Augmenting Frontal Face Models for Non-Frontal Verification

Conrad Sanderson and Samy Bengio
IDIAP, Rue du Simplon 4, CH-1920 Martigny, Switzerland
conradsand @ ieee.org, bengio @ idiap.ch

Abstract

In this work we propose to address the problem of non-frontal face verification when only a frontal training image is available (e.g. a passport photograph) by augmenting a client's frontal face model with artificially synthesized models for non-frontal views. In the framework of a Gaussian Mixture Model (GMM) based classifier, two techniques are proposed for the synthesis: UBMdiff and LinReg. Both techniques rely on a priori information and learn how face models for the frontal view are related to face models at a non-frontal view. The synthesis and augmentation approach is evaluated by applying it to two face verification systems: Principal Component Analysis (PCA) based and DCTmod2 based; the two systems are a representation of holistic and non-holistic approaches, respectively.

Results from experiments on the FERET database suggest that in almost all cases, frontal model augmentation has beneficial effects for both systems; they also suggest that the LinReg technique (which is based on multivariate regression of classifier parameters) is more suited to the PCA based system and that the UBMdiff technique (which is based on differences between two general face models) is more suited to the DCTmod2 based system. The results also support the view that the standard DCTmod2/GMM system (trained on frontal faces) is less affected by out-of-plane rotations than the corresponding PCA/GMM system; moreover, the DCTmod2/GMM system using augmented models is, in almost all cases, more robust than the corresponding PCA/GMM system.

1. Introduction

In the context of *frontal* faces, recent approaches to face recognition (here we mean both identification and verification) are able to achieve very low error rates (e.g. [19]). A more realistic and challenging task is to verify a face at a non-frontal view when only one (frontal) training image is available (e.g. a passport photograph).

While the task of view-independent recognition has been addressed through the use of training images (for the person to be recognized) at multiple views (e.g. [22]), the much harder task of using only one training image has received relatively little attention (e.g. [2, 21]). While it is possible to use 3D approaches to address the single training image problem (e.g. [1, 17]), here we concentrate on extending two well understood 2D based techniques. In particular, we will extend the Principal Component Analysis (PCA) based approach (due to its ubiquitousness) [30] and the recently proposed DCTmod2 based approach [29]. In both cases we employ a Gaussian Mixture Model (GMM) based classifier [25], which is central to our extensions.

The PCA/GMM system is an extreme example of a holistic system where the spatial relation between face characteristics (such as the eyes and nose) is rigidly kept (with the advantage of robustness to compression artefacts & additive noise [28]). Conversely, the DCTmod2/GMM approach is an extreme example of a non-holistic approach; here, the spatial relation between face characteristics is effectively lost (which results in robustness to translations [4]). In between the two extremes are systems based on multiple template matching [3], modular PCA [20, 22], Pseudo 2D Hidden Markov Models (HMMs) [10, 26] and heuristic approaches such as Elastic Graph Matching (EGM) [8, 16].

Generally speaking, an appearance based face recognition system can be thought of as being comprised of:

1. Face localization and segmentation
2. Normalization
3. Feature extraction
4. Classification

The second stage (normalization) usually involves an affine transformation (to correct for size and rotation), but it can also involve an illumination normalization (however, illumination normalization may not be necessary if the feature extraction method is robust). In this paper we shall concentrate on the last stage (and thus postulate that the preceding steps have been performed correctly).

Some approaches to addressing the single training image problem involve the synthesis of new face images (at various angles) based on *a priori* information (e.g. [2, 21]). In these approaches, the image synthesis comes before the usual step of feature extraction. The obvious question is: if we're only interested in recognition and thus we're going to extract features from synthesized images, why not synthesize the features instead? If we follow this line of thinking, a natural followup question is: instead of synthesizing features with which we're going to train a classifier, why not directly synthesize the classifier's parameters? This is in fact the crux of our proposed extensions, sketched below.

Using *a priori* information in the form of a set of faces at different views (these faces will never be used during performance evaluation), we construct face models for specific views (by "model" we mean a GMM); we then find the *differences* between the model for the frontal view and, say, the model for the $+25^o$ view. Let us now suppose that we wish to enroll a new client in our face verification system and we only have their frontal view; given a face model created from their frontal view, we can synthesize a model for $+25^o$ by applying the *a priori* differences to the client's frontal model. In order for the system to automatically handle the two views, we then augment the client's frontal model by concatenating it with the newly synthesized $+25^o$ model. We can of course repeat this procedure for other views.

The proposed synthesis and augmentation approach thus differs from the approach presented in [2, 21] where actual face images for non-frontal views were synthesized; the synthesized images shown in [2] have considerable artefacts, which can lead to a decrease in performance. The proposed approach is somewhat related to [18] where a feature transformation approach is employed in the context of an EGM based classifier. We note that in [18] manual intervention is required, while our proposed approach is automatic; moreover, unlike [18], our approach is based in a statistical framework. The augmentation part of our proposed approach is related to [14]; the main

The authors thank the Swiss National Science Foundation for supporting this work through the National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2).

Fig. 1. Images of subject 00647 from the FERET database for (from left to right) -60^o, -40^o, -25^o, -15^o and 0^o views; note that the angles are approximate.

difference being that in [14] features from the client's many *real* images are used to extend the client's face model, while in our proposed approach we synthesize the models to represent the face of a client at various non-frontal angles, without having access to the client's real images.

The rest of the paper is organized as follows. In Section 2 we briefly describe the database used in the experiments and the pre-processing of the images. In Sections 3 and 4 we overview the DCTmod2 and PCA based feature extraction techniques, respectively. Section 5 provides a concise description of the GMM based classifier and the different training strategies used when dealing with DCTmod2 and PCA based features. In Section 6 we describe two techniques used for synthesizing non-frontal models as well as a method to address the problem of correspondence between two GMMs. Section 7 details the process of concatenating two or more GMMs. Section 8 is devoted to experiments evaluating the two synthesis techniques and the use of augmented models. The paper is concluded and future work is suggested in Section 9.

2. FERET Database: Setup & Pre-Processing

In our experiments we utilized face images from the FERET database [23]. In particular, we used images from the *ba*, *bb*, *bc*, *bd*, *be*, *bf*, *bg*, *bh* and *bi* subsets, which represent views of 200 persons for (approximately) 0^o (frontal), $+60^o$, $+40^o$, $+25^o$, $+15^o$, -15^o, -25^o, -40^o and -60^o, respectively; thus for each person there are nine images. Example images are shown in Fig. 1.

The 200 persons were split into three disjoint groups: group A, group B and impostor group; the impostor group is comprised of 20 persons, resulting in 90 persons in groups A and B. Throughout the experiments, group A is used as a source of *a priori* information while the impostor group and group B are used for verification tests (i.e. clients come from group B). Thus in each verification trial there is 90 true claimant accesses and 90×20=1800 impostor attacks; moreover, in in each verification trial the view of impostor faces matched the testing view.

