Interval Research Corporation Technical Report # 1996-060



# Eigen-points: Control-point Location using Principal Component Analyses

Michele Covell (covell@interval.com)
Interval Research Corporation, 1801 Page Mill Road, Bldg. C, Palo Alto, CA 94304, USA

## ABSTRACT

*Eigen-points estimates the image-plane locations of fiduciary points on an objects. By estimating multiple locations simultaneously, eigen-points exploits the inter-dependence between these locations. This is done by associating neighboring, inter-dependent control-points with a model of the local appearance. The model of local appearance is used to find the feature in new unlabeled images. Control-point locations are then estimated from the appearance of this feature in the unlabeled image. The estimation is done using an affine manifold model of the coupling between the local appearance and the local shape.*

*Eigen-points uses models aimed specifically at recovering shape from image appearance. The estimation equations are solved non-iteratively, in a way that accounts for noise in the training data and the unlabeled images and that accounts for uncertainty in the distribution and dependencies within these noise sources.*

## 1 Introduction

Annotating images with control points is useful in many application areas. Applications include automatic lip synching [1], "in-betweening" for animation, bootstrapping annotated databases, interactive video, image segmentation, video compositing, view-based model capture [2], and automatic image morphing [3].

Each of these applications needs the image locations of fiduciary points. Control points mark these image locations.* Most of these applications use a large number of highly inter-dependent control-point locations. For example, the morphs shown in Figure 1 use 235 control points placed around the face. We could try to locate each of these points using standard feature-spotting methods (e.g. eigen-features [4]). However, such an approach does not exploit the dependences between control-point locations. Instead, eigen-points searches for image features that are associated with a group of control points and then estimates the detailed spatial distribution of the control points around that feature.

The next section of this paper reviews previous approaches to placing control points on images of deformable objects. Section 3 outlines eigen-points, our approach to placing control points automatically. Section 4 presents our results using eigen-points. Finally, Section 5 summarizes the advantages and shortcomings of our approach.

## 2 Previous approaches to point location on images of deformable objects

Our earlier work in automatic image morphing [3][5] placed control points without detailed models of the image content, using general matching techniques. These approaches fail when the images are significantly different, since they rely on direct matching between the images. Furthermore, they do not allow annotated examples to improve future analyses.

Active contour models [6][7] estimate control-point locations along a contour or snake. Bregler et al [7] propose an internal energy term to measure the distance between the estimated and expected shapes of the contour. This allows

* A fiduciary point is a specific location on an object's surface. The control point marks its image-plane location. For example, we might designate the outside left corner of the lips as a fiduciary point on a face. Then, the control point marks $(x,y) = (81,121)$, for example, as the image location where that point appears.

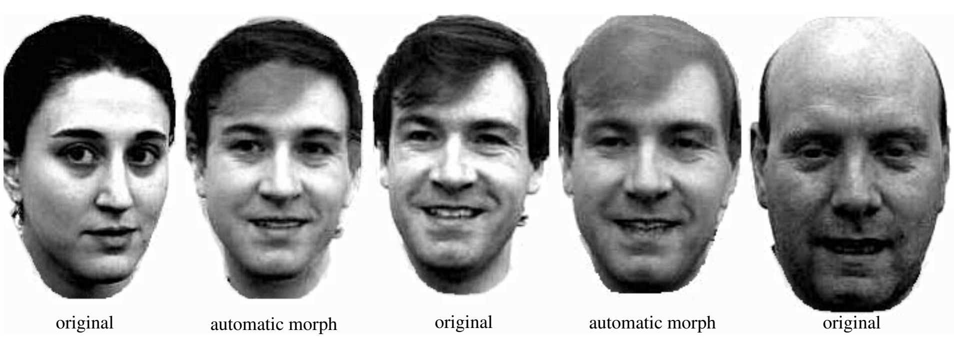


Figure 1: Examples of image morphs using automatically placed correspondences

The control-point locations for these morphs were estimated automatically by eigen-points. Constraints were placed around eyes, nose, mouth, chin and ears. No constraints were placed on the hair.

them to take advantage of example-based learning to constrain the estimated locations of these control points. However, there is no direct link between the image appearance (the external-energy term) and the shape constraints (the internal-energy term). This makes the discovery of "correct" energy functional an error-prone process.

