Face View Synthesis Across Large Angles

Jiang Ni and Henry Schneiderman

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract. Pose variations, especially large out-of-plane rotations, make face recognition a difficult problem. In this paper, we propose an algorithm that uses a single input image to accurately synthesize an image of the person in a different pose. We represent the two poses by stacking their information (pixels or feature locations) in a combined feature space. A given test vector will consist of a known part corresponding to the input image and a missing part corresponding to the synthesized image. We then solve for the missing part by maximizing the test vector's probability. This approach combines the "distance-from-feature-space" and "distance-in-feature-space", and maximizes the test vector's probability by minimizing a weighted sum of these two distances. Our approach does not require either 3D training data or a 3D model, and does not require correspondence between different poses. The algorithm is computationally efficient, and only takes 4 - 5 seconds to generate a face. Experimental results show that our approach produces more accurate results than the commonly used linear-object-class approach. Such technique can help face recognition to overcome the pose variation problem.

1 Introduction

Face recognition applications often involve pose variations. The gallery may only have the faces under a specific pose, such as the frontal view, but the probe image may be captured under a random pose, sometimes with a large out-of-plane rotation. In order to do face recognition in this scenario, we need to synthesize the new view of the probe face, such that we can compare it with the gallery images.

According to the stereopsis theory in computer vision, to recover the precise 3D geometry of an object, we need at least three images of this object. This is why some approaches use multi-view images [6], or even video sequences [13], to synthesize new views. Although a single image is insufficient to recover the precise 3D geometry, machine learning techniques can apply prior knowledge onto this single image in order to synthesize new views. In particular, Blanz and Vetter pioneered a 3D algorithm based on fitting a 3D morphable model learned from many 3D training examples, to synthesize novel views from a single image [2,3,4].

The drawback of such a 3D approach is its large computational cost of 4.5 minutes for each fitting process [3,4]. This high computational cost limits the approach's applicability to real life applications, such as in airport security. Another drawback is the need for specialized 3D scanning hardware. For practical

W. Zhao, S. Gong, and X. Tang (Eds.): AMFG 2005, LNCS 3723, pp. 364–376, 2005.

systems, it would be more attractive to implement a solution using only 2D images. The linear-object-class method proposed by Vetter and Poggio [12] and its variation [7] hold the promise of such a method. However, our own experiments show that its performance for large out-of-plane rotations, such as 45° and 90° is not satisfactory. In particular, the predicted shapes exhibit significant distortion (Fig. 3, 4, 5).

Vetter and Poggio's linear-object-class method [12] solves a set of linear equations with missing data:

$$\begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix} \mathbf{y} = \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{pmatrix}, \tag{1}$$

where $\tilde{b}_2$ is the unknown pose and $\tilde{b}_1$ is the known pose of the test example. $\Phi_M = \begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix}$ is the training set (or vectors formed from a linear combination of the training set, i.e., PCA of the training set) containing the two poses. $\begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{pmatrix}$ is represented as a linear combination of the columns in $\begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix}$. The vector $\mathbf{y}$ contains the parameters describing the linear combination. The linear-object-class method solves for $\mathbf{y} = \arg\min \left\| \Phi_1 \cdot \mathbf{y} - \tilde{b}_1 \right\|^2$, then uses it to predict $\tilde{b}_2 = \Phi_2 \mathbf{y}$. (In the view synthesis problem for faces, shape and appearance are usually analyzed and predicted seperately.)

This method has been discussed in some related problems [8,1,9]. Hwang and Lee [8] use exactly the same method as above in predicting occluded parts of human faces. Black et al. [1] and Leonardis et al.[9] slightly modify the approach, by either excluding [9] or putting less weight [1] on some rows of Φ_1 that they assume are outliers.

We believe that the problems with the linear-object-class method lie with an incorrect assumption: there are no errors inherent in the solution for $\mathbf{y}$ in $\mathbf{y} = \arg\min \left\| \Phi_1 \cdot \mathbf{y} - \tilde{b}_1 \right\|^2$. However, as it is well known, there are measurement errors in the training data due to many factors. These errors will propagate into the solution for $\tilde{b}_2$, using the linear-object-class method. We can improve upon this solution with a probabilistic formulation. This formulation combines "distance-from-feature-space" (DFFS) and "distance-in-feature-space" (DIFS) [10], whereas the linear-object-class solution is purely based on DFFS. By considering DIFS, our method penalizes for points within the subspace, $\begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix}$, that have low probability. Our representation leads to solutions that have higher probability and, as we will show, significantly better empirical performance.

