LIKELIHOOD NORMALIZATION FOR FACE AUTHENTICATION IN VARIABLE RECORDING CONDITIONS

Conrad Sanderson and Kuldip K. Paliwal

School of Microelectronic Engineering
Griffith University
Brisbane, QLD 4111, Australia

ABSTRACT

In this paper we evaluate the effectiveness of two likelihood normalization techniques, the Background Model Set (BMS) and the Universal Background Model (UBM), for improving performance and robustness of four face authentication systems utilizing a Gaussian Mixture Model (GMM) classifier. The systems differ in the feature extraction method used: eigenfaces (PCA), 2-D DCT, 2-D Gabor wavelets and DCT-mod2. Experiments on the VidTIMIT database, using test images corrupted either by an illumination change or compression artefacts, suggest that likelihood normalization has little effect when using PCA derived features, while providing significant performance improvements when using the remaining features.

1. INTRODUCTION

A face authentication system verifies the claimed identity based on images (or a video sequence) of the claimant's face. Such systems have forensic and security (ie. access control) applications.

It seems all current face-based authentication systems, eg. [1, 2, 3, 4], effectively follow a thresholding approach to make the final accept or reject decision. The result of comparison of the claimant's features (X) with a model belonging to the person whose identity is being claimed (λ_C) is a matching score or a likelihood. Let us refer to this result as $p(X|\lambda_C)$. Given a threshold t, the claim is accepted when:

$$p(X|\lambda_C) \geq t \tag{1}$$

and rejected otherwise. However, if there is a mismatch between training and testing conditions, the claim may be automatically rejected due to a low likelihood. The mismatch can occur due to, for example, different cameras being used, an illumination change (important in security applications) or compression artefacts (important in forensic work dealing with compressed video).

In speech-based verification systems it has been found that use of normalized likelihoods improves performance as well as robustness [5]. By reformulating Eqn. (1) in the Bayesian framework, the claim is accepted when:

$$\frac{p(X|\lambda_C)}{p(X|\lambda_{\overline{C}})} \geq t \tag{2}$$

where $p(X|\lambda_{\overline{C}})$ is the result of the claimant's features being compared to an anti-client model ($\lambda_{\overline{C}}$), ie. the likelihood of the claimant being an impostor. If the testing condition causes $p(X|\lambda_C)$ to decrease, then it is reasonable to suppose that $p(X|\lambda_{\overline{C}})$ will also decrease - thus the ratio of the likelihoods may remain relatively unaffected. In effect, the threshold is automatically tuned for each person to account for environmental conditions.

There are two popular approaches for finding the impostor likelihood:

1. Background Model Set (BMS) approach [6].
2. Universal Background Model (UBM) approach [7].

The most important difference between the two techniques is that in the latter approach the impostor likelihood is client independent.

We will evaluate the effectiveness of the above approaches for improving the performance and robustness of four face authentication systems in a common framework - ie. classifier, database, controlled image corruption via an illumination change and compression artefacts. The four systems differ in the feature extraction method used: eigenfaces (PCA) [8], 2-D DCT [9], 2-D Gabor wavelets [10] and DCT-mod2 [11].

The rest of the paper is organized as follows. In Section 2 we briefly review the feature extraction methods. In Section 3, we describe the Gaussian Mixture Model (GMM) based classifier which shall be used as the basis for experiments. In Section 4 we describe the two normalization approaches suited to the GMM classifier. Section 5 is devoted to experiments. The results are discussed and conclusions drawn in Section 6.

2. FEATURE EXTRACTION

In the eigenfaces approach [8], Principal Component Analysis (PCA) is used to make low dimensionality representations of face images. 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:

$$\vec{x} = \boldsymbol{U}^T(\vec{f} - \vec{f}_\mu) \tag{3}$$

where $\boldsymbol{U}$ contains D eigenvectors (with largest corresponding eigenvalues) of the training data covariance matrix, and $\vec{f}_\mu$ is the mean of training face vectors. Typically, $D = 40$. In this work we shall use the terms eigenfaces and PCA interchangeably.