Shape-plus-texture models [8][9] describe appearance using two separate reconstructive models: one for shape (e.g. contour locations) and one for shape-free texture. The shape-free texture descriptions model the grayscale values under object-centric sampling. Thus, the texture models do not describe the observed grayscale data, but instead describe the grayscale data resampled according to the estimated shape description. These shape-plus-texture approaches give simultaneous estimates for many control-point locations. They have well-defined example-based training methods and an error criteria derived from training. However, the texture models use an estimate of shape. Thus, they are forced to rely on iterative solutions to find consistent shape and texture estimates.

Another drawback to shape-plus-texture approaches is their use of *reconstructive* as opposed to *discriminative* models. The texture model capture the principal variations of the (shape-normalized) appearance, giving the minimum mean-square error reconstruction for a given description length. However, our goal is to find a good estimate for the true shape, not to find a good estimate for the true appearance (shape-normalized or otherwise). Instead of a reconstructive texture model, we need a "shape-discriminant" model of texture. That is, we need the model that best captures the principal variations of shape, as manifested in appearance.

The next section describes our approach to discriminating between shapes based on the observed image data.

## 3 Eigen-point approach to placing control points

Using eigen-points, the problem of locating fiduciary points on an unmarked image is solved in two stages. First, the location of features* are estimated; then control points are placed around that feature.

The first stage locates the feature of interest—for example, the actors' lips. This can be done using template- or model-based matching. The feature location defines both the subimage and the image-plane origin that are used in the second stage.

The second stage places the control points around the feature—for example, marking the locations in the image that show the outer boundary of the lips. The locations of the fiduciary points are estimated using an affine manifold model that couples the grayscale values within the feature to the control-point locations associated with the feature. This approach effectively assumes that there is a single $K$-dimensional vector, $x$, which drives both the feature grayscale vector and the control-point locations. The functions which transform this vector into appearance and shape are assumed to be affine.

Assuming a coupled, affine model for image-plane shape and appearance, the defining equations for the grayscale values and the control-point locations are:

$$\begin{aligned} f &= \mathbf{M_F} x + \bar{f} + n_f \\ p &= \mathbf{M_P} x + \bar{p} + n_p \end{aligned} \qquad (1)$$

where $f$ is the vector of grayscale values within the feature extent; $p$ is the vector of (x,y) control-point locations relative to the feature origin; $\bar{f}$ and $\bar{p}$ are the vectors of expected values for $f$ and $p$; $n_f$ and $n_p$ are noise vectors for $f$ and $p$; and $x$ is the vector driving both appearance and shape. Without loss of generality, $x$ is a vector of zero-mean and iid random variables. Also without loss of generality, $\begin{bmatrix} n_f^T & n_p^T \end{bmatrix}^T$ is a vector of iid random variables. This is enforced by prior rotation and re-scaling of $f - \bar{f}$ and $p - \bar{p}$ to diagonalize and equalize the noise covariance matrix.

With this underlying structure, we can relate the grayscale values within the feature to the control-point locations. In training, we use labeled data to estimate the affine manifold over which $\begin{bmatrix} f^T & p^T \end{bmatrix}^T$ varies in response to changes in $x$. This is a coupled manifold model. When labeling new images, the coupled manifold model is used to place control points around the feature location. First, the grayscale values within the feature are projected onto the coupled manifold, giving an estimate for $x$. This manifold location is then reprojected into the control-point subspace, giving estimates for the control-point locations.