This paper is organized as follows. In Section 2, a probabilistic model combining DFFS and DIFS is introduced, and the solution for equation (1) is derived. In Section 3, we explain the necessary steps of seperating the shapes from the appearance of faces, and apply the solution in Section 2 to predict a new view of faces. Section 4 shows experimental results of synthetic face images at new views. In Section 5, we discuss this approach and conclude.

2 Probabilistic Modeling

The problem of linear equations with missing data described in equation (1) is restated in the following way:

We have N_T training vectors $\{x_i\}_{i=1}^{N_T}$, each of which is an N-by-1 vector. Usually $N \gg N_T$. A test example $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ belongs to the same class defined by the training set. b is N-by-1. We only know b_1, which contains the first N_1 elements of b. The task is to predict b_2, given b_1 and $\{x_i\}_{i=1}^{N_T}$.

2.1 Probabilistic Modeling

Let's first discuss the ideal case that we have enough independent training examples to span the whole N-dimensional space, i.e., $N_T \geq N$.

Here we apply several assumptions:

1. The class defined by the training set is an M-dimensional linear subspace, denoted as F. $M < N$ and determined by PCA from the training set. PCA is computed from the training set $\{x_i\}_{i=1}^{N_T}$, and the M largest eigenvalues of the principal components $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$ are the variances along the M dimensions of F.
2. In this subspace F, samples are drawn from an M-dimensional Gaussian distribution with zero mean.
3. If the complete space has N dimensions, there is another $(N-M)$ dimensional linear subspace $\bar{F}$, which is orthogonal and complementary to the eigenspace F (Fig. 1). We assume F and $\bar{F}$ are statistically independent.
4. The samples also contain random noise distributed over all the $(N-M)$ dimensions of $\bar{F}$. Each of the $(N-M)$ dimensions of $\bar{F}$ has approximately equal non-zero variance, i.e., $\lambda_{M+1} \approx \lambda_{M+2} \approx \cdots \approx \lambda_N > 0$.

Under these assumptions, the probability of x is

$$\begin{aligned} P(x|\Omega) &= \left[\frac{\exp\left(-\frac{1}{2} \sum\limits_{i=1}^{N} \frac{y_i^2}{\lambda_i} \right)}{(2\pi)^{N/2} \prod\limits_{i=1}^{N} \lambda_i^{1/2}} \right] \\ &= \left[\frac{\exp\left(-\frac{1}{2} \sum\limits_{i=1}^{M} \frac{y_i^2}{\lambda_i} \right)}{(2\pi)^{M/2} \prod\limits_{i=1}^{M} \lambda_i^{1/2}} \right] \cdot \left[\frac{\exp\left(-\frac{1}{2} \sum\limits_{i=M+1}^{N} \frac{y_i^2}{\lambda_i} \right)}{(2\pi)^{(N-M)/2} \prod\limits_{i=M+1}^{N} \lambda_i^{1/2}} \right] \\ &= P_F(x|\Omega) \cdot P_{\bar{F}}(x|\Omega), \end{aligned}$$

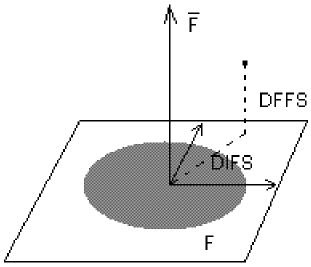

Fig. 1. Decomposition into the eigenspace F and its orthogonal subspace $\bar{F}$. The DFFS and DIFS are also shown.

where Ω denotes the class described by the training set. x is a random point from this class, and its projection onto each dimension is denoted as $\{y_i\}_{i=1}^{N}$. $P_F(x|\Omega)$ and $P_{\bar{F}}(x|\Omega)$ are two marginal Gaussian distributions, in F and $\bar{F}$ respectively.

Since N is very large, we lack sufficient data to compute each $\{\lambda_i\}_{i=M+1}^{N}$ in $P_{\bar{F}}(x|\Omega)$.

