[Dej95] Edmund X. Dejesus. Face value: Faster and more sophisticated algorithms are helping computerized facial-recognition systems come of age. *BYTE*, page State of the Art section, February 1995.

[DM84] W.R. Dilon and Goldstein M. *Multivariate Analysis, Methods and Applications*. John Wiley and Sons, 1984.

[Fur81] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-29, No. 2:254–272, April 1981.

[Fur86] Sadaoki Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-34, No. 1:52–59, February 1986.

[GG92] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Processing*. Kluwer Academic Publishers, 1992.

[GHM92] John J. Godfrey, Edward C. Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 517–520. IEEE, San Francisco, CA, 1992.

[LBG80] Yoseph Linde, Andres Buzo, and Robert M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28, No.1:84–95, 1980.

[McE94] Robert H. McEachern. Ratio detection precisely characterizes signals amplitude and frequency. *EDN*, pages 107–112, March, 3 1994.

[O'S86] Douglas O'Shaughnessy. Speaker recognition. *IEEE ASSP Magazine*, pages 4–17, October 1986.

[RS92] Aaron E. Rosenberg and Frank K. Soong. Recent research in automatic speaker recognition. In *Advances in Speech Signal Processing*, pages 701–738. Marcel Dekker, Inc., 270 Madison Avenue, New York, NY 10016, USA, 1992.

as speaker-discriminating features. Furthermore, we suggest to review the necessity of storing all the remnant pulses in the TrueSpeech encoding scheme and conjecture that additional gain in compression ratio may be achieved by reconsidering this issue.

B A refinement of the TrueSpeech 1 pitch detection technique.

As explained in subsection 4.3 using pitch as an additional feature presents an important improvement of speaker recognition performance. Any online pitch detection technique may be utilized for this purpose, for example the one found in the TrueSpeech 1 algorithm. However, the latter suffers from high noise level as well as from frequent occurrences of multiples of the actual pitch value. To remedy this situation we propose to enhance the abovementioned technique with two additional phases:

1. Pre-filter the pitch time function with energy VOX filter (in the manner we filter the input speech signal to detect and disregard any silent pauses, see Section 3), in other words leave only those pitch values computed for non-silent frames. This step will discard most of the noise present in the pitch sequence.

2. Apply a (relatively wide) median filter to the pitch time function. Due to the nature of the median filter this step will help to further remove noise as well as most of the multiples of the actual pitch.

The resultant sequence of pitch values is then suggested for use as an additional feature (i.e., an additional vector dimension) for VQ-based Speaker Recognition.

References

[CG86] De-Yuan Cheng and Allen Gersho. A fast codebook search algorithm for nearest-neighbour pattern matching. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 265–268. IEEE, Tokyo, Japan, 1986.

5. remnant excitation pulse sequence (7 pulses for each frame).

Having an efficient implementation of the TS1, we studied the above features for the purpose of speaker identification striving to make use of the existing efficient computation procedures. The following are conclusions from the study conducted:

1. Parcor and LPC coefficients are generally considered inferior to cepstral coefficients as the spectral envelope built from the latter is usually smoother ([O'S86], [RS92, Chapter 22]). On the other hand, the LPC coefficients may still be beneficial for cepstrum computation, as the following formula (found in [Fur81, p.255]) enables to compute the cepstral coefficients approximately twice as fast as from the Fourier transform:

   $$c_1 = a_1; \quad c_n = \sum_{k=1}^{n-1} (1 - \frac{k}{n}) a_k c_{n-k} + a_{n,} \ \ 1 < n \leq p \tag{10}$$

   where c_i and a_i are the ith-order cepstral coefficient and linear prediction coefficient respectively and p is the desired number of cepstral coefficients.

2. Rough pitch estimate and its second order refinement appear to be useful in speaker verification (see subsection 4.3). We therefore propose to use fine pitch estimate as an additional feature in vector quantization, after enhancing the TS1 pitch computation method as outlined in Appendix B.

3. The gain parameter by itself does not seem too useful for speaker recognition. Instead we propose to utilize another energy-related parameter, that is regression coefficients of the log-frame-energy(see also 4.3).

