M.I.T. Media Laboratory Perceptual Computing Group Technical Report No. 356.

(Submitted, CVPR'96.)

Active Face Tracking and Pose Estimation in an Interactive Room

Trevor Darrell, Baback Moghaddam, and Alex P. Pentland
trevor,baback,sandy@media.mit.edu

Abstract

We demonstrate real-time face tracking and pose estimation in an unconstrained office environment with an active foveated camera. Using vision routines previously implemented for an interactive environment, we determine the spatial location of a user's head and guide an active camera to obtain foveated images of the face. Faces are analyzed using a set of eigenspaces indexed over both pose and world location. Closed loop feedback from the estimated facial location is used to guide the camera when a face is present in the foveated view. Our system can detect the head pose of an unconstrained user in real-time as he or she moves about an open room.

1 Introduction

Faces are an important cue for systems which interact with people. To be useful, a system should know whether a user is paying attention, in particular where the user is looking during an interactive dialog. Our goal in this paper is to develop a user interface which can track the face of an unconstrained user and estimate his/her pose, as he or she walks about a room. To be successful, this analysis must occur in real-time, which places considerable constraints on the type of face processing that can be performed. Our approach is to combine active vision methods with eigenspace-based estimation of facial pose. We apply our method in an interactive domain where the user can walk freely about in a 15' by 15' space, facing a large video projection screen.

Our method uses person tracking routines run on a wide angle camera view to first locate the person in the room, and then eigenspace-based face analysis on a narrow angle camera view for accurate face tracking and pose estimation. Camera control is performed open-loop using the general person tracking

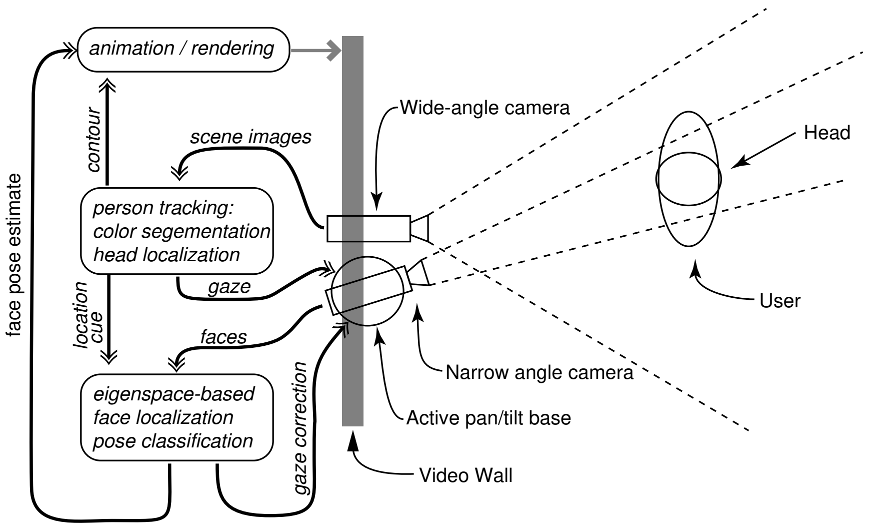


Figure 1: Overview of system for face/body tracking and pose estimation, Objects are rendered on Video Wall and react to facial pose of user. Static, wide-field-of-view, camera tracks user's head, and drives gaze control of active, narrow-field-of-view camera. Eigenspace-based pose estimation is run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

routines, and closed-loop, using feedback from eigenspace-based face and pose models (Figure 1.)

First we will review person tracking in our interactive room environment, as well as our methods for eigenspace-based pose estimation. We will then present our real-time implementation of the estimation and tracking methods, and discuss our how location-specific eigenspace learning can overcome significant variation in imaging conditions. Finally we will show results demonstrating the accuracy of pose estimation in our real-time system.

2 Person Tracking

Previously, we have implemented vision routines to track a user in an office setting as part of our ALIVE system, an Artificial Life Interactive Video Environment [5]. This system can track people and identify head/hand locations as they walk about a room, and provide foveation cues to guide an active camera to foveate head or hands. These visual routines assume only that the user is fac-

Figure 2: Person tracking in a system for vision-based interaction with a virtual environment. (a,b) A user sees him/herself in a "magic mirror", composited in a virtual environment. Computer vision routines analyze the image of the person to allow him/her to effect the virtual world through direct manipulation and/or gestural commands. (c) Results of feature tracking routine; head, hands, and feet are marked with color-coded balls.

ing the screen in front of a known background, and can operate on coarse-scale images.

