# Creating Invariance To “Nuisance Parameters” in Face Recognition

Simon J.D. Prince and James H. Elder
York University
Centre for Vision Research
Toronto, Ontario
{prince, elder}@elderlab.yorku.ca

## Abstract

*A major goal for face recognition is to identify faces where the pose of the probe is different from the stored face. Typical feature vectors vary more with pose than with identity, leading to very poor recognition performance. We propose a non-linear many-to-one mapping from a conventional feature space to a new space constructed so that each individual has a unique feature vector regardless of pose. Training data is used to implicitly parameterize the position of the multi-dimensional face manifold by pose. We introduce a co-ordinate transform which depends on the position on the manifold. This transform is chosen so that different poses of the same face are mapped to the same feature vector. The same approach is applied to illumination changes. We investigate different methods for creating features which are invariant to both pose and illumination. We provide a metric to assess the discriminability of the resulting features. Our technique increases the discriminability of faces under unknown pose and lighting compared to contemporary methods.*

## 1. Introduction

In face recognition, there is commonly only one example of an individual in the database. Recognition algorithms extract a feature vector from the probe image and search the database for the face with the closest vector. Most work in the field has revolved around selecting optimal feature sets for this process. The dominant paradigm is the “appearance based” approach in which weighted sums of pixel values are used as features on which to base the recognition decision. Turk and Pentland [9] used principal components analysis (PCA) to model image space as a multidimensional Gaussian and selected the projections onto the largest eigenvectors. Other work has used more optimal linear weighted pixel sums, or similar non-linear techniques [1, 6].

One of the greatest challenges for these methods is to recognize faces across different poses and illuminations. In this paper we address the worst case scenario in which there is only one instance of each individual in a large database and the probe image is taken with a very different pose and under very different illumination. Under these circumstances, most methods fail, since the extracted feature vector changes with these “distractor” variables. Indeed the variation attributable to these factors may dwarf the variation due to differences in identity. Our strategy is to create a many-to-one non-linear mapping from a conventional feature space to a new space which is invariant to pose and illumination.

## 2. Previous Work

The simplest approach to making recognition robust to a distractor variable is to remove all feature measurements that co-vary strongly with this variable. For example it has been suggested that the first few eigenvectors can be discarded as they mainly respond to illuminant information. A more sophisticated approach is to measure the amount of signal (inter-personal variation) and noise (intra-personal variation) along each dimension and select features for which the signal:noise ratio is optimal [1]. A problem with these approaches is that the discarded dimensions may contain a significant portion of the signal and their elimination ultimately impedes recognition performance. For example, features that are linearly invariant to horizontal translation would correspond to images where there was no variation in the horizontal dimension. In removing components that vary under horizontal translation we have also removed vital information needed for face recognition.

One obvious method to generalize across distractor variables is to record each subject in the database at each possible value of the variable, and use an appearance based model for each [7]. Another approach is to use several photos to create a 3D model of the head which can then be re-rendered at any given pose to compare with given probe [3, 10] . The disadvantage of these methods is that they require extensive recording and the cooperation of the subject.

Several previous studies have presented algorithms which take a single probe image at one pose and attempt to match to a single test image at a different pose. One approach is to create a full 3D head model based on just one image, and then re-render this at all possible poses before searching the database [8, 2]. This approach is feasible, but the computation required is very intensive. The most similar work to our approach is the work on "eigen-light fields" by Gross et al. [5]. They treat matching as a missing data problem - the single test and probe images are assumed to be parts of larger data vector containing the face viewed from all poses. The missing information is estimated from the visible data, using knowledge of the covariance structure. The complete vector is used for the matching process.

The emphasis in these algorithms is on creating a model which can predict how a given face will appear under different conditions. Our algorithm takes a different viewpoint. We aim to construct a single feature *which does not vary with pose or illumination*. This seems a natural formulation for a recognition task. Our approach is non-linear so that signal is preserved but the unwanted variation removed. This is a general learning technique which can produce features which are invariant to any dimension as long as there are sufficient training examples for which this parameter is known. This includes both continuous dimensions such as pose or age and discrete dimensions such as the presence of glasses or facial hair. Several invariances can be created *sequentially* (i.e. pose invariant features are created, which are then also made invariant to illumination) or *jointly* (features are created which are simultaneously invariant to both dimensions). It is not necessary to capture training subjects over the complete set of possible poses which makes collecting training data relatively easy.

## 3. Creating Invariant Features

Although we are concerned with 3D pose and illumination invariance, we first demonstrate our ideas using the simpler case of face recognition where test and probe faces have an unknown in-plane orientation (see Figure 1). Our task is to take a single probe face at an arbitrary orientation, and match it to a test face in the database with a different orientation. Hence, we wish to make recognition invariant to the irrelevant orientation. We term the orientation a *distractor* or *nuisance* variable, and denote its value by $\theta$.