The remainder of this section discusses these steps in more detail. Section 3.1 describes the modeling of the labeled training data. Section 3.2 describes the labeling of new image data. Section 3.2 also discusses expected noise sources and proposes variations of the labeling approach to reduce expected errors.

### 3.1 Training on coupled control-point and feature image data

The coupled grayscale/control-point model is computed from a training database. The training data include both feature images and (x,y) locations for the control-points associated with that feature, relative to the "origin" defined by the feature location.

The initial processing to derive this coupled manifold model is similar to that for eigen-features [4]. Feature subimages are analyzed to get $\bar{f}$, the $N_x N_y$-length vector of expected image values, and $\mathbf{F}$, an unbiased matrix of image data. Similarly, the $L$ control-point locations given with

* A "feature" is the image-plane appearance of the object surface surrounding one or more fiduciary points. It is an area of the image (in contrast with control-point locations and fiduciary points).

each image are analyzed to get $\bar{p}$, the *2L*-length vector of expected control-point locations, and $\mathbf{P}$, an unbiased matrix of control-point locations from the training data. These two matrices are combined into an image/control-point matrix $\begin{bmatrix}\mathbf{F}^T & \mathbf{P}^T\end{bmatrix}^T$, with each image column of $\mathbf{F}$ aligned with the corresponding control-point column of $\mathbf{P}$. The most significant left and right singular vectors and the corresponding singular values of this matrix are computed. Using the SVD, for simplicity,*

$$\begin{bmatrix}\mathbf{F}\\ \mathbf{P}\end{bmatrix} = \left[\begin{array}{c|c}\mathbf{U_F} & \\ \mathbf{U_P} & \mathbf{U}_\perp\end{array}\right]\begin{bmatrix}\Sigma_K & \mathbf{0}\\ \mathbf{0} & \Sigma_\perp\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{V} & \mathbf{V}_\perp\end{bmatrix}^T \qquad 2$$

where the first *K* components of the decomposition are considered significant and the remaining are regarded as noise dimensions, $\mathbf{U_F}$ is a $N_xN_y$ x $K$ matrix corresponding to the image subspace and $\mathbf{U_P}$ is a $2L$ x $K$ matrix corresponding to the control-point subspace.

Ideally, this analysis describes the true coupled manifold underlying the data: that is, it describes $\mathbf{M_F}$ and $\mathbf{M_P}$ from Equation 1. In fact, if $E\left\{\begin{bmatrix}n_f\\ n_p\end{bmatrix}\begin{bmatrix}n_f & n_p\end{bmatrix}\right\} = \sigma_{cn}^2\mathbf{I}$ and $\begin{bmatrix}n_f\\ n_p\end{bmatrix}$ is uncorrelated with $x$, then $\mathbf{M_F} = \mathbf{U_F}(\Sigma_K^2 - \sigma_{cn}^2\mathbf{I})^{\frac{1}{2}}$ and $\mathbf{M_P} = \mathbf{U_P}(\Sigma_K^2 - \sigma_{cn}^2\mathbf{I})^{\frac{1}{2}}$ (to within a right unitary transform). Thus, the matrices, $\mathbf{U_F}$, $\mathbf{U_P}$, and $\Sigma_K$, along with the vectors $\bar{f}$ and $\bar{p}$ and the scalar $\sigma_{cn}^2$, form our basic "coupled manifold" model. The next section discusses ways of using this model to estimate control point locations.

