(a) shows the normalized facial image extracted from Figure 3, its reconstruction using a 100-dimensional eigenspace representation (requiring only 85 bytes to encode) and, for comparison, a JPEG compressed image (encoded using 530 bytes).

5 Expression Recognition

Once we have accurately located the face and its features, we can now begin to analyze its motion to determine the

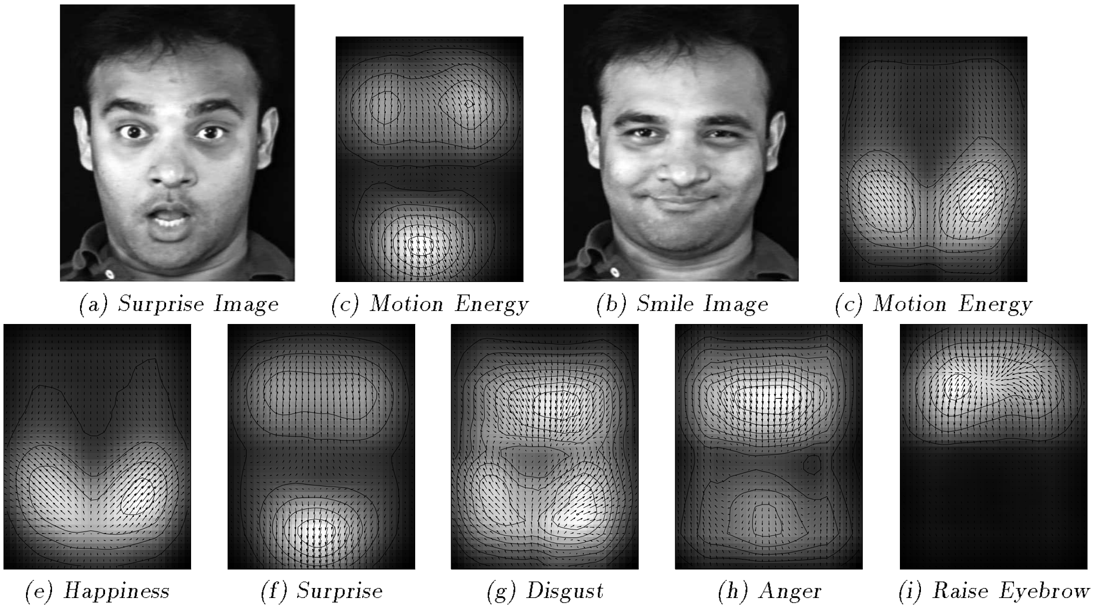


Figure 6: Determining expressions from video sequences. (a) and (c) show expressions of smile and surprise, and (b) and (d) show the corresponding spatio-temporal motion energy pattern. (e) - (i) show spatio-temporal motion-energy templates for various expressions.

facial expression. The method used in conjunction with the IVE project, developed by Essa and Pentland [6, 7], is to generate spatio-temporal motion-energy templates of the whole face for each different expression. For each facial expression, the corresponding template expresses the *peak amount* and *direction* of motion that we expect to see at each point on the face. Figures 6(a) - (d) show Irfan Essa making two expressions (surprise and smile) and the corresponding motion-energy template descriptions.

These simple, biologically-plausible motion energy "templates" can be used for expression recognition by comparing the motion-energy observed for a particular face to the "average" template for each expression. Using this simple method, we have been able to demonstrate substantially greater accuracy at expression recognition than has been previously achieved [7].

Figures 6(e) - (i) show the motion-energy templates for several expressions. Each of these templates is unique and therefore can serve as an sufficient feature for categorization of facial expression. To classify an expression we compare the observed facial motion energy with each of these motion energy templates, and then pick the expression with the most similar pattern of motion.

We have tested this expression recognition method for accuracy by applying it to a database of 52 image sequences of 8 subjects making various expressions. In each image sequence the motion energy was measured, compared to each of the templates, and the expression classified. This experiment generated the confusion matrix shown in Table 1. This table shows just one incorrect classification, giving an overall recognition rate of 98.0%. For additional details see [6, 7].

