M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 416

Perceptually-driven Avatars and Interfaces: active methods for direct control

T. Darrell(*), S. Basu(+), C. Wren(+), A. Pentland(+)
MIT Media Lab(+)
Interval Research Corp.(*)
contact:trevor@interval.com

January 14, 1997

Abstract

We show how machine perception techniques can allow people to use their own bodies to control complex virtual representations in computer graphic worlds. In contrast to existing solutions for motion capture, tracking people for virtual avatars or intelligent interfaces requires processing at multiple levels of resolution. We apply active perception techniques and use visual attention to track a user's pose or gesture at several scales simultaneously. We also develop an active speech interface that leverages this visual tracking ability; by electronically focusing a microphone array towards a particular user, speech recognition in acoustically cluttered environments is possible. Together, these methods allow virtual representations of people to be based on their actual expression, tracking their body and face gestures and speech utterances as they freely move about a room without attached wires, microphones, or other sensors.

1 Introduction

A major attraction of the internet and virtual environments is their interactive nature, allowing users to gather information without time or channel constraints and easily integrate their own content into the medium. Interactive media offer a balance between the traditional roles of information producer and consumer, allowing individual users to perform both tasks with standard computer equipment. To a large degree these goals have been achieved through the remarkable proliferation of the HTML and VRML protocols and their associated server and browser systems.

But viewed from other perspectives there is still nearly complete asymmetry in the interface between a user and digital information sources. While rich multimedia and computer graphic content from around the globe present themselves on contemporary PC screens, the relative bandwidth of information a user can generate in real-time is but a trickle: a few keystrokes per second, the occasional mouse movement or button click. When virtual environments were largely text-based chat systems, individual users had the ability to generate roughly as much bandwidth in the interface as they received. Today, this balance is not preserved.

With graphical and multimedia worlds, being a participant in a virtual sense will require much more information than a keyboard and mouse can generate, since an effective real-time presence will have many degrees of freedom which need simultaneous control. Body pose, gesture, facial expression and speech are important aspects of interpersonal communication in the real world, and are becoming equally prevalent in virtual worlds as video and human form rendering become common.

To overcome the one-way nature of image and sound use in current computer interfaces, we have been investigating the use of machine perception techniques to allow a user to directly control aspects of a computer interface with his or her body, face, and speech output, in real-time. These inputs can either drive a literal representation of the user in a virtual environment, a more comical or fantastic avatar representation, or be used for an abstract interface to a database or other digital media. Our goal is to create interfaces that are no longer deaf and blind to their users, interfaces that balance the flow of images and sound between the user and the virtual world.

2 Perception-driven user models

The idea of using machine vision to track people has been previously developed in the computer graphics literature for interactive games and experiences [20, 35, 12] and for the problem of motion or expression capture [1, 4, 15, 36, 34]. These techniques can capture a single, specific motion performed by a human user for later use in creating a special effect or animating a virtual character. However, they have several major limitations which prevent their use as a direct interface tool: they often require markers or special paint, they are often not real-time, and they are designed to capture information at only a single spatial resolution. The latter is the most fundamental problem: people express themselves at spatial scales from coarse body pose to the blink of an eye, and to capture this expression multi-resolution analysis is needed.

With conventional cameras, a single fixed viewpoint will not suffice to both track a users body as they walk about a room and to track their eye or detailed facial expression. For off-line animation or effects, this is not a problem since different captured motions (at different resolutions) can be hand-edited

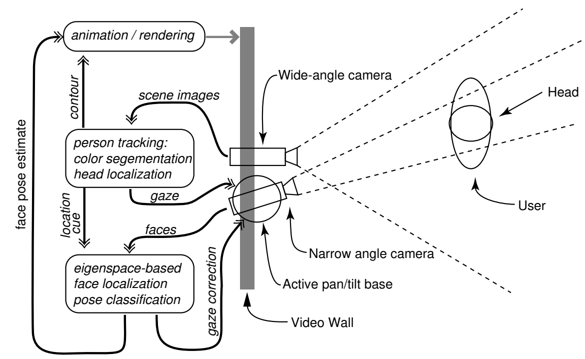


Figure 1: Overview of system for visual face/body tracking and pose estimation. Objects are rendered on a Video Wall and react to the facial pose of the user. A static, wide-field-of-view, camera tracks the user's head, and drives gaze control for an active, narrow-field-of-view camera. Eigenspace-based pose estimation is run on face images from the active camera, provides pose estimates for objects/agents to react to, and also provides closed-loop tracking feedback.

together. As an interface or avatar, however, we need to simultaneously capture in real-time these multiple scales of resolution at which people perform expressions.

This paper thus proposes active, multi-resolution tracking methods for intelligent interfaces or avatars. This is an extension of the notion of motion capture but with explicit emphasis on noninvasive and real-time processing, which leads to the requirement that multiple scales of motion be potentially tracked at the same time. Our approach is based on methods from the active perception literature; we construct an interface which has explicit modes of attention and can simultaneously attend to both coarse and fine-scale gesture performance. Further, we generalize to include audio processing and show how an active multimodal perception method can remove much of the ambiguity present in acoustic-only tracking.

The visual component of our interface implements overt attention via dual cameras, combining a fixed-view, wide-angle camera that observes an entire room with a motorized narrow field-of-view camera that can selectively obtain images of parts of a users body at high-resolution. Figure 1 presents a diagram of our visual tracking system; on the imagery from the wide-angle camera we run a tracking method that uses multi-class color segmentation to follow a users body pose, 3-D position, and the location of head and hand features, as described in the following section. This is sufficient for many applications which just need the rough 3-D position and outline of the user, but not for applications which need gestural expression from the user (e.g., detailed face or hand poses); for this we need high-resolution observations, and use the active camera.

On the active camera imagery we apply methods that are able to resolve finer details; we use parametric probability models of the appearance of images in particular expression classes, and interpolation methods to map observed class similarities to control tasks. Increased accuracy in expression modeling and direct task control comes at the expense of having to train the high-resolution portion of the system for particular users and tasks. Our systems for pose and gaze tracking assume prior training, in which examples of face and eye appearance under varying pose are initially provided.

