EURASIP Journal on Applied Signal Processing 2004:17, 2640–2649

Pitch Correlogram Clustering for Fast Speaker Identification

Nitin Jhanwar
Research and Development Division, Danlaw Technologies India Limited, Hyderabad 500 034, India
Email: nitinj@danlawinc.com

Ajay K. Raina
Research and Development Division, Danlaw Technologies India Limited, Hyderabad 500 034, India
Email: akraina@danlawinc.com

Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010, Australia

Received 24 June 2003; Revised 25 June 2004; Recommended for Publication by Bastiaan Kleijn

Gaussian mixture models (GMMs) are commonly used in text-independent speaker identification systems. However, for large speaker databases, their high computational run-time limits their use in online or real-time speaker identification situations. Two-stage identification systems, in which the database is partitioned into clusters based on some proximity criteria and only a single-cluster GMM is run in every test, have been suggested in literature to speed up the identification process. However, most clustering algorithms used have shown limited success, apparently because the clustering and GMM feature spaces used are derived from similar speech characteristics. This paper presents a new clustering approach based on the concept of a pitch correlogram that captures frame-to-frame pitch variations of a speaker rather than short-time spectral characteristics like cepstral coefficient, spectral slopes, and so forth. The effectiveness of this two-stage identification process is demonstrated on the IVIE corpus of 110 speakers. The overall system achieves a run-time advantage of 500% as well as a 10% reduction of error in overall speaker identification.

Keywords and phrases: speaker identification, clustering, pitch, correlogram.

1. INTRODUCTION

Speaker recognition aims at extracting and modeling characteristics of speech data that uniquely represent a person. These characteristics should ideally be robust to channel effects and noisy environment [1]. Cepstral coefficients [2], in Mel frequency domain, are the most robust, in this sense, among all feature vectors currently employed in speech recognition systems in a Gaussian mixture model (GMM) framework [3, 4]. At present, these features are also commonly employed for speaker identification, even though the best feature vector for speaker identification, as contrasted with speech recognition, is still an open problem. Recent papers suggest transformed feature vectors for performance enhancement [5] in speaker identification systems. Campbell's paper is still a very good reference to the problem and issues involved in speaker identification [6].

Speaker recognition is done at two levels: verification [7, 8, 9] and identification [1, 10]. Verification systems are closed set operations in which a speaker's claim to be one of the enrolled speakers is verified, generally in a cooperative text prompted mode, as in voice-based access control systems. The system conducts a binary hypothesis test, relative to the claimed identity, on the speech feature data and returns a yes/no result. Speaker identification, on the other hand, involves finding the identity of the speaker from a given test utterance. This is normally done in a text-independent manner, for example, in surveillance operations where voice channels are monitored or in other noninvasive access control applications. Identification systems, in closed set operation, identify the most likely enrolled member as the source of the utterance. In open set identification, where the utterance may or may not belong to the enrolled class, the most likely fit is further verified to return a confirmation.

A major issue in the identification problem is the GMM run-time computation, which is linear in population size. To combat this problem, systems where the enrolled speaker database is partitioned into clusters based on some proximity criteria have been proposed in the literature [11]. In these two-stage identification systems, a test utterance is first mapped into a cluster and then matched to the nearest GMM in that cluster only. This reduces the run-time computation since only a few GMMs have to be run at a time. Also, if clustering is based on features uncorrelated with the GMM

feature space and robust to channel and noise distortion, the overall system error performance improves. In addition, clustering helps in better positioning of GMM priors as well. GMM computation time reduces, on an average by a factor equal to the number of clusters. Computational advantage for complete identification will be somewhat less due to cluster identification time for a test utterance. As we will see later, for large speaker populations, clustering is particularly advantageous when the average cluster size is small, the cluster size variation is low and the cluster identification is reasonably fast. Popular approaches to clustering include vector quantization (VQ) codebook-based clustering (see [12] for comparison of various VQ approaches) and covariance-based clustering (see Wang et al. [11]). Algorithms that cluster speakers in reduced dimension spaces have also been proposed [13]. These algorithms cluster speakers based on projected individual speaker data onto reduced dimension subspaces corresponding to directions of maximum variability (eigenspaces corresponding to large eigenvalues of dataset covariance) of the speaker database. This ensures fast clustering in addition to noise immunity [14].

Most clustering algorithms, however, use speaker features (MFCCs, GMM variances, etc.) based, directly or indirectly, on short-time spectrum analysis of the signal. These features, in addition to inadequate noise and channel distortion robustness, are also the ones used for later GMM identification. This can result in deterioration of performance at times. To see this, note that in the overall two-stage identification process, an error can occur because a given test utterance is either wrongly classified (mapped into the wrong cluster) in the first stage, or correctly classified but mapped onto a wrong GMM within the cluster in the second stage. Therefore, if $P(e)$ denotes the probability of overall two-stage identification error, we have that $P(e) = P(M)\,P(e/M) + (1 - P(M))\,P(e/C)$, where $P(M)$ is the probability of cluster misclassification (the first stage error) and $P(e/C)$ is the probability of second-stage error, that is, the error in mapping onto a wrong GMM from within the correct cluster. Clearly, $P(e/M) = 1$ so that $P(e) \geq P(e/C)$. In case clustering uses the same features as the GMM, $P(e/C)$ will be approximately the same as the probability of error in an unclustered single-stage GMM-based speaker identification system. Therefore, irrespective of the run-time advantage, overall identification performance, $P(e)$, of a two-stage SI system will be good (low $P(e)$) only if the clustering feature space is independent of the GMM feature space. In fact, if a two-stage identification system yields lower $P(e)$ than a single-stage system, it is reasonable to assume that this is due to relative independence of clustering and GMM feature spaces.