In 2-D DCT feature extraction, a given face image is analyzed on a block by block basis. Each block is decomposed in terms of 2-D DCT basis functions [9], resulting in a set of coefficients. For each block, the first M coefficients are used to form an M-dimensional feature vector (typically, $M = 15$).

The DCT-mod2 approach is similar to 2-D DCT. The main difference is that the feature vector for each block also contains polynomial coefficients based on a subset of 2-D DCT coefficients extracted from spatially neighbouring blocks [11]. The dimensionality of a DCT-mod2 feature vector is $M + 3$.

In 2-D Gabor wavelet feature extraction, a coarse rectangular grid is placed over a given image. At each node of the grid, the image is analyzed by a set of biologically inspired 2-D Gabor wavelets [10], differing in orientation and scale. Responses of the wavelets are then used to form a G-dimensional feature vector (typically, $G = 18$).

It must be emphasized that in the eigenfaces approach, one feature vector represents the entire face, while in the other methods, one feature vector represents only a small portion of the face.

3. GMM BASED CLASSIFIER

Given a claim for person C's identity and a set of feature vectors $X = \{\vec{x}_i\}_{i=1}^{N_V}$ supporting the claim (which may come from a sequence of images), the average log likelihood of the claimant being the true claimant is calculated using:

$$\mathcal{L}(X|\lambda_C) = \frac{1}{N_V}\sum_{i=1}^{N_V}\log p(\vec{x}_i|\lambda_C) \tag{4}$$

$$\text{where} \quad p(\vec{x}|\lambda) = \sum_{j=1}^{N_M} m_j\,\mathcal{N}(\vec{x};\vec{\mu}_j,\mathbf{\Sigma}_j) \tag{5}$$

$$\text{and} \quad \lambda = \{m_j,\vec{\mu}_j,\mathbf{\Sigma}_j\}_{j=1}^{N_M} \tag{6}$$

Here λ_C is the model for person C. N_M is the number of mixtures, m_j is the weight for mixture j (with constraint $\sum_{j=1}^{N_M} m_j = 1$), and $\mathcal{N}(\vec{x};\vec{\mu},\mathbf{\Sigma})$ is a multi-variate Gaussian function with mean $\vec{\mu}$ and diagonal covariance matrix $\mathbf{\Sigma}$.

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 $\Lambda(X) < t$.

3.1. Model Construction

Given a set of training vectors (which may come from a sequence of images), an N_M-mixture GMM for each client can be constructed two ways:

1. Using a *k*-means clustering algorithm followed by 10 iterations of the Expectation Maximization (EM) algorithm [12]. This approach is taken when using the BMS for normalization (Section 4.1).
2. Adapting a previously constructed Universal Background Model, λ_{UBM}, using a form of *maximum a posteriori* (MAP) adaptation [7]. This is done when using the UBM approach for normalization (Section 4.2).

4. NORMALIZATION APPROACHES

4.1. Background Model Set (BMS)

In this approach, the average log likelihood that the claim for person C's identity is from an impostor is calculated using a set of background models, $B = \{\lambda_b\}_{b=1}^{N_B}$:

$$\mathcal{L}(X|\lambda_{\overline{C}}) = \log\left[\frac{1}{N_B}\sum_{b=1}^{N_B}\exp\mathcal{L}(X|\lambda_b)\right] \tag{8}$$

The set of background models for each client is selected from the pool of client models, as follows. Using training data, pair-wise distances between each client model are found. For models λ_D and λ_E with corresponding training feature vector sets X_D and X_E (which were used during the construction of the models), the distance is defined as:

$$\begin{aligned} d(\lambda_D,\lambda_E) \quad = \quad & [\mathcal{L}(X_D|\lambda_D) - \mathcal{L}(X_D|\lambda_E)] \\ & + [\mathcal{L}(X_E|\lambda_E) - \mathcal{L}(X_E|\lambda_D)] \end{aligned} \tag{9}$$

The above symmetric distance attempts to measure how similar (or close) the models λ_D and λ_E are. The background model set contains models which are the closest to, as well as the farthest from, the client model. While it may intuitively seem that only the close models are required (which represent the expected impostors), this would leave the system vulnerable to impostors which are very different from the client. This is demonstrated by inspecting Eqn. (7), where both terms would contain similar values, leading to an unreliable opinion on the claim.

