Visual processing-inspired Fern-Audio features for Noise-Robust Speaker Verification

Anindya Roy
Idiap Research Institute, Martigny
Ecole Polytechnique Fédérale de Lausanne, Switzerland
Anindya.Roy@idiap.ch

Sébastien Marcel
Idiap Research Institute
Martigny, Switzerland
Sebastien.Marcel@idiap.ch

ABSTRACT

In this paper, we consider the problem of speaker verification as a two-class object detection problem in computer vision, where the object instances are 1-D short-time spectral vectors obtained from the speech signal. More precisely, we investigate the general problem of speaker verification in the presence of additive white Gaussian noise, which we consider as analogous to visual object detection under varying illumination conditions. Inspired by their recent success in illumination-robust object detection, we apply a certain class of binary-valued pixel-pair based features called Ferns for noise-robust speaker verification. Intensive experiments on a benchmark database according to a standard evaluation protocol have shown the advantage of the proposed features in the presence of moderate to extremely high amounts of additive noise.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*Feature evaluation and selection*; I.5.3 [**Pattern Recognition**]: Applications—*Signal Processing*

General Terms

Algorithms, Performance, Design, Verification

Keywords

Speaker verification, noise robustness, additive white Gaussian noise, Local Binary Patterns, binary features, object detection

1. INTRODUCTION

Speaker verification systems have emerged as one of the major applications of speech processing technologies [1], used for secure financial transactions, access control to locations, computer systems and networks [6]. The goal of automatic speaker verification is to accept or reject a claimed identity based on the speech samples obtained from the user [2]. In this work, we considered text-independent speaker verification.

The first step in a speaker verification system is feature extraction. It extracts speaker-specific information from the speech data. Assuming speech signals are generated from quasi-stationary processes, short-term spectral analysis is applied to short speech segments, yielding a sequence of short-time spectra carrying speaker-specific information [6]. Usually, these short-time spectra are further transformed into more sophisticated feature vectors, involving further computations [1] like Mel-frequency Cepstral Coefficients (MFCC) [3]. After extraction, the features are modelled using various techniques, the most popular being Gaussian Mixture Models (GMM) [14] where generally a client-specific speaker model and a client-independent world model are trained. A score is calculated, typically a log-likelihood ratio, and a decision is obtained by comparing this score to a pre-defined threshold.

One of the main challenges to current speaker verification systems is the presence of background noise which tend to degrade verification performance. The second challenge is the reduction of computational complexity of the system at the same time.

In this work, we have addressed these challenges by viewing the problem of speaker verification from the perspective of computer vision. More precisely, we consider the spectral vectors obtained from speech data collected from a particular user as instances of a particular object class. The corruption by additive noise is considered as analogous to ambient illumination changes which change the appearance of the object. In this context, binary-valued pixel-pair based features like Local Binary Patterns (LBP) [11], [15] and Ferns [12] have been recently used successfully for fast illumination-robust object detection. By definition, their value depends on the comparison of intensities of two pixel values, thus they are independent of monotonic illumination changes. Furthermore, the final decision is based on a combination of individual decision scores calculated from a set of distinct features. Some of the individual features might produce erroneous decision scores due to noise but it is less likely that their combined decision will be erroneous. Drawing from this idea, we have applied a certain class of these features called single-pair Ferns in the problem of noise-robust speaker verification. We hypothesize that such illumination-robust features will show a more graceful degradation of verification performance with increasing amount of noise, as compared to conventional features like

SAC'10 March 22-26, 2010, Sierre, Switzerland.

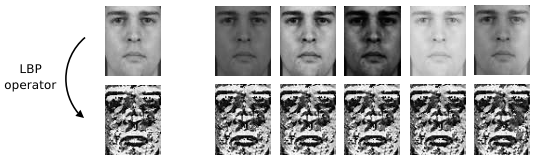

Figure 1: LBP robustness to monotonic gray-scale transformations. On the top row, the original image (left) as well as several images (right) obtained by varying the brightness, contrast and illumination. The bottom row shows the corresponding LBP images which are almost identical.