In order to create orientation invariant features, we require a training database with two important characteristics. First, the value of the distractor variable must be known for each member. Second, each individual in the training database appears with at least two different values of the distractor variable. Together these characteristics provide sufficient information to (i) learn how to estimate the distractor variable when it is not known and (ii) learn how images of the face at different distractor values are related.

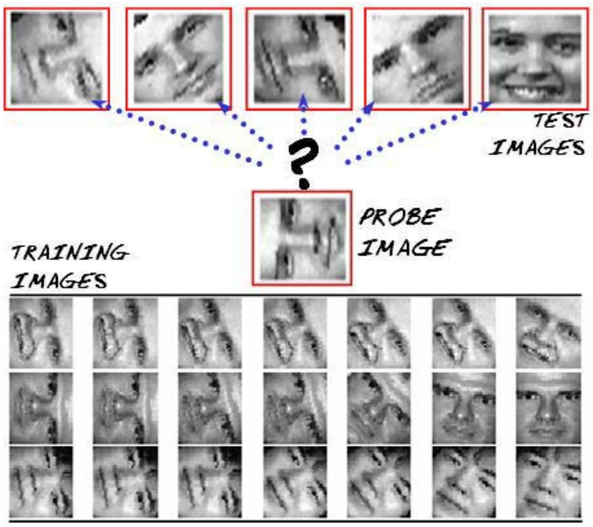

**Figure 1. We first consider face recognition in a test database of faces at an arbitrary (and different) in-plane orientation. The probe face matches a single face from the database, but is at a different orientation. Our algorithm uses a training database, containing several faces each viewed a number of angles.**

The training database consisted of 20 instances each of 200 faces taken from the BIO ID database [11]. The positions of the eyes and septum were identified by hand and defined an affine transform such that these three points were aligned for all faces. We introduced a random in-plane orientation of $[-90, 90]$. Faces were cropped to $32 \times 32$ pixels and flattened to grayscale. The lower part of Figure 1 shows examples of these training images (all with $\theta \leq 0^o$).

The grayscale values of each of the 4000 training faces, were concatenated into column vectors $\mathbf{p}$, each of length $32 \times 32 = 1024$. For each vector, the mean value is removed, and the standard deviation normalized to unity. These vectors form the $1024 \times 4000$ matrix $\mathbf{P}$. We apply principal components analysis to reduce the dimensionality of the input vectors. The principal components of the dataset are calculated by performing an eigen-decomposition of the covariance matrix, $\mathbf{P}^T\mathbf{P}$, so that $\mathbf{P}^T\mathbf{P} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$ where $\mathbf{V}$ is an orthogonal matrix of eigenvectors and $\mathbf{\Sigma}$ is a diagonal matrix consisting of the eigenvalues. We truncate the spectral composition of the matrix by discarding all but the first 100 eigenvectors and eigenvalues, to form new matrices $\mathbf{\Sigma}'$ and $\mathbf{V}'$. We then calculate initial feature vectors $\mathbf{x} = \mathbf{V}'\mathbf{p}$ by projecting the original vectors onto this reduced space.

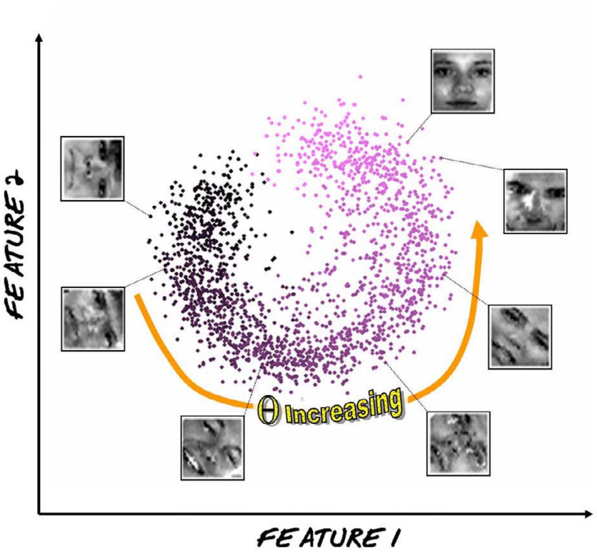

**Figure 2. Scatterplot of first two features for 2000 faces (all with $\theta \leq 0^o$). Points are color-coded as a function of the distractor orientation variable, $\theta$ so that dark points represent horizontal faces and light values vertical. The position in feature space changes systematically and smoothly as a function of $\theta$.**