In order to reduce the effects of variations possible in real life (such as facial expressions, hair styles, clothes and ornaments) closely cropped faces are used instead of full face images [5]. In particular, we used the location of the eyes to normalize the inter-ocular distance and extract a 56×64 (rows $\times$ columns) face window containing the area from the eyebrows to the nose (inclusive). Example face windows are shown in Fig. 2.

Since in this paper we are proposing extensions to existing 2D approaches, we obtain normalized face windows for non-frontal views exactly in the same way as for the frontal view; this has a significant side effect: for large deviations from the frontal view (such as -60^o and $+60^o$) the effective size of facial characteristics is significantly larger than for the frontal view. The non-frontal face windows thus differ from the frontal face windows not only in terms of out-of-plane rotation of the face, but also scale.

Fig. 2. Extracted face windows from images in Fig. 1.

Overlap (N_O)	**Vectors** (N_V)	**Spatial width**
0	30	24
1	35	22
2	56	20
3	80	18
4	143	16
5	255	14
6	621	12
7	2585	10

Table 1. Number of DCTmod2 feature vectors extracted from a 56×64 face using N_P=8 and varying overlap; also shows the effective spatial width (& height) in pixels for each feature vector.

3. Feature Extraction: DCTmod2 Based Sys.

In DCTmod2 feature extraction [29] a given face image is analyzed on a block by block basis; each block is $N_P\times N_P$ (here we use N_P=8) and overlaps neighboring blocks by N_O pixels. Each block is decomposed in terms of 2D Discrete Cosine Transform (DCT) basis functions [13]. A feature vector for each block is then constructed as:

$$\vec{x} = \left[\begin{array}{cccccccccc} \Delta^h c_0 & \Delta^v c_0 & \Delta^h c_1 & \Delta^v c_1 & \Delta^h c_2 & \Delta^v c_2 & c_3 & c_4 & \ldots & c_{M-1} \end{array} \right]^T \quad (1)$$

where c_n represents the n-th DCT coefficient, while $\Delta^h c_n$ and $\Delta^v c_n$ represent the horizontal & vertical delta coefficients respectively; the deltas are computed using DCT coefficients extracted from neighboring blocks. Compared to traditional DCT feature extraction [10], the first three DCT coefficients are replaced by their respective horizontal and vertical deltas in order to reduce the effects of illumination direction changes, without losing discriminative information. In this study we use M=15 (choice based on [29]), resulting in an 18 dimensional feature vector for each block.

The degree of overlap (N_O) has two effects: the first is that as overlap is increased the spatial area used to derive one feature vector is decreased; the second is that as the overlap is increased the number of feature vectors extracted from an image grows in a quadratic manner. Table 1 shows the amount of feature vectors extracted from 56×64 face using our implementation of the DCTmod2 extractor.

As will be shown later, the larger the overlap (and hence the smaller the spatial area for each feature vector), the more the system is robust to out-of-plane rotations.

4. Feature Extraction: PCA Based System

In PCA based feature extraction [30], a given face image is represented by a matrix containing grey level pixel values; the matrix is then converted to a face vector, $\vec{f}$, by concatenating all the columns; a D-dimensional feature vector, $\vec{x}$, is then obtained by:

$$\vec{x} = \mathbf{U}^T(\vec{f} - \vec{f}_\mu) \quad (2)$$

where $\mathbf{U}$ contains D eigenvectors (corresponding to the D largest eigenvalues) of the training data covariance matrix, and $\vec{f}_\mu$ is the mean of training face vectors. In our experiments we use frontal faces from group A to find $\mathbf{U}$ and $\vec{f}_\mu$. If robustness to illumination changes is required, an extension such as enhanced PCA can be utilized [28].

It must be emphasized that in the PCA based approach, one feature vector represents the entire face, while in the DCTmod2 approach one feature vector represents only a small portion of the face.

5. GMM Based Classifier

The distribution of training feature vectors for each person is modeled by a GMM. Given a claim for client C's identity and a set of (test) feature vectors $X = \{\vec{x}_i\}_{i=1}^{N_V}$ supporting the claim, the average log likelihood of the claimant being the true claimant is found with:

$$\mathcal{L}(X|\lambda_C) = \frac{1}{N_V} \sum\nolimits_{i=1}^{N_V} \log p(\vec{x}_i|\lambda_C) \tag{3}$$

where:

$$p(\vec{x}|\lambda) = \sum\nolimits_{j=1}^{N_G} w_j \, \mathcal{N}(\vec{x}; \vec{\mu}_j, \mathbf{\Sigma}_j) \tag{4}$$

$$\lambda = \{w_j, \vec{\mu}_j, \mathbf{\Sigma}_j\}_{j=1}^{N_G} \tag{5}$$

Here, $\mathcal{N}(\vec{x}; \vec{\mu}, \mathbf{\Sigma})$ is a D-dimensional Gaussian function with mean $\vec{\mu}$ and diagonal covariance matrix $\mathbf{\Sigma}$:

$$\mathcal{N}(\vec{x}; \vec{\mu}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(\frac{-1}{2} (\vec{x} - \vec{\mu})^T \mathbf{\Sigma}^{-1} (\vec{x} - \vec{\mu}) \right) \tag{6}$$

λ_C is the parameter set for client C, N_G is the number of Gaussians and w_j is the weight for Gaussian j (with constraints $\sum_{j=1}^{N_G} w_j = 1$ and $\forall \; j : w_j \geq 0$).

Given the average log likelihood of the claimant being an impostor, $\mathcal{L}(X|\lambda_{\overline{C}})$, an opinion on the claim is found using:

$$\Lambda(X) = \mathcal{L}(X|\lambda_C) - \mathcal{L}(X|\lambda_{\overline{C}}) \tag{7}$$

The verification decision is reached as follows: given a threshold t, the claim is accepted when $\Lambda(X) \geq t$ and rejected when $\Lambda(X) < t$. In our experiments we use a global threshold to obtain performance as close as possible to the Equal Error Rate (EER) (i.e. where the false rejection rate is equal to the false acceptance rate), following the popular practice used in the speaker verification field [7, 11].

Methods for obtaining the parameter set for the impostor model ($\lambda_{\overline{C}}$) and each client are described in the following sections.

5.1. Classifier Training: DCTmod2 Based System

First, a Universal Background Model (UBM) is trained with a form of the Expectation Maximization (EM) algorithm [6, 9] using *all* 0^o data from group A; here the EM algorithm tunes the model parameters to optimize the Maximum Likelihood (ML) criterion (i.e. so that the likelihood of the training data is maximized).

The parameters (λ) for each client model are then found by using the client's training data and adapting the UBM (the number of Gaussians is varied in the experiments); the adaptation is accomplished using a different form of the EM algorithm, often referred to as maximum *a posteriori* (MAP) estimation [12, 25]. The two instances of the EM algorithm are summarized in appendixes A and B.