The ALIVE system was originally designed to allow a user to interact with virtual creatures, through the use of a "magic-mirror" metaphor in which a user sees him/herself presented in a video display along with graphical objects and virtual creatures (Figure 2). A wide field-of-view video camera acquires an image of the user, which is then combined with computer graphics imagery and projected on a large screen in front of the user.

Vision routines acquire the image of the user, compute figure/ground segmentation, and find the location of head, hands, and other salient body features. We use only a single, calibrated, wide field-of-view camera to determine the 3-D position of these features. We do assume a fixed color background, and that the person is facing the camera/screen. For details of our method, see [13]; here we summarize the three main steps of the algorithm which are relevant to face tracking:

1. A multi-class color classification test is used to compute figure/ground segmentation, using a single Gaussian model of background pixel color and an n-class adaptive model of foreground (person) colors.

2. Region growing is performed, starting at the centroid location of the person in the previous frame, to find a single connected region. If this fails to grow a sufficiently large region, random seed points are selected until a stable region is found. The contour of the extracted region is found by chain-coding the connected foreground region.

3. Using the known camera geometery, the lowest point of the contour in the image is projected onto the known ground plane, to determine the location

in depth of the person. The contour is projected from 2-D screen coordinates into 3-D world coordinates, based on the depth value computed in the previous step. (In the ALIVE system this contour is then used to perform video compositing and depth clipping to combine the user's video image with computer graphics imagery.) The head is defined to be the highest contour point in a neighborhood directly above the centroid of the foreground region.

This monocular, wide field-of-view person tracking method 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 geometery 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 face, using a second, active camera. Since our active camera is mounted some distance from the wide angle camerea, (approx 6 ft.) we use the estimated 3-D head location and derive the active camera gaze angle with simple trigonometry using the know active camera base location. [1]

In the results shown in this paper, the wide angle camera was placed on top of a 8 ft. video projector screen and the active camera placed at the base of the screen; the 3-D method was used to determine gaze angle. We obtained reliable tracking using this method; figure 3 shows pairs of output from the wide and narrow cameras in our active-vision ALIVE 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 the eigenspace method, as presented below.

3 Pose Estimation with Eigenspaces

There are essentially two ways of approaching the problem of pose estimation in an eigenspace framework. Given N individuals under M different poses, one can do recognition and pose estimation in a universal eigenspace computed from the combination of NM images. In this way a single "parametric eigenspace" will encode both identity as well as pose. Such an approach, for example, has recently been used by Murase and Nayar [8] for general 3D object recognition and pose estimation.

Pentland *et al.* [9] have suggested a view-based approach to face recognition under varying pose. In this formulation a separate set of "eigenfaces" is computed for each possible object pose. Object pose is identified by computing the eigenspace projection of the input image onto each eigenspace and selecting the one with the lowest residual error (or "distance-from-feature-space" (DFFS)

[1] 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,

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

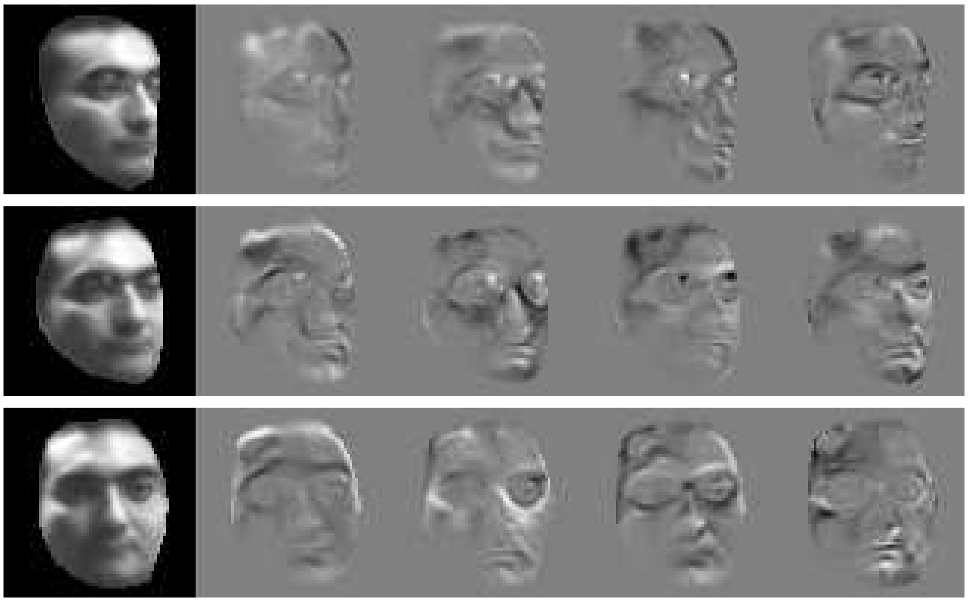


Figure 4: Multiple-Pose Eigenfaces. Mean templates (E0) are shown on the left along with the first 4 eigenvectors (E1 to E4).

metric [9]). This scheme can be viewed as a *multiple-observer* system where separate eigenspaces are simultaneously "competing" in describing the input image (see [12] and [4] for related work). Examples of eigenfaces for multiple poses (at the same spatial location) are shown in Figure 4.