4. Finally, the remnant excitation pulse sequence did not appear to be useful at all for the purpose of speaker discrimination. We conducted a set of extensive subjective tests by randomly changing remnant pulses from their original values and playing the speech waveform restored from these new values to human listeners. As a result listeners reported very little impact on their ability to recognize original speakers, hence we concluded that those remnant sequences are not worth considering

- An interesting approach to pattern recognition by decomposing the original pattern by orthogonal basis was suggested in [Dej95]. Similar techniques may useful in comparing people's voices and we thus suggest to try such methods for speaker recognition.

- People's voices change over time and a robust speaker identification system must be able to cope with this apparent problem. To this end we propose selective retraining of the original codebooks using test samples. Thus, whenever a speaker is identified by the system with a high level of confidence (which is judged upon the actual distortion values between the reference codebook and the test sample) the codebook should be retrained using this test sample for additional training material. Such continuous retraining will enable the system to keep updated with voice changes over time.

- We also propose to repeat the experiments reported on larger test corpora. For example, the Switchboard speech corpus of Texas Instruments ([GHM92]) is specifically tailored for Speaker Recognition experiments and testing any ASR system upon this data is therefore highly recommended for proper assessment of the method accuracy rate.

A Using TrueSpeech 1 data encoding scheme for Speaker Recognition.

TrueSpeech 1 (TS1 in the sequel) is a highly effective speech compression algorithm designed at the DSP Group. In the process of speech compression it analyzes the input data and for each frame stores a number of features which permit quality restoration of the original signal thereafter. Among the features inherently computed by the TS1 are:

1. Parcor coefficients (used to calculate Linear Prediction coefficients),

2. rough pitch estimate,

3. fine pitch estimate (second order correction),

4. energy gain (characterizing the frame energy),

- As mentioned above the existing MATLAB implementation is relatively slow for real-time applications and rewriting the entire system in C (especially for further porting to DSP processors) is therefore highly recommended.

- Text-dependent approaches may be necessary for endowing the system designed with Speaker Verification capabilities. To this end, Dynamic Time Warping (DTW, see also [Fur81]) or considering more discriminating features (see also 3.1.3) may be introduced.

- A lot of operations performed during the recognition process require assigning a vector to a particular cluster in the codebook. For high-dimensional vectors this frequently performed operation may place a considerable time burden on the system. Various techniques are available for reducing this time overhead by appropriately preprocessing the codebooks. We propose utilizing the Voronoi diagrams [CG86] for speeding up the recognition process, especially in the testing phase when codebooks remain (almost) unchanged and a significant performance gain may be achieved.

- In order to improve the existing GLA implementation and thus to obtain more fine-grained codebooks, the following two additions are proposed:

  - The inherent problem of the GLA is its possible convergence to a local optimum instead of the global one. To remedy this problem random "jumps" may be introduced to exit the local optimum in the hope of converging near the still better local optimum or near the global one.

  - The concept of an empty cell may be reviewed by considering a cell empty if it contains less than N (for example, $N = 4$) vectors. In our opinion, this change will result in codebooks that better grasp the very peculiarities of individual speakers.

- We conjecture that studying the formant contours in voice spectrograms may be useful for discovering additional discriminating features.

utilized when the confidence level based upon the first sentence is not high enough). We attribute this efficiency to the choice of phonetically rich sentences found in the TIMIT corpus. Specifically we use a phonetically-compact sentences in the testing phase thus introducing a lot of phoneme material in a single sentence.

- In the light of the above time efficiency testing can be performed online in systems based on DSP processors (further searching performance increase may be obtained using Voronoi Diagrams for codebook preprocessing, as outlined in subsection 4.3).

4.2 The disadvantages of the approach

- The proposed Speaker Identification system now works in the so-called "closed-set" mode, when any incoming test sample is automatically conjectured to belong to one of the known speakers. Further adjustment and refinement are required to enable system operation in an "open-set" mode when a sample to be identified can possibly belong to no known speaker.

