HIERARCHICAL TEMPORAL DECOMPOSITION: A NOVEL APPROACH TO EFFICIENT COMPRESSION OF SPECTRAL CHARACTERISTICS OF SPEECH

S. Ghaemmaghami, M. Deriche, and S. Sridharan

School of Electrical & Electronic Systems Engineering
Queensland University of Technology, Brisbane, Australia
s.ghaemmaghami@qut.edu.au m.deriche@qut.edu.au s.sridharan@qut.edu.au

10.21437/ICSLP.1998-385

ABSTRACT

We propose a new approach to Temporal Decomposition (TD) of characteristic parameters of speech for very low rate coding applications. The method models the articulatory dynamics employing a hierarchical error minimization algorithm which does not use Singular Value Decomposition. It is also much *faster* than conventional TD and could be implemented in real-time. High flexibility is achieved with the proposed method to comply with the desired coding requirements, such as compression ratio, accuracy, delay, and computational complexity. This method can be used for coding spectral parameters at rates 1000-1200 b/s with high fidelity and an algorithmic delay of less than 150 msec.

1. INTRODUCTION

Efficiency in coding of speech spectral representation is achieved basically in two directions denoted by *inter-frame* and *intra-frame* dependences. The intra-frame dependence makes *Vector Quantization* (VQ) schemes more efficient than *scalar* quantization techniques due to the non-random nature of speech spectral parameters [1]. The inter-frame dependence between spectral parameters, obtained from successive frames, results directly from the phonetic structure of speech [2].

Typically, the frame period is much smaller than the effective length of most phonetic events. This means that we need to process sufficient number of frames to employ inter-frame dependence for an efficient statistical analysis. *Temporal Decomposition* (TD) is a powerful method to achieve such an analysis [3,4]. TD resolves the overlapping structure of the speech events and represents the spectral information by a smaller set of parameters and a set of *interpolation* functions for a given block of speech. This gives a compression ratio in the range of 4.5-7 for coding the spectral parameters [5]. However, TD is computationally complex and imposes a long algorithmic delay (.5 to 1 second) on the coding process as well. Hence it is used mostly in non-real-time applications [2].

In this paper, we propose a novel method, *Hierarchical* TD (HTD), which significantly simplifies conventional TD, based on the idea of event approximation in TD-based coding we developed earlier [5,6]. The organization of the paper is as follows: section 2 describes the theory of conventional TD and its advantages in speech coding. In section 3, the proposed method is developed. Experimental results are illustrated in section 4 and a discussion on performance of the method is given in section 5.

2. TEMPORAL DECOMPOSITION

Temporal Decomposition attributes the speech spectral parameters to the speech events through linear modeling of coarticulation [3]:

$$\mathbf{Y} = \mathbf{A}\mathbf{\Phi} \tag{1}$$

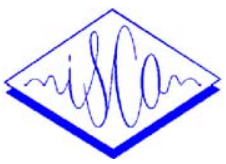
where $\mathbf{Y}$ is the matrix of spectral parameters, $\mathbf{\Phi}$ is the matrix of event functions, and $\mathbf{A}$ is the matrix of weightings.

In equation (1), only the $\mathbf{Y}$ is known. To find $\mathbf{\Phi}$ and $\mathbf{A}$ matrices, we need to decompose $\mathbf{Y}$ through orthogonalization [3]. Such a procedure is basically performed in two stages. First, the locations of event functions are detected using *Singular Value Decomposition* (SVD). Second, the event functions are refined using an iterative method, to minimize the distance (or error) between the estimated and the original parameter sets.

The refinement procedure is carried by minimizing the *Mean Square Error*, E, defined as [3]:

$$E = \sum_{n} [y_i(n) - \sum_{k=1}^{m} a_{ik}\phi_k(n)]^2 \tag{2}$$

From equation (2), event functions are extracted after elimination of their minor lobes with an insignificant



5th International Conference on Spoken Language Processing (ICSLP 98)
Sydney, Australia
November 30 - December 4, 1998

degradation in the performance [3].

We have shown earlier that events can be approximated using fixed shape/width functions, with a minor degradation in reconstructed speech quality (see [5] and [6]). This method, referred to as *Modified TD* (MTD), locates an *Event Approximating Function* (EAF) at the *centroid* of each event. This leads to a considerable reduction in the rate for coding the spectral information. In addition, it eliminates the computationally expensive *event refinement* task from TD. This idea forms the basis for the HTD method described in the next section.

3. THE PROPOSED METHOD: HIERARCHICAL TD (HTD)

Hierarchical TD (HTD) relies on the idea of MTD. It uses a specific EAF as *a priori* information about event functions. The decomposition problem is then reduced to the problem of searching for the *best* locations to locate EAF over the given speech block, in the sense of minimum *Mean Square Error* (MSE) between the approximated and the original parameter sets. The EAF is thus located at the *centroids* of the event functions, in the order of effectiveness in the approximation error reduction. Accordingly, HTD *optimizes* the event locating task.

