# Pose-Robust Face Recognition Using Geometry Assisted Probabilistic Modeling *

Xiaoming Liu
Visualization and Computer Vision Lab
General Electric Global Research Center
Schenectady, NY, 12309
liux@research.ge.com

Tsuhan Chen
Advanced Multimedia Processing Lab
Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA, 15213
tsuhan@cmu.edu

## Abstract

*Researchers have been working on human face recognition for decades. Face recognition is hard due to different types of variations in face images, such as pose, illumination and expression, among which pose variation is the hardest one to deal with. To improve face recognition under pose variation, this paper presents a geometry assisted probabilistic approach. We approximate a human head with a 3D ellipsoid model, so that any face image is a 2D projection of such a 3D ellipsoid at a certain pose. In this approach, both training and test images are back projected to the surface of the 3D ellipsoid, according to their estimated poses, to form the texture maps. Thus the recognition can be conducted by comparing the texture maps instead of the original images, as done in traditional face recognition. In addition, we represent the texture map as an array of local patches, which enables us to train a probabilistic model for comparing corresponding patches. By conducting experiments on the CMU PIE database, we show that the proposed algorithm provides better performance than the existing algorithms.*

## 1. Introduction

For decades human face recognition has been an active topic in the field of object recognition. Comprehensive surveys of human and machine recognition techniques can be found in [1]. At least two observations have been made from the previous extensive study. First, face recognition is to deal with *variations*, such as pose, illumination and expression. Among all kinds of variations, pose variation is the hardest one to model and therefore contributes most of the recognition error to a recognition system [2][3]. For example, as shown in Face Recognition Vendor Test (FRVT) 2002 [3], the recognition rate with pose variation is much lower than that with illumination variation. Second, face *registration* is the key of face recognition. This observation is a direct consequence of the first one. In dealing with different variations, if we can register face images into a canonical model, the recognition task would be simpler. In traditional face recognition, normally the face area is cropped before feeding into the recognition module. Hence, the importance of face registration has been overlooked.

The difficulty with pose variation is that, the intra-subject variations can be as large as, or even larger than the inter-subject variations when pose variation is present. To deal with the pose variation, we propose to use *geometrical mapping*, which essentially estimates the pose for each face image and maps it onto the surface of a 3D ellipsoid. All recognition is then performed on the surface of the ellipsoid. Geometrical mapping could be considered as one way of registering the faces, compensating the pose variation and as well as reducing the intra-subject variations. We also represent the facial appearance as an array of local patches and model the distance between corresponding patches from multiple poses in a probabilistic manner, which is then used to improve the pose-robust face recognition.

Many approaches have been proposed for pose-robust face recognition. The first type of approaches is to learn the dynamics/trajectories from images with continuous pose variation. And then such trajectories are used in recognizing faces from image sequences [4][5]. One drawback with these approaches is that certain application scenario, where the subject shows consistent motion in both training and test data, has to be assumed. This assumption is not true in general, which limits the popularity of this type of approaches. The second type of approaches is to treat the whole face image under a certain pose as one sample in a high-dimensional space, and learn the relation between a frontal pose image and non-frontal pose images by building a mapping function between them. Given a test image with an arbitrary pose, a recognition-by-synthesis approach is applied. That is, we can either transform this test image into the frontal view [6], or transform each of the training images into the same pose as the test image [7], based on the learned mapping function. One potential problem with this type of approaches is that it is not clear whether the rela-

*The work presented in this paper is performed in Advanced Multimedia Processing Lab, Carnegie Mellon University.

tion between different pose images can be approximated as a simple function, such as a linear transformation [6]. Since a face image is pretty complex and different parts of a face might transform in a different manner under varying poses, researchers start to look at faces as a set of parts/patches [8][9]. Kanade and Yamada [9] conduct a systematic analysis on how the discriminative power of different parts on human faces changes according to different poses, and such analysis leads to a probabilistic approach to face recognition. Since we are dealing with pose variation, which is a result of the human head's geometry projected differently, it is natural to rely on the geometric information to aid the recognition. Blanz and Vetter [10]'s approach is along this direction. Given a test image under any pose and illumination, they can fit the image with pre-trained texture and shape models by tuning the coefficients in the models. Finally the model coefficients are used for recognition.

Our approach uses less geometric information than Blanz and Vetter's. Researchers always assume that better modeling leads to better recognition performance. However, the price we have to pay for a more sophistical modeling is that model fitting becomes too difficult. For example, in [10], both the training and test images are manually labeled with 6 to 8 feature points. On the other hand, unlike rendering applications in computer graphics, we might not need a very sophistical geometric model for the recognition purpose. The benefit with a simpler geometric model is that model fitting tends to be easier and automatic, which is the goal of our approach.

The paper is organized as follows. In the next section, we first introduce how to generate a texture map from a face image. In Section 3, a basic geometry assisted face recognition approach is presented. Then we present a method of learning the probabilistic models for measuring the similarity between patches from a face database with pose variation, and how to apply it to pose-robust face recognition. The experimental results are shown in Section 6.