- Sometimes higher confidence levels are required than those normally achieved with mere speaker identification. When the latter is the case (for example, in military applications) further checks in addition to voice identification may be necessary to build a secure system.

4.3 Further research directions

At this point we see a number of possible extensions and improvements to the existing system, as their implementation is highly desirable to complete the integral powerful structure of the software created:

- Multiple studies in speech processing show that the pitch proper may account for as much as 60% accuracy in speaker verification. We hence suggest adding the pitch value to the set of speaker-discriminating features (thus increasing the dimensionality of feature vectors by one). One technique for pitch detection is outlined in Appendix B, another interesting approach to this may be found in [McE94, see also US Patent #5,214,708 of 5/25/93 by Robert H. McEachern].

3.2 Testing mode.

The testing procedure is rather straightforward. Given a speech sample of a user to be identified, we compute a sequence of feature vectors exactly the way we do this in the training phase (see also subsection 3.1). We then assess the distortion between this set of vectors and the codebooks of all the known speakers. The user will be identified with the speaker whose codebook yields the minimal distortion with the given vectors. One testing sentence usually suffices, but another one may be employed if the confidence level based on the first sentence is not high enough.

Putting this verbal description in a more formal way, the user whose feature vectors are $\{v_1, v_2, ..., v_n\}$ is identified with the known speaker i such that

$$i = \arg \min_{1 \leq j \leq Z} d(\{v_1, v_2, ..., v_n\}, CODEBOOK_j) \tag{9}$$

where $CODEBOOK_j$ is the codebook of the $j-$th known speaker and Z is the total number of such speakers.

4 Conclusions

4.1 The advantages of the approach

- The system designed is highly reliable and can provide high security levels even to most demanding applications (over 95% accuracy rate was achieved in the experiments).

- Additional security level is achieved by the system being text-independent, as random prompts may be given to users eliminating the possibility of previous recordings.

- The system is also user-friendly, as voice identification is usually non-intrusive compared to its fingerprint and retinal check counterparts.

- The system is efficient in computational load as it requires very little training and testing data for highly precise operation. In our implementation, 8 sentences (approximately 1.5 second long each) were used for training the codebooks, and 1 or 2 sentences were used for testing (one sentence usually suffices for testing, but another one may be covariance matrix of the pooled intraspeaker data we discovered that it is mainly diagonal in its nature, and the elements off the main diagonal are negligible. With this observation we employ the following formula for computing distortion between two vectors:

$$d(x,y) = \sum_{i=1}^{10} w_i(x_i - y_i)^2 \tag{6}$$

where w_i is the reciprocal of the averaged intraspeaker variance of the $i-$th cepstral coefficient (i.e., the $i-$th coordinate of feature vectors).

3.1.3 Increasing the number of features involved.

In an attempt to further increase the speaker identification accuracy rate, we introduced (following [Fur86]) two additional sets of features:

1. The so-called $\Delta - cepstrum$ or cepstrum regression coefficients. There are 10 such coefficients for each (overlapping) data frame[7] and the following formula is used for their computation:

$$\Delta c_i(t) = \frac{\sum_{j=-n_0}^{n_0} c_i(t+j) * j}{\sum_{j=-n_0}^{n_0} j^2} \tag{7}$$

   where $c_i(t)$ are the original cepstral coefficients, $\Delta c_i(t)$ are the cepstrum regression coefficients, n_0 - the number of frames to be used forward and backward in regression computations (we use $n_0 = 2$).

2. The energy regression coefficients are computed similarly, while the is one such coefficient for each (overlapping) data frame:

$$\Delta e(t) = \frac{\sum_{j=-n_0}^{n_0} \log_{10}[FrameEnergy(t+j)] * j}{\sum_{j=-n_0}^{n_0} j^2} \tag{8}$$

After incorporating those features we conducted a number of experiments with the original TIMIT speakers, but no apparent increase in the performance accuracy was noted. Nevertheless we are going to proceed with experiments utilizing those features, as we conjecture they posses a considerable speaker discrimination capability.