Since the UBM is a good representation of a general face, it is also used to find the likelihood of the claimant being an impostor, i.e.:

$$\mathcal{L}(X|\lambda_{\overline{C}}) = \mathcal{L}(X|\lambda_{ubm}) \tag{8}$$

5.2. Classifier Training: PCA Based System

The image subset from the FERET database that is utilized in this work has only one frontal image per person; in PCA-based feature extraction, this results in only one training vector, leading to necessary constraints in the structure of the classifier and the classifier's training paradigm.

The UBM and all client models (for frontal faces) are constrained to have only one component (i.e. one Gaussian). As for the DCTmod2 system (described above), the parameters for the UBM are found by running the EM algorithm on all data from group A. Instead of MAP estimation, each client model inherits the covariance matrix from the UBM; moreover, the mean of each client model is taken to be the single training vector for that client.

6. Synthesizing Models for Non-Frontal Views

6.1. UBMdiff Technique

Let us suppose that we have two UBMs, $\lambda_{ubm}^{0^o}$ and $\lambda_{ubm}^{+25^o}$ (trained using *a priori* data) that describe a general face for a view at 0^o and +25^o, respectively. Let us define the set of parameters which describes the difference between the two UBMs as:

$$\Delta^{+25^o} = \left\{ \; w_{\Delta,i}^{+25^o} \; , \; \vec{\mu}_{\Delta,i}^{+25^o} \; , \; \vec{\sigma}_{\Delta,i}^{+25^o} \; \right\}_{i=1}^{N_G} \tag{9}$$

The parameters are defined as:

$$w_{\Delta,i}^{+25^o} = w_{ubm,i}^{+25^o} / w_{ubm,i}^{0^o} \tag{10}$$

$$\vec{\mu}_{\Delta,i}^{+25^o} = \vec{\mu}_{ubm,i}^{+25^o} - \vec{\mu}_{ubm,i}^{0^o} \tag{11}$$

$$\left(\vec{\sigma}_{\Delta,i}^{+25^o} \right)^T = [\; \sigma_{\Delta,i,d} \;]_{d=1}^{D} = \left[\mathbf{\Sigma}_{ubm,i,(d,d)}^{+25^o} / \mathbf{\Sigma}_{ubm,i,(d,d)}^{0^o} \right]_{d=1}^{D} \tag{12}$$

where $\mathbf{\Sigma}_{ubm,i,(d,d)}^{+25^o}$ denotes the element at row d and column d (i.e. d-th diagonal) of $\mathbf{\Sigma}_{ubm,i}^{+25^o}$. Since the two UBMs are a good representation of a general face at the two views, and each client model is derived from the 0^o UBM, it is reasonable to assume that we can apply the above difference to client C's 0^o model to synthesize a +25^o model. Formally, the parameters for the +25^o model are:

$$\lambda_C^{+25^o} = \left\{ w_{C,i}^{+25^o} \, , \vec{\mu}_{C,i}^{+25^o} \, , \mathbf{\Sigma}_{C,i}^{+25^o} \right\}_{i=1}^{N_G} \tag{13}$$

and are synthesized using:

$$w_{C,i}^{+25^o} = \widehat{w}_{C,i}^{+25^o} / \gamma \tag{14}$$

$$\vec{\mu}_{C,i}^{+25^o} = \vec{\mu}_{C,i}^{0^o} + \vec{\mu}_{\Delta,i}^{+25^o} \tag{15}$$

$$\left. \mathbf{\Sigma}_{C,i,(d,d)}^{+25^o} \right|_{d=1}^{D} = \left. \mathbf{\Sigma}_{C,i,(d,d)}^{0^o} \sigma_{\Delta,i,d}^{+25^o} \right|_{d=1}^{D} \tag{16}$$

where the non-diagonal elements of $\mathbf{\Sigma}_{C,i}^{+25^o}$ are set to zero and

$$\widehat{w}_{C,i}^{+25^o} = w_{C,i}^{0^o} \, w_{\Delta,i}^{+25^o} \tag{17}$$

$$\gamma = \sum_{i=1}^{N_G} \widehat{w}_{C,i}^{+25^o} \tag{18}$$

As can be seen, the γ is a scale factor used to ensure that synthesized weights sum to unity. We can of course use the above procedure to synthesize models for angles other than +25^o.

6.2. LinReg Technique

Let us suppose that we have the following multi-variate linear regression model:

$$\mathbf{Y} = \mathbf{X}\mathbf{B} \tag{19}$$

$$\begin{bmatrix} \vec{y}_1^{\,T} \\ \vec{y}_2^{\,T} \\ \vdots \\ \vec{y}_n^{\,T} \end{bmatrix} = \begin{bmatrix} \vec{x}_1^{\,T} & 1 \\ \vec{x}_2^{\,T} & 1 \\ \vdots & \\ \vec{x}_n^{\,T} & 1 \end{bmatrix} \begin{bmatrix} \beta_{(1,1)} & \cdots & \beta_{(1,D)} \\ \beta_{(2,1)} & \cdots & \beta_{(2,D)} \\ \vdots & \vdots & \vdots \\ \beta_{(D+1,1)} & \cdots & \beta_{(D+1,D)} \end{bmatrix} \tag{20}$$

where $n > D + 1$, with D being the dimensionality of each $\vec{y}$ and $\vec{x}$. $\mathbf{B}$ is a matrix of unknown regression parameters; under the sum-of-least-squares regression criterion, $\mathbf{B}$ can be found using [15]:

$$\mathbf{B} = \left(X^T X\right)^{-1} X^T Y \tag{21}$$

Given a set of *a priori* models (from group A), representing faces at 0^o and $+25^o$, we can thus find the relation between the means (and diagonal covariances) for the two angles; specifically, we find $\mathbf{B}_{\mu,i}$ and $\mathbf{B}_{\Sigma,i}$ (i=1,2,$\cdots$,N_G). We can then synthesize model parameters for $+25^o$ [c.f. Eqn. (13)] from client C's 0^o model using:

$$\begin{aligned} w_{C,i}^{+25^o} &= w_{C,i}^{0^o} & (22)\\ \vec{\mu}_{C,i}^{+25^o} &= [\ (\vec{\mu}_{C,i}^{0^o})^T\ 1\ ]\ \mathbf{B}_{\mu,i} & (23)\\ \mathrm{diag}(\mathbf{\Sigma}_{C,i}^{+25^o}) &= [\ \mathrm{diag}(\mathbf{\Sigma}_{C,i}^{0^o})^T\ 1\ ]\ \mathbf{B}_{\Sigma,i} & (24) \end{aligned}$$

where the non-diagonal elements of $\mathbf{\Sigma}_{C,i}^{+25^o}$ are set to zero. It must be noted that unlike the UBMdiff technique (Section 6.1), there is no guarantee that the diagonal elements of $\mathbf{\Sigma}_{C,i}^{+25^o}$ are > 0; thus after synthesis, any diagonal elements which are ≤ 0 are set to a small positive value (1^{-25}). By the same token, the weights for the $+25^o$ model are merely copied from the 0^o model (while this seems drastic, the weights have only a minor influence on performance [25]).