The key difference between the view-based and parametric representations can be understood by considering the geometry of facespace. In the high-dimensional vector space of an input image, multiple-orientation training images are represented by a set of M distinct regions, each defined by the scatter of N individuals. Multiple views of a face form non-convex (yet connected) regions in image space [1]. Therefore the resulting ensemble is a highly complex and non-separable manifold.

The difference between the two approaches is illustrated in Figure 5. The parametric eigenspace attempts to describe this ensemble by a projection onto a single low-dimensional linear subspace (corresponding to the first n eigenvectors of the NM training images). In contrast, the view-based approach corresponds to M independent subspaces, each describing a particular region of the facespace (corresponding to a particular view of a face). The relevant analogy here is that of modeling a complex distribution by a single cluster model or by the union of several component clusters. The latter (view-based) representation can yield a more accurate representation of the underlying geometry depending on the

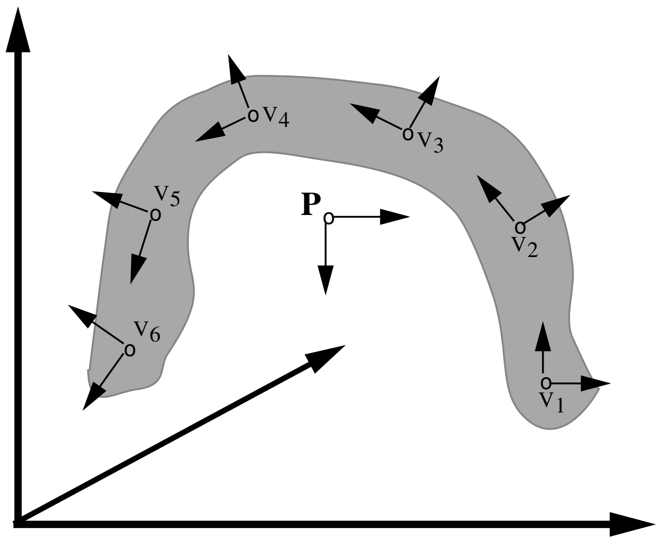


Figure 5: A schematic representation of parameteric vs. view-based eigenspaces.

degree of manifold complexity of the data.

3.1 MAP estimation with Eigenspaces

Recently Moghaddam & Pentland [7] have shown that the DFFS measure can be combined with a corresponding "distance-in-feature-space" (DIFS) to yield an estimate of the probability density function for a class of images. This *likelihood* estimate can be made optimal (with respect to information-theoretic *divergence*) and can be computed solely from the low-dimensional subspace projection coefficients, thus yielding a computationally efficient estimator for high-dimensional probability density functions.

Specifically, given a set of training images $\{\mathbf{x}^t\}_{t=1}^{N_T}$, from an object class Ω (in this case a collection of user views from a single location and pose), we wish to estimate the *likelihood* function for this data — *i.e.,* $P(\mathbf{x}|\Omega)$. We note that from a probabilistic perspective, the class-conditional density $P(\mathbf{x}|\Omega)$ is the most important data representation to be learned. This density is the critical component in detection, recognition, prediction, interpolation and general inference. In our case, having learned these densities for several pose classes $\{\Omega_1, \Omega_2, \cdots, \Omega_n\}$, we can formulate either a maximum-likelihood estimate

$$\Omega^{\mathrm{ML}}(\mathbf{x}) = \mathrm{argmax}\{P(\mathbf{x}|\Omega_i)\} \tag{1}$$

or a maximum *a posteriori* estimate

$$\Omega^{\mathrm{MAP}}(\mathbf{x}) = \Omega_j \;\; \text{s.t. } P(\Omega_j|\mathbf{x}) > P(\Omega_i|\mathbf{x}) \; \forall i \neq j \tag{2}$$

using Bayes rule

$$P(\Omega_i|\mathbf{x}) \;=\; \frac{P(\mathbf{x}|\Omega_i)P(\Omega_i)}{\sum_{j=1}^{n} P(\mathbf{x}|\Omega_j)P(\Omega_j)} \tag{3}$$

We now review how an arbitrary density estimate $P(\mathbf{x}|\Omega_i)$ can be computed using the eigenspace technique of [7] specialized to the case of a Gaussian distribution.