[7] One $\Delta - cepstrum$ coefficient correspond to each dimension of the original vector.

a codebook vector nearest to it (a tie-breaking rule is obviously necessary here to accommodate the case when a training vector is equidistant from a number of codebook vectors - our approach assigns it to the code-vector with the smallest index).

2. Centroids are computed for the newly created clusters by averaging their vectors, and the new set of centroids will form a new codebook approximation. In case an empty cell was created during phase 1 (i.e., one or more of the previous codebook vectors was assigned no training vectors) we split a cell with the highest average distortion thus far in order to maintain the codebook size constant (this process is repeated until no more empty cells remain and average distortion of cells is reassessed after each split).

The first codebook approximation to initiate the GLA iterations is computed with the Pairwise Nearest Neighbor approach. This method starts with associating a separate cluster with each of the N training vectors, and then iteratively reduces the number of clusters to the desired target codebook size. In each iteration two clusters are merged so that the increase in the overall distortion is minimal, until only the M clusters remain; the centroids of the remaining clusters will form the initial codebook guess.

As mentioned above, the GLA process always converges to a state when no further improvement is possible, but this may take a huge number of iterations. To limit this process to a reasonable time (while staying within reasonable performance accuracy) we analyze the relative change in average distortion after each iteration. By means of experiments we arrived to the following inequality, which currently serves a termination condition for the iterative process:

$$\frac{D_i - D_{i+1}}{D_i} \leq 0.001$$

that is iterations will be performed as long as the relative change in overall average distortion is larger than 0.001.

3.1.2 Mahalanobis distortion computation.

All the distortion computations throughout the system are performed using the Mahalanobis distortion measure (see also section **??**). Studying the

3. A threshold energy value is calculated as[5]

$$EnergyThreshold = \max_{all\_frames}(dB\_Frame\_Energy) - 30 \qquad (5)$$

4. All the data frames with energy less than $EnergyThreshold$ are discarded from further consideration

After filtering the source data as above we compute the cepstrum feature vectors for the remaining data[6]. All the vectors from the eight training files are then aggregated into a single training set. The first $2 * M = 128$ vectors from the training set are used to create an initial codebook approximation $CODEBOOK_0$, and the GLA is then applied to this codebook *and* the *entire* training set to obtain a (locally optimal) codebook for the given speaker.

3.1.1 The Generalized Lloyd Algorithm (GLA).

The GLA is considered one of the best approximation algorithms for building VQ codebooks. Given an initial codebook approximation and a training set of vectors, it works iteratively until obtaining a locally optimal codebook for the given data. It can be formally proven ([GG92, Lemma 11.3.2, p.368]) that for a finite training set the GLA will always converge in a finite (although possibly large) number of iterations.

Each Lloyd iteration is logically divided into two phases:

1. First, the training vectors are assigned to clusters around the previous codebook vectors (the clusters formed induce a partition on the feature space S). Each training vector is assigned to the cluster associated with

[5] This formula is an empirical one. The value of constant (30) was chosen in a series of subjective experiments, when VOX filtering was performed with various such constants and all-zero frames were substituted for those with energy below the threshold. Human listeners then reported that for the constant values of 30 or more the quality of the speech changed as above was almost unaffected. On the other hand, using the value 20 resulted in unintelligible speech for most input files.

[6] In order to exhaustively characterize the existing training data we use frame overlapping for feature vector computation. That is, after a vector is computed we move the frame pointer by half of the standard frame length (that is, by 120 samples) before the next vector computation takes place.

3 Vector Quantization for Automatic Speaker Identification.

We now proceed with detailed description of our system design. The system mainly works in two modes: the *training* mode, when speaker-dependent codebooks are computed from raw speech data, and the *testing* mode, when a given test sample is identified with the nearest speaker codebook.

3.1 Training mode.

For each known speaker we compute a codebook of size $M = 64$ code vectors. Each vector in our model holds the first ten cepstrum coefficients (the $0-$th coefficient is excluded as it contains an energy-related gain value). One such vector will be computed for each non-silent data frame containing 240 samples[3].

