Fully Automatic Pose-Invariant Face Recognition via 3D Pose Normalization

Akshay Asthana[1], Tim K. Marks, Michael J. Jones, Kinh H. Tieu[2] and Rohith MV[3]

Mitsubishi Electric Research Laboratories, Cambridge, MA, USA

aasthana@rsise.anu.edu.au, {tmarks,mjones}@merl.com, kinh.tieu@gmail.com, rohithmv@gmail.com

Abstract

An ideal approach to the problem of pose-invariant face recognition would handle continuous pose variations, would not be database specific, and would achieve high accuracy without any manual intervention. Most of the existing approaches fail to match one or more of these goals. In this paper, we present a fully automatic system for pose-invariant face recognition that not only meets these requirements but also outperforms other comparable methods. We propose a 3D pose normalization method that is completely automatic and leverages the accurate 2D facial feature points found by the system. The current system can handle 3D pose variation up to $\pm 45^\circ$ in yaw and $\pm 30^\circ$ in pitch angles. Recognition experiments were conducted on the USF 3D, Multi-PIE, CMU-PIE, FERET, and FacePix databases. Our system not only shows excellent generalization by achieving high accuracy on all 5 databases but also outperforms other methods convincingly.

1. Introduction

We present a method for improving the accuracy of a face recognition system in the presence of large pose variations. Our approach is to *pose-normalize* each gallery and probe image, by which we mean to synthesize a frontal view of each face image. We present a novel 3D pose-normalization method that relies on automatically and robustly fitting a 3D face model to a 2D input image without any manual intervention. Furthermore, our method of pose normalization handles a continuous range of poses and is thus not restricted to a discrete set of predetermined pose angles. Our main contribution is a fully automatic system for pose-normalizing faces that yields excellent results on standard face recognition test sets. Other contributions include the use of pose-dependent correspondences between 2D landmark points and 3D model vertices, a method for 3D pose estimation based on support vector regression, and the use of face boundary detection to improve AAM fitting.

To achieve full automation, our method first uses a robust method to find facial landmark points. We use Viola-Jones-type face and feature detectors (Section 3) along with face boundary finding (Section 4.2) to accurately initialize a View-Based Active Appearance Model (VAAM) (Section 4). After fitting the VAAM, we have a set of 68 facial landmark points. Using these points, we normalize the roll angle of the face and then use a regression function to estimate the yaw and pitch angles (Section 5). The estimated pose angles and facial landmark points are used to align an average 3D head model to the input face image (Section 6.1). The face image is projected onto the aligned 3D model, which is then rotated to render a frontal view of the face (Section 6.2). All gallery and probe images are pose-normalized in this way, after which we use the Local Gabor Binary Pattern (LGBP) recognizer [27] to get a similarity score between a gallery and probe image (Section 7). The entire system is summarized in Figure 1.

2. Related Research

Other papers have also explored the idea of pose normalization to improve face recognition accuracy. Examples include Chai et al. [8], Gao et al. [12], Du and Ward [10], and Heo and Savvides [15]. Unlike our method, none of these previous methods has the dual advantages of being fully automatic and working over a continuous range of poses. Chai et al. learn pose-specific locally linear mappings from patches of non-frontal faces to patches of frontal faces. Their method only handles a discrete set of poses and requires some manual labeling of facial landmarks. Gao et al. use a single AAM to fit non-frontal faces but also require manual labeling. Du and Ward require a set of prototype non-frontal face images that are in the same pose as the input non-frontal face. Heo and Savvides use a similar approach to ours for locating facial feature points but use 2D affine warps instead of our more accurate 3D warps and ap-

[1]currently at Australian National University, Canberra, ACT, Australia
[2]currently at Heartland Robotics, Boston, MA, USA
[3]currently at Dept. of Computer Science, University of Delaware, USA

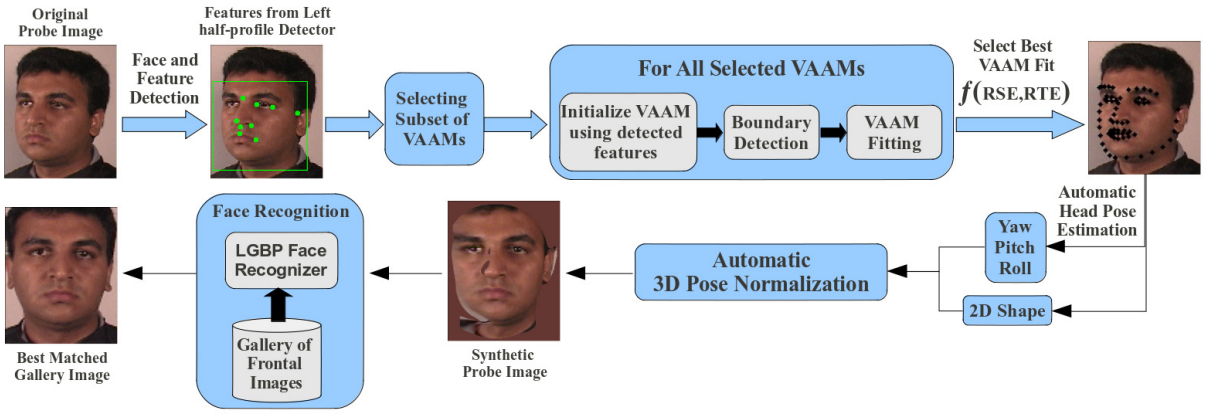

Figure 1: Overview of our fully automatic pose-invariant face recognition system.