We present these methods in detail in the following sections. Section 3 will present methods for coarse-scale tracking of users, and Section 4 will introduce active visual tracking and analysis with examples of simultaneously tracking facial and body pose with the dual camera system. Section 5 will discuss active acoustic sensing in tandem with visual person tracking, and introduce a beam-forming algorithm that focuses acoustic attention to the location of a particular users head, allowing speech recognition in an open environment that can have multiple sound sources. Finally Section 6 will present particular applications we have explored with our system, and Section 7 will offer conclusions and directions for future work.

3 Person Tracking

A multi-class color segmentation method drives the basic coarse-view person tracking routine used in our system. From a single static view, it provides estimates of the 3-D position and body pose of the user. This routine runs continuously, and provides the cues needed by the active visual and acoustic processing methods described in the following sections.

Color segmentation is but one method that can be used to track people; range estimation, motion-based segmentation, and thermal imaging are other methods that have been explored in the computer vision literature [28, 29, 25, 17, 18]. We choose to use color segmentation because accurate estimates can be obtained in real-time using an implementation with no special hardware other than a video camera and pentium-class PC processor. Color-space methods also make it relatively easy to locate head and hand features, as described below. The major restriction of our color segmentation method is the requirement of a static scene view, i.e., no moving background. We apply our method in room environments, where this is a reasonable assumption.[1]

[1] Integration of real-time motion or stereo processing into our method would allow operation in open-background environments; as computational resources become economical for these methods this will become a practical option.

Figure 2: Coarse-scale person tracking in a system for vision-based interaction with a virtual environment. (a) A user sees himself in a "magic mirror", composited in a virtual environment. Computer vision routines analyze the image of the person to allow him to effect the virtual world through direct manipulation and/or gestural commands. (b) Image presented on video wall. With this system, user can interact with virtual agents, such as the dog character in our ALIVE system [12, 6] (c) Results of feature tracking routine; head, hands, and feet are marked with color-coded balls.

Our experimental interaction domain is a "smart" room or office with an interactive graphics display. The user faces a wall-size screen which contains cameras and other sensors that observe the user. Computer-generated graphics and video images are presented on the display, along with a graphical representation of the user. The cameras are connected to a vision system which analyze in real time the state and location of the user, and update the virtual representation accordingly. This representation may consist of the user's digitized image, a 3-D model of a person, a model of a graphical cartoon character, or a combination of all of these. Objects (or agents) in the virtual world can use the vision system to "see" the user, who can in turn see the graphical representation of the objects on the display.

We have developed a set of vision routines for perceiving body pose and coarse gestures performed by a human participant in this interactive system. Vision routines acquire the image of the user, compute a figure/ground segmentation, and find the location of head, hands, and other salient body features (Figure 2). Additionally, the extracted contour can then used to perform video compositing and depth clipping to combine the user's video image with computer graphics imagery. The integration of the person and and localization of his/her head or hand features in the world are performed using the following modules: figure-ground processing, scene projection, hand tracking, and gesture interpretation.

3.1 Figure-ground processing

Figure-ground processing is accomplished by use of spatially-local pattern recognition techniques to characterize changes in the scene, followed by connected-components and morphological analysis to extract objects. We assume the background to be an arbitrary, but static, pattern. Mean and variance information about the background pattern are computed from an initial sequence of images with no person present, and these statistics are used to determine space-variant criteria for pixel class membership. We use a multi-class color classification test to compute figure/ground segmentation, using a single Gaussian model of background pixel color and an adaptive mixture model of foreground (person) colors. The color classification takes care to identify possible shadow regions, and to normalize these region's brightness before the figure/ground classification [37].

Once the set of pixels most likely belonging to the user has been found, we use connected components and morphological analysis to delineate the foreground region. This analysis begins with a seed point at the centroid location of the person in the previous frame; if this fails to grow a sufficiently large region, random seed points are selected until a stable region is found. Finally, we compute the contour of the extracted region by chain-coding the connected foreground region.

3.2 Scene projection and calibration

When the figure of the user has been isolated from the background, we compute an estimate of its 3-D location in the world. If we assume the user is indeed sitting or standing on the ground plane, and we know the calibration of the camera, then we can compute the location of the bounding box in 3-D. Establishing the calibration of a camera is a well-studied problem, and several classical techniques are available to solve it in certain broad cases [2, 16]. Typically these methods model the camera optics as a pinhole perspective optical system, and establish its parameters by matching known 3-D points with their 2-D projection.

Knowledge of the camera geometry allows us to project a ray from the camera through the 2-D projection of the bottom of the bounding box of the user. Since the user is on the ground plane, the intersection of the projected ray and the ground plane will establish the 3-D location of the user's feet. The 2-D dimensions of the user's bounding box and its base location in 3-D constitute the low-level information about the user that is continuously computed and made available to all agents in the computer graphics world. The contour is projected from 2-D screen coordinates into 3-D world coordinates, based on the computed depth location of the person. This can then used to perform video compositing and depth clipping to combine the user's video image with computer graphics imagery.

3.3 Feature and Gesture detection

Hand and head locations are found via contour analysis and by tracking flesh colored regions in the person. To initialize mixture model elements, Pfinder uses a 2D contour shape analysis that attempts to identify the head, hands, and feet locations when they are extended from the body. When this contour analysis does identify one of these locations, then a new mixture component is created and placed at that location. Likely hand and face locations are set to strong flesh-colored color priors; others are initialized to cover clothing regions. A competitive method reallocates support in the foreground image among the mixture model elements, creating or deleting new elements as needed [37]. With this framework hands can be tracked in front of the body; when one reappears after being occluded or shadowed only a few frames of video are needed to regain tracking.

Both the absolute position of hands, and whether they are performing characteristic gesture patterns, are relevant to the agents in the virtual world. The detection of coarse static gestures, such as pointing, are computed directly from the hand feature location. To recognize dynamic gestures, we use a high-resolution, active camera to provide a foveated image of the hands (or face) of the user. The camera is guided by the computed coarse feature location as well as closed-loop tracking feedback, and provides images which can be used successfully in a spatio-temporal gesture recognition method. This framework for real-time gesture processing and active camera control is described in detail in the following chapters.