## 3.2 Estimating a feature's control-point locations

When labeling new images, the coupled manifold model is used to place control points around the feature location. First, the grayscale values within the feature are projected onto the coupled manifold, giving an estimate for $x$. Assume for the moment that the training data conform to the affine manifold model and are noise free ($\sigma_{cn}^2 = 0$) and that our image data are also noise free. Our best estimate for $x$ (within a unitary transform) is given by:

$$\hat{x} = \Sigma_K^{-1}\mathbf{U_F}^{-1}(f - \bar{f}) \qquad 3$$

This estimated manifold location is then projected into the control-point subspace, giving estimates for the control-point locations:

$$\hat{p} = \mathbf{U_P}\Sigma_K\hat{x} + \bar{p} = \mathbf{U_P}\mathbf{U_F}^{-1}(f - \bar{f}) + \bar{p} \qquad 4$$

Equation 4 provides control-point estimates which are optimal under a fairly stringent set of assumptions. Even with these assumptions, the estimates from Equation 4 suffer from computational noise due to the matrix inverse and the matrix multiply. Under more realistic assumptions, the estimates from Equation 4 suffer from errors in the original manifold model, $\{\mathbf{U_P}\,\mathbf{U_P},\Sigma_K\}$ and from errors due to noise in the image data, $f$. We address each of these potential problems in turn.

### 3.2.1 Avoiding matrix inversion and multiplication

Using a matrix inverse for $\mathbf{U_F}$ followed by a matrix multiply by $\mathbf{U_P}$ introduces more computational noise than necessary. Instead we can take advantage of the special structure imposed on these two matrices by the fact that $\mathbf{U_F}^T\mathbf{U_F} + \mathbf{U_P}^T\mathbf{U_P} = \mathbf{I}$. Using this constraint, a C-S decomposition [10] is used, giving the SVDs for these two matrices which share right singular vectors:

$$\begin{aligned}\mathbf{U_F} &= \left[\mathbf{Q_F}\middle|\ \mathbf{Q_{F\perp}}\right]\begin{bmatrix}\Sigma_\mathbf{F}\\ \mathbf{0}\end{bmatrix}\mathbf{V_{FP}}^T\\ \mathbf{U_P} &= \left[\mathbf{Q_P}\middle|\ \mathbf{Q_{P\perp}}\right]\begin{bmatrix}\Sigma_\mathbf{P}\\ \mathbf{0}\end{bmatrix}\mathbf{V_{FP}}^T\end{aligned} \qquad 5$$

Using this formulation in Equation 4, the control point locations are estimated on a new image using:

$$\hat{p} = \mathbf{Q_P}(\Sigma_\mathbf{P}\Sigma_\mathbf{F}^{-1})\mathbf{Q_F}^T(f - \bar{f}) + \bar{p} \qquad 6$$

The estimation process is now a simple sequence of:

– projection onto the coupled manifold (using the left unitary matrix $\mathbf{Q_F}$);
– component scaling of the manifold coordinates (using the diagonal matrix $\Sigma_\mathbf{P}\Sigma_\mathbf{F}^{-1}$); and
– projection into the control-point space (using the left unitary matrix $\mathbf{Q_P}$).

This combination of steps has lower computational noise than Equation 4: only unitary transforms and scalar operations are used in place of general matrix operations.

### 3.2.2 Adjusting the manifold model for noisy or nonlinear training data

Inaccuracies are introduced into the labeling process by the dimension of the coupled manifold model. The number of principal components to retain in the model is a difficult and largely arbitrary choice. In Equation 4 (and Equation 6), the first *K* component directions ($\begin{bmatrix}\mathbf{U_F}^T & \mathbf{U_P}^T\end{bmatrix}^T$) are treated as if they were determined solely by the signal component of the training data, while the other component directions ($\mathbf{U}_\perp$) are ignored, as if they contain no information about the coupling data. If a small value for *K* is used, image data with valid coupling information about control-point loca-

* Both the left and right principal components can also be determined using a partial eigen-analysis of the coupled matrices.

tions are ignored. If a large value for $K$ is used, the fidelity of some of the component directions (the columns of $\mathbf{U_F}$ and $\mathbf{U_P}$) is extremely low and this corrupts the final estimate of all the control-point locations.This problem can be mitigated by replacing the hard decision with a gradual roll-off across the coupling components.