parently rely on manual initialization. Sarfraz et al. [20, 22] present an automatic technique for handling pose variations for face recognition, which involves learning a linear mapping from the feature vector of a non-frontal face to the feature vector of the corresponding frontal face. Their assumption that the mapping from non-frontal to frontal feature vectors is linear seems overly restrictive. Not only does our system remove the restrictions of these previous methods, it also achieves better accuracy on the CMU-PIE [23] and FERET [19] databases. Blanz and Vetter [5] use a 3D Morphable Model to fit a non-frontal face image and then synthesize a frontal view of the face, which is similar to our approach. However, our appearance-based model fitting is done in 2D instead of 3D, which makes it both more robust and much more computationally efficient. Furthermore, the 3D model we use does not involve texture and can be efficiently and reliably aligned to the fitted 2D facial feature points. In addition, whereas [5] relied on manual marking of several facial feature points, we automatically detect an initial set of facial feature points that ensure good initialization for the 2D model parameters. Breuer et al. [6] present a method for automatically fitting the 3D Morphable Model, but it has a high failure rate and high computational cost.

3. Face and Feature Detection

The face and feature detectors we use are Viola-Jones-type cascades of Haar-like features, trained using AdaBoost as described in [25]. To detect faces with yaw angles from $-60°$ to $+60°$ and pitch angles from $-30°$ to $+30°$, we train three face detectors: a frontal detector that handles yaw angles of roughly $-40°$ to $+40°$, a left half-profile detector that handles yaw angles of roughly $30°$ to $60°$, and a right half-profile detector that handles yaw angles of roughly $-30°$ to $-60°$. Each of these also handles pitch angles from roughly $-30°$ to $+30°$. For speed, we also trained an initial "gating" face detector on all views from $-60°$ to $+60°$. This gating detector is fast, with a very high detection rate but also a high false positive rate. If an image window is classified as a face by the gating detector, it is then passed to each of the three view-specific face detectors in sequence. The gating detector greatly increases the speed of the multi-view detector with a very small effect on accuracy. For each image window detected as a face by the multi-view detector, the rough pose class (left half-profile, frontal, or right half-profile) is also returned.

We also trained Viola-Jones-style detectors to detect facial features such as the eye corners. We have 9 different detectors for each of the three views (frontal, left half-profile, and right half-profile). The detected features for each view are illustrated in Figure 2. Each feature detector is trained using a set of positive image patches that includes about a quarter of the face surrounding the feature location. Unlike in face detection, the training patches for each feature are carefully aligned so that the feature location is at the exact same pixel position in every patch. All of the face and feature detectors are trained once on a large training set of manually labeled positive and negative image patches taken from random Web images, and they are thus very general.

Figure 2: Ground truth feature locations for right half-profile, frontal, and left half-profile faces.

4. Automatic Extraction of Landmark Points

Our system uses the Active Appearance Model (AAM) framework to find the 2D locations of landmark points in face images. Originally proposed by Cootes et al. [11], an AAM is generated by applying principal component analysis (PCA) to a set of labeled faces in order to model the intrinsic variation in shape and texture. This results in a

parametrized model that can represent large variation in shape and texture with a small set of parameters.

Fitting an AAM to a new image is generally accomplished in an iterative manner and requires accurate model initialization to avoid converging to bad local minima. Good initialization is particularly important when there is large pose variation.

4.1. Training of View-Based AAMs

In order to make the model fitting procedure robust to pose-variation, we use a View-Based AAM (VAAM) approach [9], in which the concept of a single AAM that covers all pose variations is replaced by several smaller AAMs, each of which covers a small range of pose variation. The benefits are twofold. First, the overall robustness of the fitting procedure is improved because a particular VAAM's mean shape is closer to the shapes of the faces in its range of pose variation than the mean shape of a single AAM would be. Second, the amount of shape and texture variation that is caused by changes in face pose is significantly less for a VAAM than it would be for a single, global AAM. In addition to reducing the problem of spurious local minima, VAAMs also increase the speed of model convergence.

The system presented in this paper covers poses with yaw angles from $-45°$ to $+45°$ and pitch angles from $-30°$ to $+30°$. The VAAMs in this range were trained using data from the USF Human ID 3D database [4] and the Multi-PIE database [14]. From the Multi-PIE database, we used the data of 200 people in poses 05_1, 05_0, 04_1, 19_0, 14_0, 13_0, and 08_0 to capture the shape and texture variation induced by changes in pose, and the data of 50 people in 18 different illumination conditions to capture the texture variation induced by different illumination conditions. In order to extract the 2D shapes (68 landmark point locations) for all 100 subjects from the USF 3D database, the 3D mean face was hand labeled in 199 different poses (indicated by $\times$'s in Figure 3) to determine which 3D model vertex in each pose corresponds to each of the 68 landmark points. These vertex indices were then used to generate the 2D locations of all 68 points in each of the 199 poses for all 100 subjects in the USF 3D database. Generating 2D data from 3D models in this way enables us to handle extreme poses in yaw and pitch accurately. This would not be possible using only 2D face databases for training, both because they do not have data for most of the poses marked in Figure 3 and because manual labeling would be required for each individual image. Whereas the VAAM shape models were trained on both the USF 3D and Multi-PIE data, the VAAM texture models were trained only on the Multi-PIE data.