6 Vision-Driven Audio

Audio interpretation of people is as important as visual interpretation. Although much work as been done on speech understanding, virtually all of this work assumes a closely-placed microphone for input and a fixed listening position or headphones for output. This makes audio difficult to integrate into normal human life, and so we have developed solutions that free the user from these restrictions. In this section we explain the idea of vision-directed audio; for additional detail about the IVE audio input and output systems, see references [2, 3].

6.1 Audio Input

Speech recognition applications typically require near-field, i.e. $< 1.5m$, microphone placement for acceptable performance. Beyond this distance the signal-to-noise ratio of the incoming speech affects the performace significantly. Commercial speech-recognition packages typically break down over a 4 to 6 dB range. However the IVE paradigm requires the user to be free of the constraints of near-field microphone placement and the user must be able to move around the active zone of the space with no noticable degradation in performance.

There are several potential solutions to this problem. One is to have a highly directional microphone that can be panned using a motorized control unit, to track the user's location. This requires a significant amount of mounting and control hardware, and is limited by the speed and accuracy of the drive motors. In addition, it can only track one user at a time.

It is therefore preferable to have a directional response that can be steered electronically. This can be done with the well-known technique of beamforming with an array of microphone elements, as illustrated in Figure 7(a). Though several microphones need to be used for this method, they need not be very directional and they can be permanently mounted in the environment. In addition, the signals from the microphones in the array can be combined in as many

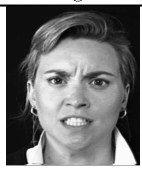
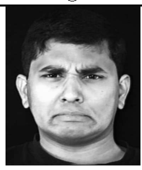
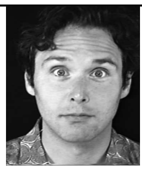
Expressions	Smile	Surprise	Anger	Disgust	Raise Brow
Template					
Smile	**12**	0	0	0	0
Surprise	0	**10**	0	0	0
Anger	0	0	**9**	0	0
Disgust	0	0	1	**10**	0
Raise Brow	0	0	0	0	**8**
Success	100%	100%	90%	100%	100%

Table 1: *Results of facial expression recognition using spatio-temporal motion energy templates. This result is on based on 12 image sequences of smile, 10 image sequences of surprise, anger, disgust, and raise eyebrow. Success rate for each expression is shown in the bottom row. The overall recognition rate is 98.0%.*

(a)

(b)

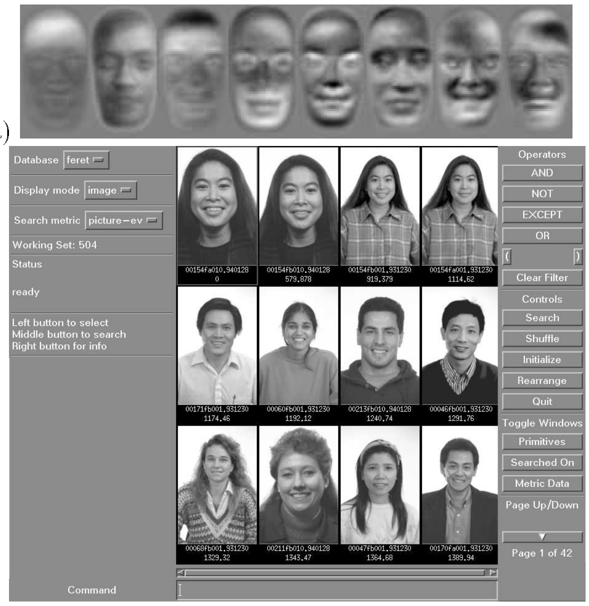


Figure 4: (a) The first 8 eigenfaces, (b) Searching for similar faces in a database, using the Photobook image database tool [10].

ways as the available computational power is capable of, allowing for the tracking of multiple moving sound sources from a single microphone array.