3.2 Principal Component Imagery

Given a set of m-by-n images $\{I^t\}_{t=1}^{N_T}$, we can form a training set of vectors $\{\mathbf{x}^t\}$, where $\mathbf{x} \in \mathcal{R}^{N=mn}$, by lexicographic ordering of the pixel elements of each image I^t. The basis functions in a Karhunen-Loeve Transform (KLT) [6] are obtained by solving the eigenvalue problem

$$\Lambda \;=\; \Phi^T \Sigma \Phi \tag{4}$$

where Σ is the covariance matrix of the data, Φ is the eigenvector matrix of Σ and Λ is the corresponding diagonal matrix of eigenvalues. In PCA, a partial KLT is performed to identify the largest-eigenvalue eigenvectors and obtain a principal component feature vector $\mathbf{y} = \Phi_{\mathrm{M}}^T\,\tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$ is the mean-normalized image vector and Φ_M is a submatrix of Φ containing the principal eigenvectors. PCA can be seen as a linear transformation $\mathbf{y} = \mathcal{T}(\mathbf{x}) : \mathcal{R}^N \rightarrow \mathcal{R}^M$ which extracts a lower-dimensional subspace of the KL basis corresponding to the maximal eigenvalues. This corresponds to an orthogonal decomposition of the vector space $\mathcal{R}^N$ into two mutually exclusive and complementary subspaces: the principal subspace (or feature space) $F = \{\Phi_i\}_{i=1}^{M}$ containing the principal components and its orthogonal complement $\bar{F} = \{\Phi_i\}_{i=M+1}^{N}$, as illustrated in Figure 6.

In a partial KL expansion, the residual reconstruction error is defined as

$$\epsilon^2(\mathbf{x}) \;=\; \sum_{i=M+1}^{N} y_i^2 \;=\; ||\tilde{\mathbf{x}}||^2 \;-\; \sum_{i=1}^{M} y_i^2 \tag{5}$$

and can be easily computed from the first M principal components and the L_2-norm of the mean-normalized image $\tilde{\mathbf{x}}$. Consequently the L_2 norm of every element $\mathbf{x} \in \mathcal{R}^N$ can be decomposed in terms of its projections in these two subspaces. We refer to the component in the orthogonal subspace $\bar{F}$ as the "distance-from-feature-space" (DFFS) which is a simple Euclidean distance and is equivalent to the residual error $\epsilon^2(\mathbf{x})$ in Eq.(5). The component of $\mathbf{x}$ which lies

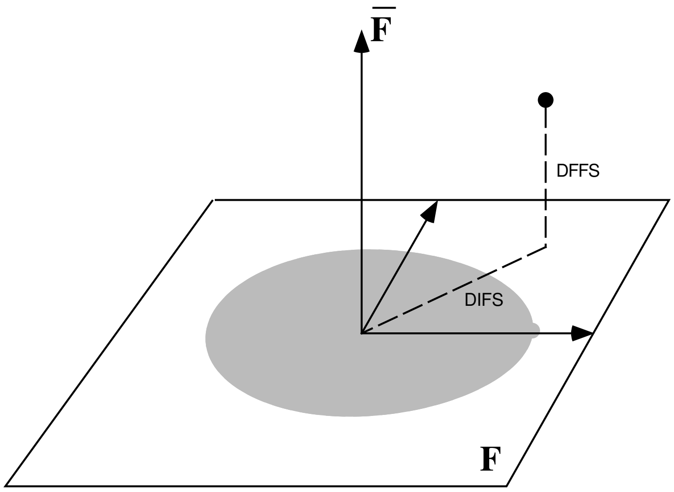


Figure 6: The principal subspace F and its orthogonal complement $\bar{F}$ for a Gaussian density.

in the feature space F is referred to as the "distance-in-feature-space" (DIFS) but is generally not a distance-based norm, but can be interpreted in terms of the probability distribution of y in F.