When the noise vector $\begin{bmatrix} n_f^T & n_p^T \end{bmatrix}^T$ is *truly* uncorrelated and identically distributed, it does not rotate the principle component directions away from the underlying manifold. In this case, $\mathbf{U_F}(\Sigma_K^2 - \sigma_{cn}^2\mathbf{I})^{\frac{1}{2}}$ and $\mathbf{U_P}(\Sigma_K^2 - \sigma_{cn}^2\mathbf{I})^{\frac{1}{2}}$ provide an optimal estimate for the underlying coupled manifold. Unfortunately, insuring that the noise vector is uncorrelated and identically distributed requires a complete model for its covariance matrix. The noise models are only roughly known and the actual noise vector, even after data rotation and rescaling, typically violates the uncorrelated/identically-distributed condition.

When the noise covariance is not a scaled identity matrix, the noise covariance rotates the principal component directions of the training data away from the component directions of the coupled manifold. These rotations are well approximated as additive noise in the "observation" of $x$ given by Equation 3. Based on subspace sensitivity analyses [10], the variance of this additive noise in $\hat{x}$ is bounded by $\sigma_{cn}^2(\Sigma_K^2 - \sigma_{cn}^2\mathbf{I})^{-1}$ where $\sigma_{cn}^2$ is the norm of the covariance of $\begin{bmatrix} n_f^T & n_p^T \end{bmatrix}^T$. The farther the covariance of $\begin{bmatrix} n_f^T & n_p^T \end{bmatrix}^T$ is from a scaled identity matrix, the tighter this bound is on the additive noise in $\hat{x}$. Using this formulation, the minimum-mean-square-error estimate for $x$ is then:

$$\begin{aligned} \hat{x} &= \left(I + \sigma_{cn}^2(\Sigma_K^2 - \sigma_{cn}^2\mathbf{I})^{-1}\right)^{-1}\Sigma_K^{-1}\mathbf{U_F^{-1}}(f - \bar{f}) \\ &= \Sigma_K^{-3}(\Sigma_K^2 - \sigma_{cn}^2\mathbf{I})\mathbf{U_F^{-1}}(f - \bar{f}) \end{aligned} \qquad 7$$

Combining this with the Equations 4 and 5 gives:

$$\hat{p} = \mathbf{Q_P}(\Sigma_\mathbf{P}\mathbf{X}_{QQ}\Sigma_\mathbf{F}^{-1})\mathbf{Q_F^T}(f - \bar{f}) + \bar{p} \qquad 8$$

where $\mathbf{X}_{QQ} = \mathbf{V_{FP}^T}\Sigma_K^{-2}(\Sigma_K^2 - \sigma_{cn}^2\mathbf{I})\mathbf{V_{FP}}$ and $\mathbf{V_{FP}}$ is defined in Equation 5.

Including $\Sigma_K^{-2}(\Sigma_K^2 - \sigma_{cn}^2\mathbf{I})$ (or $\mathbf{X}_{QQ}$) in the computation provides a gradual roll-off across the coupling components, de-emphasizing the parts of the coupled data which are likely to come from noisy components.

The estimation process still uses unitary matrices to project to and from the coupled manifold ($\mathbf{Q_F}$ and $\mathbf{Q_P}$). However, a new matrix multiply is now needed on the manifold coordinates (for $\Sigma_\mathbf{P}\mathbf{X}_{QQ}\Sigma_\mathbf{F}^{-1}$). This approach is still best, with regard to computational noise, since the general matrix multiply occurs within a lower dimensional space (only *K*x*K* dimensions, instead of $2LxN_xN_y$ dimensions).

