advantageous for the pattern to be first squared before applying DFT analysis. This minimises the spatial frequency components resulting from intermodulation products caused by the nonlinear square root operation associated with the detection and measurement of the resultant signal envelope. The procedure also uses a windowing procedure in the spatial domain, designed especially to minimise spectral leakage and optimise accuracy. To improve the accuracy of classical windowing techniques, a procedure of variable spatial domain window length is used, so that for each spatial frequency, a truncation window of an integer multiple of the period can be used. (See [2, 3] for windowing techniques.)

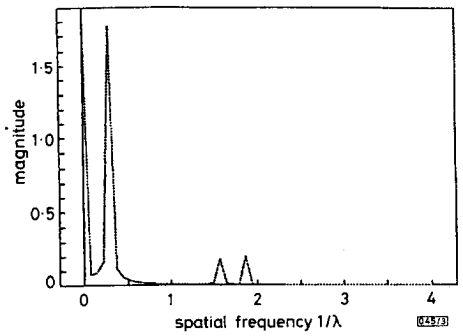

Fig. 3 *DFT of squared envelope*

Experimental verification: To verify the procedure in practice, field experiments were undertaken. To achieve controlled conditions, experiments were carried out using an open range. The different paths for the multipath were provided by reflector boards. Experiments with one and two boards were carried out, yielding two and three component multipath fields. The principal axis of the antenna was aligned with one of the paths. The antenna was displaced a fraction of a wavelength at a time over the range 300 – 800mm (11 to 30 wavelengths at a frequency of 11.2 GHz). Using reflector boards also gave the advantage of being able to measure the amplitudes of the multipath components individually by dismantling or erecting the boards accordingly.

Table 1: Results of resolution procedure compared with measurements

Example		From measurements			From DFT output		
		Component number			Component number		
		1	2	3	1	2	3
1	Amplitude [dBm]	−10.9	−14.1	–	−10.3	−15.1	–
	(2 components) Angle θ	0°	45°	–	0°	43°	–
2	Amplitude [dBm]	−3.6	−32.0	−16.3	−4.4	−27.4	−16.4
	(3 components) Angle θ	0°	22°	45°	0°	27°	44°
3	Amplitude [dBm]	−8.9	−26.3	−20.9	−9.0	−26.6	−21.0
	(3 components) Angle θ	0°	22°	43°	0°	23°	42°

Table 1 lists the results obtained from measurements of individual contributions and those obtained using the described procedure. All results represent voltage levels at the output of the low noise amplifier, connected to the antenna.

Conclusion: Results in Table 1 clearly show that the resolution procedure outlined above can yield reasonably accurate values for the resolved components. Table 1 shows the very good accuracy in determining the angle of arrival of various signal paths. A lesser degree of accuracy was achieved in the values of the component amplitudes, but it is possible that this error was exaggerated because of the difficulty in measuring individual components and errors associated with the radiation pattern of the receiving antenna and its angular orientation.

Based on these results and on others obtained from actual buildings, we can conclude that the above procedure is highly effective in identifying and resolving signal paths giving rise to multipath effects. For radio system planning, the accuracy of the procedure is considered satisfactory.

Electronics Letters Online No: 19951033 *10 July 1995*

J. Richter and M.O. Al-Nuaimi (*University of Glamorgan, Department of Electronics and Information Technology, Pontypridd CF37 1DL, United Kingdom*)

References

1 Commission of European Communities, COST 210 Report: 'Influence of the atmosphere on interference between radio communications system at frequencies above 1GHz, Brussels, Belgium, 1991

2 HARRIS, F.J.: 'On the use of windows for harmonic analysis with the discrete Fourier transform', *IEEE Proc.*, 1978, **66**, (1), pp. 51–83

3 NUTTALL, A.H.: 'Some windows with very good sidelobe behaviour', *IEEE Trans.*, 1981, **ASSP-29**, (1), pp. 84–89

Short-timed speech dynamics for speaker recognition

H. Li, J.-P. Haton and J. Su

Indexing terms: Speaker recognition, Speech processing

A temporal transition model of speech is proposed for speaker recognition and verification. The issues of model building, distance measure and implementation are addressed. A set of experiments are conducted, which give a 98.9% recognition rate and 99.5% verification rate. Short-timed dynamics of utterance well encodes the speaker specificity.