For a given client model λ_C, N_Φ closest models ($N_\Phi \geq N_B$) are placed in set Φ. Similarly, N_Ψ farthest models ($N_\Psi \geq N_B$) are placed in set Ψ. *Maximally spread* models from the Φ set are moved to set B_{close} using the following procedure:

1. Move the closest model from Φ to B_{close}.
2. Move λ_i from Φ to B_{close}, where λ_i is found using:

$$\lambda_i = \arg\max_{\lambda_j\in\Phi}\left[\frac{1}{N_{B_{close}}}\sum_{\lambda_b\in B_{close}}\frac{d(\lambda_b,\lambda_j)}{d(\lambda_C,\lambda_j)}\right] \tag{10}$$

   where $N_{B_{close}}$ is the cardinality of B_{close}.
3. Repeat step (2) until $N_{B_{close}} = \frac{N_B}{2}$.

Next, *maximally spread* models from the Ψ set are moved to set B_{far} using the following procedure:

1. Move the farthest model from Ψ to B_{far}.
2. Move λ_i from Ψ to B_{far}, where λ_i is found using:

$$\lambda_i = \arg\max_{\lambda_j\in\Psi}\left[\frac{1}{N_{B_{far}}}\sum_{\lambda_b\in B_{far}} d(\lambda_b,\lambda_j)\; d(\lambda_C,\lambda_j)\right] \tag{11}$$

   where $N_{B_{far}}$ is the cardinality of B_{far}.
3. Repeat step (2) until $N_{B_{far}} = \frac{N_B}{2}$.

Finally, $B = B_{close} \cup B_{far}$. The above procedures for selecting *maximally spread* models are required to reduce redundancy in the B set [6].

4.2. Universal Background Model (UBM)

In this approach, pooled training data from *all* clients is utilized to construct a Universal Background Model (λ_{UBM}) using a *k*-means clustering algorithm followed by 10 iterations of the EM algorithm. The average log likelihood that the claim for person C's identity is from an impostor is found using:

$$\mathcal{L}(X|\lambda_{\overline{C}}) = \mathcal{L}(X|\lambda_{UBM}) \tag{12}$$

Moreover, instead of constructing the client models directly from training data, they are generated by adapting λ_{UBM}, as follows. Given a set of training feature vectors for a specific client, $X = \{\vec{x}_i\}_{i=1}^{N_V}$, and UBM parameters, $\{\dot{m}_k, \dot{\vec{\mu}}_k, \dot{\mathbf{\Sigma}}_k\}_{k=1}^{N_M}$, estimated weights ($\hat{m}_k$), means ($\hat{\vec{\mu}}_k$), and covariances ($\hat{\mathbf{\Sigma}}_k$) are first found using (for $k = 1, ..., N_M$):

$$l_{k,i} = \frac{m_k \mathcal{N}(\vec{x}_i; \dot{\vec{\mu}}_k, \dot{\mathbf{\Sigma}}_k)}{\sum_{n=1}^{N_M} \dot{m}_n \mathcal{N}(\vec{x}_i; \dot{\vec{\mu}}_n, \dot{\mathbf{\Sigma}}_n)} \quad \text{for } i = 1, ..., N_V \quad (13)$$

$$L_k = \sum_{i=1}^{N_V} l_{k,i} \quad (14)$$

$$\hat{m}_k = \frac{L_k}{N_V} \quad (15)$$

$$\hat{\vec{\mu}}_k = \frac{1}{L_k} \sum_{i=1}^{N_V} \vec{x}_i \; l_{k,i} \quad (16)$$

$$\hat{\mathbf{\Sigma}}_k = \frac{1}{L_k} \left[\sum_{i=1}^{N_V} \vec{x}_i \vec{x}_i^T l_{k,i} \right] - \hat{\vec{\mu}}_k \hat{\vec{\mu}}_k^T \quad (17)$$