In this paper, we introduce a clustering algorithm based on speaker pitch variations for a two-stage speaker identification system. Human pitch is normally found in the 50–400 Hz range (males dominate the lower end and females the higher end). For any speaker, however, there are small (2%–10%) variations in pitch from one voiced speech frame to the next, due to random variations in the vocal fold tension [15, 16]. A frame is taken normally as 20–30 milliseconds (160 to 240 speech samples at 8-kHz rate) in length for speech stationarity. While this variation is too broad to separate nearby speakers, it can group neighborhood speakers quite well. Also, current algorithms for pitch computation are very accurate and robust to noise and channel effects [16, 17]. There have been earlier attempts to base speaker identification only on pitch properties, but these have not succeeded much (beyond separation of sexes) due to reasons mentioned above. Pitch contours, for example, were proposed way back in 1972 for automatic speaker recognition [18] but did not become popular. Attempts to use the log-normal character of pitch for open set speaker identification resulted in high false alarm or miss rates [19]. Algorithms using pitch along with other independent features show better results [20, 21, 22]; for example, it has been shown that vocal-tract-features-based identification systems perform better if pitch histogram information is utilized as well [23].

In this paper, we introduce pitch correlogram as a device to measure frame-to-frame pitch variations and use it to cluster speakers. Pitch is a property of the voiced part of speech and equals the period of the impulse train excitation of the vocal tract for voiced sound production in a short-time stationary model [24]. Given a speech tract, its silent frames are removed using a voice activity detection (VAD) algorithm, after which the voiced speech is separated from unvoiced speech using covariance thresholding, since unvoiced speech has low correlation due to its random aperiodic nature [24]. We divide the whole expected pitch region into fixed equal bands (this paper uses 40), extract pitch for each voiced frame, allocate its corresponding band, and monitor this pitch band allocation from one voiced frame to next. This information is stored in the pitch correlogram, which is a 3-dimensional matrix whose (i, j, k)th entry denotes the joint probability of pitch bands i- and j-, k-frames apart. Clearly, a pitch correlogram captures local variation within histogram bands, which makes it a better source of pitch information.

In the sequel, pitch is estimated using the mixed excitation linear prediction (MELP) speech codec pitch estimator that is based on the superresolution pitch estimation algorithm [15]. This algorithm is the most accurate known at present and only $O(N = \text{samples/frame})$ complex. Also, it is robust to noise and channel distortion and does not depend on direct short-time Fourier transform coefficients of speech. Consequently, pitch correlograms are robust to noise as well as reasonably independent of spectral features like the MFCCs or spectral slopes. This is borne out by the performance of the proposed overall two-stage identification system. Since GMM identification is dependent on speaker population and speech SNR, using correlograms is also likely to enhance the overall system performance due to low error in classification, in addition to the computational gains from clustering.

The clustering process is described in Figure 1 and discussed in Section 3. Given reference speech utterances from enrolled speakers, we extract the pitch correlogram for each utterance and use one of these as a reference correlogram and others for training of the algorithm to fix the model size. The overall two-stage identification system is depicted

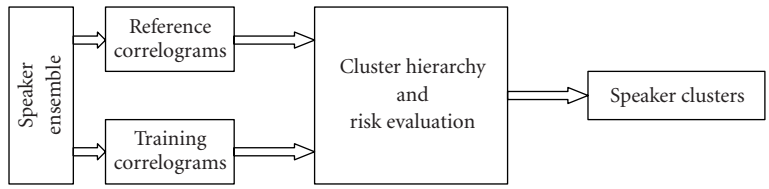


Figure 1: Clustering speaker using correlograms.

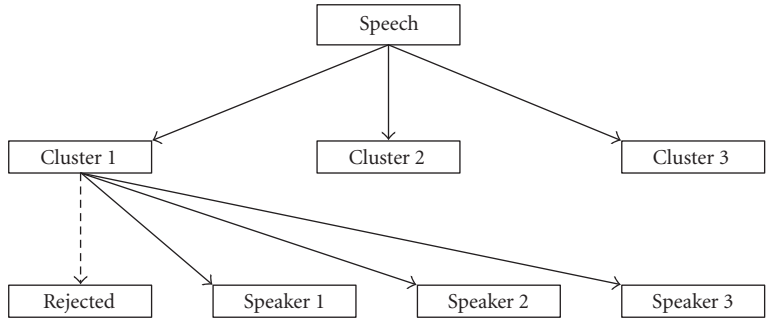


Figure 2: Schematic of the two-stage identification system.

in Figure 2. Given an unknown utterance, it is first mapped onto a cluster based on its correlogram and then passed through GMM models of speakers corresponding to the identified cluster. For closed set speaker identification, the utterance is mapped onto the most likely GMM, while for open set speaker identification, the nearest GMM is accepted only on further verification, for example, by thresholding the feature space *a posteriori* probability. This paper uses a 16-mixture GMM model with the feature space of 13th-order MFCC coefficients and spectral slope for experimental purposes, as in Murthy and Heck [10].

In this paper, "clustering" means offline partitioning of the speaker database using pitch correlograms. "Classification" and/or "misclassification" refer to mapping of a test or training speech track onto a cluster and "overall identification" refers to the complete two-stage speaker identification process in which the test track is first mapped onto a cluster and then identified within the cluster by the normal GMM identification process.

The rest of the paper is organized as follows. In Section 2, we discuss the algorithms used for the estimation of pitch and pitch correlograms. Section 3 develops the clustering algorithm for an optimal number of clusters. Section 4 discusses the experimental results and comparisons with Murthy and Heck [10] results. We conclude in Section 5.

2. PITCH ANALYSIS

Perceptually, pitch is the attribute of auditory sensation in terms of which sounds are ordered on a musical scale. In speech processing, a simplified linear prediction model for voice production is used wherein pitch equals the period of the impulse train that excites the vocal tract (modeled as a linear system) in the voiced mode of articulation [24]. This excitation is produced by vocal cords and its periodicity is reflected in the output voiced sound due to linear processing by the tract. Since speech is only a short-time stationary process (20–30 milliseconds), variation is observed in the pitch from frame to frame; this variation is the principal classification parameter in the sequel as discussed above.