6.3. The Model Correspondence Problem

The UBMdiff and LinReg synthesis techniques pre-suppose that there is a correspondence between components of the client's 0^o model, the 0^o UBM, the $+25^o$ UBM and all models for group A (loosely speaking, by correspondence we mean that corresponding components in all three models describe the same areas of the face). This is true when there is one Gaussian in each model (as for the PCA based system). However, under traditional training paradigms (as described in Section 5.1), this is generally not true when there is two or more Gaussians.

To address this issue, we propose the following modified training paradigm. Instead of training the $+25^o$ UBM directly using the ML criterion, we instead adapt the 0^o UBM using a modified form of MAP estimation; moreover, whenever adapting any client model from any UBM, the modified MAP estimation is also used.

Traditional MAP estimation by itself will not help with the correspondence problem, as for GMMs it is a form of probabilistic clustering (albeit constrained clustering). During clustering, the Gaussians tend to "wander" around before converging to a solution[1]. We illustrate the wandering problem as follows: let's say we have a 32 Gaussian 0^o UBM and we adapt it to create a $+25^o$ UBM; after convergence, it is quite possible for, say, the tenth Gaussian of the $+25^o$ UBM to be the "closest" to the first Gaussian of the 0^o UBM; moreover, it is also possible to have more than one Gaussian in the $+25^o$ UBM that is the "closest" to a given Gaussian in the 0^o UBM. Due to the "wandering" problem, there is no guarantee that the first Gaussian from the $+25^o$ UBM corresponds to the first Gaussian from the 0^o UBM (or in other words, the first Gaussian from the $+25^o$ UBM may be modeling a completely different area of the face when compared to the first Gaussian from the 0^o UBM).

Before describing the modification to the MAP estimation, let us first define a "parent UBM" as the UBM to be adapted and a "child UBM" as the UBM that resulted from adapting a "parent UBM"; in a similar vein, let us define a "parent Gaussian" as a Gaussian from the "parent UBM" and a "child Gaussian" as the Gaussian that resulted from a particular "parent Gaussian" through the process of adaptation. moreover, let us define the distance between two Gaussians as the Mahalanobis distance [9] between their means:

$$\mathcal{M}\left(\vec{\mu}_a,\ \vec{\mu}_b\right) = \left(\vec{\mu}_a - \vec{\mu}_b\right)^T \mathbf{\Sigma}_{all}^{-1} \left(\vec{\mu}_a - \vec{\mu}_b\right) \tag{25}$$

where $\mathbf{\Sigma}_{all}$ is the overall covariance matrix of the "parent UBM"; we shall assume that it is a diagonal matrix. It can be shown that the d-th diagonal element ($\mathbf{\Sigma}_{all,(d,d)}$) is found using:

$$\mathbf{\Sigma}_{all,(d,d)} = -\mu_{all,(d)}^2 + \sum_{i=1}^{N_G} w_i \left(\mathbf{\Sigma}_{i,(d,d)} + \mu_{i,(d)}\right) \tag{26}$$

where $\mu_{all,(d)}$ is the d-th element of $\vec{\mu}_{all}$, which is in turn found using $\vec{\mu}_{all} = \sum_{i=1}^{N_G} w_i \vec{\mu}_i$. Here, $\{w_i, \vec{\mu}_i, \mathbf{\Sigma}_i\}_{i=1}^{N_G}$ are the components of the "parent UBM".

Lastly, let us define a measure which will be used for checking whether any "child Gaussian" is closer to someone else's parent rather than its own parent:

$$\begin{aligned} \psi \ =\ & \sum_{i=1}^{N_G} \sum_{j=1}^{N_G} \mathcal{S}\left(k\mathcal{M}(\vec{\mu}_i^{\,child}, \vec{\mu}_i^{\,parent}) - \mathcal{M}(\vec{\mu}_i^{\,child}, \vec{\mu}_j^{\,parent})\right) \\ & -2N_G \end{aligned} \tag{27}$$

where $k > 1$ and

$$\mathcal{S}(a) = \begin{cases} +1 & \text{if} \quad a > 0 \\ -1 & \text{if} \quad a \leq 0 \end{cases} \tag{28}$$

k designates how close a "child Gaussian" can be to someone else's parent; if k=2, then it is closer than two times the distance between the parent in question and the parent's true child.

To address the "wandering" problem we modify the EM algorithm for MAP estimation (shown in Appendix B) by introducing an early stopping criterion: from the second iteration onwards, we check if $\psi \neq -N_G^2$ after each maximization step; if the condition is satisfied we restore the parameters from the last iteration and deem that we have converged. The check is enabled from the second iteration onwards since we wish at least for some adaptation to occur (otherwise it would be possible for the "child UBM" to be the same as the "parent UBM"). In this work we use k=2 (choice based on preliminary experiments).

7. Augmenting Frontal Models

A composite model for client C is created by augmenting the client's frontal model ($\lambda_C^{0^o}$) as follows:

$$\begin{aligned} \lambda_C^{aug} &= \lambda_C^{0^o} \sqcup \lambda_C^{+60^o} \sqcup \lambda_C^{+40^o} \cdots \sqcup \lambda_C^{-40^o} \sqcup \lambda_C^{-60^o} \\ &= \sqcup_{i \in A} \lambda_C^i \end{aligned} \tag{29}$$

where

$$A = \{\ 0^o, +60^o, +40^o, +25^o, +15^o, -15^o, -25^o, -40^o, -60^o\ \} \tag{30}$$

and $\sqcup$ is an operator for joining GMM parameter sets. Let us suppose we have two GMM parameter sets, λ_x and λ_y, comprised of parameters for $N_{x,G}$ and $N_{y,G}$ Gaussians, respectively. The $\sqcup$ operator is defined as follows:

$$\begin{aligned} \lambda_z &= \lambda_x \sqcup \lambda_y \\ &= \{\alpha w_{x,i},\ \vec{\mu}_{x,i},\ \mathbf{\Sigma}_{x,i}\}_{i=1}^{N_{x,G}}\ \cup\ \{\beta w_{y,i},\ \vec{\mu}_{y,i},\ \mathbf{\Sigma}_{y,i}\}_{i=1}^{N_{y,G}} \end{aligned} \tag{31}$$

where:

$$\begin{aligned} \alpha &= N_{x,G}/(N_{x,G} + N_{y,G}) & (32)\\ \beta &= 1 - \alpha & (33) \end{aligned}$$

Here the non-frontal models can be synthesized from the client's frontal model using the UBMdiff or LinReg techniques (Section 6).

[1] It must be noted that this observed behaviour is counter-intuitive; it is under further investigation.