The final parameters, $\{m_k, \vec{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^{N_M}$, are found by adapting the UBM parameters as follows:

$$m_k = [\alpha \hat{m}_k + (1-\alpha)\dot{m}_k]\,\gamma \quad (18)$$

$$\vec{\mu}_k = \alpha \hat{\vec{\mu}}_k + (1-\alpha)\dot{\vec{\mu}}_k \quad (19)$$

$$\mathbf{\Sigma}_k = \left[\alpha \left(\hat{\mathbf{\Sigma}}_k + \hat{\vec{\mu}}_k \hat{\vec{\mu}}_k^T\right) + (1-\alpha)\left(\dot{\mathbf{\Sigma}}_k + \dot{\vec{\mu}}_k \dot{\vec{\mu}}_k^T\right)\right] - \vec{\mu}_k \vec{\mu}_k^T \quad (20)$$

where γ is a scale factor to make sure all mixture weights sum to 1. $\alpha = \frac{L_k}{L_k + r}$ is a data-dependent adaptation coefficient where r is a fixed relevance factor (typically $r = 16$, Ref.[7]). It must be noted that UBM mixture components will only be adapted if there is sufficient correspondence with client training data. Thus to prevent the final client models not being specific enough, the UBM must adequately represent the general client population.

5. EXPERIMENTS AND RESULTS

5.1. VidTIMIT Audio-Visual Database

The VidTIMIT database [11], created by the authors, is comprised of video and corresponding audio recordings of 43 people (19 female and 24 male), reciting short sentences. It was recorded in 3 sessions, with a mean delay of 7 days between Session 1 and 2, and 6 days between Session 2 and 3. The mean duration of each sentence is 4.25 seconds, or approximately 106 video frames. For more information on the database, please see [11] or http://spl.me.gu.edu.au/vidtimit/

5.2. Experiment Setup

Before feature extraction can occur, the face must first be located [13]. Furthermore, to account for varying distances to the camera, a geometrical normalization must be performed. We treat the problem of face location and size normalization as separate from feature extraction.

To find the face, we use template matching with several prototype faces of varying dimensions. Using the distance between the eyes as a size measure, an affine transformation is used [9] to adjust the size of the image, resulting in the distance between the eyes to be the same for each person. Finally a 56×64 pixel face window, $w(y, x)$, containing the eyes and the nose (the most invariant face area to changes in the expression and hair style) is extracted from the image.

Fig. 1. Example face windows; **left:** clean; **middle:** corrupted with illumination change; **right:** corrupted with compression artefacts (PSNR=31.7 dB)

For PCA, the dimensionality of the face window is reduced to 40 (choice based on the work by Samaria [14] and Belhumeur [15]). For 2-D DCT and DCT-mod2 methods, each block is 8×8 pixels. Moreover, each block overlaps with horizontally and vertically adjacent blocks by 50%. The dimensionality of 2-D DCT and DCT-mod2 feature vectors is 15 and 18, respectively. For Gabor features, we follow Duc [2] where the dimensionality of the Gabor feature vectors is 18. The location of the wavelet centers was chosen to be as close as possible to the centers of the blocks used in DCT-mod2 feature extraction.

To reduce the computational burden during modeling and testing, every second video frame was used. For each feature extraction method, 8 mixture client models (GMMs) were generated from features extracted from face windows in Session 1.

For experiments involving an illumination change, the method described in [11] (using $\delta = 80$) was utilized to introduce an artificial illumination change to face windows extracted from Sessions 2 and 3.