The pitch perception models use temporal information for identifying periodicities in the signal by estimating the period of the autocorrelation function of voiced speech, given by

$$c(l) = \sum s(n)s(n-l), \tag{1}$$

where $s(n)$ is the speech sample and $c(l)$ its autocorrelation at a time lag l.

To obtain a useful set of results, the autocorrelation function is computed over a range of lag values. For periodic signals, the function attains a maximum at sample lags of 0, +/− P, +/−2 P, and so forth, where P is the pitch. This technique is most efficient at mid to low frequencies and is quite popular in speech recognition applications when the pitch range is limited. The autocorrelation function can be further used to differentiate between unvoiced and voiced frames. In the unvoiced speech, the vocal cords are not vibrating, so the resulting speech waveform is aperiodic or random in nature [24]. Therefore the autocorrelation is low for an unvoiced frame, which can be differentiated from a voiced frame using an autocorrelation threshold.

We use the MELP coder pitch extraction algorithm [25], wherein pitch is estimated first in terms of integral sample lags and later compensated for any fractional lag. The sampled speech (normally at 8 kHz) is prefiltered by a 500 Hz LPF and framed into overlapping 22.5-millisecond blocks for retaining process stationarity during autocorrelation computation. The pitch period normally is less than 20 milliseconds (160 samples) so that it is captured easily in a frame.

The normalized autocorrelation function, with lag l, is given by

$$r(l) = \frac{c_l(0,l)}{\sqrt{c_l(0,0)c_l(l,l)}}, \quad \text{where } c_l(m,n) = \sum_{k=-\lfloor l/2\rfloor-79}^{-\lfloor l/2\rfloor+79} s_{k+n}\cdot S_{k+m}. \tag{2}$$

Note that the length of the analysis window is fixed while its starting point depends on lag l. Again, since frame stationarity implies that $c_l(0,0) = c_l(l,l)$, then $r(l)$ and $c_l(0,l)$ reach their maximum at the same lag. The normalized correlation is used, however, to threshold differentiate between unvoiced and weakly voiced frames, since otherwise unstressed phonemes or certain emotional states may result in very low correlation and be taken for unvoiced sounds. The denominator in (2) nullifies the effect of low-impulse train strength voiced frames in the autocorrelation function.

$r(l)$ is computed over 20 to 159 sample lags (pitch in 50 to 400 Hz range) and the lag at which it is maximum is taken as the integral pitch estimate T. Actual pitch is normally at an offset from this value. To find the direction of this offset, we compute $r(T-1)$ and $r(T+1)$. If $r(T-1) > r(T+1)$, we decrement the integral pitch by 1; otherwise, we leave it unchanged. Let Δ be the offset required for the pitch period and denoted as fractional pitch. The actual pitch period P is then $T+\Delta$, but $r(T+\Delta)$ cannot be calculated directly from the sampled speech signal. An interpolation is therefore used to determine $r(T+\Delta)$. Since a low-passed version of the speech signal, with bandwidth much smaller than the sampling rate, is used for pitch estimation, a convex linear interpolation of the signal suffices. Therefore, we use

$$s(n+\Delta) = (1-\Delta)\cdot s(n) + \Delta\cdot s(n+1). \tag{3}$$

Since $\Delta \in [0,1)$, an optimal value of Δ is computed by maximizing the autocorrelation between $s(n)$ and $s(n+T+\Delta)$. As outlined in [15], the optimization can be carried out using the orthogonal projection theorem to give the value of Δ as

$$\Delta = \frac{c_T(0,T+1)c_T(T,T) - c_T(0,T)c_T(T,T+1)}{c_T(0,T+1)\left[c_T(T,T) - c_T(T,T+1)\right] + c_T(0,T)\left[c_T(T+1,T+1) - c_T(T,T+1)\right]}. \tag{4}$$

2.1. Correlogram

Phonetic unit of speech is a phoneme whose intonation (pitch) varies depending on factors like stressed/unstressed vowel or syllable, accent, boundary and edge tones, neighborhood speech, and so forth. The pitch variation in a phoneme is, therefore, considerably influenced by the speaker style and accent. A pitch correlogram expresses the correlation between pitch pairs at frame distances. It captures the pitch variation within a phoneme and at phoneme junctions. Assuming that a speaker employs a reasonably unique pitch variation in pronouncing particular phonemes, the pitch correlogram can be used to capture this variational characteristic (across all phonemes and their combinations if sufficient data are available) and to group speakers with adjacent behavior. This is the main speaker clustering idea of this paper.

Let $S = \{o_1, o_2, o_3, \ldots\}$ denote the voiced speech utterance of a speaker with o_i being its ith frame. From each frame o_i, the pitch is extracted as discussed earlier. Human pitch P, normally between 50–400 Hz, is then quantized into uniform nonoverlapping intervals called pitch bands. Let $\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \ldots, \mathrm{P}_B$ be the B pitch bands and $\mathrm{P} : S \to \{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_B\}$ be the utterance to pitch band map. We define

$$S_{\mathrm{P}_j} = \{o_k \mid \mathrm{P}(o_k) = \mathrm{P}_j;\ k \in \mathbb{Z}_+\}, \tag{5}$$

where $\mathbb{Z}_+$ is the set of positive integers and S_{P_j} is the set of frames with pitch P_j.