Given a test image, we use the rough pose class determined by the face detector (see Section 3) to select a subset of VAAMs that cover the relevant pose range. To initialize each selected VAAM, we use the Procrustes method [13] to

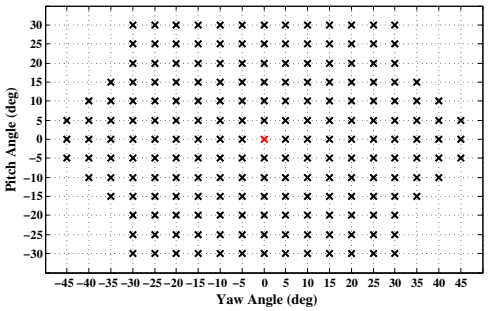

Figure 3: The VAAM shape models were trained by rotating all 100 USF 3D faces to the 199 poses indicated by $\times$'s.

find the 2D scale, rotation, and translation that best align the VAAM's mean shape with the detected facial feature points. To assist the VAAMs that cover extreme poses, we use face boundary detection to augment the fitting procedure, as described in Section 4.2. The VAAM initialization and fitting procedure is discussed in Section 4.3.

4.2. Face Boundary Extraction

Although Procrustes analysis gives a good overall initialization of the AAM, points on the side of the face are often far from the actual face boundary because of significant variation in facial structure across individuals. To mitigate this problem, we perform an additional step of extracting the boundaries of the face in an image. First, we restrict the boundary to lie inside a rectangle roughly bounded on the top, bottom, and side by the eyebrow, chin, and nose tip AAM points, respectively (see Figure 4). The problem is formulated as that of detecting a curve running from the top row of the rectangle to the bottom row that optimizes a combination of edge strength and smoothness [18]. The curve is defined by pixel coordinates (x_i, y_i), where y_i is the row index. The optimization problem is

$$\min_{\{x_i\}} \sum_i g(x_i, y_i) + \sum_i d(x_i, x_{i-1}), \qquad (1)$$

where g is the inverse of the image gradient magnitude and d constrains x_{i-1} and x_i to be within one pixel. The optimal curve is found by dynamic programming as in seam carving [2], except here we maximize instead of minimize edge strength. We then update the coordinates of an AAM point (X, Y) on the face boundary to (x_i, Y) using the curve point (x_i, y_i) whose row coordinate y_i is closest to Y.

4.3. Fitting via VAAMs

The rough pose class provided by the multi-view face detector is not accurate enough to identify the single best view-specific AAM to use for fitting. Thus, we fit a subset of VAAMs and select the one that yields the best fit. To initialize fitting for each VAAM, we use the detected facial feature points and extracted boundary points to re-align

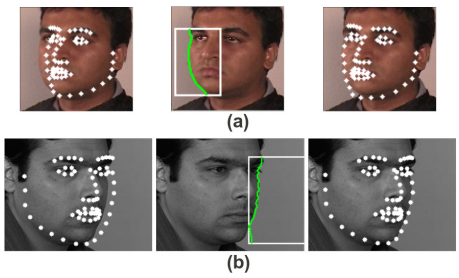

Figure 4: Face Boundary Extraction. Initial AAM fit (left) is used to constrain the search for the face boundary (middle), which is then used to generate refined AAM fit (right).

the VAAM mean shape with the face via Procrustes analysis and initialize the VAAM shape parameters. Our system uses the Efficient Approximation of Simultaneous Inverse Compositional (SIC-EA) method for face fitting [3, 21]. While standard SIC is accurate but extremely slow, SIC-EA is computationally efficient but not robust to large texture variations. Leveraging the accurate model initialization obtained above, we circumvent this problem by initializing the model texture parameters using the image texture under the current initial warp, rather than starting the fitting procedure from the mean texture. This ensures that the fitting procedure begins with reasonable initial texture parameters.

All the selected VAAMs are fitted independently of each other, and the best fit is chosen based on the final match score, which is computed as a function of the residual shape error (RSE) and residual texture error (RTE). Residual shape error is computed as the root-mean-squared error (RMSE) between the vector of points used for model initialization and the corresponding points in the converged model shape. Residual texture error is computed as the RMSE between the converged model texture and the actual texture of the image, warped according to the converged model shape.

5. Automatic Head Pose Estimation

Estimating the 3D head pose from a single 2D image automatically is a key step in our approach to pose-invariant face recognition. Our method for estimating the pose angles from the VAAM shape parameters is based on Support Vector Regression (SVR) [24], learning a general regression function from accurate training data that we generate using a large set of 3D face models. After the SVR has been trained, we estimate the pose of a new face image by using the best VAAM fitting as input to the SVR model.