For experiments involving compression artefacts, face windows extracted from Sessions 2 and 3 were processed by a JPEG codec [16], resulting in an average PSNR of 31.13 dB. Example face windows are shown in Fig. 1.

To find the performance, Sessions 2 and 3 were used for obtaining example opinions of known impostor and true claims. Four utterances, each from 8 fixed persons (4 male and 4 female), were used for simulating impostor accesses against the remaining 35 persons. For each of the remaining 35 persons, their four utterances were used separately as true claims. 10 background models were selected from the 35 client models ($N_\Phi = N_\Psi = 10$). The impostor utterances were not used during the generation of λ_{UBM}. When deriving client models from λ_{UBM}, only the weights and means were adapted - preliminary experiments showed that adapting the covariance matrices resulted in poorer performance.

For each experimental configuration, there were 1120 impostor and 140 true claims. The (person independent) decision threshold was then set so the *a posteriori* performance is as close as possible to Equal Error Rate (EER) (ie. where the False Acceptance Rate is equal to the False Rejection Rate). In all experiments, clean and corrupted face windows were used.

In the first experiment, EER performance of all face authentication systems was found without normalization ($\mathcal{L}(X|\lambda_{\overline{C}}) = 0$). Results are shown in Fig. 2.

In the second experiment, the impostor likelihood was calculated using client specific BMS. All models were constructed directly from the training data. Results are shown in Fig. 3.

In the final experiment, the impostor likelihood was calculated using the UBM approach and the client models were constructed by adapting λ_{UBM}. Results are shown in Fig. 4.

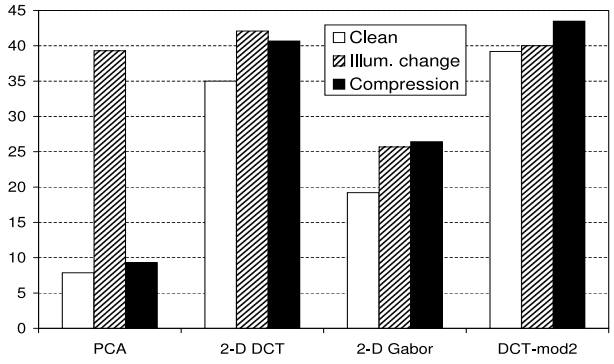

Fig. 2. EER performance without normalization

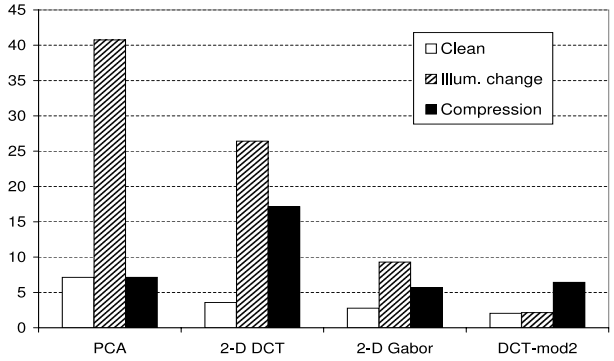

Fig. 3. EER performance using BMS normalization

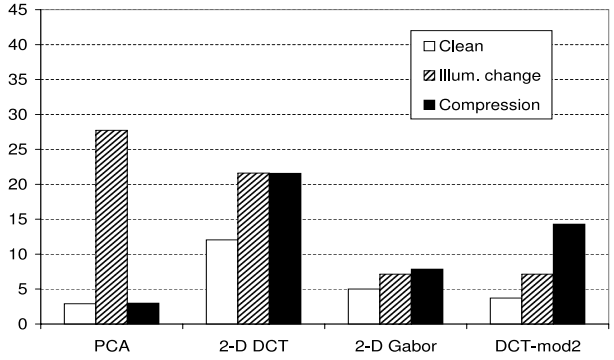

Fig. 4. EER performance using UBM normalization

6. DISCUSSION AND CONCLUSIONS

When using PCA derived features, the BMS based normalization has very little effect on the performance. This is in contrast to the UBM based normalization, where it appears that there are significant performance gains when using clean and corrupted images (eg. when using illumination corrupted images, the EER is reduced from 39.29% to 27.73%).