The $k(\geq 0)$-delay joint distribution of pitch bands P_i and P_j (in specified order) is defined as

$$\boldsymbol{\lambda}^k_{\mathrm{P}_i,\mathrm{P}_j}(S) = \text{Prob.}\left\{o_l \in S_{\mathrm{P}_j},\ o_{l-k} \in S_{\mathrm{P}_i} \mid l \in \mathbb{Z}_+\right\}. \tag{6}$$

As mentioned above, a pitch correlogram is a 3-dimensional feature matrix of speech with (i,j,k)th entry given by (6). We will, however, use only the next-frame pitch changes ($k = 1$) so that our correlogram is a two-dimensional matrix C, with $C_{ij} = \lambda_{\mathrm{P}_i\mathrm{P}_j}(S)$. Once the voiced frame pitch estimates are made and mapped onto pitch bands, the correlogram entries C_{ij} are given by the estimates

$$C_{ij} = \frac{\text{Number of times pitch band } P_i \text{ is succeeded by } P_j}{\text{Total number of voiced frames} - 1}. \tag{7}$$

Note that $\sum_{i,j=1}^{M} C_{ij} = 1$. The number B of pitch bands used depends on performance requirement and memory and computation time tradeoffs. In our experiments, we found that 35 to 45 pitch bands give reasonably good results. As already mentioned, this paper uses 40 bands.

Since pitch varies by 2%–10% in successive frames, the maximum pitch change is about 20 Hz in the mid pitch range of about 200 Hz. This implies that a pitch band i is most likely to transit to bands i, $i \pm 1$, and $i \pm 2$ in the next frame.

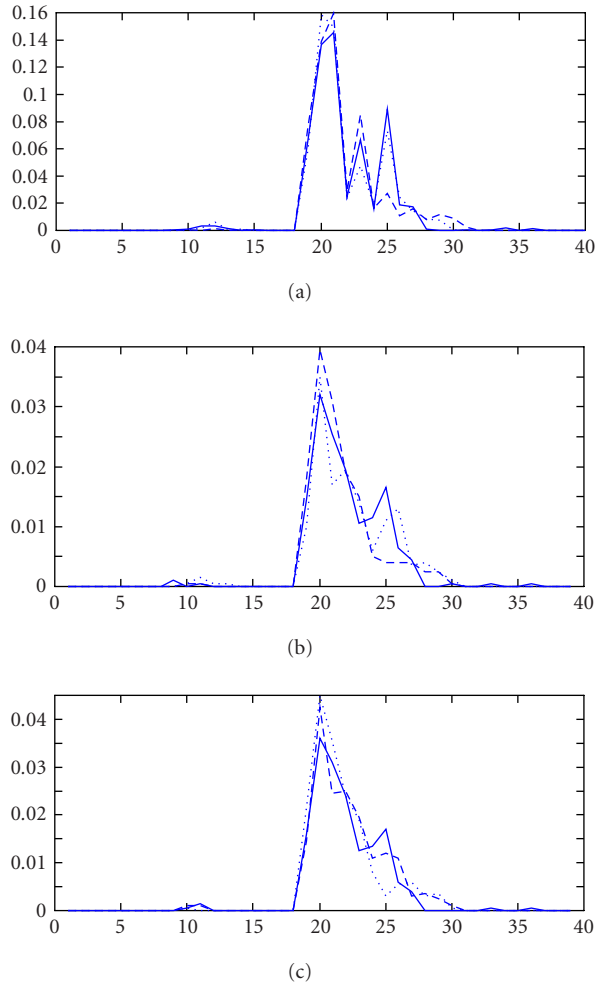


FIGURE 3: (a) Diagonal, (b) postdiagonal, and (c) prediagonal entries of correlograms (y-axis) for three different utterances of a random speaker.

Hence C is dominant on the diagonal band. Figure 3 shows the diagonal, postdiagonal, and prediagonal distribution of three different utterances of a randomly chosen speaker. The highly correlated nature of the three correlograms suggests speaker invariance of this feature. Also, as will be seen below, correlograms tend to cluster in a normed space, which makes them an ideal speaker ensemble partitioning feature. However, they are limited in scope as an indexing (speaker identification) feature because (a) pitch quantization leads to loss of information so that different speakers can occupy the same band and (b) pitch is emotion dependent [26] so that pitch bands need to be sufficiently broad to accommodate such dependence. Since it may not always be possible to reconcile these conflicting requirements, it is not advisable to use the correlogram as the sole indexing feature.

3. CLUSTERING

Clustering implies grouping of objects together based on common characteristics, that is, partitioning a large database on some proximity criteria. It is a technique to understand, simplify, and interpret large amounts of multidimensional data. An ideal clustering algorithm is one that results in high intracluster correlation and low-intercluster correlation [27], that is, it generates a sharp multimodal statistical distribution with identifiable peaks and valleys much like a Gaussian mixture with separated means and comparable covariances. This may not be always possible to achieve and, in practice, algorithms that yield reasonably separated dense groups are used.

We propose a two-step clustering algorithm. In the first step, we take reference voiced speech tracks of enrolled members, generate corresponding correlograms and continuously merge nearest neighbor correlograms. This hierarchy yields as many levels of clusters as the total speaker population, that is, level 1 contains M clusters (M equals the population ensemble size), each with population 1, and level M contains 1 cluster of population M. To fix the particular level in hierarchy for the system, that is, the number and membership of clusters to be used finally, the system is trained by another set of correlograms of the same speaker population. These are called the training correlograms. The training process involves evaluation of a Bayesian risk function—in our case, not strictly convex—for each level in hierarchy relative to the set of training correlograms. The level that results in the smallest number of clusters with minimum risk is chosen as the final model level for the system. The algorithm is explained in detailed steps below.

For the correlogram space, we use the matrix norm[1] $\|A\| = \sum_{i,j} |a_{ij}|$ so that the distance metric is given by $d(X,Y) = \sum_{i,j=1}^{L} |x_{ij} - y_{ij}|$, where X and Y are each $L \times L$. The clustering algorithm is as follows.

(1) Generate correlograms for the speech track reference ensemble. The correlograms are labeled as $C_1^1, C_2^1, C_3^1, \ldots, C_M^1$, where M is the number of reference enrolled speakers. These correlograms are generated on (at least) a 1-minute voice frame track of available speech for each speaker. This is called level-1 clustering, where each cluster has a population of 1 speaker.