MFCC. Furthermore, our binary features are simple to calculate, requiring a small number of comparisons and additions per feature vector, as compared to additional computations for the calculation of MFCC. Finally, we use a very simple feature selection method, and a Naive-Bayes classifier for obtaining the decision, as opposed to GMM which is more computationally intensive. To our knowledge, this class of pixel-pair based binary features have not been used before in the context of speaker verification. Haitsma et al. have used bit matrices obtained from comparisons of filter-banked spectrogram values as Audio Fingerprint blocks [5]. However, their work differs from ours in several aspects, in terms of methodology and application.

The rest of the paper is organized as follows. In Sec.2, we give a brief overview of binary features followed by a detailed description of the proposed Fern-Audio features in Sec.3. We describe our experiments in Sec.4 and highlight certain aspects of our work in Sec.5. Finally, Sec.6 outlines the main conclusions of our work.

2. BINARY FEATURES

Let us consider a feature vector $\overrightarrow{X} = [X(1), \cdots, X(L)]^T$. In computer vision, $\overrightarrow{X}$ is an image and each element a gray-level pixel intensity. A binary feature ϕ_i is defined by an ordered pair $\mathbf{k^{(i)}} = \{k_1^{(i)}, k_2^{(i)}\}$, with its elements varying from 1 to L. Its value $\phi_i(\overrightarrow{X})$ is 1 if $X(k_1^{(i)})$ is greater than or equal to $X(k_2^{(i)})$ and 0 otherwise. Thus it remains unchanged as long as $X(k_1^{(i)})$ and $X(k_2^{(i)})$ preserve their relationship. This makes such features robust to monotonic illumination changes. As an example, fig.1 shows the robustness of the Local Binary Pattern feature, a generalization of the simple feature ϕ_i to changes in brightness, contrast and illumination.

3. FERN-AUDIO FEATURES

3.1 Feature representation

The proposed Fern-Audio features are an adaptation of the binary features in Sec.2 to audio data. Given a speech waveform, it is broken up into overlapping frames and a Hamming window is applied to each frame. An N-point Fast Fourier Transform (FFT) is applied to each frame. The magnitudes of the Fourier coefficients are calculated and half of them are removed since they are redundant, forming the spectral vector $\overrightarrow{X}$ of length $L = \frac{N}{2} + 1$. $\overrightarrow{X}$ is considered the equivalent of images of objects of a particular class in an object detection task. As in Sec.2, each Fern-Audio feature ϕ_i is parameterized by two numbers, in this case two frequencies, $k_1^{(i)}$ and $k_2^{(i)}$ lying between 0 and $\frac{N}{2}$. The Fern feature $\phi_i(\overrightarrow{X})$ is defined as,

$$\phi_i(\overrightarrow{X}) = \begin{cases} 1 & \text{if } X(k_1^{(i)}) \geq X(k_2^{(i)}), \\ 0 & \text{if } X(k_1^{(i)}) < X(k_2^{(i)}). \end{cases} \quad (1)$$

Following the bounds for $k_1^{(i)}$ and $k_2^{(i)}$, the number of such features is $N_F = (\frac{N}{2} + 1)^2$. The features ϕ_i are binary features which simply model the interrelationship between the individual frequency components of the spectral vectors. The motivation behind using these features is that, different channel conditions and other effects might degrade the audio signal leading to a change in the shape of the spectrum emitted by the speaker. However, it is hypothesized that the precise relationship between frequency components, whether one is greater or lesser than another, will remain untouched, not for all such features but at least for a majority of selected features within a single spectral vector.

3.2 Classifier design

Given a test spectral vector $\overrightarrow{X}$, the aim is to decide if it belongs to the client class, ω_{client} or the impostor class, ω_{imp}. Let us denote by $\omega(\overrightarrow{X})$ the yet unknown class of the test vector, by $\Phi = \{\phi_1, \phi_2, \cdots \phi_{N_F}\}$ the complete Fern feature set, and by $F = \{f_1, f_2, \cdots f_{N_F}\}$ the corresponding Fern feature values extracted from $\overrightarrow{X}$, i.e. $\phi_i(\overrightarrow{X}) = f_i$, where $f_i \in \{0, 1\}$ for each $i \in \{1, \cdots, N_F\}$. We define a score $S(\overrightarrow{X})$ calculated from the vector, $\overrightarrow{X}$ as a log-likelihood ratio,

$$S(\overrightarrow{X}) = \log\left[\frac{P(\omega(\overrightarrow{X}) = \omega_{client}|\Phi = F)}{P(\omega(\overrightarrow{X}) = \omega_{imp}|\Phi = F)}\right]. \quad (2)$$