### 3.2.3 Adjusting for noise in the unlabeled image data

Equation 7 is the best estimate of $x$, assuming that there is no noise in the unlabeled image data ($f$). When there is noise in $f$ that is uncorrelated with $x$, we need to adjust $\hat{x}$ by the ratio of the signal variance to the signal-plus-noise variance. The signal variance in the feature image is captured by the training data in $\mathbf{FF}^T - \sigma_{cn}^2\mathbf{I}$. The noise variance in the unlabeled image data may be different than that in the training data (due to mismatched training/test conditions). So, we use $\mathbf{R}_{fn}$ to refer to the covariance of the noise in the unlabeled image data. (If the training and test conditions are matched, then $\mathbf{R}_{fn} = \sigma_{cn}^2\mathbf{I}$.) So, the feature-image signal-to-signal+noise ratio, in the manifold subspace, is

$$\begin{aligned} &(\mathbf{Q_F^T}(\mathbf{FF}^T - \sigma_{cn}^2\mathbf{I})\mathbf{Q_F} + \mathbf{Q_F^T}\mathbf{R}_{fn}\mathbf{Q_F})^{-1}\mathbf{Q_F^T}(\mathbf{FF}^T - \sigma_{cn}^2\mathbf{I})\mathbf{Q_F} \\ &= (\mathbf{I} + \mathbf{N})^{-1} \end{aligned}$$

where $\mathbf{N} = (\mathbf{Q_F^T}(\mathbf{FF}^T - \sigma_{cn}^2\mathbf{I})\mathbf{Q_F})^{-1}(\mathbf{Q_F^T}\mathbf{R}_{fn}\mathbf{Q_F})$ is the noise-to-signal ratio. Combining this with Equation 8 gives:

$$\hat{p} = \mathbf{Q_P}(\Sigma_\mathbf{P}\mathbf{X}_{QQ}\Sigma_\mathbf{F}^*)\mathbf{Q_F^T}(f - \bar{f}) + \bar{p} \qquad 9$$

where $\Sigma_\mathbf{F}^* = (\mathbf{I} + \mathbf{N})^{-1}\Sigma_\mathbf{F}^{-1}$ is a modified inverse of $\Sigma_\mathbf{F}$. Given the low sensitivity of $\Sigma_\mathbf{F}^*$ to the details of $\mathbf{N}$, it can also be approximated by $\Sigma_\mathbf{F}^* \approx (\mathbf{I} + \mathrm{diag}(\mathbf{N}))^{-1}\Sigma_\mathbf{F}^{-1}$. The closer the training and test conditions are to being matched, the better this final approximation is.

Equation 9, along with noise-level normalization [10] prior to the analysis, is what is used to get the results shown in Figure 1 and the results discussed in the next section.

## 4 Results

A training database was formed from images of seven people, starting with five original images of each person. These images were marked with 235 control points: 56 around the outline of the head, face, and ears; 29 each around the left and right eyes, irises and eyebrows; 31 around the nose and nostrils; and 90 around the boundaries of the lips, teeth and gums, and on the "smile lines". These control points were grouped into 18 features (using K-means clustering on the average and variance of their separation distances). For each of these clusters in each of the images, the corresponding feature location was taken to be the median (x,y) values of the control point locations in the cluster. For the sake of simplicity, the dimensions of each feature's subimage were specified manually: this required only 18 subimage specifications (one for each feature cluster). The original training database of 35 images was extended using automatic morphing to create one in-between image for each same-person pair in the original database. These in-between images did not require any additional manual labeling, since their control point locations can be computed directly from the originals. All the images were also flipped horizontally and added to the database. In this way, a database of 2*(7*(5*4)+35)=350 images was formed.

The database of 350 grayscale images were then amplitude normalized to improve the match between their histograms. The normalizing function for each image was a single affine function applied to all of its grayscale values. The affine functions were determined by first establishing a "target histogram". The final grayscale affine normalization for each image was then computed using a least-squares fit between the image's center subimage and the target histogram. The same approach was taken to normalizing the test data discussed below.

Component-wise noise-level normalization was also completed prior to the principal components analysis on the coupled data. The noise level in each image dimension was estimated from its average value, assuming that the noise variance is proportional to the average pixel value, above some lower bound. The noise variance in the control point locations was assumed to be one to four pixels, depending on how directly visible the point was in most images (four pixels for the top of the head and the gum lines; two pixels for the top and bottom of the iris and the top of the lower teeth; one pixel for everything else).