Given a set of spectral parameters represented by a $p \times N$ matrix, $\mathbf{Y}$, we rewrite the basic expression for TD, as:

$$\mathbf{Y} = \mathbf{A}\mathbf{\Psi} \tag{3}$$

where $\mathbf{\Psi}$ is an $m \times N$ matrix of EAFs and $\mathbf{A}$ is a $p \times m$ matrix of *target vectors* as weightings [6].

Event locations over the whole speech block are found by searching within shorter segments. The length of the segments is computed based on the event rate, and total width of the EAF employed. Consequently, matrix $\mathbf{Y}$ is first partitioned into a number of overlapping matrices of equal size. This is a *windowing* process applied to $\mathbf{Y}$ with an overlap equal to the width of an EAF. The total number of segments in the block is given as:

$$N_s = \lceil \frac{L_b - L_e}{L_w - L_e} \rceil \tag{4}$$

where L_b is the length of $\mathbf{Y}$ (block length), L_e is the length of event, L_w is the window length, and all length parameters are described in terms of number of frames. The symbol $\lceil\ \rceil$ represents the nearest integer greater than the expression inside the symbol.

Given $\mathbf{Y}^l$ as the parameters matrix of the lth segment in the block, we need to search for the event locations in the segment to find the corresponding target vectors. We limit, here, the problem to one event per segment. This is expressed as:

$$\hat{\mathbf{Y}}^l = \mathbf{a}^l \psi^l \tag{5}$$

where ψ^l is a row vector as the event function, $\mathbf{a}^l$ is a column vector as target vector, and $\hat{\mathbf{Y}}^l$ is a $p \times L_w$ matrix of approximated parameters.

We begin the search by locating EAF at the first position in the segment and find corresponding target vector from equation (6) reversed. This is described as:

$$\mathbf{a}^{lk} = \hat{\mathbf{Y}}^l \, (\psi^{lk})^{-1} \tag{6}$$

where ψ^{lk} is the event vector when EAF located at $n = k$ and $\mathbf{a}^{lk}$ is the corresponding target vector. k can vary in the range of $(L_e + 1)/2$ to $L_w - (L_e + 1)/2$ where L_e is chosen to be an odd number.

Vector $(\psi^{lk})^{-1}$ is obtained from:

$$(\psi^{lk})^{-1} = (\psi^{lk})^T [\psi^{lk} (\psi^{lk})^T]^{-1} \tag{7}$$

where T is for transposition. For the simple case when only one event is considered in the segment, the right bracket is simply a scalar, but for other cases, the square matrix $[\psi^{lk}(\psi^{lk})^T]$ should be inverted. It can be shown, however, that this matrix is always *non-singular*.

The procedure is repeated for all values for k allowed in the segment and, then, the *Euclidean* distance between approximated and original parameter sets in the segment is computed as:

$$\varepsilon^l = d(\mathbf{Y}^l, \hat{\mathbf{Y}}^l) = \frac{1}{L_e} \sum_{j=k-\frac{L_e-1}{2}}^{k+\frac{L_e-1}{2}} (\mathbf{y}_j - \hat{\mathbf{y}}_j)^T (\mathbf{y}_j - \hat{\mathbf{y}}_j) \tag{8}$$

where $\mathbf{y}_j$ and $\hat{\mathbf{y}}_j$ are column vectors in matrices $\mathbf{Y}^l$ and $\hat{\mathbf{Y}}^l$, respectively (superscript l dropped).

All segments in the given block are treated similarly to obtain indices of event locations from distance minimization within the segments. This information is then used for constructing the matrix of EAFs for the whole block. It is to be noted that target vectors extracted from segments are not used anymore. Instead, we need to find the matrix of weightings, $\mathbf{A}$, for the whole block through reversing equation (3), as:

$$\mathbf{A} = \mathbf{Y}\mathbf{\Psi}^{-1} \tag{9}$$

where $\mathbf{\Psi}^{-1} = \mathbf{\Psi}^T [\mathbf{\Psi}\mathbf{\Psi}^T]^{-1}$, and $[\mathbf{\Psi}\mathbf{\Psi}^T]$ is a square non-singular matrix.