4 Active Gesture Analysis

An effective gesture interface needs to track regions of a person's body at a finer resolution than is possible in a single fixed camera view of the person.[2] Detailed aspects of a person's hand or face are difficult if not impossible to track at low resolution, yet these play an important role in how people communicate – expressions, hand gestures, direction of gaze, etc. To construct a system which can recover both the body pose and position information revealed by the method in the previous section, and detailed expression and gesture information present only in high-resolution views of parts of a user, we need a system for visual attention and foveated observation (i.e., observation with a fovea: non-uniform and active sampling).

(a)

(b)

Figure 3: Images acquired from wide (a) and narrow (b) field of view cameras as user moved across room and narrow camera tracks head.

4.1 Active camera system

Many possible architectures exist for implementing active visual attention. It is theoretically possible to implement attention without actuators, by shifting a window of interest in an extremely high-resolution image sensor. While such sensors are not currently commercially available, this architecture may be viable in the future as the cost of the imaging surface is typically small compared with the cost of I/O and processing. An architecture which models the human eye, with logarithmically increasing sampling density towards a central locus on the imaging surface, is aesthetically desirable and has many benefits from an image coding standpoint. Unfortunately the construction of such a sensor is still a topic of active research, and the image processing algorithms designed for uniformly sampled grids are not directly transferable to images obtained from this type of sensor.[3] Our approach is to use a dual camera system, with two uniformly sampled imaging sensors; one static with fixed wide field-of-view, and one with active pan/tilt gaze control and narrow field-of-view.[4]

As described above, wide field-of-view tracking can locate the head of a user in the scene, and return both the 2-D image coordinates of the head, and the inferred 3-D world coordinates based on the camera geometry and the assumption that the user stands erect on the ground plane. We use the estimated head location to obtain a high resolution image of the selected feature, using

[2] Assuming of course that the camera needs to observe the entire body, as it does in our applications, and that the resolution is that of conventional cameras, e.g., roughly one megapixel or less distributed uniformly over the imaging surface.

[3] But we should note that the log-polar representation does have several desirable processing properties that Cartesian representations do not have, such as scale invariance.

[4] The most recent version of our system utilized a camera which also has active zoom control, but this camera was not used in the examples shown in this paper.

the second, active camera. Since our active camera is mounted some distance from the wide angle camera, (approx 6 ft.) we derive the active camera gaze angle with simple trigonometry using the known active camera base location.[5] Reliable head tracking results were obtained using this method; figure 3 shows pairs of output from the wide and narrow cameras in our active-vision system as the user walks across the room and has his head tracked by the narrow field-of-view camera. The narrow field-of-view camera is able to provide a high-resolution image of the users face suitable for pose estimation using an eigenspace method, as described below.

4.2 Appearance-based gesture analysis

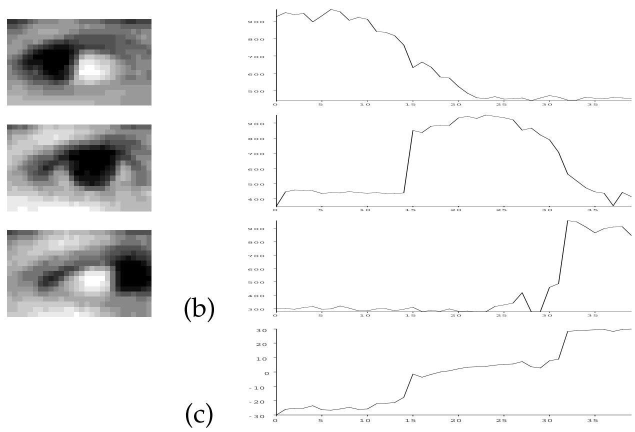
Hand and face gesture analysis has been a topic of increasing interest in the computer vision literature.[6] For tracking hand and face gesture from high-resolution data, we have adopted an appearance based representation, in which the appearance of a target object is described by its similarity to a set of iconic views that span the set of poses and configurations of the target object.

This approach is related to the idea of view-based representation, as advocated by Ullman [38] and Poggio [27], for representing 3-D objects by interpolating between a small set of 2-D views. Appearance or view-based representation achieves robust real-time performance in gesture analysis by exploiting the principle of using only as much "representation" as needed. Hands and faces are complex 3D articulated structures, whose kinematics and dynamics are difficult to model with full realism. Consequently, instead of performing model-based reconstruction and attempting to extract explicit 3D model parameters, ours is a direct approach which represents the object performing the gesture with a vector of similarity scores to a set of 2-D views. With this approach we can perform recognition and tracking on objects that are either too difficult to model explicitly or for which a model recovery method is not feasible in real time.

In general we form the set of iconic views from examples of different expressions or gestures. In the former case, we take the set of example images for a particular expression or pose class and form an model of the class probability density function (e.g., a function which returns the probability that a new image is a member of the class) using an eigenspace technique (see [11] and [23]). Our similarity function is the likelihood that a new image is the member of the given class. When provided with a sequences or unlabeled gesture examples, we first run an unsupervised clustering method to group the im-

[5] If the optical center of the active camera can be mounted close to the optical center of the fixed camera, then one could simply scale the 2-D image location of the head in the fixed view to compute a pan and tilt angle for the active camera.

[6] There are too many references to explicitly list; see the proceedings of the 1994 and 1996 workshops on Automatic Face and Gesture Recognition for a cross-section of the field. A good survey paper of work in the field remains to be written.



(a) (b) (c)

Figure 4: Example of appearance-based pose analysis and interpolation when tracking a eyeball tracking from approximately -30 to $+30$ degrees of gaze angle with two reported saccades. (a) Three spatial views of an eyeball at $+30$, 0, and -30 of gaze angle. (b) Similarity scores of these three view models (c) Interpolated gaze angle showing these saccades, using RBF method described in text.

ages into classes [9], and then construct the class density function. For reasons of computational efficiency, when searching for the spatial location which has maximal similarity across all models, we instead use the normalized correlation of the new image with the mean (zero-th order eigenvector) of the class as our similarity measure. Other similarity measures are certainly possible; a metric which included geometric distortion as well as intensity distortion would likely offer increased robustness (for example see [24] or [5], we are currently investigating the integration of these methods into our system.)