Thus, we decide that $\overrightarrow{X}$ belongs to the client class, ω_{client} if $S(\overrightarrow{X}) > \theta$, a pre-defined threshold optimized over a separate development set, and to the impostor class otherwise [1]. Assuming conditional independence between the features and a Naive Bayes probabilistic model, we can calculate,

$$\begin{aligned} S(\overrightarrow{X}) &= \log\left(\frac{P(\omega(\overrightarrow{X}) = \omega_{client})}{P(\omega(\overrightarrow{X}) = \omega_{imp})}\right) \\ &+ \log\left(\frac{\prod_{\phi_i \in \Phi} P(\phi_i = f_i|\omega(\overrightarrow{X}) = \omega_{client})}{\prod_{\phi_i \in \Phi} P(\phi_i = f_i|\omega(\overrightarrow{X}) = \omega_{imp})}\right) \\ &= C + \sum_{\phi_i \in \Phi} \sigma_i(\overrightarrow{X}) \end{aligned} \quad (3)$$

where $C = \log\left(\frac{P(\omega(\overrightarrow{X}) = \omega_{client})}{P(\omega(\overrightarrow{X}) = \omega_{imp})}\right)$ is the contribution from the client and impostor prior probabilities assumed to be constant, while σ_i is the contribution of each feature, ϕ_i,

$$\sigma_i(\overrightarrow{X}) = \log\left[\frac{P(\phi_i = f_i|\omega(\overrightarrow{X}) = \omega_{client})}{P(\phi_i = f_i|\omega(\overrightarrow{X}) = \omega_{imp})}\right]. \quad (4)$$

To calculate $\sigma_i(\overrightarrow{X})$, we approximate by counting over the training dataset,

$$P(\phi_i = f_i|\omega(\overrightarrow{X}) = \omega_{client}) = \frac{\sum_{r=1}^{N_{client}} \mathbf{1}_{\{\phi_i(\overrightarrow{X}_{client}^{(r)}) = f_i\}}}{N_{client}} \quad (5)$$

N_{client} is the number of client training samples, $\{\overrightarrow{X}_{client}^{(r)}\}_{r=1}^{N_{clients}}$ is the client training dataset. For the impostor case, we count over a single world model dataset which is client independent,

$$P(\phi_i = f_i|\omega(\overrightarrow{X}) = \omega_{imp}) = \frac{\sum_{r=1}^{N_{imp}} \mathbf{1}_{\{\phi_i(\overrightarrow{X}_{imp}^{(r)})=f_i\}}}{N_{imp}} \quad (6)$$

where N_{imp} is the number of world model training samples, $\{\overrightarrow{X}_{imp}^{(r)}\}_{r=1}^{N_{imp}}$, the world model training dataset. In both cases, the calculation is finished in only a single pass over the training data.

3.3 Feature selection

We need to select optimal features from the complete set Φ. For this, we consider the worst-case scenario, i.e. the case with the minimum magnitude of σ_i. Precisely, we select those features ϕ_i which maximize this minimum magnitude of σ_i because it can be shown that this will in turn maximize the magnitude of $S(\overrightarrow{X})$ (ref. Eqn.3). Indeed, it is clear that larger the magnitude of $S(\overrightarrow{X})$, more reliable our decision will be. We consider the minimum possible value of σ_i as the efficiency of the feature ϕ_i, denoted by E_i. Using Eqn.4, it can be calculated as,

$$E_i = \min \left[\left| \log \left(\frac{P(\phi_i = 1|\omega(\overrightarrow{X}) = \omega_{client})}{P(\phi_i = 1|\omega(\overrightarrow{X}) = \omega_{imp})} \right) \right|, \right. \quad (7)$$
$$\left. \left| \log \left(\frac{P(\phi_i = 0|\omega(\overrightarrow{X}) = \omega_{client})}{P(\phi_i = 0|\omega(\overrightarrow{X}) = \omega_{imp})} \right) \right| \right].$$

We observe that the efficiency measure E_i is somewhat related to the conditional entropy [13] associated with the feature ϕ_i with the expectation being replaced by the minimum value. In other words, given a speaker class ω_{client}, an optimal feature for that class (i.e one with a high value of E_i), is one which has a constant value (either 1 or 0 in this case) over most of the spectral vectors belonging to that class, and a different value (0 or 1 respectively) for spectral vectors of the impostor class ω_{imp}. All the features $\{\phi_i\}$ are sorted in descending order of their E_i. The best N_b features from among the complete set Φ are chosen as those with the highest N_b values of E_i. It is expected that these features will contribute to increased magnitude of the final score S, and hence will make the final classification more reliable.