## 2. Geometrical mapping

If we compare two face images of the same subject captured at two different view angles, the pixel-by-pixel difference is relatively big because these two images are not registered/aligned with respect to each other. This is also the reason why the traditional eigenface approach [11] does not work well for face images with pose variation. Image registration is the way to fix this problem, i.e., the comparison should only be conducted after two images are registered. Considering the fact that a human head has the non-planar geometry, one way to register face images is to back project them to the surface of a 3D ellipsoid based upon their specific poses. This procedure of back projection is called *geometrical mapping*, which is a key component in our face

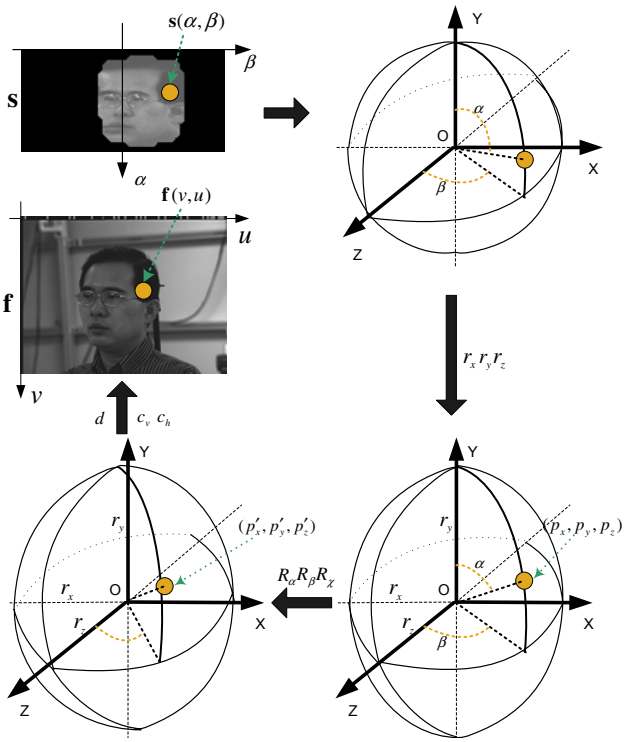


Figure 1: Geometric mapping: one point on the surface of the ellipsoid maps to a pixel on the image plane. This illustrates how to generate one texture map given a input image and a known mapping parameter.

recognition algorithm. In this section, we introduce how to generate a texture map $\mathbf{s}$ from a face image $\mathbf{f}$, given a known mapping parameter $\mathbf{x}$.

Three assumptions are made. First, a human head is a 3D ellipsoid with radiuses being $r_x$, $r_y$, and $r_z$. Second, a face image is captured with the weak-perspective camera model [12] and the camera's focal length equals to one. Third, all images are captured under the ambient lighting environment. Under these assumptions, we use a mapping parameter $\mathbf{x}$ to describe the relation between a face image and its texture map. This parameter is a 6-dimensional vector $x = [c_v\ c_h\ d\ R_\alpha\ R_\beta\ R_\chi]^T$, where $c_v$ and $c_h$ indicate the center of the face area in the image, $d$ indicates the average distance between the face and the camera, and $R_\alpha, R_\beta$ and $R_\chi$ indicate the rotation of the human head respectively. As we can see, the mapping parameter $\mathbf{x}$ includes all the information for locating the face, as well as generating a texture map from the face image.

Let a human head centered at the origin of an XYZ coordinate system and the frontal face look at the positive Z axis. Thus different views of a human face can be obtained by fixing the camera and rotating the human head with certain degrees in various directions. To generate a texture map $\mathbf{s}$ from $\mathbf{f}$, essentially for each pixel, $\mathbf{s}(\alpha, \beta)$, we need to find

its corresponding coordinate, $\mathbf{f}(v, u)$, by knowing the mapping parameter $\mathbf{x}$, which is then followed by a bilinear interpolation [13] to fill in the intensity of pixel $\mathbf{s}(\alpha, \beta)$. The parameters $v$ and $u$ are the axes of the original image; $\alpha$ and $\beta$ are the axes of the texture map. As shown in Figure 1, there are basically four steps for this mapping.

First, a pixel $\mathbf{s}(\alpha, \beta)$ in the texture map corresponds to one coordinate $(P_x, P_y, P_z)$ on the surface of a sphere, whose radius is one:

$$\begin{cases} P_x = \sin(\alpha)\sin(\beta) \\ P_y = \cos(\alpha) \\ P_z = \sin(\alpha)\cos(\beta) \end{cases}$$

As shown in the right part of Figure 1, the sphere is then converted into an ellipsoid by stretching each radius according to $r_x$, $r_y$, and $r_z$:

$$\begin{cases} P_x = r_x P_x \\ P_y = r_y P_y \\ P_z = r_z P_z \end{cases}$$

Second, we can rotate the head ellipsoid by $R_\alpha, R_\beta$ and $R_\chi$ with respect to the XYZ axes. We perform the horizontal rotation $R_\beta$ with respect to the Y axis first, then the vertical rotation $R_\alpha$ with respect to the X axis, and the in-plane rotation $R_\chi$ with respect to the Z axis. As shown in Figure 1, $(P_x, P_y, P_z)$ moves to a new coordinate $(P'_x, P'_y, P'_z)$ by the following equation:

$$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \end{bmatrix} = \begin{bmatrix} \cos(R_\chi) & \sin(R_\chi) & 0 \\ -\sin(R_\chi) & \cos(R_\chi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(R_\alpha) & -\sin(R_\alpha) \\ 0 & \sin(R_\alpha) & \cos(R_\alpha) \end{bmatrix} \begin{bmatrix} \cos(R_\beta) & 0 & \sin(R_\beta) \\ 0 & 1 & 0 \\ -\sin(R_\beta) & 0 & \cos(R_\beta) \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (1)$$