Introduction: The speaker dependent statistical model, the Gaussian speaker model trained with the maximum likelihood estimation (MLE), has been proposed in some earlier studies [1]. In this Letter, the temporal transition model (TTM) which encodes the short-timed speech dynamics is constructed based on the speaker model.

A speaker model is represented by a Gaussian mixture of J probability density function (PDF) components $\{f(x/\omega_j), P(\omega_j)\}_{j=1}^{J}$ and $f(x/\omega_j) = N(x; m_j, \Sigma_j)$. Let $X = \{x_1, ..., x_t, ..., x_T\}$ be a set of data from speaker s. The probability of drawing the sample data x_t from the mixture Ω_s can be written as

$$f(x_t/\Omega_s) = \sum_{j=1}^{J} f(x_t/\omega_j)P(\omega_j) \qquad (1)$$

where $P(\omega_j)$ is called the mixture coefficient, subject to $\Sigma_{j=1}^{J} P(\omega_j) = 1$. We obtain the *a posteriori* probability as

$$f(\omega_j/x_t) = \frac{f(x_t/\omega_j)P(\omega_j)}{\sum_{j'=1}^{J} f(x_t/\omega_{j'})P(\omega_{j'})} \qquad (2)$$

Given a set of training data, it is possible to unsupervisedly cluster all sample data from a speaker into a certain number of states, referred to as mixture component ω_j in eqn. 1. The expectation-maximisation (EM) algorithm is one of the most popular algorithms for MLE. When we consider a speech utterance as a sequence of articulatory evolution, a state could be associated with an articulatory configuration. The observation probability of mixture density could be interpreted as the occurrence probability of a certain articulatory configuration in an evolution. The speaker Gaussian mixture model here serves as the underlying definition of state for TTMs.

Temporal transition model: The TTM is motivated by two ideas. One is that most pattern classification approaches are devoted to static patterns, or patterns with a fixed size of components. The problem will become simpler if speech dynamics could be translated into a static model. Another idea is that short-timed speech dynamics carry speaker-specific information [2].

Considering a time slot of Q vectors $\mathbf{x}_t = \{x_t, ..., x_{t+Q-1}\}$, a Q-dimensional space could be spanned by the *a posteriori* state transition probabilities of $p(\omega_{q1}, ..., \omega_{qQ}/\mathbf{x}_t)$. For each $\mathbf{x}_t$, we have a Q dimensional array with J elements in each dimension, which is

called a state transition probability array and is considered to relate to the articulatory characteristics of the speakers. For simplicity, let $p_{it} = p(\omega_i/x_t)$. Thus, a set of probabilities $\{p_{1t}, ..., p_{Jt}\}$ could be obtained for each speech observation $\mathbf{x}_t$. Supposing that the occurrences of successive speech vectors are independent, we have

$$P((\omega_{d_1}, ..., \omega_{d_Q}/\mathbf{x}_t) = p_{d_1 t} \times \cdots \times p_{d_Q t+Q-1} \quad (3)$$

Furthermore

$$\sum_{d_1=1}^{J} \cdots \sum_{d_Q=1}^{J} p_{d_1 t} \times \cdots \times p_{d_Q t+Q-1} = 1 \quad (4)$$

Taking the average of all transition probability arrays over the speech session X along all Q dimensions by

$$\theta_{d_1 \ldots d_Q} = \frac{\sum_{t=1}^{T-Q+1} p_{d_1 t} \times \cdots \times p_{d_Q t+Q-1}}{T-Q+1} \quad (5)$$

a TTM with J^Q elements is obtained. There are $(T - Q + 1)$ transition probability arrays for T successive feature vectors because of the endpoint effect. We can easily verify that

$$\sum_{d_1=1}^{J} \cdots \sum_{d_Q=1}^{J} \theta_{d_1 \ldots d_Q} = 1 \quad (6)$$