Recall the assumption that $\{\lambda_i\}_{i=M+1}^{N}$ are about the same magnitude. Then it is reasonable to use the arithematic average $\rho = \frac{1}{N-M}\sum_{i=M+1}^{N}\lambda_i$ [10] to get an estimation of $P(x|\Omega)$, which is

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

The distance characterizing the $\hat{P}(x|\Omega)$ is

$$\hat{d}(x) = \left[\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right] + \frac{1}{\rho}\cdot\left[\sum_{i=M+1}^{N}y_i^2\right]. \tag{2}$$

In our problem, we only know the upper part of $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, and know it is from class Ω. In order to solve for the unknown part b_2, we want to maximize the likelihood of $\hat{P}(b|\Omega)$ by choosing $\{y_i\}_{i=1}^{N}$, where $(y_1, y_2, \cdots, y_N)^T = \Phi^T(b - \bar{x})$ and $\bar{x}$ is the mean of the training set. We then generate $b_2 = \bar{x}_2 + \Phi_2 \cdot \mathbf{y}$. This optimization depends upon three quantities: $\left[\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right]$, $\left[\sum_{i=M+1}^{N}y_i^2\right]$ and the weight ρ, . Let's look at them one by one.

2.2 $\left[\sum_{i=1}^{M} \frac{y_i^2}{\lambda_i}\right]$: DIFS

This is the Mahalanobis distance, also called the "distance-in-feature-space" (DIFS) [10]. It describes how far the projection of x onto F is from the origin.

Let $\varLambda_M^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & & & 0 \\ & \frac{1}{\lambda_2} & & \\ & & \ddots & \\ 0 & & & \frac{1}{\lambda_M} \end{bmatrix}$, and $\mathbf{y} = (y_1, y_2, \cdots, y_M)^T$, then

$$\left[\sum_{i=1}^{M} \frac{y_i^2}{\lambda_i}\right] = \mathbf{y}^T \varLambda_M^{-1} \mathbf{y}. \tag{3}$$

2.3 $\left[\sum_{i=M+1}^{N} y_i^2\right]$: DFFS

The residual reconstruction error, also called DFFS [10] is $\sum_{i=M+1}^{N} y_i^2 = \epsilon^2(x) = \|x' - x\|^2$, where x' is the projection of x on F.

The linear-object-class method [12] minimizes the DFFS to find $\mathbf{y}$ in order to predict b_2. Split the eigenvector matrix $\varPhi_M$ containing the first M eigenvectors into $\varPhi_M = \begin{pmatrix} \varPhi_1 \\ \varPhi_2 \end{pmatrix}$ and split the mean $\bar{x}$ of training data into $\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}$, where $\varPhi_1$ and $\bar{x}_1$ have the same number of rows as b_1. No matter what method we use to solve for $\mathbf{y}$, since b_2 is defined as $\bar{x}_2 + \varPhi_2 \cdot \mathbf{y}$, the residual reconstruction error of resulting $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ is

$$\begin{aligned} \sum_{i=M+1}^{N} y_i^2 = \epsilon^2(b) = \|b - (\bar{x} + \varPhi_M \cdot \mathbf{y})\|^2 &= \left\| \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - \begin{pmatrix} \bar{x}_1 + \varPhi_1 \cdot \mathbf{y} \\ \bar{x}_2 + \varPhi_2 \cdot \mathbf{y} \end{pmatrix} \right\|^2 \\ &= \|b_1 - (\bar{x}_1 + \varPhi_1 \cdot \mathbf{y})\|^2 \\ &= \left\| \tilde{b}_1 - \varPhi_1 \cdot \mathbf{y} \right\|^2 \qquad , \end{aligned} \tag{4}$$

where $\tilde{b}_1 = b_1 - \bar{x}_1$. Thus the linear-object-class method [12] solves a least square problem to solve for $\mathbf{y}$:

$$\mathbf{y} = \arg\min \left\| \varPhi_1 \cdot \mathbf{y} - \tilde{b}_1 \right\|^2 .$$

2.4 Determining ρ

Moghaddam and Pentland [10] define $\rho = \frac{1}{N-M} \sum_{i=M+1}^{N} \lambda_i$, under the assumption that the number of training examples $N_T \geq N$, and that $\{\lambda_i\}_{i=M+1}^{N}$ are about the same magnitude. However, in practice, N is very large and we have $N_T \ll N$. These N_T training examples can only span an $(N_T - 1)$ dimensional subspace, resulting in that $\lambda_{N_T} = \lambda_{N_T+1} = \cdots = \lambda_N = 0$.

