PAPER *Special Issue on Speech Information Processing*

Modified Restricted Temporal Decomposition and Its Application to Low Rate Speech Coding

Phu Chien NGUYEN[†], ***Student Member***, **Takao OCHI**[†*], ***Nonmember***, ***and*** **Masato AKAGI**[†], ***Regular Member***

SUMMARY This paper presents a method of temporal decomposition (TD) for line spectral frequency (LSF) parameters, called "Modified Restricted Temporal Decomposition" (MRTD), and its application to low rate speech coding. The LSF parameters have not been used for TD due to the stability problems in the linear predictive coding (LPC) model. To overcome this deficiency, a refinement process is applied to the event vectors in the proposed TD method to preserve their LSF ordering property. Meanwhile, the restricted second order TD model, where only two adjacent event functions can overlap and all event functions at any time sum up to one, is utilized to reduce the computational cost of TD. In addition, based on the geometric interpretation of TD the MRTD method enforces a new property on the event functions, named the "well-shapedness" property, to model the temporal structure of speech more effectively. This paper also proposes a method for speech coding at rates around 1.2 kbps based on STRAIGHT, a high quality speech analysis-synthesis method, using MRTD. In this speech coding method, MRTD based vector quantization is used for encoding spectral information of speech. Subjective test results indicate that the speech quality of the proposed speech coding method is close to that of the 4.8 kbps FS-1016 CELP coder.

key words: *temporal decomposition, LSF, STRAIGHT, speech coding*

1. Introduction

Temporal decomposition (TD) [1], which is an analysis procedure based on a linear model of the effects of co-articulation, yields a linear approximation of a time sequence of spectral parameters in terms of a series of time-overlapping event functions and an associated series of event vectors as given in Eq. (1).

$$\hat{\mathbf{y}}(n) = \sum_{k=1}^{K} \mathbf{a}_k \phi_k(n), \quad 1 \le n \le N \tag{1}$$

where $\mathbf{a}_k$ and $\boldsymbol{\phi}_k(n)$ are the kth event vector and kth event function, respectively. $\hat{\mathbf{y}}(n)$ is the approximation of $\mathbf{y}$(n), the nth spectral parameter vector, produced by the TD model. In matrix notation, Eq. (1) can be written as

$$\hat{\mathbf{Y}} = \mathbf{A}\boldsymbol{\Phi} \qquad \hat{\mathbf{Y}} \in \mathbf{R}^{P\times N}, \mathbf{A} \in \mathbf{R}^{P\times K}, \boldsymbol{\Phi} \in \mathbf{R}^{K\times N}$$

where P, N, and K are the order of the spectral parameters, the number of frames in the speech segment, and the number of event functions, respectively.

The second order TD model used in [15], where only two adjacent event functions can overlap as shown in Fig. 1, is given in Eq. (2).

$$\hat{\mathbf{y}}(n) = \mathbf{a}_k \boldsymbol{\phi}_k(n) + \mathbf{a}_{k+1} \boldsymbol{\phi}_{k+1}(n), \; n_k \le n < n_{k+1} \tag{2}$$

where n_k and n_{k+1} are the locations of event k and event $k+1$, respectively.

The restricted second order TD model was utilized in [2], [6], [9] with an additional restriction to the event functions in the second order TD model that all event functions at any time sum up to one. The argument for imposing this constraint on the event functions has not been explicitly stated in [6]. But, it has been shown in [2] that this constraint is needed to describe TD as a breakpoint analysis procedure in a multidimensional vector space, where breakpoints are connected by straight line segments (see Fig. 2). Equation (2) can be rewritten as

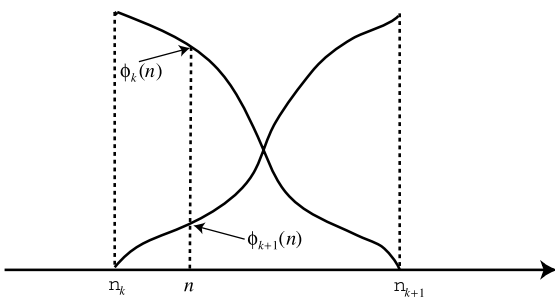

Fig. 1 Example of two adjacent event functions in the second order TD model.

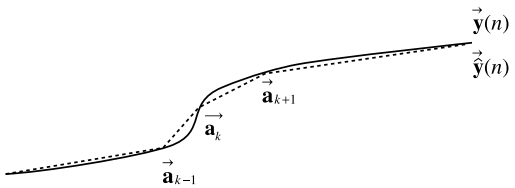

Fig. 2 The path in parameter space described by the sequence of spectral parameters $\mathbf{y}(\mathbf{n})$ is approximated by means of straight line segments between breakpoints.

Manuscript received June 30, 2002.
Manuscript revised November 14, 2002.