Furthermore, due to higher E_i values, these features are typically associated with frequency components $X(k_1^{(i)})$ and $X(k_2^{(i)})$ whose magnitudes are well separated from each other, for that particular client. Hence, when speech is corrupted by additive noise, the corresponding feature value, $\phi_i(\overrightarrow{X})$ (Eqn.1) is less likely to change, irrespective of whether the noise is wideband or narrowband.

In practice, we can reduce the feature selection time by starting with a much smaller subset of features by a random pre-selection without degradation of performance. After the N_b features are selected, the final score is obtained by summing the scores across those N_b features for every frame, and finally summing and normalizing the scores over the entire speech signal.

4. EXPERIMENTS

4.1 Database and Protocols

We tested our proposed Fern-based audio features on the XM2VTS audio database [9]. We chose this database because it is part of publicly available standard database [7], [10], [4], with a clearly defined set of protocols. Also, the speech data is relatively clean. We required clean data because we wanted to investigate the effect of additive white Gaussian noise on the speech, starting from clean speech, with a Signal to Noise Ratio (SNR) of around 30dB to very high values of noise (SNR = -10 dB). The XM2VTS database comprises of four sessions with 200 clients and 95 impostors, each session having two recordings called the first and second shot. Each recording lasts about 4 seconds. Following the Lausanne protocol 1 [8], we used the first shot of Sessions 1, 2 and 3 of the database as the training data for each client model and the second shot of Sessions 1, 2 and 3 as the development data to select the decision threshold. We selected a global threshold optimized according to Equal Error Rate (EER, i.e. Percentage of False Acceptance = Percentage of False Rejection) on the development data. We have not considered any score normalization techniques nor client-specific thresholds.

Finally, for the test we used the original data from Session 4 (both shots 1 and 2) of the XM2VTS database, following the Lausanne protocol 1. Additionally, we considered noisy speech signal by adding white Gaussian noise at SNR = 20dB, 15dB, 10dB, 5dB, 0dB, -5dB and -10dB to the original clean speech, obtaining 7 noisy speech signals for each speech file. We tested separately on these clean and noisy speech data and examine the degradation of performance of our proposed system comparing it with the performance of a baseline MFCC-GMM system. For testing, we use the global threshold obtained in the development phase to calculate the False Acceptance and False Rejection rates. Following usual practice, we report the Half Total Error Rate (HTER) which is the average of the two.

4.2 Experimental Setup

For this database, the sampling frequency f_s is $8kHz$. The speech waveform is broken up into frames of length 20 msec, with an overlap of 10 msec. For the FFT, we chose $N = 256$. For the feature selection, we examined with values of the number of features, N_b ranging from 1 to 1000.

We compared our proposed system to a baseline MFCC-GMM system with two configurations (2 and 32 Gaussians) [1]. The baseline system included a noise-removal step by Cepstral Mean Subtraction (CMS) [1]. Our proposed setup has no equivalent noise-removal step because we hypothesize it will be relatively robust to such noise. For the baseline systems, we also implemented Silence Removal by modelling the speech/non-speech segments using a Bi-Gaussian fit over the MFCC features [1]. However, for our proposed features, we cannot use such a technique, because we do not calculate MFCC and also we wanted to maintain the low computational cost requirements. As an alternative, we first sorted the frame energies, and retained 20% of the higher energy frames for training and 10% for testing. It is to be noted that these are not optimized and purely experimental.