6.2 Audio Output

The audio output system can position sounds at arbitrary azimuths and elevations around a listener's head. The system uses stereo loudspeakers arranged conventionally (at ± 30 degrees with respect to the listener). The system works by reconstructing the acoustic pressures at the listener's ears that would occur with a free-field sound source at the desired location. This is accomplished by combining a *binaural spatializer* with a *transaural audio system*, as shown in Figure

(a) (b) (c)

Figure 5: (a) aligned face, (b) eigenspace reconstruction (85 bytes) (c) JPEG reconstruction (530 bytes).

7(b).

The spatializer convolves the source sound with the direction dependent filters that simulate the transmission of sound from free-field to the two ears. The resulting binaural output of the spatializer is suitable for listening over headphones. In order to present the audio over loudspeakers, the output of the spatializer is fed to a transaural audio system which delivers binaural signals to the ears using stereo speakers. The transaural system filters the binaural signals so that the crosstalk leakage from each speaker to the opposite ear is canceled.

7 Action Recognition

Work on recognizing face, sound, and gesture is only a first step toward machine understanding of human action. To make computers really useful, these basic perceptual functions need to be integrated with higher-level models of human behavior, so that we can begin to understand what the person is *doing*.

The technical difficulty, of course, is that we generally do not understand humans well enough to model their behavior accurately. This difficulty has forced most attempts at human-machine interface to adopt the simplest possible model of the human, such as a random number generator or a temporal filter. Such simple models, however, make it impossible to build a system that takes real advantage of the

(a) (b)

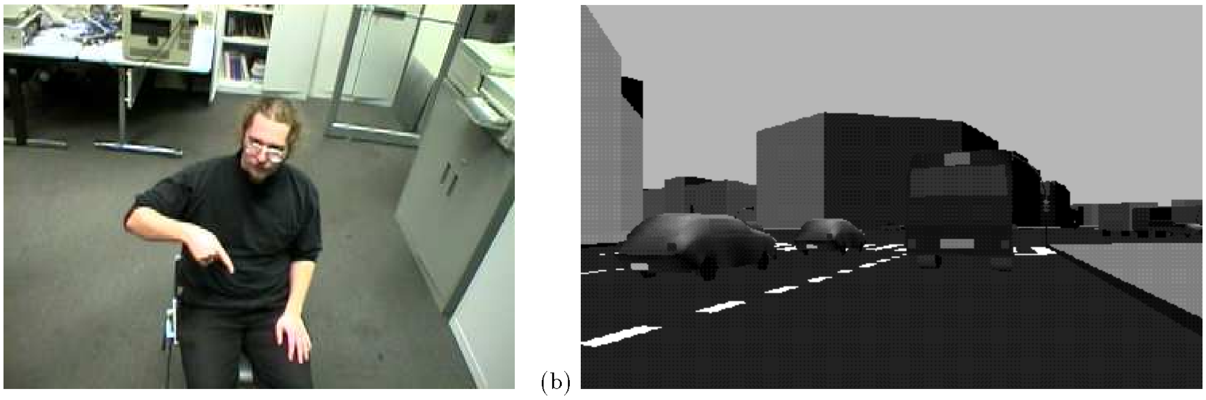


Figure 8: (a) Real-time reading of American Sign Language (with Thad Starner doing the signing), and (b) real-time classification of driver's actions in a driving simulator.