Third, we can project the coordinate $(P'_x, P'_y, P'_z)$ to the image plane by using the weak-perspective camera model and translating the resulting coordinate by $c_v$ and $c_h$ in both vertical and horizontal directions:

$$\begin{cases} v = \frac{P'_y}{d} + c_v \\ u = \frac{P'_x}{d} + c_h \end{cases}$$

Finally we get the new coordinate $(v, u)$ in the image plane. Because not all pixels on the texture map can be visible from the camera, we need to determine the visibility of each coordinate $(P'_x, P'_y, P'_z)$ by the following. That is, we rotate the normal of the point at $(P_x, P_y, P_z)$ by $R_\alpha, R_\beta$ and $R_\chi$, as done in (1). If the angle between the resulting normal and the positive Z axis is smaller than 90°, i.e., the normal points to the positive Z axis, $(v, u)$ is a valid coordinate. If it is, the bilinear interpolation result of $(v, u)$ is filled in as the intensity of the pixel $\mathbf{s}(\alpha, \beta)$. Otherwise $\mathbf{s}(\alpha, \beta)$ is considered as a missing pixel and set the intensity to be zero. To compensate the illumination variation, we also normalize the mean of the intensities of all non-missing pixels to be 128.

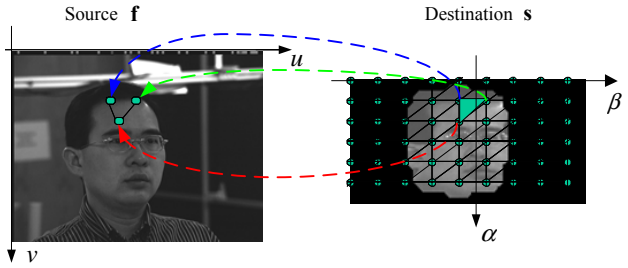


Figure 2: Triangle representation: a set of mapping equations are applied only for the vertexes of triangles, while the other mapping are obtained through affine transformations between corresponding triangles. This speeds up the geometric mapping.

One issue in the above mapping is how to determine the radiuses of a human head ellipsoid $r_x$, $r_y$, and $r_z$, which are essentially the height, width and depth of the human head. Since we are estimating the distance between the human face and the camera origin $d$, any one of the three radiuses, for example, the width $r_x$, can be set to be one. Thus we only need to determine the ratio between the width to the depth, and the ratio between the width to the height. In our algorithm, the former is set to be a fix constant 0.9 by considering that the head's depth is slightly larger than the head's width, while the latter is usually obtained from the external source, such as a face detector or hand labeling for the frontal face image. Once we obtain these two ratios, they are assumed to be constant for the same subject. Of course, we can also treat these two ratios as two additional elements in the mapping parameter $\mathbf{x}$, and estimate them using the same framework as estimating $\mathbf{x}$.

Since geometrical mapping is an essential step in our recognition algorithm, the efficiency of this step affects the speed of face registration/recognition. In practice, this step can be computationally intensive, if every pixel in $\mathbf{s}$ needs to find its corresponding coordinate in $\mathbf{f}$ using the above set of equations. To solve this problem, we approximate the texture map $\mathbf{s}$ using a triangular mesh, as shown in Figure 2. That is, for the vertexes of these triangles, we derive their corresponding coordinates in $\mathbf{f}$ using the above mapping equations. Thus the mapping between two triangles can be approximated by an affine transformation, whose six parameters are estimated via three pairs of corresponding vertexes. For the pixels inside each triangle, the scan-line algorithm [14] is used to quickly find the corresponding coordinates. The goal of this approximation is to speed up the geometrical mapping while not noticeably affecting the recognition precision. The choice of the triangle's size is a

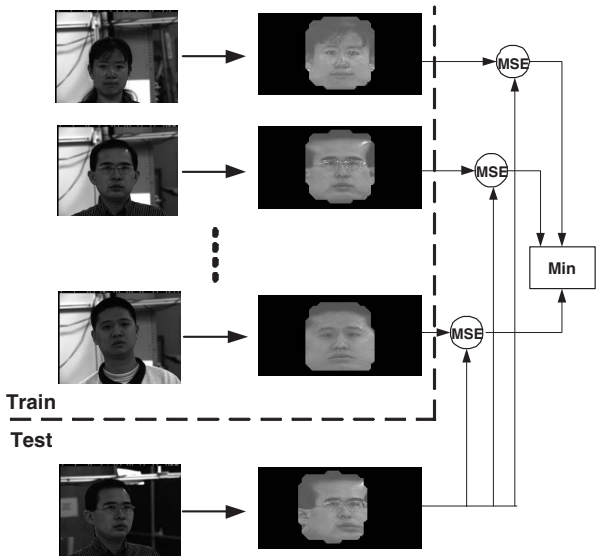


Figure 3: Geometry assisted face recognition: all training and test images are converted into the texture maps, and the distance measure is calculated based on the *overlap* area between two texture maps. Face images are better registered in the texture map space, than in the original image space.

trade-off between the mapping speed and the mapping precision. If the triangle is larger, the mapping is faster while the precision is also lower. In our implementation, the triangle size is 4 by 4 pixels.