With a smooth similarity function the similarity score of a particular view model as the object undergoes non-linear transformations will be a roughly convex function. The peak of the function will be centered at the parameter values corresponding to the pose of the object used to create the view model. For example, Figure 4(a) shows three images of an eyeball that were used to create view models for gaze tracking; one looking 30 degrees left, one looking center-on, and one looking 30 degrees to the right. Figure 4(b) shows the similarity score for each view model when tracking a eyeball rotating from left to right, with two saccades. Each view model shows a roughly convex curve centered about the gaze angle used to create the view model.

We use a task-dependent interpolation method to map from the set of view model scores to a result vector used for recognition or control. Interpolation is done using a supervised learning paradigm, based on a set of training examples which define the desired result for a particular set of view model outputs.

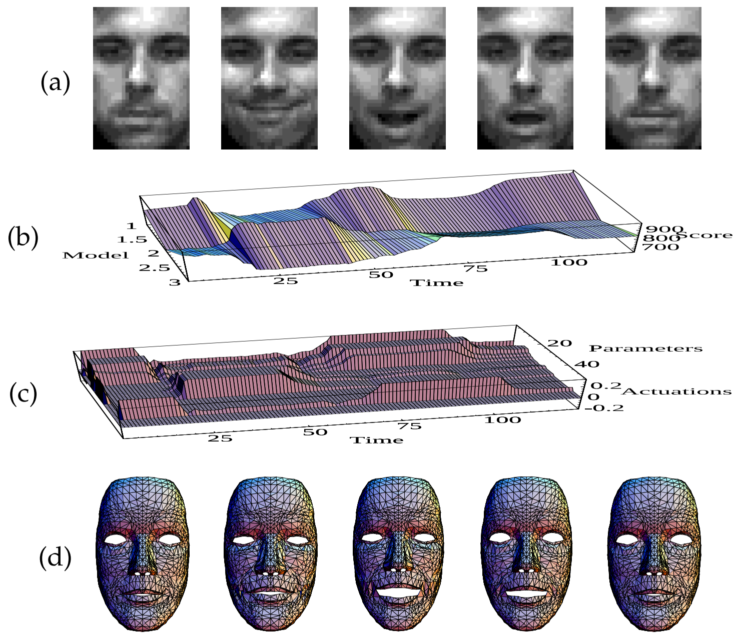


Figure 5: Interactive facial expression tracking using appearance-based analysis/interpolation with 3-D facial model as output. A set of view models are used to characterize facial state, and then used to interpolate a set of motor control parameters for a physically-based face model. View models are acquired using unsupervised clustering while the interpolation is trained using supervised learning.

Using the Radial Basis Function (RBF) method presented in [26], we compute a result vector $\hat{\mathbf{y}}$ to be a weighted sum of radial functions centered at an exemplar value:

$$\hat{\mathbf{y}}(\mathbf{g}) = \sum_{i=1}^{n} c_i \mathcal{F}(||\mathbf{g} - \mathbf{g}^{(\mathrm{i})}||) \quad , \tag{1}$$

where

$$\mathbf{c} = \mathbf{F}^{-1}\mathbf{y}, \quad (\mathbf{F})_{ij} = \mathcal{F}(||\mathbf{g}^{(\mathrm{i})} - \mathbf{g}^{(\mathrm{j})}||), \quad \mathbf{y} = [\mathbf{y}^{(1)}, ..., \mathbf{y}^{(\mathrm{n})}]^{\mathbf{T}} \quad , \tag{2}$$

$\mathbf{g}$ are the computed spatio-temporal view-model scores, and $\{(\mathbf{y}^{(\mathrm{i})}, \mathbf{g}^{(\mathrm{i})})\}$ are a set of exemplar result and view-model score pairs (which may be scalar or vector valued). $\mathcal{F}$ is the RBF, which in our implementation was $\mathcal{F}(\S) = \S$.

We use the interpolation stage to map the observed view model scores into a quantity which is directly useful for a particular task. For example, if we wanted to estimate the eye gaze angle for the example in Figure 4, we could

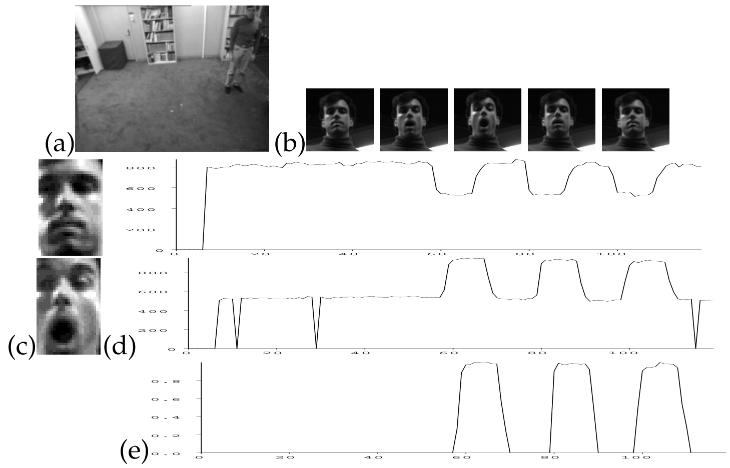


Figure 6: View-based expression tracking using foveated face images. (a) Wide-angle view of scene with person standing in corner. (b) Foveated images of face while user performs “surprise” expression; this is subsequence of entire run, in which user repeated this expression three times. (c) Exemplars of neutral and surprise expression classes. (d) Similarity score of expression classes evaluated on full sequence. (e) Plot of surprise measure interpolated from view template scores. Three peaks are present corresponding to the three surprise expressions.

use an RBF interpolator with a one dimensional output space corresponding to gaze angle and three exemplars, containing the view model scores corresponding to each view model angle:

$$\{(y^{(i)}, \mathbf{g}^{(i)})\} = \{(-30, [1.0, 0.3, 0.3]^T), (0, [0.3, 1.0, 0.3]^T), (30, [0.3, 0.3, 1.0]^T)\}$$

Using this RBF configuration, it is straightforward to recover an estimate of the underlying eye gaze angle from the three spatial view model outputs. The interpolated gaze angle is shown in Figure 4(c).