As shown in Appendix A, an estimator for $\hat{P}(\mathbf{x}|\Omega)$ is given by:

$$\begin{aligned}\hat{P}(\mathbf{x}|\Omega) &= \left[\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{1/2}}\right]\cdot\left[\frac{\exp\left(-\frac{\epsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(N-M)/2}}\right] \\ &= P_F(\mathbf{x}|\Omega)\ \hat{P}_{\bar{F}}(\mathbf{x}|\Omega)\end{aligned} \tag{6}$$

In general, brute-force computation of pose likelihoods in real-time is computationally infeasible. Fortunately, most of the information computed by a brute force evaluation of DFFS is of little importance–what is of interest is the location of the minima of the distance function. Following [2], we use the zero-th order eigenvectors, E0, to perform spatial localization within the foveated camera view. We compute a coarse to fine search using the E0 template for each pose, and find the pose and offset which has maximal normalized correlation response. We then fully evaluate the higher order eigenvectors at this location for each pose, and compute the pose class likelihood as given above.

4 Location-dependent Eigenspace Learning

Face images obtained from our active camera can be used to compute pose estimates, using the eigenspaces technique described above. However, with a user moving in 3-space, we have to deal with considerable variations in scale (size of head), and illumination changes (such as shadows) that are not well modeled by a single eigenspace. These variations have large-scale geometric effects, just as do changes in pose. Our approach is to define multiple *sets* of eigenspaces, indexed over both pose and location in the world. A set of eigenspaces is constructed corresponding to each facial pose and world location. Each pose class is defined by a set of location specific pose class statistics:

$$\Omega_i = \{\Omega_{i,l}\}, l \in \mathcal{L}, \tag{7}$$

where the set of world locations is given by

$$\mathcal{L} = \{\mathbf{z}_0, \mathbf{z}_1, ...\mathbf{z}_L\}, \tag{8}$$

where $\mathbf{z}$ is a 3-D coordinate vector. To compute a composite pose class likelihood, we consider the estimation problem to be a case of estimation given spare observations. We approximate the probability at locations where no training data is available. Given an observed face image x at a world location z^*, we compute an approximate probability via interpolation among the K nearest locations which have actual proability estimates. Using a linear interpolant, we have

$$\hat{P}(\mathbf{x}, z^*|\Omega_i) \approx \sum_{k=0}^{K} w(k)\hat{P}(\mathbf{x}|\Omega_{i,n(k)}), \tag{9}$$

where $n(k)$ is a function that returns the k-th nearest location to z^* in $\mathcal{L}$, and $w(k)$ weights the distance of each location

$$w(k) = \frac{||z^* - n(k)||^2}{\sum_{j=0}^{K} ||z^* - n(j)||^2} \tag{10}$$

This offers much increased accuracy over computing a single set of pose eigenspaces for use over the entire room environment. Figure 7 shows sets of eigenspaces for three different poses collected at three different world locations.

Note that we need not evaluate the eigenspaces for each possible world location, since the person tracking routines provide an estimate of the users position that is sufficient to restrict the set of eigenspaces used by the system. The run-time computational burden of having L different world locations each with a separate set of pose templates is k times the cost of a single location, since we need not evaluate the eigenspace likelihoods that for locations that are not in the nearest neighbor set [11].