Recall that data from all clients is used to find λ_{UBM}. In the UBM approach, client models are created by adapting λ_{UBM} (via MAP) using client specific data. This is in contrast to directly computing the client models using the EM algorithm, where only client specific data is used. Effectively there is approximately 30 times more data used during MAP based training than in direct EM based training. Thus the apparent performance improvement when using the UBM based normalization can be attributed to MAP training of the client models rather than the process of likelihood normalization. Further experiments (not reported here) support this assertion.

The rest of the discussion concerns 2-D DCT, 2-D Gabor and DCT-mod2 features. When using these features with the GMM classifier, the spatial relation between major face features (eg. eyes and nose) is lost. While this inherently allows a degree of robustness to image translation, it results in poor performance when compared to the PCA/GMM combination. Thus in these cases, use of likelihood normalization is important in order to obtain good performance. The gains are quite staggering - eg. for DCT-mod2 features, the EER drops from 39.2% to 2.05% when using clean images and the BMS normalization approach. It can be observed that the BMS approach generally provides the most performance gain. The UBM approach is only better for two cases: 2-D DCT and 2-D Gabor features with face windows corrupted with the illumination change.

These experiments also allow us to compare the relative robustness of all the features. We can observe that PCA derived features are the most affected by the illumination change, while being the least affected by compression artefacts. When employing likelihood normalization, DCT-mod2 features are generally the least affected by the illumination change, closely followed by 2-D Gabor wavelets. However, 2-D Gabor wavelets, compared to DCT-mod2 features, are less affected by compression artefacts.

7. REFERENCES

[1] A. Tefas et al., "Using Support Vector Machines to Enhance the Performance of Elastic Matching for Frontal Face Authentication", *IEEE Trans. Patt. Analysis and Machine Intell.*, Vol. 23, No. 7, 2001.

[2] B. Duc et al., "Face Authentication with Gabor Information on Deformable Graphs", *IEEE Trans. Image Proc.*, Vol. 8, No. 4, 1999.

[3] F. Smeraldi et al., "Face Authentication by retinotropic sampling of the Gabor decomposition and Support Vector Machines", *Proc. 2nd Int. Conf. AVBPA*, Washington, 1999.

[4] S. Pigeon and L. Vandendorpe, "Image-based multimodal face authentication", *Signal Processing*, Vol. 69, No. 1, 1998.

[5] A. Rosenberg and S. Parthasarathy, "Speaker Background Models for Connected Digit Password Speaker Verification", *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Atlanta, 1996.

[6] D. Reynolds, "Speaker Identification and Verification Using Gaussian Mixture Speaker Models", *Speech Communication*, Vol. 17, No. 1-2, 1995.

[7] D. Reynolds et al., "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, Vol. 10, No. 1-3, 2000.

[8] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991.

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

[10] T. S. Lee, "Image Representation Using 2D Gabor Wavelets", *IEEE Trans. Patt. Analysis and Machine Intell.*, Vol. 18, No. 10, 1996.

[11] C. Sanderson and K. K. Paliwal, "Polynomial Features for Robust Face Authentication", *Proc. Int. Conf. Image Processing*, Rochester, New York, 2002.

[12] T. K. Moon, "Expectation-maximization Algorithm", *IEEE Signal Processing Magazine*, Vol. 13, Iss. 6, 1996.

[13] L-F. Chen et al., "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.

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

[15] P. N. Belhumeur et al., "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Trans. Patt. Anal. and Machine Intell.*, Vol. 19, No. 7, 1997.

[16] G. K. Wallace, "The JPEG Still Picture Compression Standard", *Communications of the Association for Computing Machinery*, Vol. 34, No. 4, 1991.