4.3 Face gesture tracking

The interpolation paradigm is quite powerful as it allows us to map to arbitrarily complex control dimensions. For example, when tracking facial expressions, we can directly control the motor parameters of a face. Figure 5 shows an example where the user presented the system with a rigid face in known position and varied his facial expression. The top row of this figure shows a sequence of faces performing a surprise expression; the second row shows the similarity score for smile, surprise and neutral expression classes. In this system the users face was restricted to a small workspace in front of the camera;

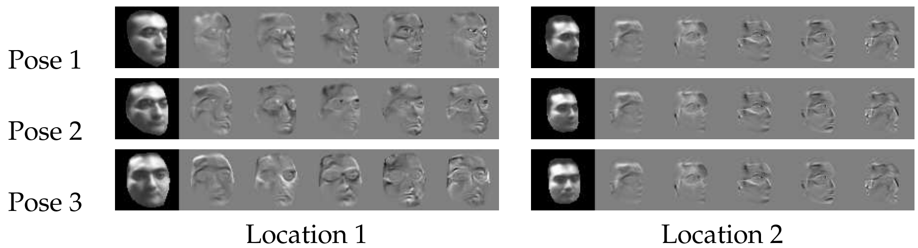


Figure 7: Multiple-pose eigenspaces used to compute pose class probability density function, shown for three different poses at two different world locations. The truncated eigenvector sequence allows rapid computation of the likelihood a new image is in a particular pose class, and thus efficient computation of our similarity score vector.

see [10] for details. The similarity scores were mapped to a computer graphics face model (described in [14]) with physiologically-valid muscle parameters corresponding to different facial muscle groups; the large number of parameters makes this a difficult model to animate in real time via conventional control. The interpolation method was trained with examples of the appearance of the users face and the synthetic muscle parameters for each expression. The bottom two rows of the figure show the interpolated motor control parameters during a real-time run of the system, and the generated synthetic faces.

With an active vision approach, we relax the restriction that the users face be rigid and in a known position to track gestures. We have used similarity-based gesture analysis on the active camera output, and can track a users expression or pose as they freely walk about a room, unencumbered, physically far (10') from the actual sensors. Figure 6 shows the results of tracking expressions using the narrow angle camera input when the user is in two different locations in the scene, using a single set of view models. In this case a scalar surprise measure was chosen as the output dimension in the interpolator; we could equivalently driven the synthetic facial model parameters as in the previous example. For interface applications a scalar output is more appropriate; for re-animation or avatar applications the full motor state is more appropriate.

In this paper we focus on face pose estimation for updating a virtual user representation, and track the direction of head gaze as a user walks about the interactive room. We then allow agents and objects in the virtual world react directly to the users direction of head gaze in addition to overall body pose. With this facility a virtual agent can know when the user is looking at them, or know where the user is looking when referencing an object.

To accomplish this task, we trained appearance classes for three different

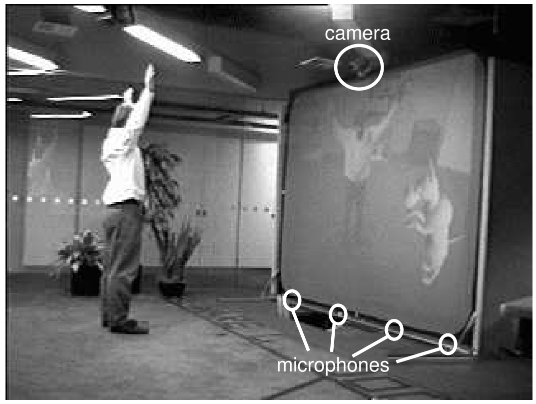


Figure 8: Location of the camera and microphone array in the virtual environment. Virtual agents such as animated dog character on video wall react to gesture and speech of user.

facial pose conditions: looking at the left, center, or right of the screen. Further, we trained pose classes separately at different physical locations in the scene. Figure 7 shows the eigenvectors used to represent the probability density function for these pose classes at three example locations. Location-dependent training is possible since we know the 3-D position of the user from the coarse tracking system, and simply gather separately the statistics for each pose class at each location. Multiple location training allows us to model variation in the observed faces due to changing scale and global illumination, which are not well modeled by a single eigenspace technique. There is considerable additional training time, but no additional run-time cost, and the system becomes much more robust to global illumination and other effects. In [11] we evaluated the recognition performance of our system using just the similarity scores, and found recognition rates in excess of 84% for the task of discriminating amonst the three head gaze poses were possible. (See video for demonstration.)

5 Active Speech Recognition

Speech is an essential element of a complete perception-based interface. Our goal is to provide a mechanism for speech recognition in unconstrained environments, so that a user need not be encumbered by a microphone or wires. Conventional speech recognition methods have great difficultly in acoustic environments with multiple speakers or other sources of noise. Even without other speakers in the room, effective speech recognition typically requires the high signal to noise ratio of a near-field (i.e., clip-on or noise-cancelling) microphone. However, we are unwilling to encumber the user with such devices,

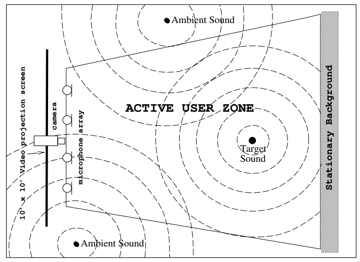


Figure 9: Target and Ambient Sound in our Virtual Environment

and thus face the problem of getting high quality audio input from a distance. Our approach is to provide a method to isolate the acoustic signal being produced by the user in an otherwise cluttered environment, by focusing acoustic attention in the same way we focus visual attention to obtain high-resolution imagery of parts of the user. This goal can be achieved with the well-known technique of beamforming with an array of microphone elements.[7] The basic principle is to combine the signals from omnidirectional or partially directional (i.e., cardioid) microphones to form a more directional response pattern. 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 ways as the available computational power is capable of, allowing for the tracking of multiple moving sound sources by a single microphone array. The setup of the array used in our implementation is shown in Figure 8 and Figure 9.