(a) (b)

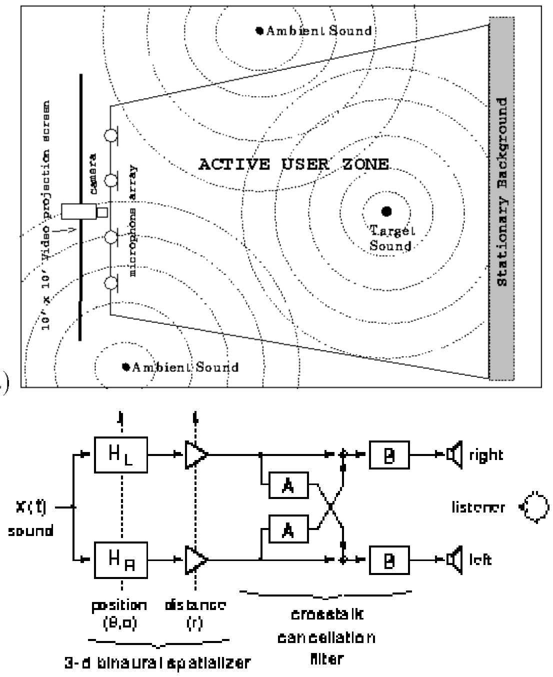


Figure 7: (a) Geometry of the vision-directed phased-array microphone system, (b) Block diagram of the 3-D audio output system.

human's abilities.

The general approach we have developed is to instead model the human as a Markov device with a (possibly large) number of internal mental states, each with its own particular behavior, and inter-state transition probabilities. However the internal states of the human are not directly observable, so they must be determined through an indirect estimation process, using the person's movement and vocalizations as measurements. One efficient and robust method of accomplishing this is to use the expectation-maximization methods developed for use with Hidden Markov Models (HMM).

This general approach is similar to that taken by the speech recognition community. The difference is that we do not think of the person's internal state as being just words or sentences; the internal states can also be actions or intentions. Moreover, we do not think of the input as being only audio filter banks; we have also used hand movement, leg movement, and pitch as inputs to infer the user's internal state.

To date we have used this approach to build two systems for interpreting human action. The first system, developed by Starner and Pentland [14], reads American Sign Language (ASL). This real-time system has allowed us to perform near-perfect classification of a forty-word subset of ASL using only the hand measurements provided by Pfinder (e.g., hand position, orientation, and width/height ratio). Thad Starner is shown using this system in Figure 8(a). The accurate classification performance of this system is particularly impressive because in ASL the hand movements are rapid and continuous, and exhibit large coarticulation effect.

The second system, developed by Pentland and Liu [11], interpreted people's actions while driving a car. In this system the driver's hand and leg motions were observed while driving in the Nissan Cambridge Basic Research Lab's driving simulator (see Figure 8(b)). These observations used to classify the driver's action as quickly as was possible. The goal is to develop safer cars by having the automobile "observe" the driver, continuously estimate the driver's internal state (what action they are taking), and respond appropriately.

We found that the system was surprisingly accurate at identifying which action the driver was beginning to execute (e.g., passing, turning, stopping, car following, lane change, or speeding up). Accuracy was 86% within 0.5 seconds of the beginning an action, rising to 97% accuracy within two seconds. This sort of accuracy at determining the driver's intention should allow us to more accurately and quickly adapt the car's performance to suit the situation.

8 Applications

Although interesting by itself, the full implications of real-time interpretation of human action only become concrete when the information is used to create an interactive application. We have explored several different human interface applications.

8.1 Gesture Control

In many applications it is desirable to have an interface that is controlled by gesture rather than by a keyboard or mouse. One such application is the Artificial Life IVE (ALIVE) system, a joint research project with Professor Pattie Maes and her students [8]. ALIVE utilizes Pfinder's support map polygon to define alpha values for video compositing (placing the user in a scene with some artificial life forms in real-time). Pfinder's gesture tags and feature positions are used by the artificial life forms to make decisions about how to interact with the user, as illustrated in Figure 9(a) [4].

Pfinder's output can also be used in a much simpler and direct manner. The position of the user and the configuration of the user's appendages can be mapped into a control space, and sounds made by the user are used to change the operating mode. This allows the user to control an application with their body directly. This interface has been used to navigate a 3-D virtual game environment as in SURVIVE (Simulated Urban Recreational Violence IVE) [12] (illustrated in Figure 9(b)), and an information landscape / virtual museum [13].