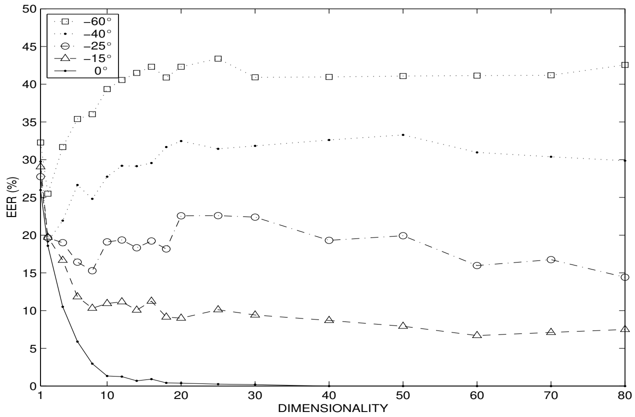

Fig. 3. Performance of PCA based system (trained on frontal faces) for increasing dimensionality and the following angles: -60^o, -40^o, -25^o, -15^o and 0^o (frontal).

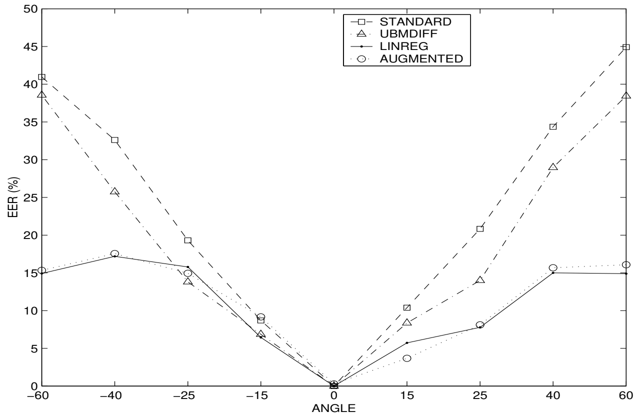

Fig. 4. Performance of various PCA based systems: standard, UBMdiff, LinReg and augmented; the standard system used original frontal client models only; UBMdiff and LinReg systems used client models synthesized specifically for a given test angle; the augmented system used client models comprised of original frontal and synthesized side models (via LinReg technique).

8. Experiments and Discussion

8.1. PCA Based System

In the first experiment we studied how the dimensionality of the feature vectors used in the PCA system affects robustness to varying pose. The higher the dimensionality, the more accurately the face image is represented; we conjecture that as more accurately the face is represented, the more the system will be affected by varying pose. Client models were trained on frontal faces and tested on faces from -60^o to +60^o views; impostor faces matched the testing view. Results for -60^o to 0^o are shown in Fig. 3 (the results for 0^o to +60^o, not shown here, have very similar trends).

As can be observed, a dimensionality of 40 is required to achieve perfect verification on frontal faces (this agrees with results presented in [26]). For non-frontal faces at $\pm 60^o$ and $\pm 40^o$, the error rate generally increases as the dimensionality is increased, saturating when the dimensionality is about 15; hence there is somewhat of a trade-off between the error rate on frontal faces and non-frontal faces, controlled by the dimensionality. Since in this work we are pursuing extensions to standard 2D approaches, the dimensionality has been fixed at 40 for further experiments (using a lower dimensionality of, say, 4, offers better (but still poor) performance for non-frontal faces, however it comes at the a cost of an EER of about 10% on frontal faces, which is unacceptable in real life applications).

In the second experiment we evaluated the performance of models synthesized using UBMdiff and LinReg techniques; The client models were synthesized for a given test angle; this pre-supposes that we know what the test angle is *a priori*, but is nevertheless useful for comparing performance with augmented models. As can be seen from the results presented in Fig. 4, both techniques perform better than the standard system and the LinReg technique offers significantly better performance than UBMdiff. We conjecture the reason for the betterness of the LinReg technique as follows: the UBMdiff technique only utilizes the difference between two general models, while the LinReg technique utilizes the differences between two sets of models (90 models for a frontal view and 90 models for a non-frontal view); in effect, the LinReg technique utilizes more information than the UBMdiff technique (in the form of 180 mean vectors instead of two) and is thus able to synthesize the non-frontal models more accurately. While the LinReg technique does not guarantee that valid covariance matrices will be generated, for the case of the PCA based system no such problem occurred; we conjecture that this is due to the constrained training strategy (Section 5.2), where client models inherited their covariance matrix from the UBM; in effect the LinReg technique uses information from two covariance matrices instead of 180.

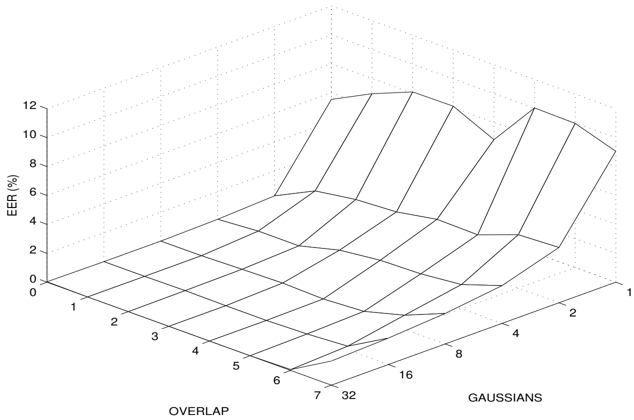

Fig. 5. Performance of standard DCTmod2 based system trained and tested on frontal faces, for varying degrees of overlap and number of Gaussians.

In the third experiment we augmented each client's frontal model with models (for the eight non-frontal views) synthesized by the LinReg technique; since each frontal model was constrained to have one Gaussian, each resulting augmented model had nine Gaussians. From the results shown in Fig. 4, we can see that there is little difference between using client models specifically synthesized for a given test angle and the augmented models, which cover all the test angles. These results thus support the use of frontal models augmented with synthesized models.

8.2. DCTmod2 Based System

In the first experiment we studied how the overlap setting in the DCTmod2 feature extractor and number of Gaussians in the classifier affects performance & robustness. Client models were trained on frontal faces and tested on faces at 0^o and +40^o views; impostor faces matched the testing view. Results are shown in Figs. 5 and 6.