Beamforming is formulated in two flavors: fixed and adaptive. In fixed beamforming, it is assumed that the position of the sound source is both known and static. An algorithm is then constructed to combine the signals from the different microphones to maximize the response to signals coming from that position. This works quite well, assuming the sound source is actually in the assumed position. Because the goal is to have a directional response, this method is not robust to the sound source moving significantly from its assumed position. In adaptive beamforming, on the other hand, the position of the source is neither known nor static. The position of the source must continuously be estimated by analyzing correlations between adjacent microphones, and the corresponding fixed beamforming algorithm must be applied for the

[7] Another potential solution is to have a highly directional microphone that can be panned using a motorized control unit such as drives the high-resolution camera. In our implementation we found the drive motors generated undesirable acoustic interference, which would prevent tracking a moving acoustic source. With a directional response that can be steered electronically this is not a problem. In addition, with a directional microphone, only one sound source can be tracked at a time. With an electronically-steered array we can overcome both of these problems.

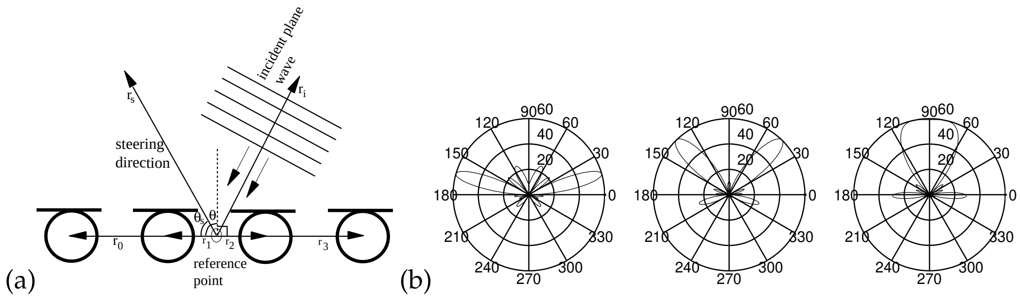


Figure 10: (a) Broadside Microphone Array Geometry and Notation (b) Directivity Pattern of Broadside Array with Cardioid Elements steered at 15, 45, and 75 degrees. Note that the reference point of the broadside array geometry should be aligned with the center of each polar plot

estimated position. This works well when the signal is known as with radar or sonar, but quite poorly when it is unknown and wideband, as is the case with speech. This also does not tend to work well when there are multiple sources of sound, since there are high correlations for multiple possible sound source positions. It is difficult and often impossible to tell which of these directions corresponds to the sound of interest, e.g., the voice of the user.

Our solution to this problem is a hybrid of these two flavors which leverages the information available in the visual domain. Instead of using the audio information to determine the location of the sound source(s) of interest, we use the vision system, which exports the 3-D position of the user's head. Using this information, we formulate the fixed beamforming algorithm for this position to combine the outputs of the microphone array. This algorithm is then updated periodically (5 Hz) with the vision information. As a result, we have the advantages of a static beamforming solution that is adaptive through the use of vision information.

Beamforming is a relatively old technique; it was developed in the 1950's for radar applications [22]. In addition, its use in microphone arrays has been widely studied [8, 19, 31, 33]. In our implementation, four cardioid microphones were placed 0.5 meters apart from one another in a broadside configuration due to space constraints 10(a). The geometry of our microphone array is represented by the set of vectors $\mathbf{r}_n$ which describe the position of each microphone n relative to some reference point (e.g., the center of the array), see Figure 10(a). The array is steered to maximize the response to plane waves coming from the direction $\mathbf{r}_s$ of frequency f_o. The polar response patterns for this arrangement are shown in Figure 10(b). A detailed examination of the response patterns with different array geometries and element responses is developed in [7].

With this beamforming implementation, we were able to sufficiently increase the signal to noise ratio to successfully feed the output into a commercial speech recognition system. This is particularly impressive in that these systems are typically designed to work only with noise-cancelling headset mics. In other words, while the systems usually require the microphones to be within inches of the user's mouth to maximize the signal to noise ratio, we were able to achieve good recognition results with the microphones at several meters away.

An application of this result was built into the ALIVE system in 1995. We combined this active acoustic sensor with the HARK recognition system from BBN [3], constructed a restricted grammar for the recognizer, and then trained a virtual dog character [6] to respond to the recognition results. At ICCV95, we demonstrated unencumbered speech recognition in an open environment using this technique. The system filtered out noise from other observers/sound sources in the room, and users were able to successfully command the dog character using both speech and gesture. In the presence of crowd noise, the recognition rate with a single microphone was approximately 30%; with the vision-steered beamforming, we obtained results in excess of 80% correct. While this is insufficient for complex sentence recognition, it is sufficient for single word interfaces.

6 Perceptive Interfaces and Avatars

Many applications are possible with a perception-based user interface. Together with other colleagues at the MIT Media Lab, we have developed systems for interacting with autonomous virtual agents [12], browsing a multimedia database using arm pointing gestures [32], playing interactive 3-D game environments using a full body interface for navigation [30], and a system for multi-user interaction in a shared virtual world [13] (Figure 11).

As first implemented these systems used only the fixed camera person tracking system and could thus only detect coarse body gestures. With active visual tracking, information about detailed hand or face expression of a user can now be used in the interface. All of these systems can benefit from hand and face gesture input; for example, in the ALIVE system virtual agents or characters can be designed to respond to different facial expression, or to respond based on whether the user is looking at the creature. We are designing characters which respond to gaze; currently we show this ability in our virtual environment with a moving block demonstration, where the block follows the direction of the users facial gaze on the screen. (See video tape for demonstration.)