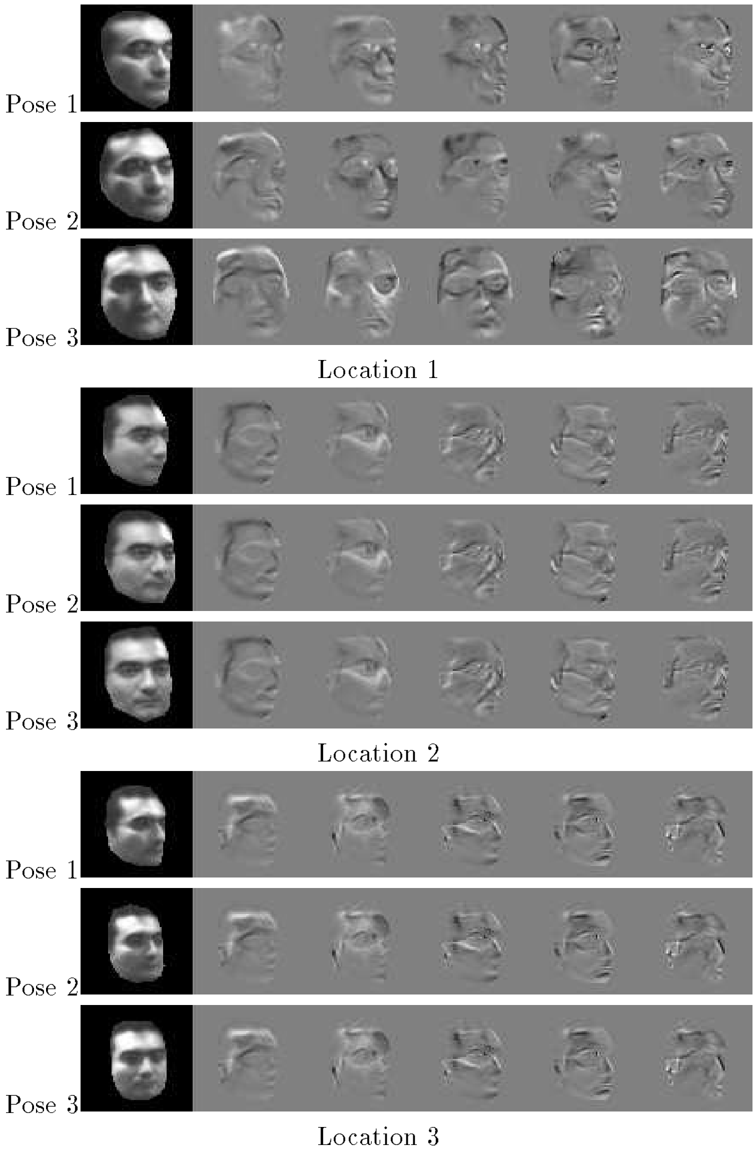
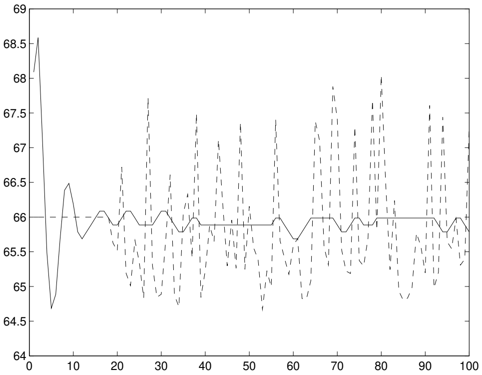


Figure 7: Multiple-Pose Eigenspaces for 3 different spatial locations

(a) 

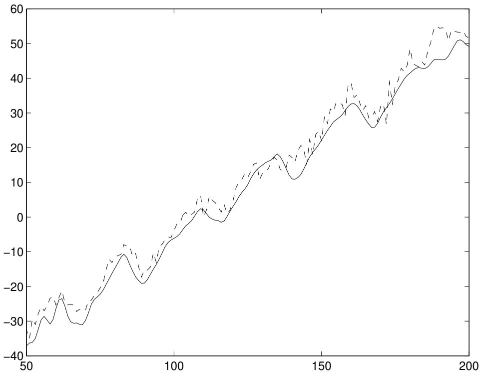
(b) 

Figure 8: Tracking results: plot of pan angle for (a) stationary user, and (b) moving user who walked across room while oscillating head. Dashed line shows pan position under open-loop control; solid line shows pan position under closed-loop control.

5 Performance

We evaluated the tracking and pose estimation performance of our system. Eigenspaces were trained for each of 3 poses at 10 different world locations, using a sample set size of 10 images at each location. The locations were set to be in two concentric semi-circles on the floor of the workspace, at camera pan angles of -32, -16, 0, 16, and 32 degrees, and ranges of 80 and 120 inches. The active camera was fitted with a lens of 50mm focal length (c-mount type). Figure 7 shows three of the eigenspaces that were trained.

In these experiements we have used multiple views of a *single* user to construct eigenspaces. In our data set, each eigenspace describes variations in appearance due to expressions, slight mis-alignments and with and without glasses. (The method, however, can easily be extended to multiple users.)

We then evaluated the performance of the system against new images of the same user both at the locations where the eigenspaces were trained, and at randomly selected floor locations. We used the spatial localization method described above, evaluating eigenspaces at the location at which the corresponding E0 template had maximal normalized correlation response.

First, we note that eigenspace face analysis can improve head tracking accuracy using closed-loop feedback to guide the active camera. Figure 8 compares the camera position in the case of open loop control, when the gaze angle is determined only by the wide-angle person finder, and closed loop control, when the gaze angle is corrected by the offset of the face in the current foveated image. During normal system operation, we set a threshold on DFFS value to determine the transition between open and closed loop state, so that the closed loop

(a) trained locations	(actual) Ω_1	Ω_2	Ω_3
(observed)			
Ω_1	10	0	0
Ω_2	0	10	3
Ω_3	0	0	7

(b) untrained locations	(actual) Ω_1	Ω_2	Ω_3
(observed)			
Ω_1	14	2	2
Ω_2	1	12	3
Ω_3	0	1	10