Once the TTMs for given speech sessions have been built, the distance measures are required for evaluating the differences between TTMs. To deal with two probability distribution models, one possibility is to use the information divergence (ID), which is usually used to measure the difference between distributions [3]. Given two TTMs $A = \{a_{d1 \ldots dQ}\}$ and $B = \{b_{d1 \ldots dQ}\}$, we have

$$D(A,B) = \sum_{d_1=1}^{J} \cdots \sum_{d_Q=1}^{J} a_{d_1 \ldots d_Q} \log \left(\frac{a_{d_1 \ldots d_Q}}{b_{d_1 \ldots d_Q}} \right) \quad (7)$$

It is known that $D(A, B) \geq 0$ but that it is asymmetric, i.e. $D(A, B) \neq D(B, A)$ [3]. An alternative is to combine two of them as $D'(A, B) = [D(A, B) + D(B, A)]/2$, which gives a symmetric one with more computational load. We can still obtain consistent results with $D(A, B)$ instead of $D'(A, B)$ if the same order of A and B, i.e. the order of reference patterns and test patterns in the distance measure, is kept.

For simplicity, we next transform the Q-dimensional array of TTM into a one-dimensional vector by letting $\theta_i = \theta_{d1 \ldots dQ}$ where $i = \{\{\{d_1 \times J + d_2\} \times J\} + \cdots + d_Q\}$ in the cardinal set $[1, I]$ and $I = J^Q$. In the case of multiple reference patterns, we are usually asked to find the centroid in a pattern cluster. A centroid Θ^c in a cluster of K templates could be obtained by minimising the objective function

$$f(\theta^c) = \sum_{k=1}^{K} \sum_{i=1}^{I} \theta_i^c \log \frac{\theta_i^c}{\theta_i^k} \quad (8)$$

for $D(A, B)$, and similarly

$$f(\theta^c) = \sum_{k=1}^{K} \sum_{i=1}^{I} \theta_i^k \log \frac{\theta_i^k}{\theta_i^c} \quad (9)$$

for $D(B, A)$, subject to the constraint of eqn. 6. The centroids for the two measures under discussion could be derived as eqn. 10 for $D(A, B)$ and eqn. 11 for $D(B, A)$ [3]. Unfortunately, it is not straightforward to give a centroid with $D'(A, B)$.

$$\theta_i^c = \frac{\sqrt[K]{\prod_k \theta_i^k}}{\sum_{i'=1}^{I} \sqrt[K]{\prod_k \theta_{i'}^k}} \quad (10)$$

$$\theta_i^c = (1/K) \sum_k \theta_i^k \quad (11)$$

Experiments: A database from 72 French speakers, with 36 male and 36 female speakers, was used in the experiments. It consisted of three-word French phrases naturally uttered five times by each speaker. Using one of the five sessions as the test data and others as the training data rotating the orders resulted in five assessment sets. The results were reported as the average of the five sets. A 12th mel-scale cepstral analysis was performed after each 10ms time interval with a 32ms Hamming window over each session. All silences and stops in the sessions were eliminated, since they are not discriminative. To build the mixture model for a speaker, eight-mixture Gaussian PDFs were trained ($J = 8$) with the EM algorithm. TTMs with different Q settings were built for each session.

Table 1: Recognition accuracy with $D(A, B)/D(B, A)$

	$Q = 1$	$Q = 2$	$Q = 3$
	%	%	%
NC	84.8/85.3	98.9/97.2	98.7/97.5
C	85.9/85.5	98.5/97.1	98.5/97.7
	$J = 32$	$J = 64$	$J = 128$
	%	%	%
$C/Q = 2$	70.6/77.4	96.5/97.4	98.2/97.2

Speaker recognition: Three experiments were carried out in the recognition test. In Table 1, *C* is referred to as the case where TTM centroids for $D(A, B)$ or $D(B, A)$ are used. *NC* means no centroid is used and all four TTM references are evaluated in each recognition trial.