Of the ten sentences[4] available for each speaker in the TIMIT corpus we use eight sentences for training (in the TIMIT classification - if we denote the ten sentences for each speaker as $S?1, ..., S?0$ - those are *dialect* sentences SA1 and SA2, *diverse* sentences SI3 and SI4 and *compact* sentences SX6, SX7, SX8 and SX9; one *diverse* sentence SI5 and one *compact* sentences SX0 are reserved for testing). After reading the eight files corresponding to the sentences chosen, we perform decimation to reduce the sampling frequency from $16kHz$ to $8kHz$ (the latter is sufficient for our purposes) and subtract the mean value of the original signal to get rid of any DC component present in the raw data.

We then filter the original signal with an energy VOX filter to discard silent pauses that carry no speaker dependent information. The following steps are performed during the filtering:

1. The energy value is computed for each frame of input data.

2. All energy values thus computed are converted to the dB scale according to the formula

$$dB\_Frame\_Energy = 10 * \log_{10}(Frame\_Energy) \qquad (4)$$

[3] The number 240 was chosen for compatibility purposes with the existing TrueSpeech 1 embodiment (see also Appendix A).

[4] Each TIMIT sentence is approximately 1.5 second long.

we would like to partition the latter into a set M of pairwise disjoint cells $\{S_1, S_2, ..., S_{|M|}\}$ (i.e., for any $i \neq j$, $1 \leq i, j \leq |M|$ holds $S_i \cap S_j = \emptyset$, and also $S_1 \cup S_2 \cup ... \cup S_{|M|} = S$). Each cell from among $S_1, S_2, ..., S_{|M|}$ will then be assigned a code vector $c_1, c_2, ..., c_{|M|}$ respectively, while each c_i is chosen to be a centroid of all the training vectors in the cell S_i. An optimal such set $\{c_1, c_2, ..., c_{|M|}\}$ will then comprise the desired codebook.

To define the notion of optimality of a codebook we first define a *distortion measure* $d(\circ\ ,\circ\ )$ between two data vectors, which may be computed as a (possibly weighted) distance between the vectors in the space S. The average distortion of a codebook given a training set T is given by the following formula:

$$D = \frac{1}{|T|} \sum_{i=1}^{|T|} \min_{1 \leq j \leq |M|} d(t_i, c_j) \tag{1}$$

A codebook for which the above value of D is minimal will be considered optimal[1]. Finding an overall optimum codebook is apparently a computationally intractable problem, as the number of possible space partitions grows exponentially in the size of the training set. We therefore have to employ a heuristic algorithm for finding some sub-optimal codebook. A number of algorithms are available for this purpose, among them the $k-means$ statistical analysis algorithm [DM84] and the iterative Generalized Lloyd Algorithm [LBG80]; in scheme describe we make use of the latter one.

A note on distortion computation is in order here. Two distance measures are usually used for distortion computation:

1. The Euclidean distance assumes orthogonality of the basis and defines

$$d(x, y) = (x - y)^T \times (x - y) \tag{2}$$

2. The general Mahalanobis distance permits different weighting for various features (dimensions) of reference vectors:

$$d(x, y) = (x - y)^T \times W^{-1} \times (x - y) \tag{3}$$

   where W is the autocovariance matrix averaged over a representative set of reference vectors[2].

[1] Note that a number of equally-optimal codebooks is possible.

[2] Actuall, the Euclidean distance may be viewed as a particular case of the general Mahalanobis distance if we choose $W = I$ (the identity matrix).

an impostor is extremely unlikely to have them previously recorded.

The system described has been actually implemented in MATLAB. Although inferior in its performance to compiler languages (e.g., C), MATLAB enables easy manipulation of large quantities of data with many useful scientific routines supplied with or incorporated into the system. MATLAB's graphic capabilities have also been indispensable for visual representation and analysis of the speech data. Having completed the system design and with the MATLAB implementation in hand, we now suggest to rewrite it in C for PC or DSP processor in order to boost the performance.