The performance in placing control points, given the correct feature location, was tested on the same set of images as were in the training database. The feature location was, again, the median control-point (x,y) values for the feature cluster. The average error in control point location (relative to the marked location) was less than one pixel (0.8 pixels). The maximum error was16 pixels, as shown in Figure 2. These results are obviously best-case results, since the test data was included in the training data and since the correct feature locations were given.

The performance in locating features and placing control points was tested on a separate set of images, showing new shots of the people shown in the training set. Eigenfeatures were used for feature location. The training and test data was normalized prior to analysis for eigen-features, using the same methods described above. Spatial dependencies between features were captured and exploited through conditional distributions. The average error was 1.0 pixel in feature location and 1.5 pixels in control point location. Examples are shown in Figure 3.

## 5 Conclusions

Eigen-points estimates the image-plane locations of fiduciary points on an objects. By estimating multiple locations simultaneously, eigen-points exploits the inter-dependence between these locations. This is done by associating neighboring, inter-dependent control-points with a model of the local appearance. The model of local appearance is used to find the feature in new unlabeled images. Control-point locations are then estimated from the appearance of this feature in the unlabeled image. The estimation is done using an affine manifold model of the coupling between the local appearance and the local shape.

Some advantages of eigen-points for estimating control point locations are:

- the solution is non-iterative;
- the models are derived from examples and labeling accuracy can be improved by adding more training data;
- the models are aimed specifically at recovering shape from image appearance, instead of being pure shape or pure appearance models;
- the estimation equations account for noise in the training data and the unlabeled images; and
- the estimation equations allow for uncertainty in the distribution and dependencies within these noise sources.

Future work should include:

- developing higher-order or piecewise affine models, beyond the simple affine model currently used;
- exploiting the inter-dependencies between feature locations in a more disciplined manner;
- exploiting the inter-dependencies between control-point locations for control points associated with different features; and
- improving the feature definition process.

## References

[1] C. Bregler, S. Omohundro, M. Covell, M. Slaney, S. Ahmad, D. Forsyth, J. Feldman, "Probabilistic Models of Verbal and Body Gestures," to appear in *Computer Vision in Man-Machine Interfaces* (R. Cipolla and A.Pentland, eds), Cambridge University Press, 1996.

[2] D. Beymer, T. Poggio, "Face Recognition from One Example View," MIT AI Memo 1536, 1995.

[3] M. Covell, M. Withgott, "Spanning the gap between motion estimation and morphing," *Proc IEEE International Conference on Acoustics, Speech and Signal Processing,* 1994

[4] B. Moghaddam, A. Pentland, "Maximum likelihood detection of faces and hands," *Proc International Workshop on Automatic Face and Gesture Recognition,* 1995.

[5] M. Covell, "Autocorrespondence: Feature-based Match Estimation and Image Metamorphosis," *Proc IEEE International Conference on Systems, Man and Cybernetics,* 1995.

[6] M. Kass, A. Witkin, D. Terzopoulous, "Snakes, Active Contour Models." *Proc IEEE International Conference on Computer Vision,* 1987.

[7] C. Bregler and S. Omohundro, "Surface Learning with Applications to Lipreading," *Neural Information Processing Systems,* 1994.

[8] D. Beymer, "Vectorizing Face Images by Interleaving Shape and Texture Computations," MIT AI Memo 1537, 1995.

[9] A. Lanitis, C.J. Taylor, T.F. Cootes, "A Unified Approach to Coding and Interpreting Face Images," *Proc International Conference on Computer Vision,* 1995.

[10] G. Golub, C. Van Loan, *Matrix Computations,* Johns Hopkins University Press, 1989.