# 3. Geometry assisted face recognition

In many face recognition systems, there is only one face image, normally the frontal view face image, during the training stage. However, in the test stage, there might be test images that correspond to different poses of human faces. This is a hard problem because the same face might appear very differently under various poses. In this section, we present our geometry assisted approach to deal with this case.

As shown in Figure 3, given a face database with $L$ subjects, there is only one frontal view face image, $\mathbf{f}_l (l = 1, 2, \ldots, L)$, for each subject that is available for training. During the training stage, the optimal mapping parameter $\mathbf{x}_l$ is estimated for each training image $\mathbf{f}_l$ based on a universal *mosaic* model, which is generated by combining texture maps from multiple subjects, using the condensation method [15]. Essentially this estimation process is trying to minimize the difference between the universal mosaic model and the texture map controlled by the mapping parameter, which provides information about the position, the distance, and the pose of the face. Notice that some of the parameters might be known from external sources. For example, if we know all training images have frontal view faces, their pose parameters, $R_\alpha, R_\beta$ and $R_\chi$, are known

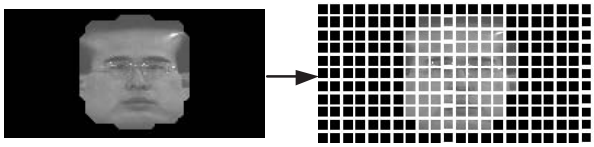

Figure 4: Patch representation: a texture map is evenly decomposed into an array of local patches, which enables the appearance modeling of local patches and the potential movement of patches.

to be zero. Once the estimation is done, the corresponding texture map $\mathbf{s}_l$ is generated from each training image $\mathbf{f}_l$. It is obvious that in the texture map $\mathbf{s}_l$, only part of the pixels are valid information of the facial appearance, while the rest are missing pixels since each face image only corresponds to one portion of a 3D head ellipsoid's surface. To describe this missing pixel information, we also generate a mask map, $\mathbf{a}_l$, which has the same dimension as the texture map $\mathbf{s}_l$. For all missing pixels in $\mathbf{s}_l$, the corresponding pixels in $\mathbf{a}_l$ are zero and the others are one.

During the test stage, given one test image $\mathbf{f}_t$, first we estimate the optimal mapping parameter based on the universal mosaic model. Second, the resulting texture map $\mathbf{s}_t$ and the mask map $\mathbf{a}_t$ are compared with each of the training texture maps as the following:

$$d_l = \frac{1}{\|\mathbf{a}_t \circ \mathbf{a}_l\|} \|(\mathbf{s}_t - \mathbf{s}_l) \circ \mathbf{a}_t \circ \mathbf{a}_l\|^2 \qquad (2)$$

where $\circ$ refers to the element-wise multiplication. Basically $d_l$ is the normalized mean-square-error (MSE) between the overlap area of the test texture map $\mathbf{s}_t$ and the training texture map $\mathbf{s}_l$, and $\|\mathbf{a}_t \circ \mathbf{a}_l\|$ indicates the size of the overlap area between two texture maps. There is a degeneration case when the two texture maps have a very small overlap area, which leads to a small $d_l$. Because in our estimation algorithm, the mapping parameter changes slowly, there is a very low chance that we will fall into this degeneration case. Eventually, the test image is recognized as the subject with the minimal $d_l$.

# 4. Probabilistic modeling for patches

Researchers have considered that different parts of a human face contribute differently for face recognition. For example, Pentland et. al. [16] propose to use modular eigenspaces to model the appearance of facial features, such as eyes, mouth, etc. Kanade and Yamada [9] perform discriminative analysis for all sub-regions in the face area and obtain a pose-robust face recognition algorithm.

We extend the idea of sub-region analysis and apply it to the geometry assisted approach. As shown in Figure 4, for each texture map $\mathbf{s}_l$, we represent it as an array of local patches $\mathbf{s}^l_{i,j}$. There are a number of benefits of using

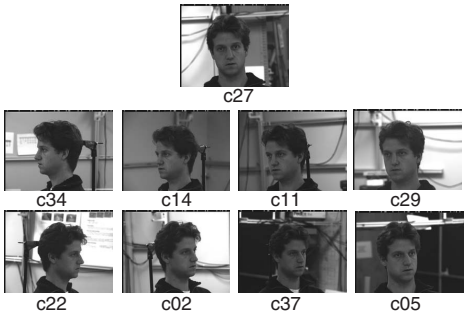


Figure 5: Sample images of one subject from the PIE database: the image in the first row is the training image, while the others are test images.