These systems generally used live video as the representation of the user in the virtual environment (i.e., by compositing). However this is impractical when one wants to implement a distributed environment, where a single virtual world is shared by participants at several remote physical sites. To allow this type of interaction, we have developed a method for animating an image-based representation of the user, constructed using a real-time inter-

(a)

(b)

(c)

Figure 11: Applications of perceptual interface: (a) environment for interacting with virtual agents or animated characters, (b) browsing information databases, (c) interactive 3-D game environments.

(a) (b) (c)

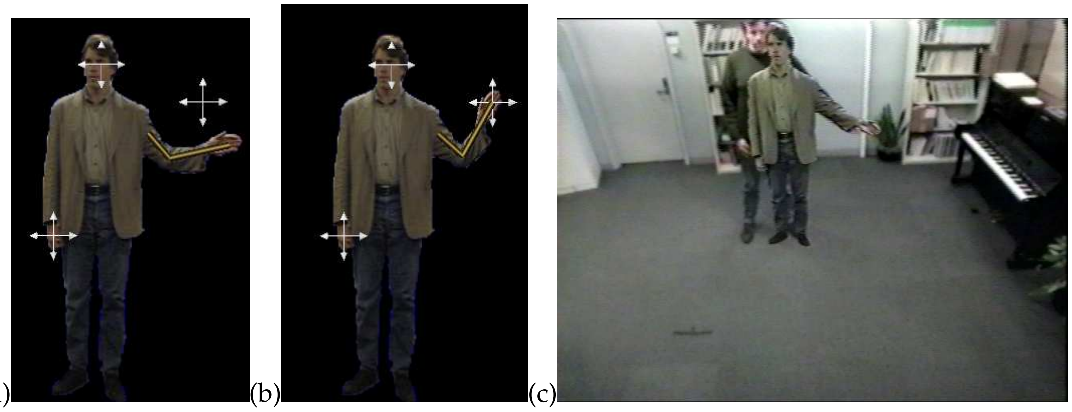


Figure 12: Avatar constructed from interpolated video keyframes. Approximately 10 annotated example images of user in fronto-parallel poses were used to render avatar. (a) Keyframe with head/hand configuration nearest to goal state indicated by crosses. (b) Result after warping arm to goal state. (c) Avatar rendered with user (slightly smaller) to show real-time tracking ability.

Figure 13: Avatar driven by multiple scales of visual tracking; subwindows show silhouette from coarse-scale figure/ground segmentation and fine scale face pose analysis on active camera imagery.

polated keyframe technique (see [21] for a related system). We collect a set of keyframes (typically ten or fewer) of the user in different poses, annotate the head, hand, and arm joint positions [8], and at run-time select the keyframe which has the closest head and hand configuration to the user's current body pose. We scale and translate this image to have the correct 3-D position and pose to mirror the user's position in the real world. Further, we warp the users arm in the nearest keyframe image to be as close as possible to the desired configuration. We model the arms with a planar two joint kinematic chain and perform the warping via texture-mapping in Open Inventor using linearly interpolated coordinates along the arm contour.[9] Figure 12 shows a keyframe used in the construction of the video avatar, the arm annotation used to construct the kinematic chain, and a warped version of the keyframe. Using this warping technique dramatically reduces the number of example keyframes needed to render the set of body poses the user performs.

We note that the planar arm model is quite restrictive, and is only designed to render arm configurations when the user is largely in a fronto-parallel plane with respect to the video wall. This is the normal configuration in our system, and is also the assumption used by the coarse person tracker when identifying head and hand locations. While our avatar rendering system can present arbitrary poses by simply adding keyframes (without interpolated arms), it is not really designed for the task. We are currently researching methods for more general image-based rendering of articulated human forms that can capture and generate general 3-D configurations.

But there is another major limitation in this representation; while it tracks the hand and body pose of the user, it lacks any facial expression or hand gesture. These are critical flaws for a system of virtual communication; coarse body pose alone is hardly an expressive interface.

To solve this, we use the information from our active camera and view-based analysis of facial pose and hand gesture. We train our system to learn the face and hand appearance for certain pose and gesture classes as performed by particular users of the system. We estimate pose and gesture class likelihood scores, and interpolate a gaze angle or expression class value function as described in the previous sections. We then re-render our avatar to express the appropriate facial expression or hand gesture being performed by the user.[10] Figure 13 shows a still of the video demo of our foveated avatar. The avatar is shown, together with windows that show the coarse and fine scale visual tracking. The overall body pose is driven by the coarse tracking process, and

[8] Head and hand positions are automatically found using the tracking method in Section 2, while arm joint positions are currently hand annotated.

[9] Since this implementation only uses warping and does not do a full morph from multiple examples, some artifacts are visible at keyframe transitions.

[10] As we are limited by the keyframe rendering implementation, we cannot smoothly render transitions between expressions and gestures without an excessive number of keyframes; the implementation currently in progress will remove this restriction.

the hand and gestures are updated based on the fine-scale gesture analysis. (See video tape for demonstration.)

7 Conclusion

This paper argues direct perception is possible for user interface and avatar creation tasks, and exploits active methods in perceptual processing. No single spatial scale or perceptual routine will likely suffice for perception of users in general interface tasks, but through the use of an active architecture we can begin to capture the range of user expression needed to implement a remote user presence. The use of machine perception techniques offers the potential for users to accomplish control tasks in real time that would otherwise require bulky, invasive body instrumentation.

We demonstrated multi-scale user representations in virtual spaces driven by active noninvasive estimates of the actual position and pose of a user. Systems such as ours could be used for a variety of telepresence tasks and interfaces for which a keyboard and mouse are either inappropriate or do not generate enough output bandwidth to drive the desired representation.

References

[1] Akita, K., Analysis of body motion image sequences, Proceedings of the 6th International Conference on Pattern Recognition, pp. 320-327, October 1982.

[2] Ballard, D., and Brown, C., (1982) Computer Vision, Prentice-Hall, Englewood

[3] BBN HARK Systems, HARK Users Guide, Cambridge MA, 1994.

[4] Bergeron, P., Lachapelle, P., Techniques for Animating Characters, Advanced Computer Graphics and Animation, ACM SIGGRAPH 85 Tutorial Notes, pp. 61–79, 1985.