As we can see, when testing with frontal faces, the general trend is that as the overlap increases more Gaussians are needed to decrease

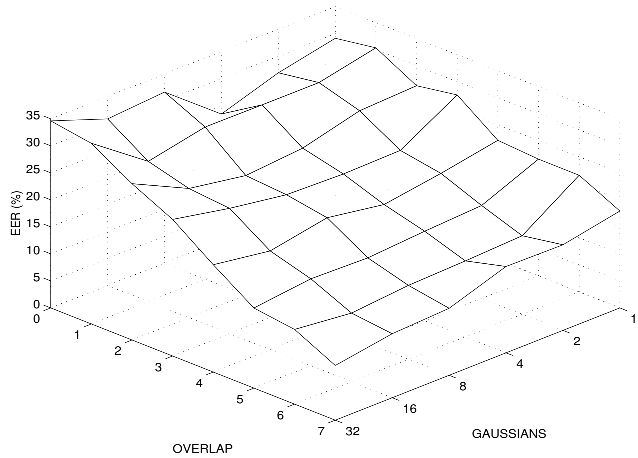

Fig. 6. Performance of standard DCTmod2 based system trained on frontal faces and tested on +40^o faces, for varying degrees of overlap and number of Gaussians.

the error rate (which can be interpreted as follows: the smaller the spatial area used by the features, the more Gaussians are required to adequately model the face). When testing with non-frontal faces, the general trend is that as the overlap increases, the lower the error rate; there is also a less defined trend when the overlap is 4 pixels or greater: the more Gaussians, the lower the error rate[2]. While not shown here, the DCTmod2 based system obtained similar trends for non-frontal views other than +40^o.

The best performance for +40^o faces is achieved with an overlap of 7 pixels and 32 Gaussians, resulting in an EER close to 10%. This is quite impressive. considering that the EER of the standard PCA based system is around 35%; for the PCA system utilizing synthesized models the EER is around 15%. The robustness of the standard DCTmod2/GMM system can be attributed to two aspects:

1. The small spatial area (especially with an overlap of 7) used by each feature vector, results in out-of-plane rotations having a smaller effect on DCTmod2 features when compared to PCA based features (which describe the entire face).
2. The loss of spatial relation between face characteristics (due to use of the GMM classifier), resulting in the "movement" of facial characteristics (due to out-of-plane rotations) having relatively little effect.

For further experiments we have chosen the configuration of 7 pixel overlap and 32 Gaussians. While this does not achieve perfect verification rate on frontal faces, the EER is quite low at 1.67%; moreover, as will be shown in the next experiment, the EER drops to 0% when the modified MAP estimation is used (described in Section 6.3).

In the second experiment we evaluated the effects of modified MAP estimation. From the results presented in Fig. 7 we can see that utilizing the modified training has no adverse effects on the performance when compared to original MAP estimation.

In the third experiment we evaluated the performance of models synthesized via the UBMdiff technique, using both original and modified training. In order to provide a fair comparison with the LinReg technique in later experiments, synthesis of weights was not done; instead, the weights for non-frontal models were copied from the frontal model. As shown in Fig. 7, using original training causes the UBMdiff technique to fall apart (the results are worse than the standard approach); in contrast, using the UBMdiff technique with

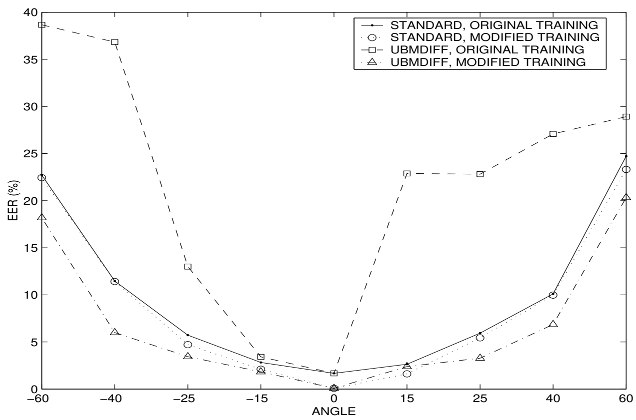

Fig. 7. Performance of various DCTmod2 based systems: standard (using original & modified training) and UBMdiff (also using original & modified training).

modified MAP estimation reduces the error rate in almost all cases. These results thus suggest that the model correspondence problem (described in Section 6.3) is effectively addressed via the modified MAP estimation; the results also suggest that the UBMdiff technique is useful for synthesizing models.

In the fourth experiment we evaluated the use of the LinReg technique for synthesizing models; results are presented in Fig. 8. It can be seen that the performance is worse than the UBMdiff technique; a possible cause of this has been alluded in Section 6.2: there is no guarantee that valid covariance matrices will be generated. Indeed, during model synthesis it was found that many elements of the covariance matrices had negative values, and were thus set to a small positive value; this obviously has the effect of making any model less precise, leading to worse performance.

In the fifth experiment we augmented each client's frontal model with models synthesized by the UBMdiff technique for the following angles: $\pm 60^o$, $\pm 40^o$ and $\pm 25^o$. Synthesized models for $\pm 15^o$ were not used since they provided no performance benefit over the 0^o model. Since each frontal model was set to have 32 Gaussians, each resulting augmented model had 224 Gaussians. From the results shown in Fig. 8, we can see that there is little difference between us-

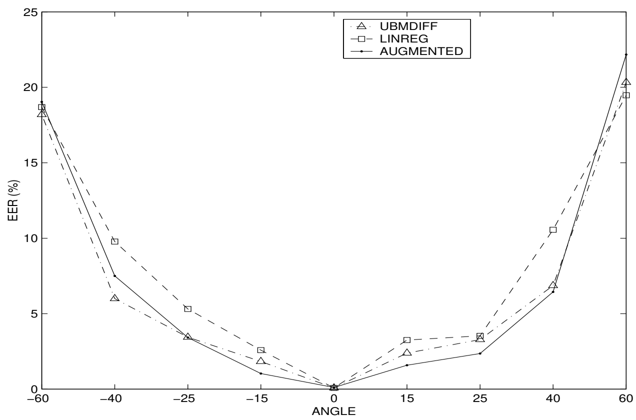

Fig. 8. Performance of various DCTmod2 based systems: UBMdiff, LinReg and augmented; UBMdiff and LinReg systems used client models synthesized specifically for a given test angle; the augmented system used client models comprised of original frontal and synthesized side models (via UBMdiff technique).

[2]This is true up to a point: eventually the error rate will go up as there will be too many Gaussians to train adequately with the limited amount of data.

ing client models specifically synthesized for a given test angle and the augmented models, which cover all the test angles. Like in the case for the PCA based system, these results support the use of frontal models augmented with synthesized models.

8.3. PCA/GMM vs DCTmod2/GMM

Since in this work we have evaluated two significantly different face verification systems (PCA based and DCTmod2 based), it would be interesting to compare their performance. The results shown in Fig. 9 (created by recycling some of the results from previous experiments) suggest the following:

1. The standard DCTmod2/GMM system (trained on frontal faces) is less affected than the corresponding PCA/GMM system.
2. In almost all cases, frontal model augmentation has beneficial effects for both systems.
3. Except for the extreme views at $\pm 60^o$, the DCTmod2/GMM system using augmented models is more robust than the corresponding PCA/GMM system.