## ACKNOWLEDGEMENTS

I would like to thank Malcolm Slaney of Interval Research Corporation for his comments both on this paper and on the underlying research and A. Lanitis of University of Manchester for the use of his database of face images.

Figure 2: Examples of re-labeled training data.

The images that were used as training data were themselves re-labeled, using the coupled models which were derived from them. The new estimates of control point locations were compared to the original values used in training. The new estimates are shown in white; the original training data in black. The average location error on this set of inputs was 0.8 pixels.

This image shows the worst errors, with offsets of 16 pixels at the top of the head. Other locations where errors tended to occur were at the gum lines and at the top of the forehead. It is not clear how much of this error is due to poor training data (i.e. inconsistent original labeling) and how much is due to the reduced "compliance" of the manifold models at these control points. The compliance was effectively reduced by the increased noise variance estimates for these locations. Another possible source of error is an incorrect model of the way in which the image noise varies with amplitude.

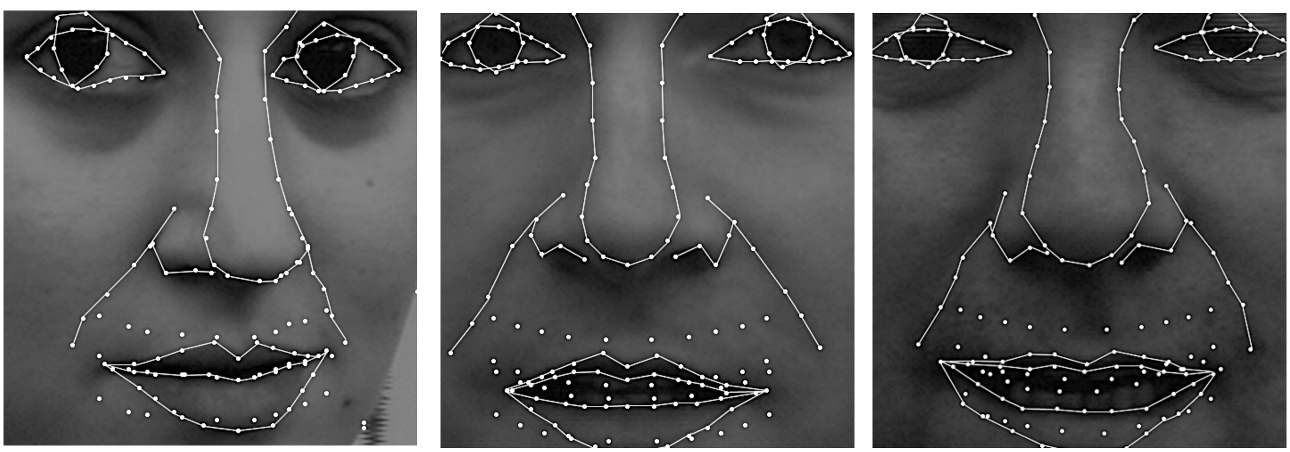

Figure 3: Labeling images that were not in the training data base.

These images, along with others, were used as a disjoint testing set: none were included in the training database. (Other images of these same people were included in the training database.) Labeling (using eigen-features for feature location) averaged 1 pixel in feature-location error and 1.5 pixels in control-point-location error.

Lines are drawn in the images between control-point locations to simplify their interpretation. Control points around the iris are meant to mark its outline, even when occluded by the upper or lower lid. Similarly, control points on the teeth are meant to mark their boundaries, even when occluded by the lips. (The disconnected control points in the mouth area were estimated as the top and bottom of the upper and lower teeth.)

The errors in the left image can be seen most clearly at: the bottoms and tops of the irises, on the top right eyelid, on the upper boundary of the top lip and along the right smile line.

The errors in the center image can be seen most clearly at: the bottom of the center part of the nose, the bridge of the nose, and the outline of the right nostril.

The errors in the right image can be seen most clearly at: the tops and sides of both irises, the inside and outside boundaries of both lips, the right side of the nose, and the left smile line.