5.1. Training SVR for pose estimation

Our head pose estimation system was trained only on the 2D data generated from the USF 3D database (see Section 4.1). A set of shape vectors, representing 68 2D landmark points on the face, was generated by rendering 100 face models at 199 different poses (Figure 3). These 19,900 shape vectors were used to create a shape model by aligning them via Procrustes analysis and applying PCA for dimensionality reduction, just like training a shape model for AAM [11]. We use the first 5 shape parameters to train the pose estimator using SVR.

For our SVR training set, we augment the ground truth shape parameters described above with the shape parameters obtained by fitting the correct VAAM (see Section 4) to each of the 19,900 rendered USF images. This has two main advantages. First, it doubles the amount of training data at our disposal. Second, and more important, it helps to model the noisy shape parameters that result from fitting 2D models automatically to face images. To estimate yaw and pitch angles independently, we train two separate SVMs, both using the Gaussian Radial Basis Function (RBF) kernel.

5.2. Pose Estimation Results

We tested the accuracy of our pose estimation system on the USF Human ID 3D database [4] and FacePix(30) database [17]. Since our SVR was trained using the USF 3D data, our USF pose estimation tests use a 5-fold cross-validation scheme. Table 1 gives the mean error obtained for yaw and pitch angle across all 199 poses of 100 subjects.

For FacePix, we trained the pose estimator on all of the data obtained from the USF 3D database. Since our pose estimator covers the range of $-45°$ to $+45°$ in yaw, our test set from FacePix consisted of all 91 images within that range (every $1°$) for each of the 30 subjects. Table 1 shows our pose estimator's overall mean error in yaw and pitch.

Mean Error	USF 3D database	FacePix Database
Yaw	2.61°	3.96°
Pitch	4.66°	2.84°

Table 1: Pose estimation Errors on USF 3D and FacePix.

6. 3D Pose Normalization

Our pose normalization method, outlined in Figure 6, utilizes both the 2D VAAM fitting (see Section 4) and the 3D pose estimate (see Section 5) of each input image to align a 3D face model (the mean face shape from the USF 3D database [4]) to the input image. We then project the input image onto the 3D face model to create a textured 3D model, on which we perform a 3D rigid transformation to pose-normalize the face into the canonical frontal pose.

One problem with AAM-based pose synthesis methods such as [1, 12, 15] is that even if the AAM points themselves are moved from the correct location in one view to the correct location in another view, the rest of the points on the face surface (the points in the interiors of the triangles whose vertices are the AAM points) will not be warped correctly. The points in the interiors of the AAM triangles are typically warped in a piecewise-affine fashion (linear within each triangle), whereas the true 2D projection of the 3D ro-

tation of a face surface corresponds to a nonlinear warp in 2D. Since we synthesize the frontal pose from a non-frontal pose via a 3D rigid rotation of a 3D face model, our method is not subject to the limitations of linear 2D warping.

6.1. Aligning the 3D Face Model with a 2D image

Perhaps a larger problem for AAM-based pose synthesis methods is that the correspondence between AAM points and points on the 3D surface of a face changes depending on the pose. For a single AAM that is correctly fitted to a single face in two different poses, the points of the AAM in the first pose and the same points of the AAM in the second pose can actually correspond to very different points on the surface of the 3D face. In particular, AAM points that reside on the left and right visible boundaries of the face will correspond to quite different points on the surface of the face when the head is pointing to the left versus to the right. This is illustrated in Figure 5, which shows the same 3D face in three different poses: yaw = $-45°, 0°$, and $45°$. The 3D model vertices corresponding to the AAM boundary points that fit the left pose are shown in green in all three images; the 3D model vertices corresponding to the AAM boundary points that fit the right pose are shown in red in all three images. Notice that the same AAM points correspond to different points on the 3D face model, depending on the pose of the head fitted by the AAM.

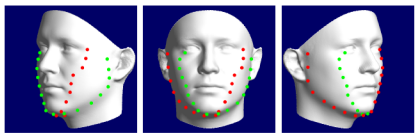

Figure 5: AAM to 3D model correspondence is pose-dependent.

This pose-dependency of the correspondence between points of a 2D AAM and points of a 3D face model will cause inaccuracies in any AAM-based pose synthesis method (such as [1, 12, 15]) that warps the texture from an AAM fitted point in one view to the corresponding AAM point's location in a different view. It would also cause inaccuracies in a model based on a 2D+3D AAM [26], which constructs its 3D model based on the assumption that the same AAM point in different views corresponds to a single point on the 3D model.

We used our hand-labeling of the mean USF face in 199 poses (Section 4.1) to generate a lookup table that, for each pose represented by an $\times$ in Figure 3, tells which vertex of the 3D head model corresponds to each of the 68 points of our VAAM. Given an input image, our system takes the estimated pose (Section 5), finds the nearest pose in the lookup table, and uses that pose to obtain the 3D model vertices that correspond to the VAAM fit of the input image.

Next, we find the 3D rigid transformation $\mathbf{T}$ of the mean 3D head model that minimizes the squared distance between the 2D point locations fitted by the VAAM and the 2D projection of the corresponding 3D model vertices.