9. Conclusions and Future Work

In this work we proposed to address the problem of non-frontal face verification when only a frontal training image is available (e.g. a passport photograph) by augmenting a client's frontal face model with artificially synthesized models for non-frontal views. In the framework of a GMM based classifier, two techniques were proposed for the synthesis: UBMdiff and LinReg. Both techniques rely on *a priori* information and learn how face models for the frontal view are related to face models at a non-frontal view. The synthesis and augmentation approach was evaluated by applying it to two face verification systems: PCA based and DCTmod2 based; the two systems are a representation of holistic and non-holistic approaches, respectively.

Experimental results suggest that in almost all cases, frontal model augmentation has beneficial effects for both systems; they also suggest that the LinReg technique (which is based on multivariate regression of classifier parameters) is more suited to the PCA based system and that the UBMdiff technique (which is based on differences between two general face models) is more suited to the DCTmod2 based system. The results also support the view that the standard DCTmod2/GMM system (trained on frontal faces) is less affected by out-of-plane rotations than the corresponding PCA/GMM system; moreover, except for the extreme views at $\pm 60^o$, the DCTmod2/GMM system using augmented models is more robust than the corresponding PCA/GMM system.

Currently in the DCTmod2/GMM approach each Gaussian often models disjoint face areas that are similar in texture (see Appendix A in [27]). This may not be optimal when dealing with out-of-plane face rotations as different parts of face may very well undergo different transformations. Better performance may be obtained if the Gaussians are constrained to model non-disjoint areas; to some extent this could be achieved by incorporating positional information in each feature vector (i.e. augmenting each DCTmod2 vector with the row and column of where it comes from); another possibility it to use a 2D Hidden Markov Model (HMM) based classifier [10, 26] in place of the GMM (since a GMM can be considered to be a special case of an ergodic HMM).

Finally we note that, in the context of generative models (such as the GMM), there are probably more principled ways (than UBMdiff and LinReg) of utilizing *a priori* information; however, the techniques presented here show that it's possible to effectively utilize *a priori* information directly in the model domain, rather than in the image domain.

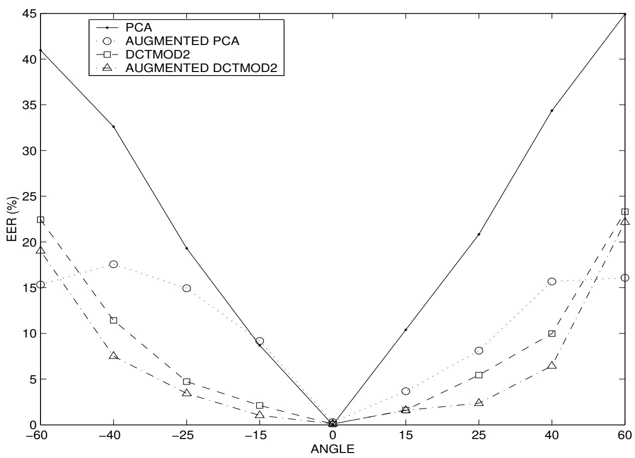

Fig. 9. Performance comparison of standard PCA, augmented PCA, standard DCTmod2 and augmented DCTmod2.

Appendix A. EM: Maximum Likelihood

Given a set of training vectors, $X = \{\vec{x}_i\}_{i=1}^{N_V}$, the GMM parameters ($\lambda$) are estimated using the Maximum Likelihood (ML) principle:

$$\lambda = \arg\max_{\hat{\lambda}} \ p(X|\hat{\lambda}) \tag{34}$$

The estimation problem can be solved using a form of the Expectation Maximization (EM) algorithm [6, 9]. The EM algorithm for GMMs is comprised of iterating two steps: the *expectation* step, followed by the *maximization* step. GMM parameters generated by the previous iteration (λ^{old}) are used by the current iteration to generate a new set of parameters (λ^{new}), such that:

$$p(X|\lambda^{new}) \geq p(X|\lambda^{old}) \tag{35}$$

The process is usually repeated until convergence (the parameters have not changed from one iteration to the next), or until the increase in the likelihood after each iteration falls below a pre-defined threshold, or until the number of iterations is equal to a pre-defined maximum. Reynolds [24] showed that the EM algorithm generally converges in 10 to 15 iterations, with further iterations resulting in only minor increases of the likelihood $p(X|\lambda)$; this has also been the authors' experience with various types of data. In our implementation we have therefore limited the number of iterations to 20. The algorithm is summarized as follows:

Expectation step:

for $k = 1, \cdots, N_G$: for $i = 1, \cdots, N_V$:

$$l_{k,i} = \frac{w_k \mathcal{N}(\vec{x}_i; \vec{\mu}_k, \mathbf{\Sigma}_k)}{\sum_{n=1}^{N_G} w_n \mathcal{N}(\vec{x}_i; \vec{\mu}_n, \mathbf{\Sigma}_n)} \tag{36}$$

for $k = 1, \cdots, N_G$:

$$L_k = \sum_{i=1}^{N_V} l_{k,i} \tag{37}$$

$$\hat{w}_k = L_k / N_V \tag{38}$$

$$\hat{\vec{\mu}}_k = \frac{1}{L_k} \sum_{i=1}^{N_V} \vec{x}_i \, l_{k,i} \tag{39}$$

$$\hat{\mathbf{\Sigma}}_k = \frac{1}{L_k} \left(\sum_{i=1}^{N_V} \vec{x}_i \vec{x}_i^T \, l_{k,i} \right) - \hat{\vec{\mu}}_k \hat{\vec{\mu}}_k^T \tag{40}$$

Maximization step:

$$\{w_k, \vec{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^{N_G} = \{\hat{w}_k, \hat{\vec{\mu}}_k, \hat{\mathbf{\Sigma}}_k\}_{k=1}^{N_G} \tag{41}$$

The initial estimate (i.e. the seed) is typically provided by the k-means clustering algorithm [9]. It must be noted that the above implementation of EM can also be interpreted as an unsupervised probabilistic clustering procedure, with N_G being the assumed number of clusters.

Appendix B. EM: MAP Estimation

The main difference between ML and MAP estimation is in the use of *a prior* distribution ($f(\hat{\lambda})$) of the parameters to be estimated [c.f. Eqn. (34)]:

$$\lambda = \arg \max_{\hat{\lambda}} \; p(X|\hat{\lambda}) \; f(\hat{\lambda}) \tag{42}$$

The above estimation problem can be also solved using the EM algorithm, albeit in a different form to the one described in Appendix A; this form is often referred to as maximum *a posteriori* estimation [12, 25], and is summarized as follows.