(2) Compute pairwise correlogram distances at level N; merge the cluster pair with the least distance into a single cluster and calculate its new representative correlogram (for $(N+1)$th level) as $C_r^{(N+1)} = (\mathrm{P}_s^N C_s^N + \mathrm{P}_t^N C_t^N)/(\mathrm{P}_t^N + \mathrm{P}_t^N)$, where C_s^N and C_t^N are the nearest neighbors at Nth level with respective probabilities (relative sizes of clusters) P_s^N and P_t^N. Relabel the clusters at $N+1$th level as $C_1^{(N+1)}, C_2^{(N+1)}, \ldots, C_{M-N}^{(N+1)}$.

(3) Continue cluster merging up to level M when there is only a single cluster left.

(4) Generate correlograms for cluster training utterances, that is, for all speakers, use some utterances for training the system (these could come from the main

[1] Since the space is finite dimensional, all norms will induce the same topology. Note that all correlograms are on the unit circle. We choose this norm for ease of calculation and interpretation.

ensemble or from outside). Let T_i^N represent the set of training correlograms belonging to speakers from cluster i at level N. We use the same number of training correlograms for each speaker. This implies that at level 1, all T_i^1 have equal number of correlograms in their respective sets. If C_s^N and C_t^N are merged to create level $N+1$, then the training correlogram set for the new cluster, say T_r^{N+1}, will be the union of the training correlograms of the underlying sets, that is, $T_r^{N+1} = T_{Ns}^N \cup T_{Nt}^N$.

(5) Evaluate the following Bayesian risk function at each level (level N has $M - N + 1$ clusters):

$$\text{Risk}_N = \sum_{i,j=1}^{M-N+1} R_{ij}^N \, \text{P}^N(i,j). \tag{8}$$

Bayesian risk is the expected value of a random cost variable, given in our case by R_{ij} as below:

$$R_{ij}^N = \begin{cases} 1, & i \neq j, \\ \text{P}_i, & i = j, \end{cases} \tag{9}$$

P_i is the probability of cluster i, which, for equiprobable speakers, equals the relative cluster size P_i^N. The cost of all wrong decisions is equal, while correct decisions are rewarded in inverse proportion to cluster size. This is done in order to reduce the computational complexity (Section 3.1) of the overall speaker identification system. Let $\text{P}^N(i,j)$ be the probability that an utterance from cluster i is mapped onto cluster j, that is, the training correlogram from cluster i turns out to be closest to the representative correlogram for cluster j. $\text{P}^N(i,j)$ can be estimated as

$$\text{P}^N(i,j) = \frac{\text{Number of training utterances from } T_i^N \text{ mapped into cluster } j}{\text{Total number of training utterances}}. \tag{10}$$

(6) The level at which the risk is the least represents an optimal number of clusters and, in case there are multiple global minima, we use the one with the smallest cluster size.

Experimental results (Section 5) show that the algorithm meets with the properties of a good practical clustering algorithm. Note also that once the algorithm has been executed, each cluster is represented by a single correlogram, which is the mean of all its element correlograms.

3.1. Computational complexity

As mentioned earlier, in the two-stage speaker identification (see also Figure 2) process, an unknown utterance is first mapped onto a cluster and then the speaker from the cluster is identified using standard GMM identification. The overall complexity of the two-stage identification process can be computed in terms of complexity of its subprocesses and compared with a single-stage GMM only identification process. First, some notations are mentioned.

(1) α denotes the computational complexity of calculating a pitch correlogram.
(2) β denotes the computational complexity of comparing two correlograms.
(3) N_0 denotes the number of clusters and, equivalently the number of representative correlograms.
(4) N_i denotes the number of speakers in cluster i.
(5) M denotes the total number of speakers.
(6) γ denotes the computational complexity of running a single GMM.

Clearly, the complexity of a single-stage GMM-based identification process equals $M\gamma$. In the two-stage scenario, the computational complexity of the first stage, that of finding the cluster to which an unknown speaker belongs equals $\alpha + N_0\beta$. In the second stage, for a given identified cluster with population N_i, the complexity equals $N_i\gamma$. Therefore, the expected value of computation in the overall two-stage process is given by

$$\alpha + N_0\beta + \sum_{i=1}^{N_0} \text{P}_i \cdot N_i \cdot \gamma. \tag{11}$$

P_i is the probability of cluster i and equals N_i/M when all the speakers are equiprobable, in which case (11) simplifies to

$$\alpha + N_0\beta + \frac{\gamma \sum_{i=1}^{N_0} N_i \cdot N_i}{M}. \tag{12}$$

The complexity is minimum when all clusters are equal, that is, $N_i = M/N_0$, for all i. In this case it is $M\gamma/N_0 + N_0\beta + \alpha$, which, under the reasonable assumption that $\beta \ll \gamma$, is much smaller than $M\gamma$. This computational advantage makes a two-stage speaker identification system preferable to a single-stage one.

Observe that fewer clusters implies large average cluster size, which in turn, leads to small misclassification (mapping of a test/unknown utterance to a wrong cluster) probability and large GMM computational load. The converse is equally true. Computationally, the worst and the best cases

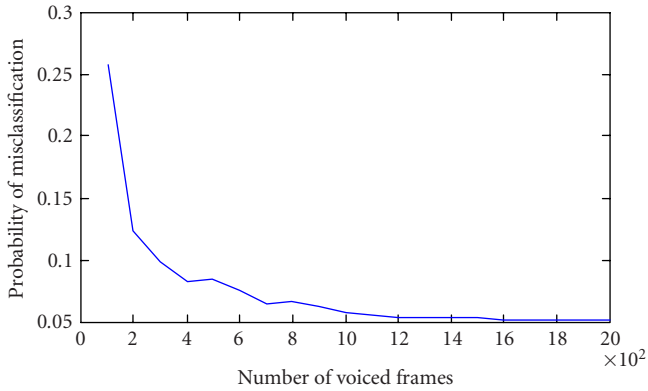


FIGURE 4: Probability of misclassification versus the number of voiced frames used for training.