We shown briefly that the proposed method captures correlated parameters in $\mathbf{Y}$, in fact in segments of $\mathbf{Y}$, using the EAF as the *basis* function. From equation (8), the total approximation error within each segment can be expressed as:

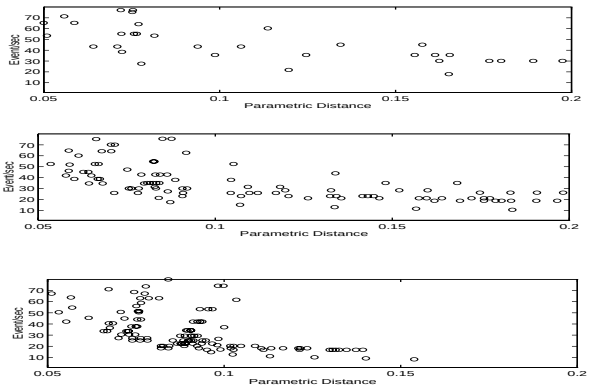


Figure 1. Event rate versus parametric distance for different values for the block length: 100, 150, and 200 msec, from top to bottom.

$$\varepsilon(k) = \sum_{i=1}^{p} \sum_{j=1}^{L_w} [y_i(j) - a^k(i)\psi^k(j)]^2 \qquad (10)$$

where $\varepsilon(k)$ is the approximation error, $y_i(j)$ is the (i,j)th element in the segmented $\mathbf{Y}$, $\psi^k(j)$ is the jth element of vector ψ^k, $a^k(i)$ is the ith element of vector $\mathbf{a}^k$, and superscript k represents the frame index at which EAF is located. We need to find $\mathbf{a}^k$ such that $\varepsilon(k)$ is minimized with a given value for k.

By making the partial derivatives of $\varepsilon(k)$ with respect to $a^k(i)$ equal to zero, we get:

$$\varepsilon_m(k) = \alpha^2 - \beta^2 \sum_i u^2(i,k) \qquad (11)$$

where $\alpha^2 = \sum_i \sum_j [y_i(j)]^2$, $\beta^2 = [\sum_j [\psi^k(j)]^2]^{-1}$ and $u(i,k) = \sum_r y_i(r)\psi^k(r)$.

Both α and β in equation (11) are constants for a given subset of parameters. Hence, the error is minimized when $u(i,k)$, representing *cross-correlation* between the EAF and the part of $\hat{\mathbf{Y}}^l$ selected, reaches its maximum value. This happens when the peak of the EAF coincides nearly with the *most-steady* points in the selected subset of the spectral parameters, over i dimension $(i = 1, ..., p)$.

4. EXPERIMENTS

We conducted a number of experiments, using different speech samples from the TIMIT database.

Figure 1 illustrates the effects of the system parameters on the event rate and the parametric distance for different block lengths (L_b). The points shown on the figures are obtained from averaging the results processing a large number of speech samples.

To measure the distance between the original and the approximated parameter sets, we used *Euclidean* metric in the LAR space which resembles Log Likelihood Ratio (LLR) [1]:

L_b	L_w	L_e	Event Rate	Parametric Distance
20	20 STD=1	13 STD=2	27	.078
30	22 STD=2	14 STD=2	24	.072
40	25 STD=3	17 STD=2	23	.071

Table 1. Best overall values for system parameters, corresponding STDs over speech samples, and resulting parametric distances. Length parameters are in terms of the number of frames.

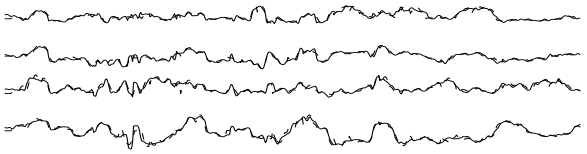


Figure 2. Time trajectories of the first four LAR parameters for $L_b = 30$. Solid: original, dashed: approximated.

$$d_p = \overline{[\sum_{i=1}^{p} |par_{1i}(n) - par_{2i}(n)|^2]^{1/2}} \qquad (12)$$

where n is the frame index, p is the number of parameters for each frame, and $par_1(n)$ and $par_2(n)$ are the original and the approximated LAR parameters for frame n, respectively. Acceptable values for the system parameters and their corresponding *Standard Deviation* (STD) over speech samples are shown in table 1. Note that the dispersion of points in the figures is due to the changes in the system parameters because of rounding errors. A sample of spectral parameters trajectories approximated by the method, obtained at rate 1.2 kb/s, is displayed in figure 2.

In all experiments, *Log Area Ratio* (LAR) parameters were used for HTD analysis based on our previous findings [7]. The frame length and frame period were 40 and 5 msec, respectively, where *Hanning* windowing was used.

5. DISCUSSION

Results presented in table 1 and figure 2 show that HTD is able to conform with the large variability found in the LAR trajectories. This is achieved given two major features of the method. First, the method is not sensitive to the frame rate. Hence, as long as the computational complexity is acceptable, the temporal resolution of the system can be improved. In practice, a resolution of 5 msec could give excellent performance, while 10 msec resolution could still yield adequate accuracy. Second, the method, unlike most other interpolation methods [8], models co-articulation which occurs at different levels of the articulation. At event rates

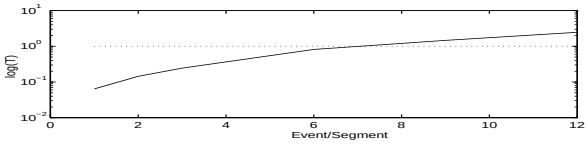


Figure 3. Normalized implementation time on log scale in terms of number of events located within each segment, where total number of events for the whole block is taken fixed. Dotted line shows implementation time for TD.