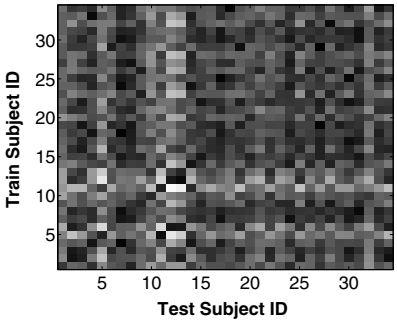


Figure 6: A 2D distance map: the intensity of each element indicates the distance measure of the same patch (around the right eye) at two poses (pose c29 and c27). It can be observed that diagonal elements (intra-subject) are darker (smaller valued) than the off-diagonal elements (inter-subject).

the patch representation instead of the whole texture map. First, when combing the texture maps from multiple poses to generate a map that covers larger pose views, patches can move locally to find better matching with other poses. Hence the moving of local patches compensates when the assumption of the ellipsoid human head is not perfect. Second, instead of treating each pixel equally by using (2), we can modify the distance measure of each patch according to the pose changes. A probabilistic model can be trained to model such changes and improve face recognition under pose variation.

Let us introduce how to train a probabilistic model for the distance measures of patches from a face database with pose variation. In this paper we train such a model using the CMU PIE database [17]. The PIE database consists of face images of 68 subjects under different combinations of poses and illuminations. We use part of this database in this paper, which are 9 pose images for 68 subjects. These are the images with multiple poses under the neutral illumination condition. Sample images from one subject are shown in Figure 5, where the numbers, c27, c34, c14, c11, c29, c22, c02, c37, c05, are the pose labels for each image. We choose c27 as the training pose and the other eight poses as the test poses.

We take 9 pose images of 34 subjects for training the probabilistic model. We denote each of the images as $\mathbf{f}(l, \phi_m)$, where $\phi_m$ is one of the eight test pose labels. We obtain the texture maps of all images, and have the patch representation as $\mathbf{s}_{i,j}(l, \phi_m)$, where $i$ and $j$ are the index of patches vertically and horizontally.

Since we treat the frontal pose, c27, as the training image, we need to study how the distance measure of corresponding patches between c27 and all other eight poses changes. This is done by fixing one patch and one particular pose, and calculating the distance measure of one patch (MSE) between all subjects in the pose c27 and all subjects in that particular pose. For example, Figure 6 is the result of such calculation for one patch closer to the right eye and the pose c29. In this 2D map, the vertical axis represents all the training images from 34 subjects under the pose c27, while the horizontal axis represents all test images from 34 subjects under the pose c29. Each entry indicates the distance measure of the same patch between one pair of subjects. For each combination of all other patches and other eight test poses, we should generate one of such 2D map.

Ideally we should expect that the diagonal elements of this 2D map are darker than the off-diagonal elements because the former is an indication of the intra-subject variations, while the latter is an indication of the inter-subject variations. In order to verify such expectation, we can plot the histograms of the diagonal elements and off-diagonal elements respectively. Also, for explicitly modeling these two types of variations, we approximate them as two Gaussian distributions. That is, we estimate the mean and stand deviation of intra-subject variations from the diagonal elements, and the mean and stand deviation of inter-subject variations from the off-diagonal elements. The resulting two Gaussian distributions are denoted as the following:

$$p(d_{i,j}|\text{same}, \phi_m) = \frac{1}{\sqrt{2\pi}\sigma^{\text{same}}_{i,j}} exp[-\frac{1}{2}(\frac{d_{i,j} - \mu^{\text{same}}_{i,j}}{\sigma^{\text{same}}_{i,j}})^2]$$

$$p(d_{i,j}|\text{diff}, \phi_m) = \frac{1}{\sqrt{2\pi}\sigma^{\text{diff}}_{i,j}} exp[-\frac{1}{2}(\frac{d_{i,j} - \mu^{\text{diff}}_{i,j}}{\sigma^{\text{diff}}_{i,j}})^2] \quad (3)$$

where $\mu^{\text{same}}_{i,j}$,$\sigma^{\text{same}}_{i,j}$, $\mu^{\text{diff}}_{i,j}$, $\sigma^{\text{diff}}_{i,j}$ are the mean and stand deviation of intra-subject and inter-subject variations for the patch $(i, j)$ under the test pose $\phi_m$. Let us denote the probabilistic model as $\mathbf{P}_d = \{\{\mu^{\text{same}}_{i,j}, \mu^{\text{diff}}_{i,j}, \sigma^{\text{same}}_{i,j}, \sigma^{\text{diff}}_{i,j}\}_{\phi_m}\}$. Notice that all four parameters depend on the test pose $\phi_m$. For example, the most left plot of Figure 7 is the Gaussian approximation of two distributions in Figure 6. The solid and broken lines are the histograms of two distributions, and the dotted curves are the approximated two Gaussian distributions. These four plots are from the two distributions of

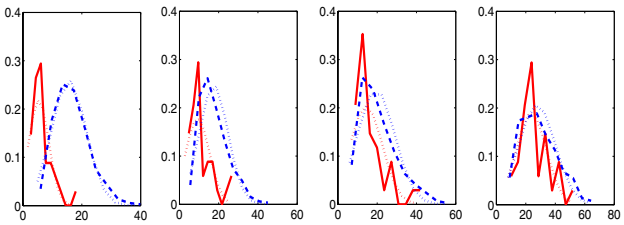

Figure 7: Gaussian approximation: each figure has two histograms (solid and broken curves) and two Gaussian approximations (dotted curves); four figures are from the two distributions of the same patch (around the right eye) with four different poses, namely c29, c11, c14, c34 from left to right. This shows that the discriminative power of distance measures decreases as the pose changes from the frontal view to the profile view.

the same patch with four different test poses: slightly right (c29), more right (c11), further right (c14), profile (c34). We can see that as the pose changes from the frontal view to the profile view, the discriminative power decreases since these two distributions are harder to be classified.