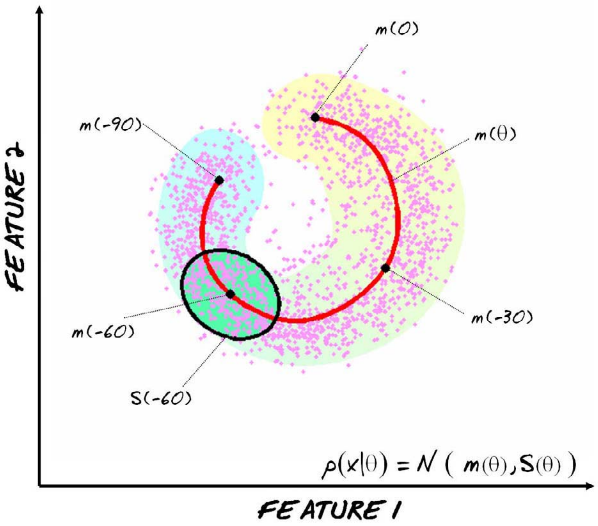

**Figure 3. The shape of the manifold is parameterized as a function of the distractor variable, $\theta$. For each value of $\theta$ we calculate the mean feature vector (red line). We also calculate the covariance (ellipse presents two standard deviations for $\theta = -60^o$). Shaded region represents the region within two standard deviations of the mean at some $\theta$.**

### 3.1. Modelling Manifold Shape

For most choices of feature vector, the majority of positions in the vector space are unlikely to have been generated by face images. The subspace to which faces commonly project is termed the face manifold. In general this manifold is a complex non-linear probabilistic region tracing through multi-dimensional space. The mean position of this manifold changes systematically as a function of the distractor variable, $\theta$. Figure 2 plots the first two components of the feature vectors $\mathbf{x}_i$ for 2000 points with $\theta \leq 0^o$. Each point is color coded by the distractor variable, $\theta$, so that dark points have orientations close to $-90^o$ and light points have orientations near $0^o$. Notice that there is a systematic and smooth change in the mean position of the manifold as a function of the distractor variable $\theta$ so that near-horizontal faces are represented mainly in the bottom left-quadrant and near-vertical faces in the top-right quadrant.

We aim to implicitly parameterize the shape of the face manifold as a function of the nuisance variable, $\theta$. We model the shape of the manifold at a given orientation, $\theta$ as a multivariate Gaussian, represented by a mean vector $\mathbf{m}(\theta)$ and a covariance matrix, $\mathbf{S}(\theta)$, so that the probability of generating a vector $\mathbf{x}$ is given by

$$p(\mathbf{x}|\theta) = \frac{1}{\sqrt{2\pi}|\mathbf{S}(\theta)|^{\frac{d}{2}}} \exp\left[-\tfrac{1}{2}(\mathbf{x}-\mathbf{m}(\theta))^T\mathbf{S^{-1}}(\mathbf{x}-\mathbf{m}(\theta))\right] \quad (1)$$

Figure 3 replots the first two features and shows the value of the mean, $\mathbf{m}$ parameterized by the distractor variable, $\theta$ (red line). The covariance is depicted for the value $\theta = -60$ by an ellipse representing a Mahalanobis distance of 2. The shape of the manifold in this two-dimensional subspace is visualized by displaying the envelope of all of these covariance ellipses (shaded area). The mean at each value of $\theta$ is calculated by Gaussian weighting the data points by their $\theta$-proximity.

$$\mathbf{m}(\theta) = \frac{\sum_{i=1}^{n} \mathbf{x}_i \exp\left[-(\theta_i-\theta)^2/2\sigma_m^2\right]}{\sum_{i=1}^{n} \exp\left[-(\theta_i-\theta)^2/2\sigma_m^2\right]} \quad (2)$$

where $\mathbf{x}_i$ is the i'th training data vector, $\theta_i$ is the associated value of the distractor variable, $\sigma_m$ is a smoothing parameter and $n$ is the total number of training vectors. This weighted/regularized estimate has two advantages. First, it ensures that the mean and the covariance of the manifold change smoothly with $\theta$. Second, it ensures that the estimates can always be calculated, even when data around this

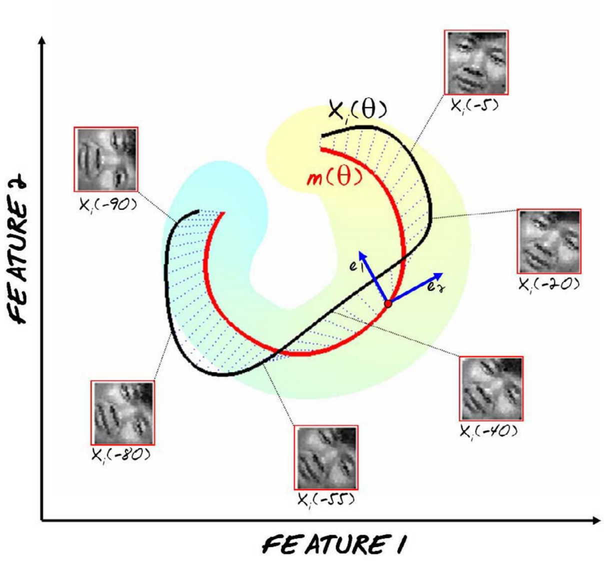

**Figure 4. We plot the feature vector for a single face as a function of orientation. Notice that while it follows the general shape of the manifold (shaded area), it's position relative to the manifold varies. For some values of the distractor variable it is above the manifold, whereas for others it is to the left or right. The blue axes represent a local change in co-ordinates for a given value of $\theta$.**

value of $\theta$ is sparse. A similar formulation is used to calculate a smoothly varying covariance function.