Throughout this paper, *pitch* refers to a rotation of r_x° about the x-axis, given by a rotation matrix $\boldsymbol{R}_x$. Similarly, *yaw* corresponds a rotation of r_y° about the y-axis (given by rotation matrix $\boldsymbol{R}_y$), and *roll* corresponds to a rotation of r_z° about the z-axis (matrix $\boldsymbol{R}_z$). (For a head in frontal position, the x-axis is parallel to a line passing through the left and right ears, and the y-axis runs parallel to a vertical line through the center of the head.) The overall rotation $\boldsymbol{R}$ of a vector $\boldsymbol{x} = [x\ y\ z]^{\mathrm{T}}$ is the following matrix product of these three component rotations: $\boldsymbol{R}\boldsymbol{x} = (\boldsymbol{R}_z\boldsymbol{R}_x\boldsymbol{R}_y)\boldsymbol{x}$.

The mean 3D head model has N vertices, of which $n = 68$ correspond to the VAAM landmark points. Their 3D locations in canonical frontal pose are given by the $3 \times n$ matrix $\boldsymbol{H}$. The weak perspective projection is expressed using a scale k, the projection matrix $\boldsymbol{P} = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right]$, and a $2 \times n$ translation matrix $\boldsymbol{D} = \left[\begin{smallmatrix} d_x & d_x & d_x & \cdots \\ d_y & d_y & d_y & \cdots \end{smallmatrix}\right]$, which simply consists of n copies of the 2D translation vector $[d_x\ d_y]^{\mathrm{T}}$. To find the rigid 3D transformation $\mathbf{T} = \{\boldsymbol{R}, \boldsymbol{D}, k\}$ that best aligns the 3D head model to the 2D points, we solve the following nonlinear least squares optimization problem:

$$\underset{r_x, r_y, r_z, d_x, d_y, k}{\operatorname{argmin}} \left\| \boldsymbol{S} - (k\boldsymbol{P}\boldsymbol{R}\boldsymbol{H} + \boldsymbol{D}) \right\|_{\mathrm{F}}^2, \qquad (2)$$

where $\boldsymbol{S}$ is a $2 \times n$ matrix containing the 2D locations of the n fitted VAAM points, and $\| \cdot \|_{\mathrm{F}}^2$ is the Frobenius norm. We solve this optimization using the Levenberg-Marquardt method. Good initialization is required to prevent the optimization from ending in the wrong local minimum; we initialize using our 3D pose estimate (see Section 5).

6.2. Synthesizing the Frontal Pose

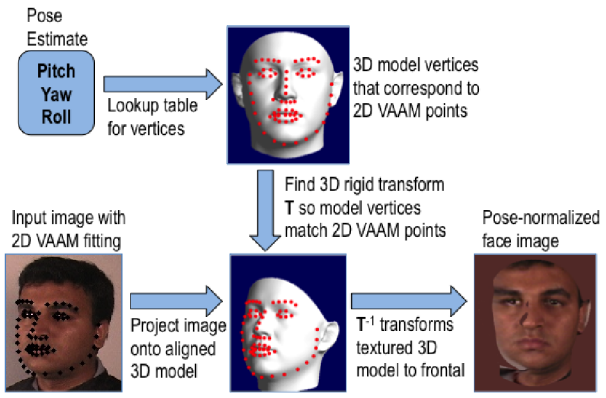

Figure 6: Overview of 3D pose normalization.

The 3D rigid transformation $\mathbf{T}$ transforms the mean 3D head model from its canonical frontal pose to a pose that optimally matches the 2D VAAM points that were fitted to the input image. In other words, we have aligned the

3D head model with the input image. Next, we project the image onto the 3D model to obtain a textured 3D model. We then apply the inverse 3D rigid transformation, $\mathbf{T}^{-1}$, to transform this textured 3D model back to the frontal pose. Figure 7 gives examples of images from several data sets that have been pose-normalized using our fully automatic method. In each part there are two rows and five columns of images. Each image in the top row is an original input image, and the image beneath it in the bottom row is the synthetic frontal face that results from our pose normalization. The left column in each part shows the frontal (gallery) image, while the other 4 columns show probe images of the same person in various poses. In a face recognition experiment, the pose-normalized frontal image is compared with the pose-normalized images from other poses.

7. Local Gabor Binary Patterns

We have chosen to use the Local Gabor Binary Pattern (LGBP) recognizer [27] for comparing two face images. Briefly, this recognizer works by computing histograms of oriented Gabor filter responses over a set of non-overlapping regions that tile an input image. The concatenation of all histogram bins forms a feature vector. Two feature vectors are compared by summing the histogram intersections over all the bins. This yields a value indicating how similar the two face images are. The LGBP recognizer has the advantage that there are no parameters to set, so no training is involved, and yet its performance is comparable to other state-of-the-art recognizers on many test sets.

8. Recognition Experiments and Results

We conducted face recognition experiments on the USF Human ID 3D [4], Multi-PIE [14], CMU-PIE [23], FERET [19], and FacePix(30) [17] databases. The CMU-PIE and FERET databases are the most commonly used databases for face recognition across pose variation, so they are best for comparison with previous approaches. The experiments on FacePix highlight the ability of our system to handle continuous pose variation accurately. The USF database enables us to generate images at any 3D pose, demonstrating our system's ability to handle combined pose variation in yaw and pitch. Multi-PIE is the most recent of the databases, and our experiments on this challenging data set will facilitate comparisons with future methods.