We use the non-zero eigenvalues, $\{\lambda_i\}_{i=M+1}^{N_T-1}$, to guess what $\{\lambda_i\}_{i=N_T}^{N}$ would be like had we been given sufficient training data. Here we add another assumption:

- We assume that the actual values of $\{\lambda_i\}_{i=N_T}^{N}$ will be about the same magnitude as the average of the known eigenvalues $\{\lambda_i\}_{i=M+1}^{N_T-1}$.

Under this assumption, $\rho = \frac{1}{N_T-M-1} \sum\limits_{i=M+1}^{N_T-1} \lambda_i$.

2.5 Solving the Optimization Problem

Given b_1, we want to find b_2 that minimizes $\hat{d}(b)$. Substituting equations (3) and (4) into (2),

$$\begin{aligned} \hat{d}(b) &= \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} + \frac{\epsilon^2(b)}{\rho} \\ &= \mathbf{y}^T \Lambda_M^{-1} \mathbf{y} + \frac{1}{\rho} \left\| \tilde{b}_1 - \Phi_1 \cdot \mathbf{y} \right\|^2 \\ &= \mathbf{y}^T \Lambda_M^{-1} \mathbf{y} + \frac{1}{\rho} \left(\tilde{b}_1 - \Phi_1 \cdot \mathbf{y} \right)^T \left(\tilde{b}_1 - \Phi_1 \cdot \mathbf{y} \right) \\ &= \frac{1}{\rho} \left(\mathbf{y}^T \rho \Lambda_M^{-1} \mathbf{y} + \mathbf{y}^T \Phi_1^T \Phi_1 \mathbf{y} - 2 \left(\Phi_1^T \tilde{b}_1 \right)^T \cdot \mathbf{y} + \tilde{b}_1^T \tilde{b}_1 \right) . \end{aligned}$$

Letting the partial derivative to be zero,

$$\begin{aligned} 0 = \frac{\partial \hat{d}(b)}{\partial \mathbf{y}} &= 2\rho \Lambda_M^{-1} \mathbf{y} + 2\Phi_1^T \Phi_1 \mathbf{y} - 2\Phi_1^T \tilde{b}_1 \\ &= 2 \left[\left(\rho \Lambda_M^{-1} + \Phi_1^T \Phi_1 \right) \mathbf{y} - \Phi_1^T \tilde{b}_1 \right] . \end{aligned}$$

The solution of $\mathbf{y}$ is

$$\mathbf{y} = \left(\rho \Lambda_M^{-1} + \Phi_1^T \Phi_1 \right)^{-1} \cdot \Phi_1^T \tilde{b}_1. \tag{5}$$

And the unknown b_2 can be predicted as $b_2 = \bar{x}_2 + \Phi_2 \cdot \mathbf{y}$.

3 Seperating Shape and Appearance

Let's use the above technique to solve the problem of synthesizing new views of human faces. The problem is described as follows. Given a probe face image I under pose 1, we need to synthesize a new image J of this person's face under pose 2. The training set consists of N_T pairs of face images, $\{[I_1, J_1], [I_2, J_2], \cdots, [I_{N_T}, J_{N_T}]\}$. I_i and J_i are faces of the ith subject in the training set. $\{I_i\}_{i=1}^{N_T}$ are under pose 1, and $\{J_i\}_{i=1}^{N_T}$ are under pose 2.

In our approach, we make the common assumption [5,12,7] that the characteristics of shape can be seperated from appearance.

3.1 Shape

On each face image, a set of landmarks are labeled by hand. For the ith training image under pose 1, denote the coordinates of each landmark as (x_j, y_j), $j = 1, \cdots, L_1$, where L_1 is the number of landmarks on the faces under pose 1. Define the shape vector of this ith face image under pose 1 as

$$s_{i,1} = (x_1, x_2, \cdots, x_{L_1}, y_1, y_2, \cdots, y_{L_1})^T .$$

A similar vector $s_{i,2}$ can also be defined in the same way for pose 2. Concatenating these two vectors, we get a vector

$$s_i = \begin{pmatrix} s_{i,1} \\ s_{i,2} \end{pmatrix}$$

as a combined shape vector for the ith subject in the training set.

Thus, for the N_T subjects in the training set, we get a training set of shape vectors $\{s_i\}_{i=1}^{N_T}$.