To illustrate the relation among these parameters for all test poses, we plot them in Figure 8. In total, there are five columns and eight rows, where each row corresponds to the statistical information of each test pose, namely c34, c14, c11, c29, c05, c37, c02, c22 from top to bottom. The first four columns are the plots of $\mu_{i,j}^{\text{same}}, \mu_{i,j}^{\text{diff}}, \sigma_{i,j}^{\text{same}}, \sigma_{i,j}^{\text{diff}}$ for all eight test poses. The intensity of each pixel indicates the value of the parameter. The brighter the intensity is, the larger the parameter is. In order to illustrate the difference between these two distributions, we normalize the intensity of the first and second column, as well as the intensity of the third and fourth column. Naturally, we can observe that the second column, $\mu_{i,j}^{\text{diff}}$, is brighter than the first column, $\mu_{i,j}^{\text{same}}$, and the fourth column, $\sigma_{i,j}^{\text{diff}}$, is brighter than the third column, $\sigma_{i,j}^{\text{same}}$, which means the inter-subject variations have larger mean and stand deviation than those of the intra-subject variations. The last column is the Fisher ratio [18] between two Gaussian distributions defined as the following:

$$f_{i,j} = \frac{(\mu_{i,j}^{\text{diff}} - \mu_{i,j}^{\text{same}})^2}{\sigma_{i,j}^{\text{same}^2} + \sigma_{i,j}^{\text{diff}^2}}$$

Since the fisher ratio is a good indication of the discriminative power, we can study that among all patches in the texture map, which patches provide more discriminative power than the others. From the last column of Figure 8, we see that the nose and forehead seem to have more discriminative power. This observation might not be true in general. However, it seems to be a right conclusion for this particular dataset.

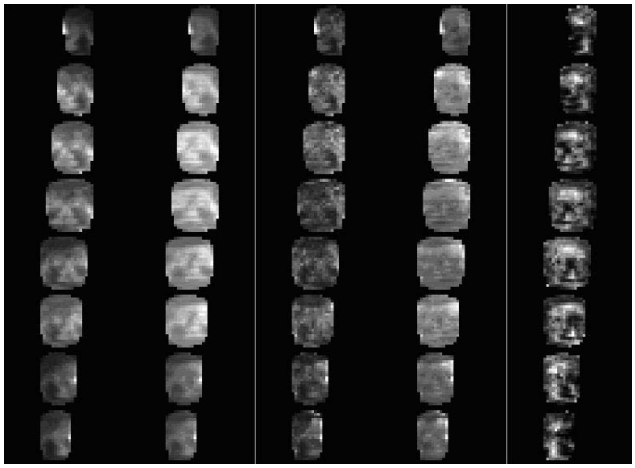

Figure 8: Probabilistic modeling for patches: the first four columns are plots of $\mu_{i,j}^{\text{same}}, \mu_{i,j}^{\text{diff}}, \sigma_{i,j}^{\text{same}}, \sigma_{i,j}^{\text{diff}}$ for all eight test poses; the last column is the fisher ratio of two distributions for all eight poses; each row corresponds to the statistical information of each test pose, namely c34, c14, c11, c29, c05, c37, c02, c22 from top to bottom.

# 5. Probabilistic geometry assisted face recognition

After introducing how to train a probabilistic model, let us focus on how to utilize it in improving the pose-robust face recognition. Given a face database with $L$ subjects, only one frontal view face image, $\mathbf{f}_l (l = 1, 2, \ldots, L)$, of each subject is available for training. During the training stage, the geometry assisted algorithm estimates the optimal mapping parameter $\mathbf{x}_l$ for each training image $\mathbf{f}_l$ based on the universal mosaic model. The resulting texture map from each training image is represented as an array of local patches, $\mathbf{s}_{i,j}^l$.

Given a test image, we also generate its texture map $\mathbf{s}_{i,j}^t$ based on the universal mosaic model. For the test texture map $\mathbf{s}_{i,j}^t$ and one of the training texture map $\mathbf{s}_{i,j}^l$, we compute the distance measures of all corresponding patches, $\{d_{i,j}\}$. Since we have developed the probabilistic models of distance measures of each local patch, they enable us to properly combine these distance measures, one computed for each corresponding patches, to reach to the local decision for recognizing whether the two texture maps/faces are from the same subject or not.