Given a test image, our system automatically detects the face and facial features (Section 3) that will be used to initialize and robustly fit the 2D VAAM (Section 4). If no face or fewer than 3 facial features are detected, a Failure To Acquire (FTA) has occurred, and no pose-normalized face image is output. For all other images, the extracted 2D shape is used to compute the shape parameters that are used for head pose estimation (Section 5). This head pose

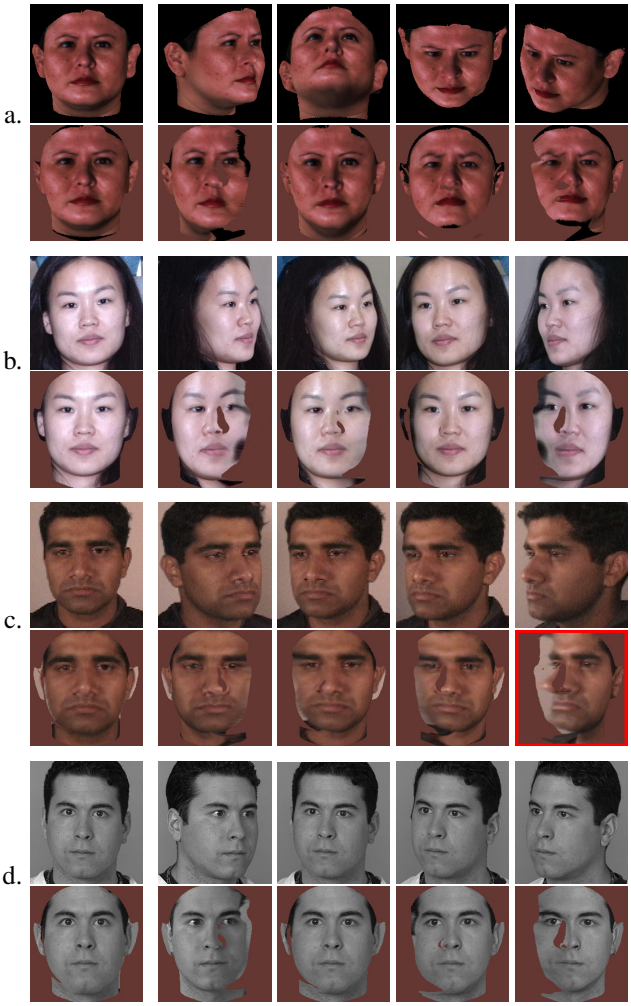

Figure 7: Examples of 3D Pose Normalization from (a) USF 3D, (b) Multi-PIE, (c) FacePix, and (d) FERET. Each part's top row contains the input images, and the bottom row contains the corresponding pose-normalized images. The left column contains a gallery image; the other columns contain example probe images. Errors, as in the red box in part (c), can occur for extreme poses.

information, along with the 2D VAAM landmark locations, is used by our 3D pose normalization system (Section 6) to generate the synthetic frontal face image that is passed to the LGPB recognizer (Section 7) to compare with all the gallery images. The result is a robust pose-invariant face recognition system that requires absolutely no manual intervention. The entire fitting and pose normalizing process takes about 6 seconds on a modern Pentium processor.

All automatic systems have the issue of FTAs. In each of the recognition experiments described below, we report the percentage of FTA cases. FTA cases are removed from the test set and thus are not counted as recognition errors. This allows us to clearly distinguish between failures of our detector and failures of our recognizer.

CMU-PIE

Method	Alignment	Trained on PIE	Gallery/Probe Size	Poses Handled	c11 -45°	c29 -22.5°	c07 up 22.5°	c09 down 22.5°	c05 +22.5°	c37 +45°	**Avg**
Kanade03*[16]	manual	yes	34	discrete set	96.8	100.0	100.0	100.0	100.0	100.0	99.5
Chai07* [8]	manual	no	68	discrete set	89.8	100.0	98.7	98.7	98.5	82.6	94.7
Castillo07 [7]	manual	no	68	continuous	100.0	100.0	90.0	100.0	100.0	99.0	98.2
Sarfraz10*[20, 22]	automatic	yes	34	continuous	87.9	89.2	99.8	92.8	91.5	87.9	91.5
Sarfraz10*[20, 22]	automatic	no	68	continuous	84.0	87.0	–	–	94.0	90.0	88.8
LGBP [27]	automatic	no	67	N/A	71.6	87.9	78.8	93.9	86.4	75.8	82.4
Ours	automatic	no	67	continuous	98.5	100.0	98.5	100.0	100.0	97.0	99.0

FERET

Method	Alignment	Trained on FERET	Gallery/Probe Size	Poses Handled	bh -40°	bg -25°	bf -15°	be +15°	bd +25°	bc +40°	**Avg**
Gao09[12]	manual	yes	200	continuous	78.5	91.5	98.5	97.0	93.0	81.5	90.0
Asthana09[1]	manual	yes	200	discrete set	87.0	93.0	98.0	98.5	95.5	74.0	91.0
Sarfraz10*[20]	automatic	yes	200/100	continuous	92.4	89.7	100.0	98.6	97.0	89.0	94.5
LGBP [27]	automatic	no	200	N/A	62.0	91.0	98.0	96.0	84.0	51.0	80.5
Ours	automatic	no	200	continuous	90.5	98.0	98.5	97.5	97.0	91.9	95.6