[†]The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Tatsunokuchi, Ishikawa-ken, 923–1292 Japan.
[*]Presently, with Matsushita Communication Industrial Co., Ltd.

$$\hat{\mathbf{y}}(n) = \mathbf{a}_k \boldsymbol{\phi}_k(n) + \mathbf{a}_{k+1}\left(1 - \boldsymbol{\phi}_k(n)\right), \; n_k \le n < n_{k+1} \tag{3}$$

The spectral parameters used in the original TD method by Atal [1] were the log-area parameters. Some other spectral parameter sets such as log area ratios, cepstrum, and so forth have also been considered as input for TD [3]. Due to the stability problems in the linear predictive coding (LPC) model, not all types of parametric representations can be used. This is because there is no guarantee that the selected spectral parameters are valid after spectral transformation performed by TD. Thus, the line spectral frequency (LSF) parameters [4] have not been used for the conventional TD method although they have several properties that make them more suitable for interpolation [11] and quantization [12].

An important property of LSFs $\{\omega_i\}$ is that they are ordered in $(0, \pi)$ as follows.

$$0 < \omega_1 < \omega_2 < \cdots < \omega_P < \pi \tag{4}$$

Also, (4) means that the difference of two consecutive LSFs (dLSF) $\{d_i = \omega_i - \omega_{i-1}\}$ with $d_1 = \omega_1$ and $d_{P+1} = \pi - \omega_P$ are always greater than zero. This ordering property is a necessary and sufficient condition for the stability of the corresponding LPC synthesis filter. It implies that TD can be applied to analyzing the LSF parameters if the ordering property of LSFs is guaranteed for the event vectors.

Kim and Oh [6] have introduced a method of temporal decomposition for the LSF parameters, called "Restricted Temporal Decomposition" (RTD), based on the restricted second order TD model. The RTD method enforces a minimum dLSF constraint on the event vectors in order to preserve their LSF ordering property. Originally, RTD was proposed in narrowband speech coding for significantly reducing the bit rate for spectral parameters [6]. Subsequent research [13] investigated on its application to wideband speech coding and found that RTD is a promising approach to low rate wideband speech coding also. However, both have not reported any drawback, from which RTD is being suffered.

In this paper we claim that the RTD method, however, has not completely guaranteed the LSF ordering property for the event vectors; instead, we propose an improved algorithm, namely modified RTD (MRTD), to solve this problem. Additionally, we impose a new property, the well-shapedness property, on the event functions to model the temporal structure of speech more effectively and reduce the quantization error when vector quantized.

We have investigated the application of MRTD to speech coding. In this paper a method for low rate speech coding based on STRAIGHT [5], a high quality speech analysis-synthesis method, using MRTD is also presented. Here, spectral information of speech is encoded using MRTD based vector quantization (VQ), whilst other speech parameters are encoded using scalar quantization (SQ). As a result, low bit rate speech coders operating at rates around 1.2 kbps have been realized. Subjective test results indicate that the speech quality of this speech coding method is close to that of the 4.8 kbps US Federal Standard (FS-1016) CELP coder.

2. MRTD of LSF Parameters

2.1 Additional Constraints on Event Functions

Based on the geometric interpretation of TD described in [2], we impose a new property, namely, the well-shapedness property on the event functions. Here, by a well-shaped event function we mean an event function having only one peak, as depicted in Fig. 3 (a). Those event functions having more than one peak are called ill-shaped event functions (see, e.g., Fig. 3 (b)). Well-shaped event functions are desirable from speech coding point of view. Further, the well-shapedness property helps to describe the temporal structure of speech by means of straight line segments between breakpoints more effectively.

TD yields an approximation of a sequence of spectral parameters by a linear combination of event vectors. Since TD's underlying distance metric is Euclidean, a natural requirement is to have this approximation be invariant with respect to a translation or rotation of the spectral parameters. Dix and Bloothooft [2] considered the geometric interpretation of TD results and found that TD is rotation and scale invariant, but it is not translation invariant.

In order to overcome this shortcoming and describe TD as a breakpoint analysis procedure in a multidimensional vector space, Dix and Bloothooft enforced two

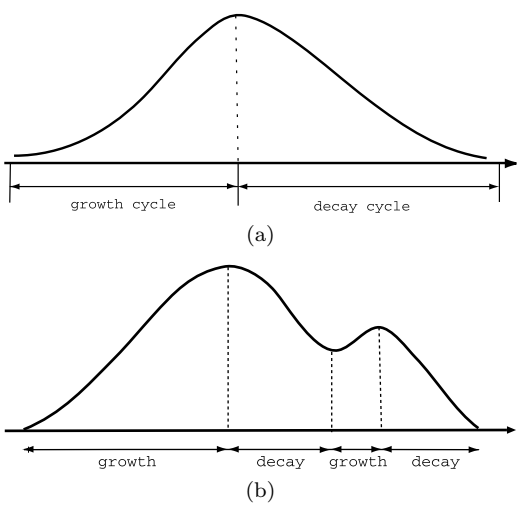

Fig. 3 Examples of a well-shaped event function (a) and an ill-shaped event function (b).