The abovementioned ideas are presented in detail in what follows. The ideas of vector quantization and distortion computation are briefly reviewed in Section 2. Section 3 describes our vector quantizer for speaker identification. Finally, Section 4 contains conclusions and outlines further research directions. Furthermore, *Appendix A* describes the study of the DSP Group *TrueSpeech 1* algorithm with respect to the Speaker Recognition problem. We also present a short memo on pitch detection (which refines the pitch computation technique implemented in the TrueSpeech 1 algorithm) in *Appendix B.*

2 Vector Quantization and Distortion Computation.

In this section we briefly review the notions of Vector Quantization and Distortion Computation (see [GG92] for more comprehensive treatment of the issue).

Generally speaking, vector quantization is representing an infinite variety of possible data vectors with a relatively small set of the most frequent or characteristic ones. This set is the referred to as a *codebook* while its constituents are called *code vectors.* A representative set of actual data used to compute the codebooks is called a *training set.* In our approach code vectors do not necessarily have to appear in the training set; instead they are supposed to grasp the training vectors distribution patterns in the best way possible.

Let us have a more close look at the process of codebook computation. Given a set T of $n-$dimensional training vectors $\{t_1, t_2, ..., t_{|T|}\}$ in a space S,

1 Introduction

With the number of multi-user applications and facilities growing constantly nowadays, the problem of identifying users and granting system access only to those properly authorized has become extremely important. A number of approaches are currently being used separately or combined to address this problem, while the more security is required the more sophisticated and computationally expensive techniques are used. Among the most popular techniques today are login/password checks, studying of fingerprints and retinal blood vessel patterns, and voice identification. The first approach is not considered secure enough for numerous applications as sophisticated electronic eavesdropping permits intruders to steal passwords and then use them illegally. The major drawback of fingerprints or retinal checks is their intrusiveness, as an individual to be identified by those methods has to be willing to undergo the tests and not get offended by the procedures. On the other hand, voice identification enables non-intrusive monitoring while achieving very high accuracy rate which complies with most security requirements.

The proposed scheme utilizes the Vector Quantization approach along the guidelines of [GG92]. Given a speech file, we first filter it to discard silent pauses using a special energy VOX filter. We then compute a set of feature vectors to be used in building speaker-dependent codebooks. In our scheme each feature vector contains ten first cepstral coefficients for each input data frame (i.e., it is a ten-dimensional vector). Using a Generalized Lloyd Algorithm (GLA in the sequel; see [GG92] and [LBG80]) we build a set of codebooks which characterize each speaker (this is a so-called "training" phase when codebooks are computed given voice samples uttered by respective codebook "owners"). The training process usually involves a considerable amount of data and is performed off-line (this is usually not a problem as training is only performed once for each new speaker). In a "testing" phase the system is given a speech sample and has to identify a speaker. This is done by assessing the average distortion between the given sample and all the reference codebooks in the database. The user requesting system access is thus identified with the person whose codebook yields the minimum average distortion versus the given sample. The access rights are then granted within the permissions specified for this particular user. An important additional feature of our approach is its text-independence which enables to request a user to utter random prompts and be reasonably sure

Speaker Recognition:
Using a Vector Quantization Approach for Robust Text-Independent Speaker Identification.

Technical Report DSPG-95-9-001

Evgeniy Gabrilovich* and Alberto D. Berstein
DSP Group, Inc.
3120 Scott Blvd., Santa Clara, California 95054, USA
gabr@cs.technion.ac.il aberstein@dspg.com

September 28, 1995

Abstract

In this paper we propose a particular use of the Vector Quantization technique for Speaker Identification. A representative set of codebooks is created for each speaker and is then used as characteristical reference to discriminate among speakers. The experiments conducted with 20 speakers (10 female and 10 male) from the TIMIT speech corpus yielded success ratio of over 95% with training/testing sets of very small size. The entire system is text-independent thus enabling higher security in host applications (preventing an impostor from recording a legal user and then playing his/her voice back to the speaker identification system).

*First author's main affiliation is with the Department of Computer Science, Technion - Israel Institute of Technology, Technion City, 32000 Haifa, Israel.