(c) all trials	(actual) Ω_1	Ω_2	Ω_3
(observed)			
Ω_1	24	2	2
Ω_2	1	22	6
Ω_3	0	1	17

Table 1: Results of pose classification experiment determing the pose of a user facing a display screen as the user stood at various locations in an interactive room. The task was to classify where on the video screen the user was looking; left (Ω_1), center (Ω_2), or right (Ω_3). A multiple location/multiple pose eigenspace technique was used on the output of an active camera tracking the users head, as described in the text. The confusion matrix was computed for (a) trials at trained locations, (b) trials at non-trained locations, (c) all trails. An overall success rate of 84% was achieved.

signal does not contribute when there is no face in the active camera field of view. During these runs, the user was approximately twelve feet from the camera, and walked freely in approximately a ten by ten foot area. Total time for computing pose estimates and active tracking, including closed loop feedback, was less than 1/5 second.

Second, we show the pose classification rate for our system. In a trial with $n = 25$ observations, where 10 of these observations were at the training locations and the remainder at locations chosen with a uniform probability across the workspace, we computed the pose class confusion matrix. Three pose classes were used, one for looking to the left of the screen (Ω_1), one for looking at the center of the screen (Ω_2), and one for looking at the right of the screen (Ω_3). Recall that the screen was situated in front of the 15'x15' space, and was itself 8'x10'. Results of our system on this experiment are shown in Table 1. We obtained an overall success rate of 84% (63/75) for all trails, which breaks down to a success rate of 90% (27/30) on the trails at the locations were the eigenspaces were trained, and 80% (36/45) on the trails at randomly selected locations.

6 Conclusion

In conlusion, we have shown that by intergrating person tracking routines, an active camera, and multiple eigenspace pose models, we can accurately estimate the direction of gaze of a user interacting with a large screen video display. In the experiment described here, the user was on average 15' from the cameras and the display, and yet our system could discriminate pose classes which amounted to 10-15 degrees of gaze angle. Our system runs in real time, and is used in applications for interacting with virtual environments or agents that can respond appropriately to the users gaze, such as showing more information about an object of interest.

Appendix A. Gaussian F-Space Densities

We now consider an optimal approach for estimating high-dimensional Gaussian densities. We assume that we have (robustly) estimated the mean $\bar{\mathbf{x}}$ and covariance Σ of the distribution from the given training set $\{\mathbf{x}^t\}$. Under this assumption, the likelihood of a input pattern $\mathbf{x}$ is given by

$$P(\mathbf{x}|\Omega) = \frac{\exp\left[-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T\Sigma^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right]}{(2\pi)^{N/2}\ |\Sigma|^{1/2}} \tag{11}$$

The sufficient statistic for characterizing this likelihood is the *Mahalanobis* distance

$$d(\mathbf{x}) \ = \ \tilde{\mathbf{x}}^T\Sigma^{-1}\tilde{\mathbf{x}} \tag{12}$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$. Using the eigenvectors and eigenvalues of Σ we can rewrite Σ^{-1} in the diagonalized form

$$\begin{aligned} d(\mathbf{x}) &= \tilde{\mathbf{x}}^T\Sigma^{-1}\tilde{\mathbf{x}} \\ &= \tilde{\mathbf{x}}^T\left[\Phi\Lambda^{-1}\Phi^T\right]\tilde{\mathbf{x}} \\ &= \mathbf{y}^T\Lambda^{-1}\mathbf{y} \end{aligned} \tag{13}$$

where $\mathbf{y} = \Phi^T\tilde{\mathbf{x}}$ are the new variables obtained by the change of coordinates in a KLT. Because of the diagonalized form, the *Mahalanobis* distance can also be expressed in terms of the sum

$$d(\mathbf{x}) \ = \ \sum_{i=1}^{N}\frac{y_i^2}{\lambda_i} \tag{14}$$

We now seek to estimate $d(\mathbf{x})$ using only the M principal projections. Therefore, we formulate an estimator for $d(\mathbf{x})$ as follows

$$
\begin{aligned}
\hat{d}(\mathbf{x}) &= \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} \;+\; \frac{1}{\rho}\left[\sum_{i=M+1}^{N} y_i^2\right] \\
&= \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} \;+\; \frac{1}{\rho}\,\epsilon^2(\mathbf{x})
\end{aligned}
\tag{15}
$$

where the term in the brackets is the DFFS $\epsilon^2(\mathbf{x})$, which as we have seen can be computed using the first M principal components. We can therefore write the form of the likelihood estimate based on $\hat{d}(\mathbf{x})$ as the product of two marginal and independent Gaussian densities