### 3.2. Creating Invariance

Figure 4 depicts a single face projected into this two dimensional subspace, also parameterized by the distractor variable $\theta$. As this example face varies from horizontal to vertical the projection, $\mathbf{x}(\theta)$ (black line) moves around the manifold (shaded area). On average the movement follows that of the mean, $\mathbf{m}(\theta)$ of the manifold (red line), but at any given value of $\theta$ the position relative to the mean changes. Let us consider the vector $\mathbf{y}(\theta)$ from the mean at a given value of the distractor variable to the feature vector at the same value of the distractor variable:

$$\mathbf{y}(\theta) = \mathbf{x}(\theta) - \mathbf{m}(\theta) \tag{3}$$

The vectors $\mathbf{y}$ are depicted as the "spokes" connecting the mean (red line) to the path of the particular face (black line) in Figure 4. Notice that this vector is not constant as a function of the distractor variable $\theta$. At $\theta = -90$, $\mathbf{x}$ is to the left (lower value of feature 1) of the mean $\mathbf{m}$, whereas at $\theta = -20$, $\mathbf{x}$ is to the right (higher value of feature 1) of the mean. In general there is no particular reason why the feature vector $\mathbf{y}$ should be invariant to the distractor variable.

The core of our method is to define a local transformation $f$ which depends on $\theta$ and acts on $\mathbf{y}_i(\theta)$ to form a new vector $\mathbf{c}_i$ which is always constant:

$$f(\mathbf{y_i}(\theta), \theta) = \mathbf{c_i} \qquad \forall \quad \theta, i \tag{4}$$

The transformation is defined so that it produces a different constant vector, $\mathbf{c}_i$ for each individual, $i$ in the database.

We might consider a number of different forms for the function $f(\theta)$, but a reasonable starting point is to consider a rigid rotation, $\mathbf{R}(\theta)$ around the local origin, $\mathbf{m}(\theta)$. In other words, we define a change in co-ordinate system that depends on $\theta$. The local co-ordinates change with $\theta$ so that $\mathbf{y}(\theta)$ remains constant when expressed in the new frame.

In Figure 4, the local co-ordinate frame is depicted by a set of blue axes for $\theta(-30)$. These axes rotate as they traverse along the mean $\mathbf{m}(\theta)$ (red line). The rotation is chosen so that for every $\theta$ the local value of $\mathbf{y}$ (black spoke from the current point on the manifold mean) is a constant vector in the local frame of reference defined by the axes. The problem of calculating invariant feature vectors is now re-cast as finding the rotation $\mathbf{R}$ as a function of the distractor variable, $\theta$, and the constant vectors, $\mathbf{c}_i$.

## 4. Invariance To In-Plane Orientation

In this section we provide a concrete example of our technique. We quantize the distractor dimension into $n_k$ evenly spaced bins $\theta_k$, and represent the function in each bin by the rotation matrix $\mathbf{R}_k$. Consider the case where we have $n_f$ training feature vectors of dimension $n_d$ from each of $n_i$ individuals. This provides a total of $n_y = n_f \times n_i$ feature vectors $\mathbf{y}$, each of which has an associated value of the distractor variable, $\theta$. Let $\mathbf{Y}$ be the $n_d \times n_y$ matrix containing these column vectors. Similarly, let $\mathbf{C}$ be a $n_d \times n_i$ matrix where each column represents the invariant feature associated with individual $i$. We seek:

$$\underset{\mathbf{C},\mathbf{R}_k}{\arg\min}\, \mathbf{E} = \sum_{k=0}^{n_k} ||\mathbf{R}_k\mathbf{Y}\mathbf{W}_k - \mathbf{C}\mathbf{T}\mathbf{W}_k||_F \tag{5}$$

where the operator $||.||_F$ denotes the Frobenius norm. $\mathbf{W}_k$ are diagonal $n_y \times n_y$ weight matrices in which the entries depend on the distance between the current distractor variable $\theta_k$ and the distractor value for each of the $n_y$ training vectors, $\theta_y$. These weights are calculated as in Equation 2. The $n_i \times n_y$ identification matrix $\mathbf{T}$ contains only zeros and ones and indicates which feature vectors in $\mathbf{Y}$ belong to which individual (and hence to which invariant vector in $\mathbf{C}$). Assuming that the vectors in $\mathbf{Y}$ are ordered by individual, it will have the form:

$$\mathbf{T} = \begin{bmatrix} 1 & 1\cdots1 & & & \cdots & & \\ & & 1 & 1\cdots1 & \cdots & & \\ \vdots & & \vdots & & \ddots & & \vdots \\ & & & & \cdots & 1 & 1\cdots1 \end{bmatrix} \tag{6}$$

Our approach is to alternately minimize the error, $E$ with respect to the functions $\mathbf{R}_k$ and the constant vectors, $\mathbf{c}_i$. The algorithm is as follows:

**Algorithm 1** Learn Invariant Mapping

```
$\mathbf{R}_k \Leftarrow \mathbf{I}$
$\mathbf{C} \Leftarrow \frac{1}{n_f}\mathbf{Y}\mathbf{T}^T$
while $\Delta E \geq \Delta E_{tol}$ do
    for all $k$ do
        $\mathbf{R}_k \Leftarrow$ PROCUSTES$(\mathbf{Y}\mathbf{W}_k, \mathbf{C}\mathbf{T}\mathbf{W}_k)$
    end for
    $\mathbf{C} \Leftarrow \frac{1}{n_f n_k}\sum_{k=1}^{n_k}\mathbf{R}_k\mathbf{Y}\mathbf{W}_k\mathbf{T}^T$
    $E \Leftarrow \sum_{k=0}^{n_k}\|\mathbf{R}_k\mathbf{Y}\mathbf{W}_k - \mathbf{C}\mathbf{T}\mathbf{W}_k\|_F$
end while
```

The function $\mathbf{R} =$ PROCRUSTES$(\mathbf{A}, \mathbf{B})$ returns the solution to the orthogonal Procrustes problem (see [4]), which is the rotation matrix $\mathbf{R}$ which most closely fulfils $\mathbf{RA} = \mathbf{B}$. The result of this algorithm is a set of co-ordinate transformations $\mathbf{R}_k$ which vary as we iterate along the quantized distractor variable, $\theta_k$. The Gaussian weighting matrix ensures that the transformation changes smoothly. The invariant features of the training faces, $\mathbf{C}$ may be discarded.

## 4.1. Calculating Invariant Features

To calculate the invariant feature vector associated with a new face, we first project into the reduced eigenspace, to form the data vector $\mathbf{x}$. Then we estimate the value of the distractor variable using Bayes rule.

$$p(\theta_k|\mathbf{x}) = \frac{p(\mathbf{x}|\theta_k)p(\theta_k)}{\sum_{l=1}^{n_k} p(\mathbf{x}|\theta_l)p(\theta_l)} \tag{7}$$

where $p(\mathbf{x}|\theta_k)$ is given by Equation 1. For simplicity we select the maximum a posteriori (MAP) value of k. We calculate the vector $\mathbf{y}$ relative to the local mean, $\mathbf{m}_k$ and transform by the local transformation $\mathbf{R}_k$. The complete algorithm is:

**Algorithm 2** Calculate MAP Invariant Feature

```
$\mathbf{x} \Leftarrow \mathbf{V}'\mathbf{p}$
$k \Leftarrow \arg\max_k\{p(\theta_k|\mathbf{x})\}$
$\mathbf{y} \Leftarrow \mathbf{x} - \mathbf{m}_k$
$\hat{\mathbf{c}} \Leftarrow \mathbf{R}_k\mathbf{y}$
```

## 4.2. Results

We calculated the invariant mapping for the in-plane orientation example using $n_f = 20$ images from each of $n_i = 200$ individuals. We quantized the distractor variable (orientation) into $n_k = 181$ bins in $1^o$ increments from $-90$ to $+90^o$. We formed a test database containing one example each of 200 different faces. Each test face had a different random orientation. We calculated the pose invariant vector for each face, $\mathbf{c}$. We used 200 probe faces, which came from the same individuals as the test dataset, but were at a different random orientation. We calculate the pose invariant vectors for each of the probe faces. We use nearest neighbor matching in the 100 dimensional invariant space to identify which test face is most similar to each probe face.

For almost all cases, the MAP estimate of the distractor variable was very close to the true value, with a mean error of the order of $2^o$. Recognition performance was very good. For 192 out of 200 probe faces, the first choice match was correct. Visual inspection of the failures indicated that the chosen match was very similar to the probe face. This may partly reflect the limitations of our low resolution $32 \times 32$ grayscale input images.

## 4.3. Visualizing Invariant Vectors

In this section we describe a method for visualizing the invariant features. Consider a given face feature vector, $\mathbf{p_i}$, which maps to the invariant feature $\mathbf{c_i}$. We aim to visualize all other face feature vectors which project to the same invariant feature $\mathbf{c_i}$. Intuitively, these faces should look like the original face under all possible values of the distractor variable, $\theta$.

**Algorithm 3** Visualize Invariant Feature

```
$\mathbf{c} \Leftarrow INVARIANT(\mathbf{p})$
for all $k$ do
    $\mathbf{y}_k \Leftarrow \mathbf{R}_k^T\mathbf{c}$
    $\mathbf{x}_k \Leftarrow \mathbf{m}_k + \mathbf{y}_k$
    $\mathbf{p}_k = \mathbf{V}'^T\mathbf{x}_k$
end for
```

where INVARIANT corresponds to Algorithm 2.

Figure 5 shows the results of this process for three faces from the test database (top). The first row consists of a number of face images which map to exactly the same vector $\mathbf{c}_i$ as the first face. The second and third rows correspond to the second and third faces respectively. In each case, the faces look like rotated versions of each other. The mapping of these images to the same vector constitutes invariance to in-plane orientation. Note that this is not mere interpolation since only a single instance of each of these faces was used, and even this was not in the training database.