[5] Beymer, D.J., Face recognition under varying pose, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 756-761, Seattle, WA, 1994.

[6] Blumberg, B and Tinsley Galyean. Multi-level Direction of Autonomous Creatures for Real-Time Virtual Environ ments. Proceedings of SIGGRAPH 95 , 1995.

[7] Casey, M., Gardner, W., and Basu, S., "Vision Steered Beam-forming and Transaural Rendering for the Artificial Life Interactive Virtual Environment (ALIVE)". In *Proc. Audio Eng. Soc. Conv.*, 1995.

[8] Cox, H., "Robust Adaptive Beamforming" *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(10):1365-1376, 1987.

[9] Darrell, T., and Pentland, A., Space-Time Gestures. Proceedings IEEE CVPR-93, New York, IEEE Comp. Soc. Press, 1993.

[10] Darrell, T., Essa, I., and Pentland, A., Correlation and Interpolation Networks for Real-Time Expression Analysis/Synthesis, Advances in Neural Information Processing Systems-7, Tesauro, Touretzky, and Leen, eds., MIT Press. Conference, 1994.

[11] Darrell, T., Moghaddam, B., and Pentland, A., Active Face Tracking and Pose Estimation in an Interactive Room, Proc. Computer Vision and Pattern Recognition, CVPR-96, San Francisco. 1996.

[12] Darrell, T., Maes, P., Blumberg, B., Pentland, A. P., A Novel Environment for Situated Vision and Behavior, Proc. IEEE Workshop for Visual Behaviors, IEEE Comp. Soc. Press, Los Alamitos, CA, 1994

[13] Darrell, T., Blumberg, B., Maes, P., and Pentland, A., ALIVE: dreams and illusions, Visual Proceedings of SIGGRAPH 95, ACM Press, 1995.

[14] Essa, I., and Pentland, A. P., A vision system for observing and extracting facial action parameters, In Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1994.

[15] Ginsberg, C., and Maxwell, D., "Graphical Marionette", In Proc. SIGGRAPH / SIGART Motion Workshop, pp. 172-179, Toronto, 1983.

[16] Horn, B.K.P., Robot Vision, M.I.T. Press, Cambridge, MA, 1991.

[17] Kakadiaris, I. and Metaxas, D. and Bajcsy, R., Active Part-Decomposition, Shape and Motion Estimation of Articulated Objects: A Physics-based Approach, Proc. CVPR '94, pp. 980-984, 1994.

[18] Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M., A Stereo Engine for Video-rate Dense Depth Mapping and Its New Applications, Proceedings of Conference on Computer Vision and Pattern Recognition, pp. 196-202, June 1996.

[19] Khalil, F., Jullien, J.P., and Gilloire, A., "Microphone Array for Sound Pickup in Teleconference Systems". *Journal of the Audio Engineering Society*, 42(9):691-699, 1994.

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

[21] Litwinowitcz, P., Williams, L., Animating Images with Drawings, Proc. ACM SIGGRAPH, pp. 409–412, 1994.

[22] Mailloux, R.J., *Phased Array Antenna Handbook*. Artech House, Boston, 1994.

[23] Moghaddam, B. and Pentland, A., "Probabilistic Visual Learning for Object Detection," *Proc. of Int'l Conf. on Comp. Vision*, Camb., MA, June 1995.

[24] Moghaddam, B., Nastar, C., and Pentland, A., "A Bayesian Similarity Metric for Direct Image Matching," *Proc. of Int'l Conf. on Pattern Recognition*, Zurich, September 1996.

[25] Ohya, J., Ebihara, K., Kurumisawa, J., and Nakatsu, R., " Virtual Kabuki Theater: Towards the realization of human metamorphosis system", Proc. of 5th IEEE International Workshop on Robot and Human Communication, pp.416-421, November 1996.

[26] Poggio, T., and Girosi, F., A theory of networks for approximation and learning, MIT AI Lab TR-1140, 1989.

[27] Poggio, T., and Edelman, S., A Network that Learns to Recognize Three Dimensional Objects, *Nature*, Vol. 343, No. 6255, pp. 263-266, 1990.

[28] Rehg, J.M. and Kanade, T., Visual Tracking of High DoF Articulated Structures: An Application to Human Hand Tracking, Proc. ECCV '94, pp. B:35-46, 1994.

[29] Rohr, K., Towards Model-Based Recognition of Human Movements in Image Sequences, Comp. Vision, Graphics, and Image Processing: Image Understanding, Vol. 59, no. 1, pp. 94-115, Jan 1994.

[30] Russell, K., Starner, T., and Pentland, A., Unencumbered Virtual Environments. MIT Media Laboratory Perceptual Computing Group Technical Report 305. Appears in IJCAI '95 Workshop on Entertainment and AI/Alife. Montreal, Canada, August 20-25, 1995.

[31] Soede, W., Berkhout, A., and Bilsen, F. "Development of a Directional Hearing Instrument Based on Array Technology". Journal of the Aoustical Society of America, 94(2):785-798, 1993.

[32] Sparacino, F., Wren, C., Pentland, A., Davenport G., HyperPlex: a World of 3D Interactive Digital Movies, Appears in IJCAI '95 Workshop on Entertainment and AI/Alife. Montreal, Canada, August 20-25, 1995.

[33] Stadler, R., and Rabinowitz, W., "On the Potential of Fixed Arrays for Hearing Aids". *Journal of the Acoustical Society of America*, 94(3):1332-1342, 1993.

[34] Terzopoulos, D., and Waters, K., Physically-based Facial Modeling, Analysis, and Animation, The Journal of Visualization and Computer Animation, vol. 1, pp. 73–80, 1990.

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

[36] Williams, L., Performance-driven facial animation, Proc. ACM SIGGRAPH, Vol. 24, No. 4, pp. 235-242, 1990.

[37] Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. pfinder: Real-Time Tracking of the Human Body, Second Int'l Conf. on Automatic Face and Gesture Recognition, Killington, VT, October 1996.

[38] Ullman, S., and Basri, R., Recognition by Linear Combinations of Models, *IEEE PAMI*, Vol. 13, No. 10, pp. 992-1007, 1991.