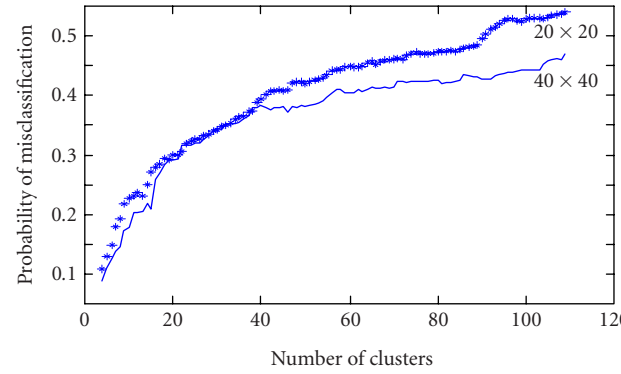


FIGURE 5: Probability of misclassification versus the number of clusters for 110 speakers.

would occur when there are one and M clusters, respectively, but classification errors follow exactly the reverse path. The optimum number is actually a trade-off between misclassification error and total computational load. At the optimal point, the decrease in misclassification error (further merging) does not compensate for the increase in computation caused by larger average cluster size.

Note that, at any level in the clustering hierarchy, when the number of clusters is N_0, the mean cluster size equals M/N_0 and the cluster size sample variance equals

$$\sigma_{N_0}^2 = \frac{\sum_{i=1}^{N_0} N_i^2}{M} - \frac{M}{N_0}. \tag{13}$$

Equation (12) can be represented in terms of sample variance as

$$\alpha + N_0\beta + \gamma\left(\sigma_{N_0}^2 + \frac{M}{N_0}\right). \tag{14}$$

Clearly for a given number of clusters, the computation complexity is a function of cluster size sample variance. If this variance is large (a case encountered fairly often), so the complexity is. Therefore, clustering algorithms that yield small cluster size sample variance achieve better complexity advantage. In general, the cluster size sample variance tends to be large, because some clusters are likely to deviate much from the mean in the algorithm. Therefore, computational complexity of the system can be further reduced only if the clustering algorithm is such that it limits the deviation of the cluster sizes.

4. EXPERIMENTAL RESULTS

The database used in the study is the IVIE corpus, publicly available www.phon.ox.ac.uk/~esther/ivyweb/download1.html. It has 110 speakers (55 male and 55 female) with 12 utterances from every speaker, each of 15- to 60-second duration.

The performance curves for the clustering algorithm are given below. Figure 4 gives the probability of misclassification versus the number of voiced frames used in training for 36 speakers. The probability of misclassification for N clusters equals $\sum_{i,j=1,i\neq j}^{N} \mathrm{P}^N(i,j)$. It shows asymptotic behavior that reaches a tractable minimum at around 1 000 frames. This suggests that sufficient statistic for clustering is grabbed in about 1 000 voiced speech frames.

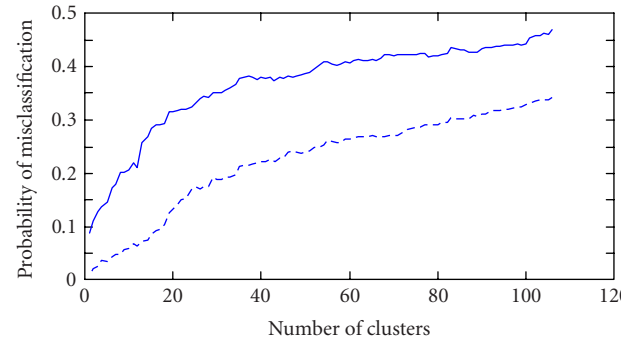


FIGURE 6: Probability of misclassification verses the number of clusters when an unknown utterance was mapped to the nearest cluster (solid) and two nearest clusters (dashed).

Figure 5 shows for 110 speakers the probability of misclassification versus the number of clusters for 20×20 and 40×40 pitch correlograms. As the number of clusters becomes larger—for example, when the data base size increases—the performance difference between the two diverges. This is because smaller pitch bands capture more local information that can differentiate nearby clusters.

The primary aim of introducing clusters in the speaker identification system is to decrease the computational complexity. However, since the overall SI system performance (probability of overall speaker identification error) needs to be kept above a threshold, the probability of misclassification should be small. One approach to this problem is to map an unknown speech utterance onto two nearest neighbors (see Figure 6). The improved performance, however, comes at the expense of running two-cluster GMMs for SI.

Figures 7, 8, and 9 give the risk versus the number of clusters (solid lines) as well as the relative computational

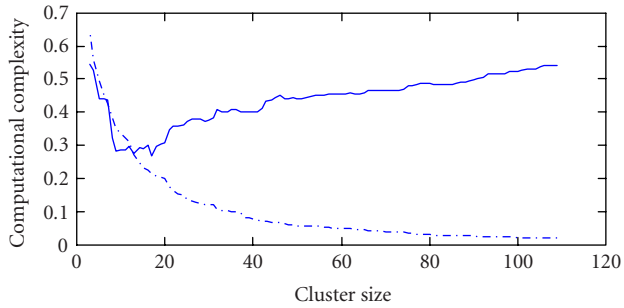


Figure 7: The solid line plots risk and the dashed line plots relative computational complexity versus cluster size of our system. Number of speakers = 110.

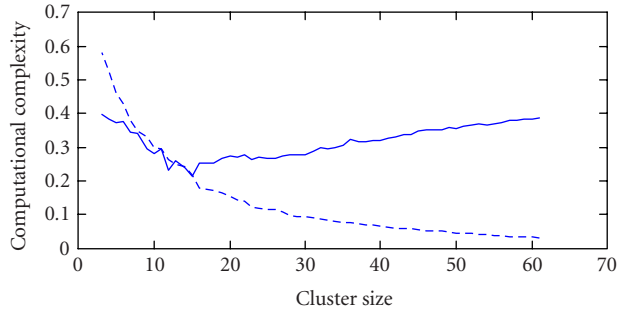


Figure 8: The solid line plots risk and the dashed line plots relative computational complexity versus cluster size of our system. Number of speakers = 62.