Multi-PIE

Method	080_05 -45°	130_06 -30°	140_06 -15°	051_07 0°	050_08 +15°	041_08 +30°	190_08 +45°	**Avg**
LGBP [27]	37.7	62.5	77.0	92.6	83.0	59.2	36.1	64.0
Ours	74.1	91.0	95.7	96.9	95.7	89.5	74.8	87.7

FacePix

Method	Left 45°−31°	30°−16°	15°−1°	Right 1°−15°	16°−30°	31°−45°	**Avg**
LGBP [27]	30.9	58.9	93.5	93.3	68.0	40.3	64.2
Ours	71.6	90.0	97.3	95.8	92.7	74.8	87.0

Table 2: Pose-wise rank-1 recognition rates (%) for CMU-PIE, FERET, Multi-PIE, and FacePix databases. The numbers for the starred(*) methods were estimated from plots in [16, 8, 22]. To get LGBP baseline results, we first performed 2D alignment using our automatic feature detectors, then used code from the authors of [27].

USF 3D Database : We rendered images of all 94 unique subjects of the USF Human ID 3D database [4] at 199 different poses (Figure 3), ranging from $+45°$ to $-45°$ in yaw angle and $+30°$ to $-30°$ in pitch angle, for our recognition test. The frontal image for each subject was used as a gallery image (94 total), and the remaining images (18,612) were used as probes. The FTA rate was 3.37%. (This rate of detection failure is higher than on other data sets due to the combinations of extreme yaw and pitch included in this set.) The overall rank-1 recognition rate obtained by our system on this test set was **98.8%**. Table 3 shows a pose-wise breakdown of recognition accuracy.

Pitch Range (°)	-15 to +15		-30 to -20 and +20 to +30	
Yaw Range (°)	LGBP	**Ours**	LGBP	**Ours**
-15 to +15	97.1	99.7	84.4	98.7
-30 to -20 and +20 to +30	88.8	99.4	67.2	98.9
-45 to -35 and +35 to +45	78.3	97.4	–	–

Table 3: Pose-wise rank-1 recognition rates (%) for USF 3D

Multi-PIE Database : For our recognition experiment, we used 137 subjects (*Subject ID 201 to 346*) with neutral expression from all 4 sessions at 7 different poses, with illumination that is frontal with respect to the face (see Table 2). Note that 200 subjects (*Subject ID 001 to 200*) were used for training the VAAM (Section 4.1) and were therefore not used for the recognition experiments. The frontal image (*Pose ID 051*) from the earliest session for each subject was used as the gallery image (137 total), and all of the remaining images per subject (including frontal images from other sessions) were used as probes (1,963 total). The FTA rate was 1.2%. The overall rank-1 recognition rate obtained by our system on this test set was **87.7%**. Table 2 shows a pose-wise breakdown of recognition accuracy.

CMU-PIE Database : We used all 68 subjects with neutral expression at 7 different poses (see Table 2) for our recognition experiment. The frontal image (*Pose ID c27*) for each subject was used as the gallery image (68 total) and the remaining 6 images per subject were used as probes (408 total). One of the FTA cases was the gallery image for one subject (*Subject ID 04021*). Thus there was no gallery image for that subject, so we removed that subject from our results. We used the remaining 67 subjects for our recognition test, which had an FTA rate of 1.1%. Our system's overall rank-1 recognition rate on this set was **99.0%**. Table 2 shows a pose-wise breakdown of recognition accuracy.

FERET Database : We used all 200 subjects at 7 different poses (see Table 2) for our recognition experiment. Frontal image *ba* for each subject was used as the gallery image (200 total) and the remaining 6 images per subject were used as probes (1,200 total). The FTA rate was 0.29%. Our system's overall rank-1 recognition rate was **95.6%**. Table 2 shows a pose-wise breakdown of recognition accuracy.

FacePix Database : This test set contains images of 30 subjects with yaw angle ranging from $+90°$ to $-90°$, at $1°$ intervals. Since our current system can handle poses ranging from $+45°$ to $-45°$ in yaw angle, we used only these images for our recognition test. The frontal ($0°$) image for each subject is used as the gallery image (30 total), and the

remaining 90 images per subject, with pose ranging from $+45°$ to $-45°$ in yaw angle, were used as probe images (2,700 total). The FTA rate was 0.62%, and our system's overall rank-1 recognition rate was **87.0%**. Table 2 shows a pose-wise breakdown of recognition accuracy.

Summary of Results : Our results show that our system acheives state-of-the-art recognition for data sets with wide pose variation. Unlike most previous methods, our system is fully automatic, handles continuous pose variation, and generalizes well to unseen data sets. The only other method with similar properties, Sarfraz et al. [20, 22], performs significantly worse than ours on CMU-PIE and slightly worse on FERET (despite their advantage of having trained on part of FERET). We significantly outperform other methods on FERET, which is a well-tested data set. On CMU-PIE, Kanade and Yamada [16] perform 0.5% better than us, but they train on part of CMU-PIE, have only 34 gallery subjects (as opposed to our 67), and require manual alignment. Castillo and Jacobs [7] perform slightly worse than us but require manual alignment. Our LGBP baseline results on Multi-PIE and FacePix show that they are more difficult test sets, and yet our system shows even greater improvement over the baseline on these.