The Gaussian PDF computation is time consuming. We have proposed a tied speaker model and its MMI learning algorithm [1] for tying the mixtures across speakers into a sharing Gaussian kernel, where the speakers are only discriminated by the mixture coefficients so that the computation of PDFs can be significantly reduced. Tying PDFs will reduce the discriminability of the system to some extent. A compensation can be made by increasing the number of sharing components. To show the impact of tying mixture on the performances, the experimental results with respect to different mixture number J are summarised in Table 1.

Speaker verification: It is known that a normalised distance outperforms an unnormalised distance by introducing a group of speakers as the background speakers for each speaker [2]. We used a new cohort selecting approach to select 24 cohort speakers as presented in [2]. The cohort speakers were excluded as imposters to avoid unexpected optimistical biasing of the results. The verification accuracy was defined as the average of the 'true-accept' (true claim goes with acceptance) and 'false-reject' (false claim goes with rejection) rates, as shown in Table 2. Only $D(A, B)$ distance was evaluated in verification experiments, since it was shown to be more effective than $D(B, A)$.

Table 2: Speaker verification accuracy

	$Q = 1$	$Q = 2$	$Q = 3$
	%	%	%
NC	90.8	99.4	99.5
C	92.8	98.9	98.9
	$J = 32$	$J = 64$	$J = 128$
	%	%	%
$C/Q = 2$	83.6	98.5	99.3

Conclusions: The two motivations of TTM are implemented in this Letter. We have successfully transformed the short-timed speech dynamics into a static space. The model enables us to apply some other pattern recognition techniques to speaker recognition, for instance, vector quantisation and linear discriminant analysis. The effectiveness of TTM has also been confirmed.

According to the modelling procedure, a larger Q implies an average transition probability over a longer time slot. For cases $Q = 2$ and 3, the two cases perform almost the same and outperform the case $Q = 1$, where no speech dynamics is taken into account. Other than the recognition/verification accuracy, one critical factor that we should consider is the computational complexity. When Q increases, the components in the model will augment exponentially; not only excessive storage is required but also more computation. Thus, Q should be carefully selected in practice.

In conclusion, the short-timed speech dynamics well-encodes the speaker specificity. The TTM with $Q = 2$ is a reasonable model

setting for the database. By using the tied speaker model, the computation is significantly reduced, whereas the performances remain almost the same as the speaker model.

© IEE 1995 *14 June 1995*
Electronics Letters Online No: 19950962

H. Li and J.-P. Haton (*CRIN-CNRS/INRIA-Lorraine, Campus Scientifique, F-54506 Vandoeuvre-lès-Nancy, France*)

J. Su (*Institute of Radio Engineering and Automation, South China University of Technology, Peoples' Republic of China*)

References

1 LI, H., HATON, J.-P., and GONG, Y.: 'On MMI learning of Gaussian mixture for speaker models'. Proc. European Conf. on Speech Technology, Madrid, Spain, 1995

2 NG, K.T., LI, H., HATON, J.-P., and SU, J.: 'Nonparametric distance measures of speaker verification', *Electron. Lett.*, 1995, **31**, (9), pp. 700–701

3 LI, H., HATON, J.-P., SU, J., and GONG, Y.: 'Speaker recognition with temperal transition models'. Technical Report of CRIN, Vandoeuvre les Nancy, France, 1995)

VQ codebook design using genetic algorithms

J.S. Pan, F.R. McInnes and M.A. Jack

Indexing terms: Genetic algorithms, Vector quantisation

A codebook design approach for vector quantisation using genetic algorithms is proposed. This novel approach provides superior performance compared with the generalised Lloyd algorithm (GLA).