$$
\begin{aligned}
\hat{P}(\mathbf{x}|\Omega) &= \left[\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{1/2}}\right] \cdot \left[\frac{\exp\left(-\frac{\epsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(N-M)/2}}\right] \\
&= P_F(\mathbf{x}|\Omega)\;\hat{P}_{\bar{F}}(\mathbf{x}|\Omega)
\end{aligned}
\tag{16}
$$

where $P_F(\mathbf{x}|\Omega)$ is the true marginal density in F-space and $\hat{P}_{\bar{F}}(\mathbf{x}|\Omega)$ is the estimated marginal density in the orthogonal complement $\bar{F}$-space. The optimal value of ρ can now be determined by minimizing a suitable cost function $J(\rho)$. From an information-theoretic point of view, this cost function should be the Kullback-Leibler divergence [3] between the true density $P(\mathbf{x}|\Omega)$ and its estimate $\hat{P}(\mathbf{x}|\Omega)$

$$
J(\rho) \;=\; \mathrm{E}\left[\log \frac{P(\mathbf{x}|\Omega)}{\hat{P}(\mathbf{x}|\Omega)}\right] \tag{17}
$$

Using the diagonalized forms of the *Mahalanobis* distance $d(\mathbf{x})$ and its estimate $\hat{d}(\mathbf{x})$ and the fact that $\mathrm{E}[y_i^2] = \lambda_i$, it can be easily shown that

$$
J(\rho) \;=\; \frac{1}{2}\sum_{i=M+1}^{N}\left[\frac{\lambda_i}{\rho} - 1 + \log\frac{\rho}{\lambda_i}\right] \tag{18}
$$

The optimal weight ρ^* can be then found by minimizing this cost function with respect to ρ. Solving the equation $\frac{\partial J}{\partial \rho} = 0$ yields

$$
\rho^* \;=\; \frac{1}{N-M}\sum_{i=M+1}^{N}\lambda_i \tag{19}
$$

which is simply the arithmetic average of the eigenvalues in the orthogonal subspace $\bar{F}$. In addition to its optimality, ρ^* also results in an *unbiased* estimate of the *Mahalanobis* distance — *i.e,* $\mathrm{E}[\hat{d}(\mathbf{x};\rho^*)] = \mathrm{E}[d(\mathbf{x})]$. What this derivation shows is that once we select the M-dimensional principal subspace F (as indicated, for example, by PCA), the optimal density estimate $\hat{P}(\mathbf{x}|\Omega)$ has the form of Eq.(16) with ρ given by Eq.(19).

References

[1] Bichsel, M., and Pentland, A., "Human Face Recognition and the Face Image Set's Topology," *CVGIP: Image Understanding*, vol. 59, no. 2, pp. 254-261, 1994.

[2] Burl, M.C., et al., "Automating the Hunt for Volcanos on Venus", *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, Seattle, WA, June 1994.

[3] Cover, M. and Thomas, J.A., *Elements of Information Theory*, John Wiley & Sons, New York, 1994.

[4] Darrell, T., and Pentland, A., "Space-Time Gestures," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, New York, NY, June 1993.

[5] Darrell, T., Maes, P., Blumberg, B., and Pentland, A. P., "A Novel Environment for Situated Vision and Behavior", *Proc. IEEE Wkshp. for Visual Behaviors (CVPR-94)*, IEEE C.S. Press, Los Alamitos, CA, 1994

[6] Loeve, M.M., *Probability Theory*, Van Nostrand, Princeton, 1955.

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

[8] Murase, H., and Nayar, S.K., "Visual Learning and Recognition of 3D Objects from Appearance," *Int'l Journal of Computer Vision*, vol. 14, no. 1, 1995.

[9] Pentland, A., Moghaddam, B. and Starner, T., "View-based and modular eigenspaces for face recognition," *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, June 1994.

[10] Turk, M., and Pentland, A., "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience,* Vol. 3, No. 1, 1991.

[11] Weng, J.J., "On Comprehensive Visual Learning", *Proc. NSF/ARPA Workshop on Performance vs. Methodology in Computer Vision*, Seattle, WA, June 1994.

[12] Wilson, A., and Bobick, A., "Learning visual behavior for gesture analysis", to appear, *Proc. International Symposium on Computer Vision*, Coral Gables, November 1995.

[13] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body", in *Proc SPIE Photonics East 1995*, also available as MIT Media Lab Perceptual Computing Technical Report TR-353.