Given the distance measure and the pose of the test image, the posteriori probability that the test image and the training image belong to the same subject is:

$$P(\text{same}|d_{i,j}, \phi_t) = \frac{p(d_{i,j}|\text{same}, \phi_t)P(\text{same})}{p(d_{i,j}|\text{same}, \phi_t)P(\text{same}) + p(d_{i,j}|\text{diff}, \phi_t)P(\text{diff})} \quad (4)$$

where $\phi_t$ is the pose of the test image, which can be obtained during the estimation of the mapping parameter, $P(\text{same})$ and $P(\text{diff})$ are the prior probabilities of being the same subject or not given any test image. For a database with $L$ subjects, normally we set $P(\text{same}) = \frac{1}{L}$ and $P(\text{diff}) = \frac{L-1}{L}$. Notice that in order to calculate $p(d_{i,j}|\text{same}, \phi_t)$ using (3), $\phi_t$ needs to be equal to one of the test poses $\phi_m$. This issue is dealt with in two different ways.

First, if the pose of the test image $\phi_t$ is similar to one of the eight test poses $\phi_m$, we can approximate $\phi_t$ using the most similar test pose. Second, if $\phi_t$ is not similar to any one of test poses $\phi_m$, we can compute the marginal distributions of (4) over $\phi_m$:

$$p(d_{i,j}|\text{same}) = \sum_m P(\phi_m) p(d_{i,j}|\text{same}, \phi_m)$$

$$p(d_{i,j}|\text{diff}) = \sum_m P(\phi_m) p(d_{i,j}|\text{diff}, \phi_m)$$

$$P(\text{same}|d_{i,j}) = \frac{p(d_{i,j}|\text{same})P(\text{same})}{p(d_{i,j}|\text{same})P(\text{same}) + p(d_{i,j}|\text{diff})P(\text{diff})}$$

Here we assign a uniform distribution for $P(\phi_m)$. It could be non-uniform if we consider the probability of each pose presenting in the test set. Finally, the sum rule is applied. That is, the averaged probability measure of all patches $P(\text{same}|d_{i,j})$ is the similarity measure between the test image and one of the training subjects. Basically different combination rules, such as the sum rule, the product rule, the max rule, etc, can be applied here. Kittler et. al. conclude that in general the sum rule outperforms other combination rules because the sum rule is more resilient to estimation errors [19]. The test image is recognized to be the subject that gives the highest similarity measure.

In addition to being applied in image-based face recognition, our algorithm can also be used in video-based pose-robust face recognition. Given a video sequence containing human faces, video-based recognition involves both face tracking and face recognition. One computational efficient way is to combine these two by using the same model for performing both tasks simultaneously. As an extension of our approach, the individualized face mosaic model, which combines multiple texture maps from various poses of the same subject, can serve this purpose.

Face tracking is to determine the location, pose of a face in each frame. Since the mapping parameter $\mathbf{x}$ contains all these information, the face tracking is equivalent to estimating $\mathbf{x}$. Given one video frame, we use the condensation method [15] for this estimation. After estimating $\mathbf{x}$ and generating a texture map, we use the distance between the texture map and the individualized mosaic model for the recognition purpose. Using our mosaic model, we have observed satisfying tracking and recognition performance from video sequences with face images.

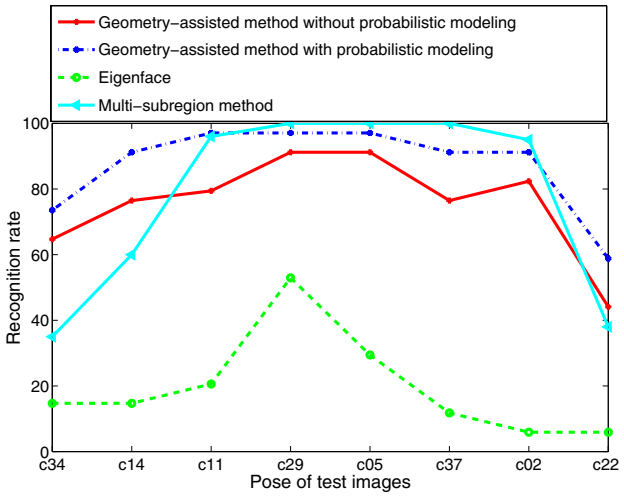


Figure 9: Recognition performances of four algorithms on the PIE database with one frontal training image. Our algorithm performs better than others, especially for the difficult scenario (poses closer to profile views).

# 6. Experimental results

We evaluate our algorithm by comparing its performance on the CMU PIE database with a traditional eigenface method [11]. We use half of the subjects (34 subjects) in the PIE database for training the probabilistic model as presented above. The 9 pose images per subject from remaining 34 subjects are used for the recognition experiments.

The frontal view image (c27) is used for the training, and the other 8 images are used for test. As shown in Figure 9, the horizontal axis represents the labels of 8 test poses, c34, c14, c11, c29, c05, c37, c03, c22, from the right profile to the left profile. The vertical axis shows the recognition rate of four algorithms for each specific pose. The first is the traditional eigenface approach [11], where the nearest neighbor classifier is applied. We manually crop the human face for both the training and test images, and normalized them to the size of 64 by 64 pixels. Since there are 34 training images in total, it is possible to use an eigenspace whose number of eigenvectors varies from 1 to 33. We test all these possibilities and plotted the one with the best recognition performance. The second algorithm is our geometry assisted method without probabilistic modeling, which is presented in Section 3. The third algorithm is the geometry assisted method with probabilistic modeling.