3.2 Appearance

For each pose, a reference face is chosen, so that every face under this pose is warped to the shape of the reference face, giving a normalized image. The warping is done via a triangulation algorithm with the landmarks [5] assuming that the faces have lambertian surfaces. On each normalized image, only the pixels within the convex hull of the landmarks are kept and all other pixels are discarded. This is done to remove the unnecessary variations of the hair or the background scenery. Let's call the resultant normalized images under pose 1 as $\widetilde{I_1}, \widetilde{I_2}, \cdots, \widetilde{I_{N_T}}$, and those under pose 2 as $\widetilde{J_1}, \widetilde{J_2}, \cdots, \widetilde{J_{N_T}}$. Reshape them into vectors as $\{t_{i,1}\}_{i=1}^{N_T}$ and $\{t_{i,2}\}_{i=1}^{N_T}$ for pose 1 and pose 2 respectively.

For the ith subject in the training set, define

$$t_i = \begin{pmatrix} t_{i,1} \\ t_{i,2} \end{pmatrix}$$

as a combined appearance vector. Thus, for the N_T subjects in the training set, we get a training set of appearance vectors $\{t_i\}_{i=1}^{N_T}$.

3.3 Probe Image and Prediction

Given a probe face image I under pose 1, we need to synthesize a new image J of this person's face under pose 2. With a set of landmarks on I and the reference face under pose 1, we can again decompose I into its shape vector $\hat{s}_1$ and appearance vector $\hat{t}_1$. The landmarks can be obtained using AAM fitting [5]. In our experiments, we hand labeled these landmarks on the probe image I.

If we can predict the shape vector $\hat{s}_2$ and the appearance vector $\hat{t}_2$ of the unknown image J, by warping $\hat{t}_2$ from the reference face under pose 2 back to the shape defined by $\hat{s}_2$, we will be able to get the synthesized new image J.

So the problem turns into: How to predict $\hat{s}_2$, given $\hat{s}_1$ and the training set $\{s_i\}_{i=1}^{N_T}$? And how to predict $\hat{t}_2$, given $\hat{t}_1$ and the training set $\{t_i\}_{i=1}^{N_T}$? They are the same mathematical problem. Using exactly equation (5) that we described