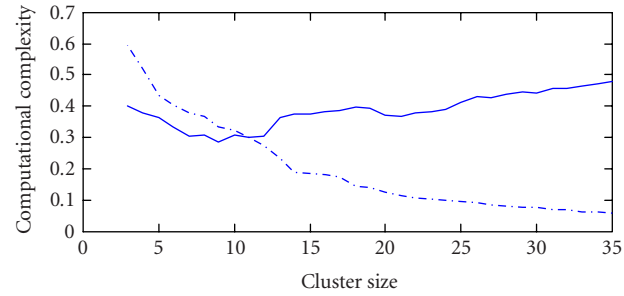


Figure 9: The solid line plots risk and the dashed line plots relative computational complexity versus cluster size of our system. Number of speakers = 36.

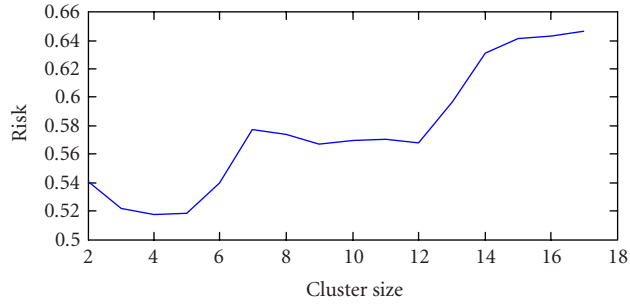


Figure 10: Risk versus cluster size when the speakers belonging to the largest cluster (cluster size is 18) were treated as an ensemble and clustered.

complexity of the two-stage identification system with respect to a single-stage one (dashed lines), when the numbers of speakers is 110, 62, and 36, respectively. The optimum number of clusters is 18, 15, and 9 for 110, 62, and 36 speakers, respectively and is clearly not a linear function of the ensemble size. This is because some new speakers get classed into present clusters themselves.

Clearly, the computation required varies inversely relative to the number of clusters. At the optimum point, the computational complexity of the proposed two-stage system is only 0.2 times the computational complexity of a single-stage system that uses the same GMM feature space (see Figure 7).

For a given number of clusters, the computational complexity of the two-stage system (see (12)) is minimum when all the clusters have the same population. However, since the cluster formation is data dependent, the cluster sizes are often different. As suggested earlier, the intracluster distance should be small. To test the algorithm's efficiency from this point of view, the largest cluster obtained was taken as new seed population and the clustering algorithm was reapplied afresh to this subensemble of 18 speakers. The process returned 4 clusters as optimal (Figure 10) showing the robustness of the algorithm. Therefore, speakers close in the correlogram space get clustered together (intracluster distance is small) in our process, which is a major desired behavior of any clustering process as discussed in Section 3.

Table 1 shows the misclassification error obtained at different SNRs at the optimal cluster level. Note that the misclassification error remains constant for a large SNR range, showing the robustness of the algorithm to noise. Again, it is well known that identifying females is tougher than identifying males. Table 2 shows that our algorithm is not biased in favor of either sex in terms of performance.

Table 1: Misclassification errors at the optimal point for different SNRs when the number of speakers is 110.

SNR	35 dB	20 dB	13 dB
Error	8.79%	8.91%	8.91%

Table 2: Misclassification errors for male and female speakers.

No. of speakers	Male error (%)	Female error (%)
110	9.7	7.0
62	7.1	8.2
36	5.0	1.5

To test the relative performance of the one-and two-stage SI systems, individual speaker GMMs with 16 mixtures with a feature space of 13th-order MFCC coefficients and spectral slope were trained and then run in a closed set framework. The results are listed in Table 3. The two-stage system returned the overall speaker identification error 10% lower than the single-stage one. It also used only about 23%

Table 3: Comparison of SI systems with and without clustering.

SI system	Average no. of GMMs to run per utterance	Overall identification error probability (%)	Computation time (CPU)
Single-stage system	110	34.09	2736 s
Two-stage system using pitch correlogram clustering	21.34	24.59	618 s

of the computation time. The performance improvement suggests the relative independence of the pitch correlogram and GMM feature space. The proposed two-stage SI system is efficient in both the performance and computation time dimensions.

The apparently high error rates (34% in single-stage and 24% in two-stage systems) are because of the small feature space dimensionality (13 MFCC and spectral slope) used. Normally, a minimum of 39 features (13 MFCC, 13 delta and 13 acceleration coefficients) is used. Still, errors tend to be high, for example, Reynolds reports around 26 percent error for telephone quality speech with this dimensionality [2]. The feature space used in this paper is the same as the one used by Murthy and Heck [10], who obtained an error of 42% on telephone quality speech with 64 GMMs on 100 speakers (we use the clean speech of the IVRS database and only 16 GMMs for our experiments). Our aim is to reduce computation time without performance compromise (and noise robustness) and this will certainly hold better on larger feature space dimensionality.

5. CONCLUSION

This paper suggests using a pitch-correlogram-based front-end database classifier to speed up text-independent SI systems based on Gaussian mixture models. We have shown that, in addition to a large computational advantage, better robustness to noise and distortion and a better overall performance are ensured by clustering. This is due to robustness as well as relative independence of the clustering statistic from the GMM space. The run-time advantage is particularly significant for surveillance and access control SI systems where live, or near-live, identification is desirable.

The pitch correlograms are computationally inexpensive and can be easily implemented on a real-time platform. Though the pitch correlogram has been used with GMMs in this paper, the framework allows these to be directly integrated with any SI system for computational gain and increased robustness. The effectiveness of the pitch-correlogram-based classifier has been demonstrated in a closed set SI system. It can also be used with similar effect in an open set SI solution.

The clustering algorithm yields the optimal number of clusters in Bayesian framework. Dynamic enrolment and distortion channel issues for clustering are under study and will be reported elsewhere.