To justify our feature selection scheme, we also compared our proposed system with a similar system using the same Fern-Audio features but where the features are selected uni-

Table 1: *Speaker Verification performance (HTER%) on the XM2VTS database.*

	System	Configuration	Silence removal	clean	20dB	15dB	10dB	5dB	0dB	-5dB	-10dB
1	MFCC-GMM	32 Gaussians	Bi-Gaussian	**1.4**	**9.5**	**20.0**	33.7	41.1	45.4	47.7	**49.1**
	(Baseline)	2 Gaussians		7.8	20.5	26.1	**32.2**	**35.7**	**39.6**	**46.0**	49.6
		32 Gaussians	Frame energy	1.9	10.7	20.3	33.1	42.1	46.5	48.6	49.7
		2 Gaussians	sorting	8.6	18.8	26.3	32.1	39.6	44.3	47.3	48.8
2	Fern-Audio	$N_b = 5$		17.1	18.2	18.9	19.5	21.7	21.6	24.2	37.1
	features	$N_b = 10$	Frame energy	16.2	17.8	18.0	18.7	19.3	20.5	21.6	34.8
	with optimized	$N_b = 20$	sorting	**14.6**	**15.3**	**15.8**	**16.9**	**18.7**	**20.3**	**21.7**	**37.2**
	feature selection	$N_b = 50$		15.2	16.5	17.1	18.3	20.0	21.7	23.9	37.3
3	Fern-Audio	$N_b = 5$		33.5	40.0	41.7	43.8	46.0	47.0	48.0	48.9
	features	$N_b = 10$	Frame energy	25.5	36.2	39.1	42.2	44.4	46.2	47.9	48.9
	with random	$N_b = 20$	sorting	23.4	35.0	39.1	43.0	45.8	48.1	48.4	49.3
	feature selection	$N_b = 50$		21.3	34.1	39.2	43.9	45.5	48.7	48.9	49.7

formly randomly from the feature set Φ instead of using the optimality criterion in Eqn.7.

4.3 Results

In Table 1, we report the verification performance of different systems, in terms of the HTER% on the test data, using a global threshold optimized according to EER criterion on the development data (ref. Sec.4.1). A total of 3 systems are represented. System 1 is the baseline, with (16 MFCC + 16 delta-MFCC) features [4], modelled by GMM (either with 2 Gaussians or 32 Gaussians) [1]. It has a CMS step, and silence removal is either by bi-Gaussian or by sorting on frame energies. System 2 shows our proposed approach, with the number of optimally chosen Fern-Audio features N_b varying from 5 to 50. System 3 uses the proposed Fern-Audio features but uses random selection. The HTER% for the best performing configuration is highlighted in bold for the first two systems.

5. DISCUSSIONS

Performance From Table 1, we observe that the baseline system performs far better than the proposed system on clean speech. However, our proposed system begins to outperform the best among the baseline system configurations (highlighted in bold, system 1, Table 1) as the SNR falls below 20dB. At SNRs at and below 10dB, the baseline system quickly approaches pure chance performance (50%) while the performance of our proposed system degrades much more slowly, the HTER remaining below 25% till SNR=-5dB which is a statistically significant improvement over the baseline system. The best performance for the Fern features occurs at an N_b of 10 to 20. We also examined cases of N_b from 1 to 1000, however, the performance degrades in both directions.

System requirements Our proposed system is computationally much faster because 1) there is no computation of summation over bands, logarithms and inverse DCT as in the baseline MFCC system, 2) there is no CMS step, 3) feature modelling is by a Naive Bayes probabilistic model involving only a small number of additions (10 to 20 corresponding to the optimal N_b), 4) feature selection is by a simple counting procedure and requires only one pass over the feature set, and 5) during testing of the model, each Fern can be obtained by one simple comparison operation in contrast to GMM-based systems which involve more intensive calculations. Our proposed system is also less intensive on storage space requirements because only $2 \times N_b$, i.e. 20 to 40 σ_i values need to be stored per client model, while for the baseline system, $2 \times 32 \times 32$, i.e 2048 values need to be stored.

Feature selection The role of our feature selection strategy is justified because our proposed approach performs significantly better than a similar system with randomly chosen features (system 3 in the table).

Comparison details It cannot be argued that the Fern features are performing better than the baseline system due to over-learning by the baseline system because even with a very weak baseline system (system 1, configuration : 2 Gaussians), the HTER% increases rapidly with decreasing SNR. Similarly, it cannot be argued that the Fern features are performing better due to the different silence removal technique used (sorting on frame energies), because an MFCC-GMM system with the same silence removal technique as ours (system 1, 32 and 2 Gaussians, Silence Removal method : sorting) performs similarly as the MFCC-GMM system with silence removal by bi-Gaussian, and is worse than the proposed system with various high levels of noise.