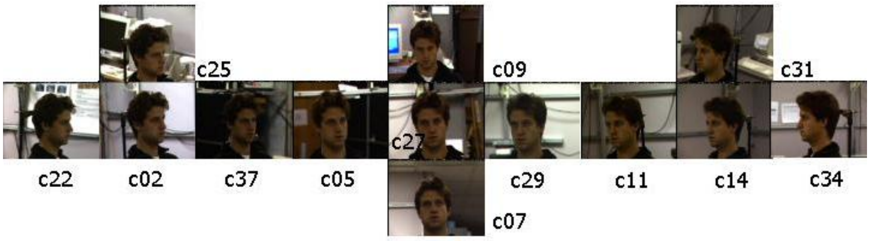

Fig. 2. The 13 poses in CMU PIE database

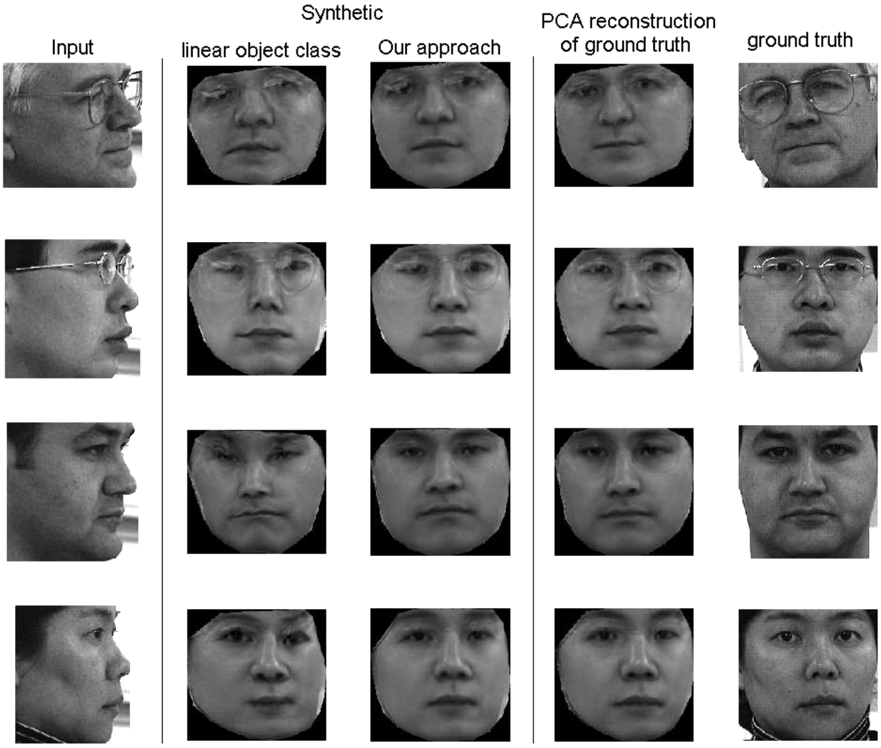

Fig. 3. Synthesizing a frontal view from a given profile. Column 1 to 5: (1) input image under pose 1 (2) synthetic image using linear-object-class. (3) synthetic image using our approach. (4) PCA reconstruction of ground truth of pose 2. (5) ground truth of pose 2.

in Section 2 will predict the unknown shape $\hat{s}_2$ and the unknown appearance $\hat{t}_2$. Then we can combine them to get the synthesized new image J, which is the new view of the probe face under pose 2 .

4 Experimental Results

We tested the performance of this method on the CMU PIE database [11]. The database contains 68 subjects. We chose 64 subjects as the training set, and 4 subjects (04016, 04022, 04026 and 04029) as the test set. Our experiments were performed on the "expression" subset including those images with neutral expressions, and those images containing glasses if the subject normally wears glasses. All images were converted to gray-scale images. The database contains 13 poses, illustrated in Fig. 2. We used combinations of 'c27' (frontal view), 'c37' (45° view) and 'c22' (profile) to test our algorithm. The landmarks were

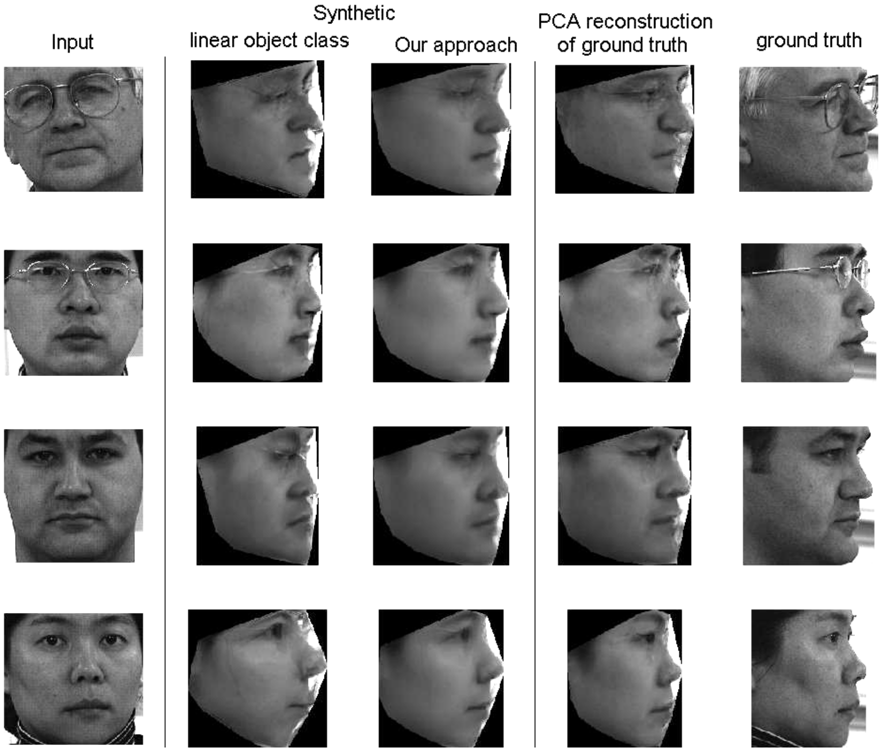

Fig. 4. Synthesizing a profile from a given frontal view. Column 1 to 5: (1) input image under pose 1 (2) synthetic image using linear-object-class. (3) synthetic image using our approach. (4) PCA reconstruction of ground truth of pose 2. (5) ground truth of pose 2.