8.2 Avatars and Telepresence

Using Pfinder's estimates of the user's head, hands, and feet position it is possible to create convincing shared virtual spaces. The ALIVE system, for instance, places the user at a particular place in the virtual room populated by virtual occupants by compositing real-time 3-D computer graphics with live video. To make a convincing 3-D world, the video must be placed correctly in the 3-D environment, that is, video of the person must be able to occlude, or be occluded by, the graphics. The method of accomplishing this sophisticated compositing operation is described in Darrell *et al* [4].

If you share Pfinder's information about the user between geographically separate locations it is possible to create convincing telepresence without shipping video to the remote site, thus providing very low-bandwidth coding of human action, as in Darrell *et al* [5]. On the remote end, information about the user's head, hand, and feet position is used to drive a video avatar that represents the user in the scene. One such avatar is illustrated in Figure 10. It is important to note that the avatars need not be an accurate representation of the user, or be human at all.

References

[1] Azarbayejani, A., and Pentland, A. (1995) "Camera self-calibration from one point correspondence," Technical Report 341, MIT Media Lab Vision and Modeling Group, 1995.

[2] Azarbayejani, A., and Pentland, A. (1995) "Recursive estimation of motion, structure, and focal length," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(6), June 1995.

[3] Casey, M.A., Gardner, W.G., and Basu, S., (1995) "Vision Steered Beam-Forming and Transaural Rendering for the Artificial Life Interactive Video Environment (ALIVE)," *99th Convention of the AES*, October 6-9 1995, New York.

[4] Darrell, T., Maes, P., Blumberg, B., and Pentland, A., (1994) "A Novel Environment for Situated Vision and Behavior," *IEEE Workshop on Visual Behaviors* pp. 68-72, Seattle, WA., June 19, 1994.

[5] Darrell, T., Blumberg, B., Daniel, S., Rhodes, B., Maes, P., and Pentland, A. (1995) "Alive: Dreams and illusions," *Visual Proceedings, ACM Siggraph*, July 1995.

[6] Essa, I., Pentland, A., (1994) A Vision System for Observing and Extracting Facial Action Parameters, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 76-83, Seattle, WA., June 1994.

[7] Essa, I., and Pentland, A. (1995) "Facial Expression Recognition Using a Dynamic Model and Motion Energy," *Int'l Conference on Computer Vision*, Cambridge, MA, June 20-23 1995.

[8] Maes, P., Blumburg, B., Darrell, T., and Pentland, A., (1995) "The ALIVE System: Full-body Interaction with Autonomous Agents." Proceedings of Computer Animation 95, IEEE Press, April 1995.

[9] Moghaddam, B., and Pentland, A., (1995) "Probabalistic Visual Learning for Object Detection," *Int'l Conference on Computer Vision*, Cambridge, MA, June 20-23 1995.

[10] Pentland A., Picard, R., Sclaroff, S., (1994) "Photobook: Tools for Content-Based Manipulation of Image Databases," *SPIE Conf. on Storage and Retreval of Image Databases II*, San Jose, CA, Feb 6-10, 1994