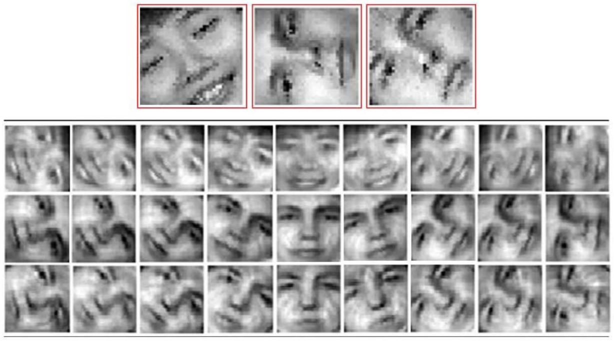

Figure 5. An invariant feature vector was calculated for each of the three faces at the top of the figure. The three rows of faces underneath correspond to other faces which project to the exactly the same vector.

## 4.4. Functional Forms for Transformation

At the heart of the technique for calculating invariant feature vectors is an orthogonal transormation $\mathbf{R}_k$ which varies smoothly as a function of the distractor variable index, $k$. This corresponds to a change of co-ordinates relative to the local manifold mean, as depicted in Figure 4. There is no reason why this transformations should be limited to pure rotation. However, we do expect mappings to be smooth since faces which are similar at one value of the distractor variable will tend to look similar at other values.

In order to incorporate a more sophisticated function $f(\mathbf{y})$ we make three substitutions. First, the forward model, $\mathbf{R}_k\mathbf{y}$ is replaced by the new function $f_k(\mathbf{y})$. Second, PROCRUSTES$(A, B)$ is replaced by an estimation procedure for the parameters of $f_k$ which aims to minimize $\|f_k(\mathbf{A}) - \mathbf{B}\|_F$. Finally, in order to visualize the invariant feature, we replace $\mathbf{R}^T$ in Algorithm 3 by the function inverse $f^{-1}(\mathbf{c})$. This final substitution is not required for recognition performance, so non-unique functions are permitted. We have experimented with using a general linear transformation instead of a rotation, and this improves recognition rates to 196/200.

## 5. Invariance to 3D Pose and Lighting

We now consider the more challenging case of face recognition under different poses and lighting conditions. We restrict our discussion to variations in horizontal pose and horizontal lighting direction, although the same ideas can be applied to the more general problem if enough data is gathered. We have collected a database in which the lighting direction, $\phi$ varied from $-90^o$ to $+90^o$ in $30^o$ increments. The camera pose, $\theta$ varied from $-90^o$ to $+90^o$ for each of these lighting directions (see Figure 6). Ten unique positions on the face were identified by hand. Combinations of the features were used to extract pixel regions corresponding to the nose, eyes and mouth of the object. Pixels from these regions were concatenated to create the initial feature vectors, $\mathbf{p}$. The pixel data was projected onto the first 20 eigenvectors to form a reduced feature vector, $\mathbf{x}$.

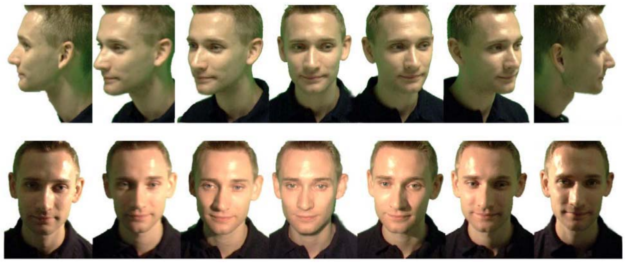

Figure 6. Training faces were captured with systematically varying pose (top) and illumination (bottom). All combinations of pose and illumination were captured.

## 5.1. Quantifying Invariance

We propose a measure of the feature discriminability, which quantifies the success of our method, even when the test database is small, and recognition rates are close to 100%. We compare within-class to between-class variance. We might expect our technique to reduce the within-class variance since the feature vectors are now relatively constant as a function of 3D pose and illumination. Let the $n_d \times n_y$ matrix $\mathbf{Y}$ denote a feature matrix where each column represents one $n_d$ dimensional feature vector. This might contain the features before or after invariance is introduced. We assume that this matrix contains $n_f$ features from $n_i$ individuals so that $n_y = n_f \times n_i$. Let $\mathbf{T}$ represent the $n_i \times n_y$ identification matrix mapping individuals to data vectors (see Equation 6). If we assume that the mean feature vector, $\mathbf{y}$ is zero, then the within- and between-class variances, $\mathbf{S}_W$ and $\mathbf{S}_B$ are as follows:

$$\mathbf{S}_W = \frac{1}{n_f n_i}(\mathbf{Y} - \frac{1}{n_f}\mathbf{Y}\mathbf{T}^T\mathbf{T})(\mathbf{Y} - \frac{1}{n_f}\mathbf{Y}\mathbf{T}^T\mathbf{T})^T \quad (8)$$

$$\mathbf{S}_B = \frac{1}{n_f^2 n_i}\mathbf{Y}\mathbf{T}^T\mathbf{T}\mathbf{Y}^T \quad (9)$$