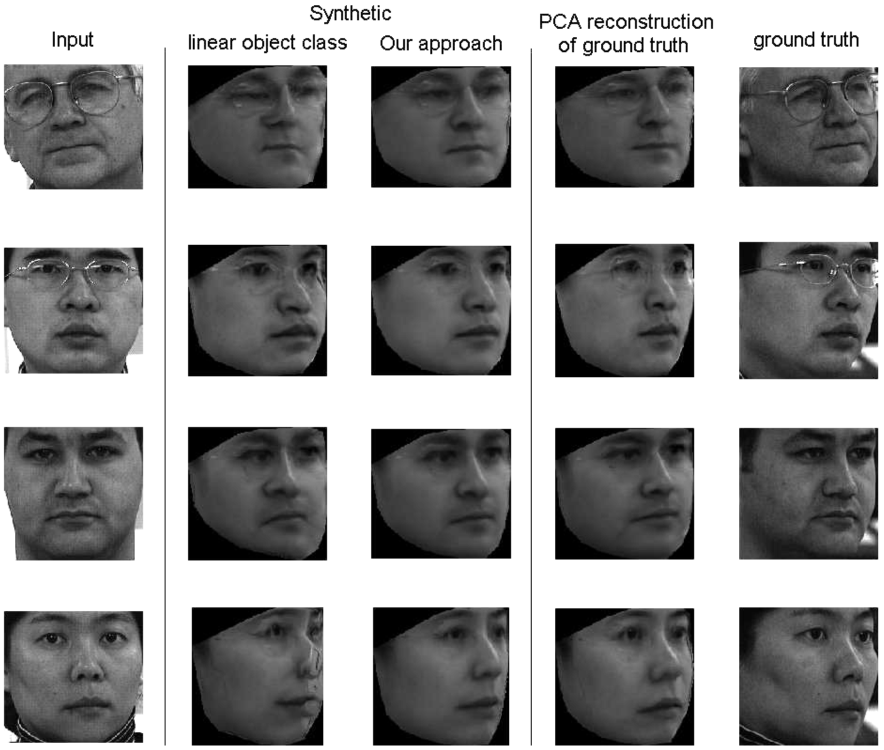

Fig. 5. Synthesizing a 45° view from a given frontal view. Column 1 to 5: (1) input image under pose 1 (2) synthetic image using linear-object-class. (3) synthetic image using our approach. (4) PCA reconstruction of ground truth of pose 2. (5) ground truth of pose 2.

provided courtesy of Ralph Gross [7]. The number of landmarks vary depending on the pose, from 39 landmarks to 54 landmarks.

We performed 3 sets of experiments, including predicting frontal view from profile (Fig. 3), predicting profile from frontal view (Fig. 4), predicting 45° view from frontal view (Fig. 5). These experiments all involve large out-of-plane rotations, such as 90° or 45°. In each experiment, the result of our approach is compared with that of linear-object-class method. We also computed the PCA reconstruction of the ground truth, by projecting the true $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ onto the eigenspace, to show the best possible reconstruction under the linear eigenspace assumption. In each experiment, for either the shape or the appearance, we always choose the number of principal eigenvectors that occupies 98% of energy. Each synthesis takes an average of 4 - 5 seconds on a PC with a 3GHz Pentium 4 processor, including predicting shape and appearance and also warping the appearance to the shape. More specificly, the prediction of shape and appearance takes about 0.3 second, and the warping takes about 4 seconds.

Fig. 3 - 5 show how our approach improves upon the results of the linear-object-class method, especially in predicting the shapes and handling large out-

Table 1. Sum of squared errors in shape prediction from profile to frontal view

linear-object-class method	our approach	PCA reconstruction of ground truth
49.5860	36.4763	12.8272
31.8927	27.6268	11.6914
51.8873	34.2422	11.6563
74.9253	49.6667	13.1261

Table 2. Sum of squared errors in shape prediction from frontal view to profile

linear-object-class method	our approach	PCA reconstruction of ground truth
67.1898	38.1367	15.9177
42.6668	32.0460	11.9452
47.4225	31.5215	13.5370
50.4465	28.2523	16.2700

Table 3. Sum of squared errors in shape prediction from frontal view to 45° view

linear-object-class method	our approach	PCA reconstruction of ground truth
46.4735	34.7182	16.0734
42.6817	31.4329	12.1678
56.5308	35.1610	18.0164
65.3341	41.0252	15.2125

of-plane rotations. Although our synthetic images are not perfect replicas of the ground truth, they are similiar to the PCA reconstructions of the ground truth, which are the best possible synthetic images under the linear eigenspace assumption. We performed these experiments using a training set of only 64 subjects. With more training data, the eigenspace would be more accurately described and better results could be expected.