A number of observations are made from these results. First, when the pose of the test image is more toward the profile view, the recognition rate decreases. Second, both our algorithms perform much better than the baseline algorithm. Third, the geometry assisted method with probabilistic modeling works better than the one without probabilistic modeling. We can see that with one training image,

our algorithm presents satisfying recognition performance: it recognizes all face views with more than 90% correct rate except the two most extreme profile views. Even for the two profile views, around 70% and 60% recognition rates are achieved.

We also plot the results of the multi-subregion method reported in [9]. We can see that the performance of our algorithm is comparable with the multi-subregion method for test images closer to the frontal view. For test images closer to profile views, our algorithm performs noticeably better. For example, in their report, the recognition rates of two profile views are both lower than 40%. There are a few reasons why our method works better for profile views. One is that we utilize more appearance information instead of only using the area bounded by facial features, such as eyes and the mouth, as done in [9]. Also, the geometrical mapping greatly compensates the pose variation and reduces the intra-subject variations.

# 7. Conclusions

In this paper we introduce a probabilistic geometry assisted approach and apply it to pose-robust face recognition. All training and test images are projected to the surface of a 3D ellipsoid by estimating the pose and position information, and represented as texture maps. The distance measure is conducted in the overlap area between any two texture maps. Also by representing the texture map as an array of local patches, it enables us to develop a probabilistic model for the distance measures of patches from a face database with pose variation. Eventually we are able to utilize the Bayesian framework to evaluate the distance measure of corresponding patches. Comparing it with the existing algorithms, we observe significant improvement when performing experiments on the CMU PIE database.

Suppose we only have one frontal training image for one subject. Can we estimate/anticipate the profile of this subject? We can approach this problem by studying the relationship between the patches from the frontal view and the patches from the profile view. Our algorithm has already modeled the statistical of the within-patch appearance, and we can extend it by modeling statistics of the between-patch.

# References


[1] Wen-Yi Zhao, Rama Chellappa, P.J. Jonathon Phillips, and Azriel Rosenfeld, "Face Recognition: A Literature Survey", *ACM Computing Survey*, Vol. 35, No 4, pp. 399-458, 2003.

[2] Ralph Gross, Jianbo Shi, and Jeff Cohn, "Quo Vadis Face Recognition", *Third Workshop on Empirical Evaluation Methods in Computer Vision*, 2001.

[3] P.J. Phillips, P. Grother, R.J Micheals, D.M. Blackburn, E Tabassi, and J.M. Bone, "FRVT 2002: Evaluation Report", March 2003.

[4] Yongmin Li, Shaogang Gong and Heather Liddell, "Recognizing the Dynamics of Faces Across Multiple Views", *In Proc. British Machine Vision Conference*, Bristol, England, pp. 242-251, September 2000.

[5] Xiaoming Liu and Tsuhan Chen, "Video-Based Face Recognition Using Adaptive Hidden Markov Models", *In the Proceeding of the IEEE International Conference on Computer Vision and Pattern Recognition* 2003, Madison, Wisconsin, 2003.

[6] Hyung-Soo Lee and Daijin Kim, "Pose Invariant Face Recognition Using Linear Pose Transformation in Feature Space", *ECCV 2004 Workshop on Computer Vision in Human-Computer Interaction*, Czech Republic, 2004.

[7] K. Okada and C. von der Malsburg, "Pose-Invariant Face Recognition with Parametric Linear Subspaces", *In Proc. of Fifth International Conference on Automatic Face and Gesture Recognition*, Washington DC, pp. 64-69, 2002.

[8] Simon Lucey and Tsuhan Chen, "A GMM Parts Based Face Representation for Improved Verification through Relevance Adaptation", *Proc. of IEEE Conference on Computer Vision and Pattern Recognition 2004*, Washington, DC, 2004.

[9] T. Kanade and A. Yamada, "Multi-subregion Based Probabilistic Approach toward Pose-invariant Face Recognition", *Proc. of 2003 IEEE International Symposium on Computational Intelligence in Robotics Automation*, Vol. 2, Kobe, Japan, pp.954-959.

[10] V. Blanz and T. Vetter, "Face Recognition Based on Fitting a 3D Morphable Model", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 9, 2003.

[11] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp.71-86, 1991.

[12] O. D. Faugeras, "*Three-Dimensional Computer Vision*", MIT Press, Cambridge, 1993.

[13] A. K. Jain, "*Fundamentals of Digital Image Processing*", Prentice Hall, Englewood Cliffs, NJ, 1989.

[14] George Wolberg, "*Digital image warping*", published by IEEE Computer Society Press, 1990.

[15] Michael Isard and Andrew Blake, "*Active Contours*", Springer-Verlag, 1998.

[16] Alex Pentland, Baback Moghaddam, and Thad Starner, "View-Based and Modular Eigenspaces for Face Recognition", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition* 1994.

[17] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, pp. 1615 - 1618, 2003.

[18] R.O. Duda, P.E. Hart and D.G. Stork, "*Pattern Classification*", 2nd edition. John Wiley & Sons. Inc., New York, 2001.

[19] J Kittler, M Hatef, R P W Duin, and J Matas, "On combining classifiers", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp. 226-239, 1998.