We define our measure of discriminability as:

$$D = \frac{\text{Trace}(\mathbf{S}_B)}{\text{Trace}(\mathbf{S}_B) + \text{Trace}(\mathbf{S}_W)} \quad (10)$$

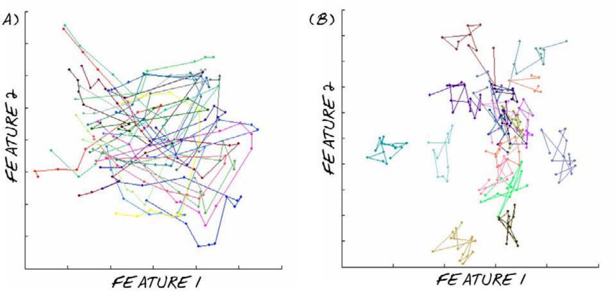

**Figure 7. (A) Scatterplot of first 2 dimensions of original space. Each continuous line represents data from one subject, ordered by horizontal pose. Variation due to pose change is much greater than variation due to identity. (B) First 2 components of the invariant vector. There is little variation with pose, but different individuals (line segments) are separated.**

This index compares the between-subject variance to the within-subject variance. If the between-subject variance dominates the index will be near one. If the within-subject variance dominates, the index will be near zero.

### 5.2. Pose Only Results

We first test our algorithm on a subset of data in which only pose was varied. The training dataset consisted of 225 images of 20 different individuals at random poses, $\theta$, between $\pm 90^o$. Test data consisted of 235 images of 18 different individuals also at random orientations, but with the same lighting conditions. The pose dimension was subsampled into 181 distinct bins. The weighting standard deviation was set to $40^o$. The pose of the test samples was assumed known to make comparison with [5] fair. The transformations $\mathbf{R}_k$ were linear.

Figure 7 (A) shows a scatter plot of the first two components of the orignal vectors $\mathbf{X}$. Connected points originate from the same subject and are plotted in order of $\theta$. In Figure 7 (B) the first two components of the invariant features $\mathbf{C}$ are shown. Points originating from the same individual project to nearly the same place. Figure 8 (A) plots the discriminability index in the original and invariant spaces. The initial eigenvectors, $\mathbf{x}$ are dominated by within-subject variance ($D \leq 0.5$), whereas the invariant features, $\mathbf{c}$ are dominated by between subject variance ($D$ close to 1). We also plot gold standard discriminability. This was calculated by extracting features from several images of each subject in which pose and lighting were constant. The remaining within-subject variation is due to expression changes and variability in the feature extraction process. Finally, we plot the results of the algorithm of Gross et al. [5]. Here the pose was discretized into 8 bins. Each training subject contributed one image to each bin. Twenty "eigen-light field vectors" were calculated. The same test dataset was used to calculate the final feature vectors. The discriminability was less than for our algorithm.

We also compared these algorithms for using a subset of the FERET database. The training set consisted of 200 faces each viewed from 10 different poses in the range [-90, 90]. We tested each algorithm using 100 different faces across all pairs of poses from the set, using a nearest-neighbour classification algorithm. The eigen-light field algorithm produced features with a discriminability of 0.6395 and 54% recognition performance across pose. This is similar to result presented in [5] for weakly registered data. The invariant features produced by our algorithm had a discriminability of 0.8755. Recognition performance across pose was 61%. We conclude that our algorithm produces comparable or better results than that of [5].

### 5.3. Pose and Illumination Results

There are two distinct approaches to making the extracted feature vectors invariant to pose and lighting direction. In the *joint* approach, we model the manifold as a 2D surface and calculate a mapping $\mathbf{R}_{k,l}$, which depends on the two dimensional position on the manifold, $(\theta_k, \phi_l)$. If we quantize each dimension into $n_k$ bins, this technique requires us to estimate $(n_k)^d$ transformations. Hence, it may not be tractable in high dimensions unless the per-dimension sampling is considerably reduced. We term the second approach *sequential* invariance. Recall that our algorithm maps one feature vector $\mathbf{x}$ to another vector $\mathbf{c}$ of the same dimension where this feature is invariant to the distractor variable, $\theta$. We use this $\theta$-invariant feature $c$ as the input for a second process in which we make the vector invariant to a second distractor variable, $\phi$. This process can be repeated until the discriminability, no longer increases.