ACKNOWLEDGMENT

The authors gratefully acknowledge the advice and the numerous suggestions provided by the reviewers, which have vastly improved the clarity and organization of this paper.

REFERENCES

[1] R. I. Damper and J. E. Higgins, "Improving speaker identification in noise by subband processing and decision fusion," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2167–2173, 2003.

[2] D. A. Reynolds, "Experimental evaluation of features for robust speaker identification," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 4, pp. 639–643, 1994.

[3] H. Gish and M. Schmidt, "Text-independent speaker identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 18–32, 1994.

[4] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[5] U. V. Chaudhari, J. Navratil, and S. H. Maes, "Multigrained modeling with pattern specific maximum likelihood transformations for text-independent speaker recognition," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 1, pp. 61–69, 2003.

[6] J. P. Campbell Jr., "Speaker recognition: a tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[7] A. E. Rosenberg, "Automatic speaker verification: A review," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 475–487, 1976.

[8] T. Matsui, T. Nishitani, and S. Firui, "Robust methods of updating model and a priori threshold in speaker verification," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '96)*, vol. 1, pp. 97–100, Atlanta, Ga, USA, May 1996.

[9] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," to appear in IEEE Trans. on Speech and Audio Processing.

[10] H. A. Murthy, F. Beaufays, L. P. Heck, and M. Weintraub, "Robust text-independent speaker identification over telephone channels," *IEEE Trans. Speech and Audio Processing*, vol. 7, no. 5, pp. 554–568, 1999.

[11] Z. Wang, Y. Liu, P. Ding, and X. Bo, "Covariance-tied clustering method in speaker identification," in *Proc. 4th IEEE International Conference on Multimodal Interfaces (ICMI '02)*, pp. 81–84, Pittsburgh, Pa, USA, October 2002.

[12] T. Kinnunen, T. Kilpeläinen, and P. Fränti, "Comparison of clustering algorithms in speaker identification," in *Proc. IASTED International Conference on Signal Processing and Communications (SPC '00)*, pp. 222–227, Marbella, Spain, September 2000.

[13] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[14] R. Kuhn, P. Nguyen, J-C. Junqua, et al., "Eigenvoices for speaker adaptation," in *Proc. International Conference on Spoken Language Processing (ICSLP '98)*, pp. 1771–1774, Sydney, Australia, November–December 1998.

[15] Y. Medan, E. Yair, and D. Chazan, "Super resolution pitch determination of speech signals," *IEEE Trans. Signal Processing*, vol. 39, no. 1, pp. 40–48, 1991.

[16] W. Hess, *Pitch Determination of Speech Signals*, Springer-Verlag, New York, NY, USA, 1983.

[17] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 399–418, 1976.

[18] B. S. Atal, "Automatic speaker recognition based on pitch contours," *Journal of the Acoustical Society of America*, vol. 52, no. 6, pp. 1687–1697, 1972.

[19] M. K. Sönmez, L. Heck, M. Weintraub, and E. Shriberg, "A lognormal tied mixture model of pitch for prosody-based speaker recognition," in *Proc. Eurospeech 1997*, vol. 3, pp. 1391–1394, Rhodes, Greece, September 1997.

[20] F. Weber, L. Manganaro, B. Peskin, and E. Shriberg, "Using prosodic and lexical information for speaker identification," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '02)*, vol. 1, pp. 141–144, Orlando, Fla, USA, May 2002.

[21] B. Wildermoth and K. K. Paliwal, "Use of voicing and pitch information for speaker recognition," in *Proc. 8th Australian International Conference Speech Science and Technology (SST '00)*, pp. 324–328, Canberra, Australia, December 2000.

[22] H. Ezzaidi, J. Rouat, and D. O'Shaughnessy, "Towards combining pitch and MFCC for speaker identification systems," in *Proc. Eurospeech 2001*, Aalborg, Denmark, September 2001.

[23] F. Jauquet, P. Verlinde, and C. Vloeberghs, "Histogram classifiers using vocal tract and pitch information for text-independent speaker identification," in *Proc. ProRISC 8th Annual Workshop on Circuits, Systems and Signal Processing*, Mierlo, the Netherlands, November 1997.

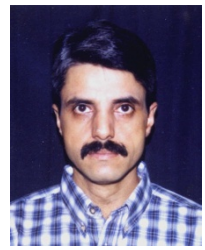
[24] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[25] A. V. McCree and T. P. Barnwell, "A mixed excitation LPC vocoder model for low bit rate speech coding," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 4, pp. 242–250, 1995.

[26] F. Dellaert, T. Polzin, and A. Waibel, "Recognizing emotion in speech," in *Proc. 4th International Conference on Spoken Language Processing (ICSLP '96)*, pp. 1970–1973, Philadelphia, Pa, USA, October 1996.

[27] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1988.

Nitin Jhanwar was born in 1977 in Bhilwara, India. He received his B.S. and M.S. degrees in electrical engineering from the Indian Institute of Technology Bombay in 2001. He was a Visiting Research Scholar at the University of Paris XI in 2001, where he worked in the computer vision area. He moved to the National University of Singapore in 2002 as a Research Assistant working on speaker verification problems. Since 2003, he has been with the R&D Division, Danlaw Technologies India, working in the speaker identification and applied signal processing areas. His interests are in applied signal processing and computer vision.



Ajay K. Raina obtained his Ph.D. degree in electrical engineering from the Indian Institute of Technology, Delhi, in 1983. He then joined the Faculty of Electrical Engineering, Indian Institute of Technology Kanpur, where he became a Professor of electrical engineering in 1997. Later in 2001, he joined Danlaw Technologies Limited to found the R&D Division of the company. His research interests are in the areas of discrete event processes and applied signal processing. He is a Member of the IEEE and a Fellow of the Institute of Electronics and Telecommunication Engineers (IETE). He is currently living in Australia.