Introduction: Vector quantisation (VQ) [1] is a very efficient approach to data compression. The encoder of VQ encodes a given set of k-dimensional data vectors $X=\{X_j|X_j \in R^k; j = ,...,T\}$ with a much smaller set of codewords $C=\{C_i|C_i \in R^k; i = 1,...,N\}(N << T)$. Only the index i is sent to the decoder. The decoder has the same codebook as the encoder, and decoding is operated by a table look-up procedure. The performance of data compression depends on a good codebook of representative vectors.

Lloyd [2] showed two conditions are necessary but not sufficient for the existence of an optimal minimum mean squared error (MSE) quantiser:

(i) The codewords should be the centroids of the partitions of the vector space.

(ii) The centroid is the nearest neighbour (NN) for the data vectors in the partitioned set.

These conditions have been applied to codebook design by Linde *et al.*, and called the generalised Lloyd algorithm (GLA) [3]. Since these conditions are necessary but not sufficient, there is no guarantee that the resulting codebook is optimal. The generalised Lloyd algorithin is widely used in codebook generation for vector quantisation. It is a descent algorithm in the sense that at each iteration, the average distortion is reduced. For this reason, GLA tends to get trapped in local minima. The performance of the GLA depends on the number of minima and choice of the initial conditions.

Genetic algorithms refer to a model introduced and investigated by Holland [4] and by students of Holland. A genetic algorithm is any population-based model that uses selection and recombination operators to generate new sample points in a search space. Genetic algorithms [5] have been proven to be powerful methods in search, optimization and machine learning. They encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures to achieve optimisation. To our knowledge, no one has applied genetic algorithms to optimisation of codebooks. We describe the GA-GLA algorithm derived by applying genetic algorithms to codebook design to produce globally optimum VQ codebook vectors. The four main steps involved in genetic algorithms are evaluation, selection, crossover and mutation. In this Letter, only evaluation, selection and crossover are adopted in combination with GLA to produce a superior GA-GLA codebook design algorithm.

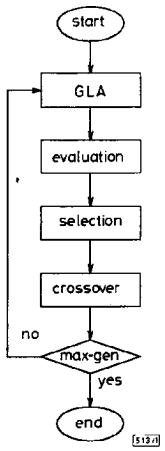

Fig. 1 *Flowchart of GA-GLA algorithm*

GA-GLA algorithm: The fitness of genetic algorithms can be represented by the mean squared error (MSE). In the VQ operation, the chromosome is designated as the centroid of the cluster. The individual of the population is the codebook. As shown in Fig. 1, the proposed algorithm consists of the following steps:

(i) *Initialisation:* Calculate the central chromosome (centroid) G_0 from the training vectors X_i (i=1,2,...,T). Select N chromosomes G_j (j=1,2,...,N) for every member of the population using a random number generator, where N is the codebook size, so that each codebook consists of N single-vector chromosomes. P sets of N chromosomes are generated in this step, where P is the population size.

(ii) *Update:* GLA - GLA is used to update N chromosomes for every member of the population.

(iii) *Evaluation:* The fitness (or MSE) of every member of the population is evaluated.

(iv) *Selection:* The survivors of the current population are decided from the survival rate P_s. A random number generator is used to generate random numbers whose values are between 0 and 1. If the random number is $< P_s$, this codebook survives; otherwise, it does not survive. The best fitness of the population always survives. Pairs of parents are selected from these survivors and undergo a subsequent crossover operation to produce the child chromosomes that form a new population in the next generation.

(v) *Crossover:* The chromosomes of each survivor are sorted in decreasing order according to the squared error between the chromosome G_j of the current population and the central chromosome G_0. Without sorting at this stage, it is difficult to jump out of the local minima. The single point crossover technique [5] is used to produce the next generation from the selected parents.

(vi) *Termination:* Steps (ii)–(v) are repeated until the predefined number of generations has been reached. After termination, the optimal codebook is generated from N chromosomes in the best member of the current population.

Experimental results: The test materials for these experiments consisted of nine words recorded from one male speaker. The speech