The pose dimension was subsampled into 31 distinct bins and the lighting dimension into 7 bins. The weighting standard deviation was $40^o$ for each dimension. The first 60 eigenvectors were used as the initial features.The training dataset consisted of a total of 1846 images of 20 individuals under different lighting and pose conditions. The test dataset consisted of 168 images of 18 different individuals. Figure 8 (B) shows the results of our algorithm on the pose and lighting dataset illustrated in Figure 6. The plot shows the discriminability, $D$ of the resulting data vectors for the original dataset, feature vectors invariant to pose alone, feature vectors invariant to lighting alone, the jointly invariant data vectors and the sequentially invariant feature vectors. Figure 8 (B) shows that our method significantly increases the discriminability of face vectors under unknown pose and

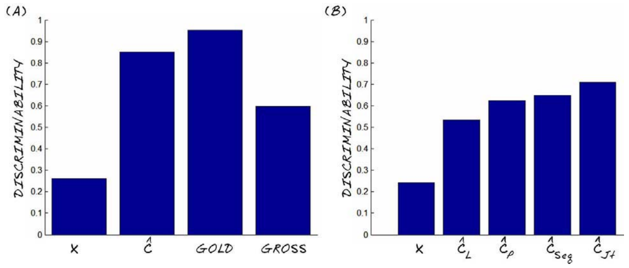

**Figure 8. (A) “Pose-Only” results. From L to R, Original features $\mathrm{x}$, Invariant Features, $\hat{\mathrm{c}}$, Gold Standard, Eigen-Light Fields approach of Gross et al. (B) “Pose and Illumination” results. From L to R, Original Features, $\mathrm{x}$, features invariant to illumination, $\hat{\mathrm{c}}_L$, invariant to pose $\hat{\mathrm{c}}_p$, both sequential $\hat{\mathrm{c}}_{seq}$, both joint $\hat{\mathrm{c}}_{jt}$**

lighting. The most optimal condition is to render features jointly invariant to pose and orientation.

## 6. Discussion

In this paper we have presented a supervised method for rendering face recognition features invariant to nuisance parameters such as pose and lighting. We have demonstrated the results on the toy-problem of in-plane orientation and for horizontal pose and lighting-direction. First, we estimate the value of the nuisance variable(s). We then define a transform upon the data which is contingent on the value of the nuisance variable and produces a invariant new vector. We present a technique for visualizing invariant vectors, by showing the set of the original vectors which project to it.

The most closely related approach to our work is the “eigen-light fields” of Gross et al. [5]. They utilize an extended feature vector in which the image data from several different poses is concatenated. The eigenvectors of this extended space are calculated and used as the input features. A given probe face is considered as an input vector with missing data (the other poses). The expected value of this missing data can be predicted based on the known data (the image at a given pose), and is determined by a linear transform determined by the data covariance.

This is similar to our formulation in which the pose-invariant vector $\mathbf{c}$ is related to the input vector $\mathbf{y}_k$ at a given value of pose ($\theta_k$) by a transformation, $\mathbf{R}_k$. However, our approach is potentially more expressive as this transformation may be an arbitrary function. Our approach has several advantages. First, we provide a coherent mechanism for estimating the (potentially unknown) value of the distractor variable. Second, we can apply more complicated transformations to model the variation as a function of the nuisance variable. Third, for comparable amounts of training data, our algorithm gives better performance. This may reflect the regularizing effect of weighting the data points based on their proximity in “nuisance space”. Fourth, our algorithm does not require training data to be coarsely binned, nor does it require complete training data for any given subject. Importantly, the philosophy of our approach differs. Gross et al. learn how images at one pose are related to images at another with a view to predicting the full light-field from one pose. Our approach aims to eliminate the pose variable from the feature vector entirely.

## References

[1] P.N. Belhumeur, J. Hespanha and D.J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 19, pp. 711-720, 1997.

[2] V. Blanz, S. Romdhani and T. Vetter, “Face identification across different poses and illumination with a 3D morphable model,” *Proc. 5’th Int’l Conf. Face and Gesture Recognition* pp. 202-207, 2002.

[3] A. Georghiades, P. Belhumeur and D. Kriegman, “From few to many: illumination cone models and face recognition under variable lighting and pose,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 129-139, 2001.

[4] G.H. Golub and C.F. Van Loan “*Matrix Computations*”, Johns Hopkins University Press, Baltimore, 1989.

[5] R. Gross, I. Matthews and S. Baker, “Appearance-Based Face Recognition and Light Fields.” *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 26, pp.449-465, 2004.

[6] M.H. Yang “Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods” *Proc. Fifth IEEE Int’l Conf. Automatic Face and Gesture Recognition (RGR ’02)*, pp. 215-220, 2002.

[7] A. Pentland, B. Moghaddam and T. Starner, “View-based and modular eigenspaces for face recognition,” *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994.

[8] S. Romdhani, V. Blanz and T.Vetter, “Face identification by fitting a 3D morphable model using linear shape and texture error functions,” *Proc. European Conference on Computer Vision*, 2002.

[9] M. Turk and A. Pentland, “Face Recognition using Eigenfaces,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp.586-591, 1991.

[10] W. Zhao and R. Chellappa, “SFS based view synthesis for robust face recognition,” *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp 285-292, 2002.

[11] http://www.humanscan.de/support/downloads/facedb.php.