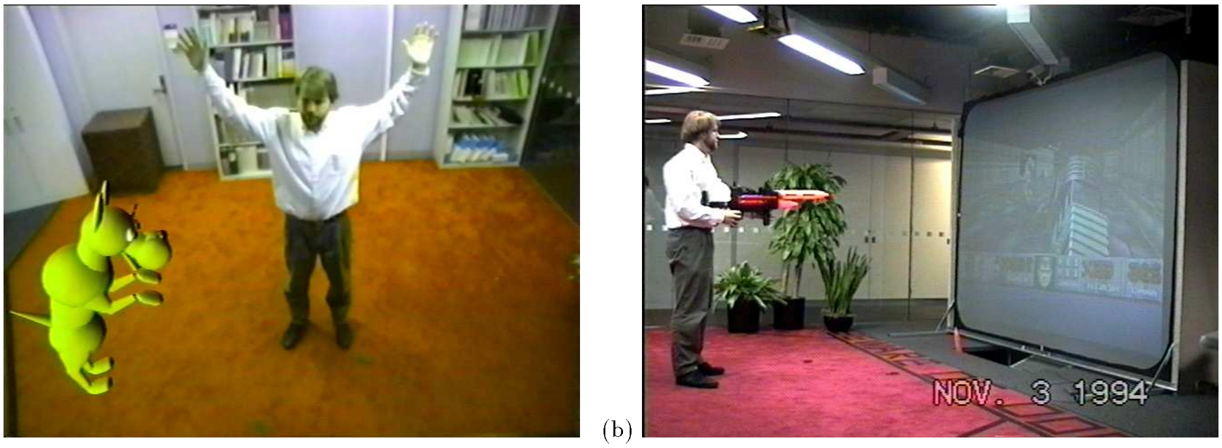
[11] Pentland, A., and Liu, A., (1995) Toward Augmented Control Systems, *IEEE Intelligent Vehicle Symposium 95*, September 25-26, Detroit, MI.

[12] Russell, K., Starner, T., and Pentland, A. (1995) "Unencumbered virtual environments," *IJCAI-95 Workshop on Entertainment and AI/Alife*.

[13] Sparacino, F., Wren, C., Pentland, A., and Davenport, G., (1995) "Hyperplex: a world of 3d interactive digital movies," *IJCAI-95 Workshop on Entertainment and AI/Alife*.

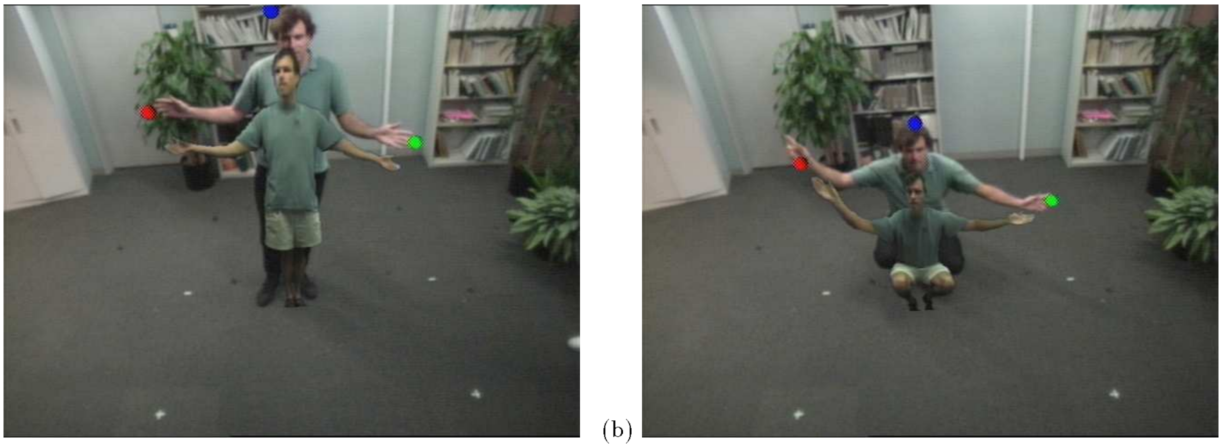
[14] Starner, T., and Pentland, A., Visual Recognition of American Sign Language Using Hidden Markov Models, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition 1995*, Zurich, Switzerland, June 26-28, 1995.

[15] Wren, C., Azarbayejani, A., Darrell, T.,and Pentland, A., (1995) "Pfinder: Real-Time Tracking of the Human Body," *SPIE Conference on Real-Time Image Processing*, Philadelphia, PA., Oct 27, 1995



(a) (b)

Figure 9: (a) Chris Wren playing with Bruce Blumberg's virtual dog in the ALIVE space, and (b) playing SURVIVE.



(a) (b)

Figure 10: Trevor Darrell controlling an avatar of himself.