Other reported systems Experiments have been carried out on additive white Gaussian noise-corrupted speech data from the XM2VTS database using conventional MFCC features by Nefian et al.[10] and Fox et al.[4]. Our system outperforms theirs by a significant margin.

Robustness to voice synthesis A useful aspect of the proposed framework is that the Fern models do not store any client-specific spectral shape information. They only store discriminative frequency points and the table of log-likelihood ratios, $\{\sigma_i\}$. Thus, the proposed models may be more robust against efforts to reconstruct a synthetic voice model from stolen model parameters than an equivalent MFCC-GMM model. This could be an advantage because this means that such a verification system is not susceptible to attacks using stolen model data.

Future work As a part of future work, we shall investigate the effect of other noise types on the proposed method, for example, babble noise and pink noise. It is to be noted that the proposed features individually examine only a distinct and small part of the whole spectrum (two frequency

components). Hence, additive noise might affect some of the selected features, but it is less likely that it will affect too many of them all at the same time, so as to change the final decision. This will be true irrespective of the type of noise, whether wideband or narrowband, corrupting the speech data. This is likely to make the proposed method robust even in such cases. In future, we shall also use other more challenging speaker corpora for the speaker verification experiments and investigate more sophisticated feature selection strategies than the current Naive-Bayes framework.

6. CONCLUSIONS

Inspired by recent advances in computer vision, we have proposed an alternative feature set called Fern-Audio features for speaker verification. Using a Naive-Bayes classifier and a simple feature selection procedure, our system outperforms the baseline system in an unmatched noisy test scenario involving additive white Gaussian noise across a wide range of SNRs. At the same time, it is less intensive on computation and storage requirements than the baseline system showing promise for use in real-time systems.

7. ACKNOWLEDGEMENTS

The authors would like to thank the Swiss National Science Foundation (projects 200020-122062 and 51NF40-111401) and the FP7 European MOBIO project (IST-214324) for their financial support. The authors would like also to thank Dr Mathew M. Doss for his helpful comments and advice.

8. REFERENCES

[1] F. Bimbot et al. A Tutorial on Text-Independent Speaker Verification. *EURASIP Journal on Applied Signal Processing*, (4):431–451, 2004.

[2] J. Campbell. Speaker recognition : A tutorial. In *Proc. IEEE*, volume 85, pages 1437–1462, 1997.

[3] S. Davies and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continously spoken sentences. *IEEE Trans. on ASSP*, 28(4):357–366, 1980.

[4] N. Fox et al. Person identification using automatic integration of speech, lip and face experts. In *Proc.ACM SIGMM 2003 Multimedia Biometrics Methods and Applications Workshop(WBMA'03),Berkeley,CA*, pages 25–32, 2003.

[5] J. Haitsma and T. Kalker. A Highly Robust Audio Fingerprinting System. In *International Symposium on Music Information Retrieval*, 2002.

[6] S. Kung, M. Mak, and S. Lin. *Biometric Authentication*. Prentice Hall, 2005.

[7] J. Luettin. Speaker Verification Experiments on the XM2VTS database. Idiap Research Report 99-02, Idiap, 1999.

[8] J. Luettin and G. Maitre. Evaluation protocol for the extended M2VTS database (XM2VTSDB). Idiap Communication 98-05, Idiap, 2000.

[9] K. Messer, J. Matas, J. Kittler, J. Lüttin, and G. Maitre. XM2VTSDB: The Extended M2VTS Database. In *AVBPA*, pages 72–77, 1999.

[10] A. Nefian et al. A Bayesian approach to audio-visual speaker identification. In *AVBPA*, pages 761–769, 2003.

[11] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29, 1996.

[12] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *CVPR*, 2007.

[13] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill International Editions, 1999.

[14] D. Reynolds. Experimental evaluation of features for robust speaker identification. *IEEE Trans. on Speech and Audio Processing*, 2(3):639–643, 1995.

[15] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li. Face detection based on Multi-Block LBP representation. In *ICB*, pages 11–18, 2007.