9. Discussion

We presented a fully automatic system for pose-invariant face recognition that handles a wide range of poses and achieves excellent results on five publicly available databases. Our proposed method for 3D pose normalization leverages accurate 2D feature points provided by our 2D model fitting system, which makes it computationally very efficient compared to other 3D model-based methods [5]. Moreover, our system is designed to handle continuous pose variation, unlike 2D pose-normalization methods such as [8, 1] that assume a fixed set of discrete poses for the probe images. A detailed comparison with several previous methods is given in Table 2. Note that unlike many of the comparison methods tested on the CMU-PIE and FERET databases (for example, Sarfraz et al. [20, 22] that use half of the FERET database for training and half for testing), we treat both CMU-PIE and FERET databases as completely unseen databases. Our method did not use either of these databases in the training phase. Nonetheless, our method outperforms these previous methods on the same data sets.

In future work, we plan to extend the system to an even wider range of poses. In more extreme poses, large occluded regions can be exposed after rotation to frontal, in which case additional strategies such as texture synthesis, image completion, or inpainting might be helpful.

References

[1] A. Asthana, C. Sanderson, T. Gedeon, and R. Goecke. Learning-based face synthesis for pose-robust recognition from single image. In *BMVC09*, 2009. 4, 5, 7, 8

[2] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26, July 2007. 3

[3] S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 3. Technical report, RI, Carnegie Mellon University, USA, 2003. 4

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*. 1999. 3, 4, 6, 7

[5] V. Blanz and T. Vetter. Face Recognition Based on Fitting a 3D Morphable Model. *IEEE PAMI*, 25(9), 2003. 2, 8

[6] P. Breuer, K. Kim, W. Kienzle, B. Scholkopf, and V. Blanz. Automatic 3D Face Reconstruction from Single Images or Video. In *FG 2008*. 2

[7] C. D. Castillo and D. W. Jacobs. Using stereo matching with general epipolar geometry for 2D face recognition across pose. *IEEE PAMI*, 31(12), 2007. 7, 8

[8] X. Chai, S. Shan, X. Chen, and W. Gao. Locally linear regression for pose-invariant face recognition. *IEEE Trans. Image Proc.*, 16(7), 2007. 1, 7, 8

[9] T. Cootes, K. Walker, and C.J.Taylor. View-Based Active Appearance Models. In *Proc. FG'00*, 2000. 3

[10] S. Du and R. Ward. Component-wise pose normalization for pose-invariant face recognition. In *IEEE ICASSP*, 2009. 1

[11] G. Edwards, C. Taylor, and T. Cootes. Interpreting Face Images Using Active Appearance Models. In *FG 1998*. 2, 4

[12] H. Gao, H. K. Ekenel, and R. Stiefelhagen. Pose normalization for local appearance-based face recognition. In *Proc. 3rd Intl. Conf. on Advances in Biometrics*, 2009. 1, 4, 5, 7

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1989. 3

[14] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. In *FG*, 2008. 3, 6

[15] J. Heo and M. Savvides. Face recognition across pose using view based active appearance models (VBAAMs) on CMU Multi-PIE dataset. In *ICVS*, 2008. 1, 4, 5

[16] T. Kanade and A. Yamada. Multi-Subregion Based Probabilistic Approach Toward Pose-Invariant Face Recognition. In *IEEE CIRA*, 2003. 7, 8

[17] G. Little, S. Krishna, J. Black, and S. Panchanathan. A methodology for evaluating robustness of face recognition algorithms with respect to changes in pose and illumination angle. In *ICASSP*, 2005. 4, 6

[18] U. Montanari. On the optimal detection of curves in noisy pictures. *Commun. ACM*, 14:335–345, May 1971. 3

[19] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET Evaluation Methodology for Face Recognition Algorithms. In *IEEE PAMI*, pages 1090–1104, 2000. 2, 6

[20] M. Saquib Sarfraz and O. Hellwich. Probabilistic learning for fully automatic face recognition across pose. *Image Vision Comput.*, 28:744–753, May 2010. 2, 7, 8

[21] J. M. Saragih and R. Goecke. Learning AAM fitting through simulation. *Pattern Recognition*, 42(11):2628–2636, Nov. 2009. 4

[22] M. S. Sarfraz. *Towards Automatic Face Recognition in Unconstrained Scenarios*. PhD thesis, Technische Universität Berlin, 2008. 2, 7, 8

[23] T. Sim, S. Baker, and M. Bsat. The CMU Pose, Illumination, and Expression Database. *IEEE PAMI*, 25(12), 2003. 2, 6

[24] V. N. Vapnik. The Nature of Statistical Learning Theory. In *Springer-Verlag, New York, ISBN 0-387-94559-8*, 1995. 4

[25] P. Viola and M. Jones. Robust real-time face detection. *Intl. Journal of Computer Vision*, 57:137–154, 2004. 2

[26] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2D+3D active appearance models. In *CVPR*, 2004. 5

[27] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition. *IEEE ICCV*, 2005. 1, 6, 7