Given UBM parameters $\lambda_{ubm} = \{\tilde{w}_k, \tilde{\vec{\mu}}_k, \tilde{\mathbf{\Sigma}}_k\}_{k=1}^{N_G}$ and a set of training feature vectors for a specific client, $X = \{\vec{x}_i\}_{i=1}^{N_V}$, the estimated weights ($\hat{w}_k$), means ($\hat{\vec{\mu}}_k$), and covariances ($\hat{\mathbf{\Sigma}}_k$) are found as per Eqns. (38)-(40). The maximization step (for $k = 1, \cdots, N_G$) is then defined as:

$$w_k = \left[\alpha \hat{w}_k + (1-\alpha)\tilde{w}_k\right] \gamma \tag{43}$$

$$\vec{\mu}_k = \alpha \hat{\vec{\mu}}_k + (1-\alpha)\tilde{\vec{\mu}}_k \tag{44}$$

$$\mathbf{\Sigma}_k = \left[\alpha\Big(\hat{\mathbf{\Sigma}}_k + \hat{\vec{\mu}}_k\hat{\vec{\mu}}_k^T\Big) + (1{-}\alpha)\Big(\tilde{\mathbf{\Sigma}}_k + \tilde{\vec{\mu}}_k\tilde{\vec{\mu}}_k^T\Big)\right] {-} \vec{\mu}_k\vec{\mu}_k^T \tag{45}$$

where γ is a scale factor to make sure the weights sum to one. $\alpha{=}\frac{L_k}{L_k+r}$ is a data-dependent adaptation coefficient [L_k is found using Eqn. (37)], where r is a fixed relevance factor [25]; in our experiments we used r=256 (choice based on preliminary experiments).

As can be seen, the new parameters are simply a weighted sum of *a priori* statistics and new statistics. Here, α can be interpreted as the amount of faith we have in the new statistics. The choice of $\alpha{=}\frac{L_k}{L_k+r}$ causes the adaptation of only the Gaussians for which there is "sufficient" data; in other words, the MAP estimation approach for finding GMM parameters should be robust to limited amount of training data.

Since the ML EM algorithm for GMMs is a form of unsupervised probabilistic clustering, the MAP EM algorithm is also a form of unsupervised probabilistic clustering, albeit it is constrained.

References

[1] J. J. Atick, P. A. Griffin and A. N. Redlich, "Statistical Approach to Shape from Shading: Reconstruction of Three-Dimensional Face Surfaces from Single Two-Dimensional Images", *Neural Computation*, Vol. 8, 1996, pp. 1321-1340.

[2] D. Beymer and T. Poggio, "Face Recognition From One Example View", *Proc. 5th Int. Conf. Computer Vision (ICCV)*, Cambridge, 1995, pp. 500-507.

[3] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, No. 10, 1993, pp. 1042-1052.

[4] F. Cardinaux, C. Sanderson and S. Marcel, "Comparison of MLP and GMM Classifiers for Face Verification on XM2VTS", *Proc. 4th Int. Conf. Audio- and Video-Based Biometric Person Authentication (AVBPA)*, Guildford, 2003, pp. 911-920.

[5] L-F. Chen, H-Y. Liao, J-C. Lin and C-C. Han, "Why recognition in a statistics-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof", *Pattern Recognition*, Vol. 34, No. 7, 2001, pp. 1393-1403.

[6] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *J. Royal Statistical Soc., Ser. B*, Vol. 39, No. 1, 1977, pp. 1-38.

[7] G. R. Doddington, M. A. Przybycki, A. F. Martin and D. A. Reynolds, "The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective", *Speech Communication*, Vol. 31, No. 2-3, 2000, pp. 225-254.

[8] B. Duc, S. Fischer and J. Bigün, "Face Authentication with Gabor Information on Deformable Graphs", *IEEE Trans. Image Processing*, Vol. 8, No. 4, 1999, pp. 504-516.

[9] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, John Wiley & Sons, USA, 2001.

[10] S. Eickeler, S. Müller and G. Rigoll, "Recognition of JPEG Compressed Face Images Based on Statistical Methods", *Image and Vision Computing*, Vol. 18, No. 4, 2000, pp. 279-287.

[11] S. Furui, "Recent Advances in Speaker Recognition", *Pattern Recognition Letters*, Vol. 18, No. 9, 1997, pp. 859-872.

[12] J.-L. Gauvain and C.-H. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains", *IEEE Trans. Speech and Audio Processing*, Vol. 2, No. 2, 1994, pp. 291-298.

[13] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, Massachusetts, 1993.

[14] R. Gross, J. Yang and A. Waibel, "Growing Gaussian Mixture Models for Pose Invariant Face Recognition", *Proc. 15th Int. Conf. Pattern Recognition*, Barcelona, 2000, pp. 1088-1091 (Vol. 1).

[15] D.-Y. Huang and K.-C. Liu, "Some variable selection procedures in multivariate linear regression models", *J. Statistical Planning and Inference*, Vol. 41, 1994, pp. 205-214.

[16] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R. P. Würtz and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture", *IEEE Trans. Computers*, Vol. 42, No. 3, 1993, pp. 300-311.

[17] M. W. Lee and S. Ranganath, "Pose-invariant face recognition using a 3D deformable model", *Pattern Recognition*, Vol. 36, No. 8, 2003, pp. 1835-1846.

[18] T. Maurer and C. v.d. Malsburg, "Learning Feature Transformations to Recognize Faces Rotated in Depth", *Proc. Int. Conf. Artificial Neural Networks (ICANN)*, Paris, 1995, pp. 353-358.

[19] K. Messer et al., "Face Verification Competition on the XM2VTS Database", *Proc. 4th Int. Conf. Audio- and Video-Based Biometric Person Authentication (AVBPA)*, Guildford, 2003, pp. 964-974.

[20] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, 1997, pp. 696-710.

[21] P. Niyogi, F. Girosi and T. Poggio, "Incorporating Prior Information in Machine Learning by Creating Virtual Examples", *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, pp. 2196-2209.

[22] A. Pentland, B. Moghaddam and T. Starner, "View-Based and Modular Eigenspaces for Face Recognition", *Proc. Int. Conf. Computer Vision and Pattern Recognition*, Seattle, 1994, pp. 84-91.

[23] P. J. Phillips, H. Moon, S. A. Rizvi and P. J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, 2000, pp. 1090-1104.

[24] D. A. Reynolds, "A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification", *Technical Report 967*, Lincoln Laboratory, Massachusetts Institute of Technology, 1993.

[25] D. Reynolds, T. Quatieri and R. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, Vol. 10, No. 1-3, 2000, pp. 19-41.

[26] F. Samaria, *Face Recognition Using Hidden Markov Models*, PhD Thesis, University of Cambridge, 1994.

[27] C. Sanderson, "Face Processing & Frontal Face Verification", IDIAP-RR 03-20, Martigny, Switzerland, 2003. (see *www.idiap.ch*)

[28] C. Sanderson and S. Bengio, "Robust Features for Frontal Face Authentication in Difficult Image Conditions", *Proc. 4th Int. Conf. Audio- and Video-Based Biometric Person Authentication (AVBPA)*, Guildford, 2003, pp. 495-504.

[29] C. Sanderson and K. K. Paliwal, "Polynomial Features for Robust Face Authentication", *Proc. IEEE Int. Conf. Image Processing*, Rochester, 2002, pp. 997-1000 (Vol. 3).

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