We have also included the numerical comparison of errors for shape prediction in each set of experiments (Table 1, 2, 3). In each table, we compare the sum-of-squared-errors in the coordinates of the predicted shape, using the linear-object-class method, our approach, and the PCA reconstructions of ground truth, respectively. From the numerical errors, we can see our approach is efficient in reducing the errors by at least 30%.

Such technique can be used in face recognition which involves pose variation of large out-of-plane rotations.

5 Discussion and Conclusion

In this paper, we proposed an approach that can efficiently synthesize accurate new views of faces across large out-of-plane rotation, given only a single image. In our approach, we formulate a probabilistic model combining the "distance-from-feature-space" and the "distance-in-feature-space", and minimize the weighted sum of the two distances, in order to maximize the likelihood of the test example with missing data. Experimental results show that our approach produces more accurate results than the commonly used linear-object-class approach which is the basis of many 2D approaches.

Moreover, if compared with the 3D approaches, our method is also attractive in that it is computationally efficient and fast. With no need for a 3D model or 3D training data, it does not construct 3D geometry at all, thus it avoids expensive 3D rendering and fitting. It directly synthesizes 2D images, and requires using only pairs of 2D images. Nor does our approach require correspondence between different poses. Only correspondence between faces under the same pose is used. As a tradeoff, our approach needs the same number of views for each subject in the database, and can only synthesize those views which are in the database.

Pose variations, especially large out-of-plane rotations, make face recognition a difficult problem. Our algorithm of synthesizing new views of a person's face given a single input face image, can enable face recognition systems to overcome the pose variation problem.

Acknowledgements

We would like to thank Ralph Gross for providing us the hand labeled landmarks. We would also like to thank Simon Baker for providing us the CMU PIE database.

References

1. Black, M., Jepson, A., "Eigen-tracking: Robust matching and tracking of articulated objects using a view-based representation," International Journal of Computer Vision, Vol.36, No.2, pp.101-130, 1998.
2. Blanz, V., Vetter, T., "A Morphable Model for the Synthesis of 3D Faces," ACM Siggraph, 1999.
3. Blanz, V., Vetter, T., "Face Recognition Based on Fitting a 3D Morphable Model," IEEE TPAMI, Vol.25, No.9, pp.1063-1074, 2003.
4. Blanz, V., Grother, P., Phillips, P.J., Vetter, T., "Face Recognition Based on Frontal Views generated from Non-Frontal Images," CVPR, 2005.
5. Cootes, T.F., Edwards G.J., and Taylor, C.J., "Active Appearance Models," IEEE TPAMI, Vol.23, No.6, pp.681-685, 2001.
6. Debevec, P., Taylor, C., and Malik, J., "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," Computer Graphics (SIGGRAPH), pages 11-20, 1996.
7. Gross R., Matthews, I., Baker, S., "Appearance-Based Face Recognition and Light-Fields," IEEE TPAMI, Vol.26, No.4, pp.449-465, 2004.
8. Hwang, B-W., Lee S-W., "Reconstruction of Partially Damaged Face Images Based on a Morphable Face Model," IEEE TPAMI, Vol.25, No.3, pp.365-372, 2003.
9. Leonardis, A., Bischof, H., "Robust recognition using eigenimages," Computer Vision and Image Understanding," Vol.78, No.1, pp.99-118, 2000.

10. Moghaddam, B., Pentland, A., "Probabilistic Visual Learning for Object Representation," IEEE TPAMI, Vol.19, No.7, pp.696-710, 1997.
11. Sim T., Baker, S., Bsat, M., "The CMU Pose, Illumination, and Expression Database," IEEE TPAMI, Vol.25, No.12, pp.1615-1618, 2003.
12. Vetter, T., Poggio, T., "linear object classes and Image Synthesis From a Single Example Image," IEEE TPAMI, Vol.19, No.7, pp.733-742, 1997.
13. Xiao, J., "Reconstruction, Registration, and Modeling of Deformable Object Shapes," CMU RI PhD thesis, 2005.