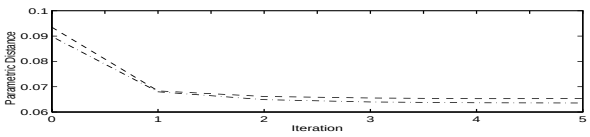


Figure 4. Parametric distance (distance between original and approximated parameters) in TD and HTD versus number of iterations in event refinement through mean square error minimization. TD: dashed, HTD: dashed-dotted.

higher than true phoneme rate, the proposed method characterizes co-articulation between consecutive subphonemes through adaptation with statistical characteristics of spectral parameters by adjusting the event locations over the speech block. Increasing the number of EAFs within each block could improve performance of the method at the expense of increasing the coding rate.

Table 1 also shows that HTD, unlike TD, does not need long blocks of speech to process the signal. This arises from the flexibility of HTD which is achieved by adjusting the method parameters, L_b, L_e, and L_w, based on the requirements in the coding system. Henceforth, the algorithmic delay can be reduced to about 100 msec, with very low distortion, as indicated in the table. This makes HTD usable in many voice communication and storage applications.

The elimination of two time consuming tasks, SVD and event refinement, from TD is another significant advantage of HTD over conventional TD. Figure 3 shows the implementation time of HTD, in terms of the number of events per segment, with respect to that of TD. For the simplest case considered in section 3, when only one event per segment is to be located, HTD is more than ten times faster than TD. This makes it plausible for real-time implementation using most PCs.

At lower event rates, when the length of EAF is comparable to the length of phonemes, HTD simulates TD. The optimal performance is achieved using an event refinement algorithm, as that in conventional TD. In this case, HTD again outperforms TD in almost all conditions from both viewpoints of accuracy and computational complexity. Figure 4 shows the distance between the original and the approximated spectral parameters, using both TD and HTD, after performing event refinement. The drawback of TD stems from its distant sidelobes which are likely missed in sidelobe removal at each iteration in the refinement process [3]. This prevents TD to reach an optimal state in the sense of minimum distance between original and approximated parameters. HTD is free from this shortcoming, as it uses monotonic, smooth functions at the initial state [6].

6. CONCLUSION

We have proposed, in this paper, a new method for temporal decomposition of speech spectral parameters using a hierarchical error minimization algorithm, applied directly to the matrix of parameters, for the purpose of speech compression. The method reduces significantly the computational load compared to that with conventional TD. It also gives high flexibility in adaptation with the coding system specifications, such as delay and accuracy. This makes the method applicable in many speech coding applications such as voice communication and voice storage.

7. REFERENCES

[1] Deller, J. R., Jr., J. G. Proakis, J. H. L. Hansen, "Discrete-Time Processing of Speech Signals", *MacMillan Pub. Co.*, 1993.

[2] Y.M. Cheng, D. O'Shaughnessy, "On 450-600 b/s Natural Sounding Speech Coding", *IEEE, Trans. SAAP*, Vol. 1, No. 2, pp. 207-219, 1993.

[3] B.S. Atal, "Efficient Coding of LPC Parameters by Temporal Decomposition", *Proc. ICASSP 83*, pp. 81-84, 1983.

[4] Bimbot, F., B. S. Atal, "An Evaluation of Temporal Decomposition", *Proc. EUROSPEECH'91*, pp. 1089-1092, Sep. 1991.

[5] S. Ghaemmaghami, M. Deriche, "A New Approach to Very Low-Rate Speech Coding Using Temporal Decomposition", *Proc. ICASSP'96*, Vol. 1, pp. 224-227, May 1996.

[6] Ghaemmaghami, S., M. Deriche, and B. Boashash, "On Modeling Event Functions in Temporal Decomposition Based Speech Coding", *EUROSPEECH'97*, Vol. 3, pp. 1299-1302, 1997.

[7] Ghaemmaghami, S., M. Deriche, B. Boashash, "Comparative Study of Different Parameters for Temporal Decomposition Based Speech Coding", *Proc. ICASSP'97*, Vol. 3, pp. 1703-1706, Apr. 1997.

[8] Lopez-Soler, J. M., N. Farvardin, "A Combined Quantization-Interpolation Scheme for Very Low Bit Rate Coding of Speech LSP Parameters", *Proc. ICASSP 93*, Vol. 2, pp. 21-24, 1993.