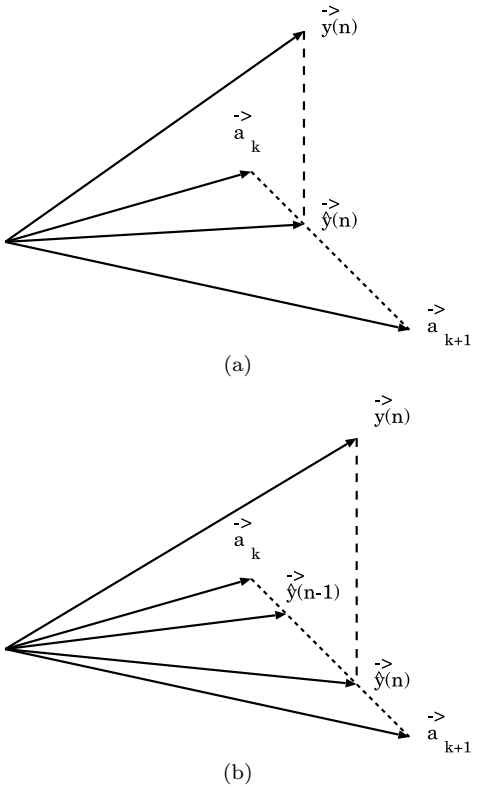

Fig. 4 Determination of event functions in the transition interval $[n_k, n_{k+1}]$. The point of the line segment between $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$ (a), between $\hat{\mathbf{y}}(n-1)$ and $\mathbf{a}_{k+1}$ (b) with minimum distance from $\mathbf{y}(n)$ is taken as the best approximation.

constraints, which are identical to those in the RTD method [6], on the event functions: (i) at any moment of time only two event functions, which are adjacent in time, are non-zero; and (ii) all event functions at any time sum up to one. In other words, the restricted second order TD model was utilized in both [2] and [6]. Geometrically speaking, the two event vectors $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$ define a plane in P-dimensional vector space. The determination of event functions $\phi_k(n)$ and $\phi_{k+1}(n)$ in the interval $[n_k, n_{k+1}]$ is now depicted in Fig. 4 (a) as the projection of vector $y(n)$ onto this plane and is also equivalent to that in [6]. Clearly the following holds: $\boldsymbol{\phi}_k(n_k) = 1$, $\boldsymbol{\phi}_k(n_{k+1}) = 0$, and $0 \leq \boldsymbol{\phi}_k(n) \leq 1$ for $n_k \leq n \leq n_{k+1}$.

The TD model is based on the hypothesis of articulatory movements towards and away from targets. An appealing result of the above properties of event functions is that one can interpret the values $\boldsymbol{\phi}_k(n)$ as a kind of activation values of the corresponding event. During the transition from one event towards the next the activation value of the left event decreases from one to zero, whilst the right event increases its activation value from zero to the value of one. Note that to model the temporal structure of speech more effectively no backwards transitions are allowed. Therefore, each event function should have a growth cycle; during which the event function grows from zero to one and a decay cycle; during which the event function decays from one to zero. In other words, each event function should have the well-shapedness property. In contrast, an ill-shaped event function can be viewed as an event function which has several growth and decay cycles.

However, the determination of event functions in [2], [6], [13] has not guaranteed the well-shapedness property for them since their changes during the transition from one event towards the next may not be monotonic, which results in ill-shaped event functions. In particular, one may wonder that if an event function has some values of one interlaced by other values, it will cause the next event function to have more than one lobe, which is not acceptable in the conventional TD method. Ill-shaped event functions are undesirable from speech coding point of view also. They increase the quantization error when vector quantized because the uncharacteristic valleys and secondary peaks are not normally captured by the codebook functions.

Taking into account the above considerations, we have determined the event functions corresponding to the point of the line segment between $\hat{\mathbf{y}}(n-1)$ and $\mathbf{a}_{k+1}$ instead of $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$ as considered in [2], [6], [13], with minimum distance from $\mathbf{y}(n)$ (see Fig. 4 (b)). This determination of event functions can be written in mathematical form as

$$\boldsymbol{\phi}_k(n) = \begin{cases} 1 - \boldsymbol{\phi}_{k-1}(n), & \text{if } n_{k-1} < n < n_k \\ 1, & \text{if } n = n_k \\ \min(\boldsymbol{\phi}_k(n-1), \max(0, \hat{\boldsymbol{\phi}}_k(n))), & \\ & \text{if } n_k < n < n_{k+1} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where

$$\hat{\boldsymbol{\phi}}_k(n) = \frac{\langle (\mathbf{y}(n) - \mathbf{a}_{k+1}), (\mathbf{a}_k - \mathbf{a}_{k+1}) \rangle}{\| \mathbf{a}_k - \mathbf{a}_{k+1} \|^2} \tag{6}$$

Here, $\langle ., . \rangle$ and $\| \cdot \|$ denote the inner product of two vectors and the norm of a vector, respectively.

2.2 Refinement of Event Vectors

The event vectors are estimated corresponding to the determined event functions in the least mean square sense using the following formula [1], [6], [8], [13], [15].

$$\mathbf{A} = \mathbf{Y}\boldsymbol{\Phi}^T(\boldsymbol{\Phi}\boldsymbol{\Phi}^T)^{-1} \tag{7}$$

The estimated event vectors may violate the ordering property of LSFs since the error criterion does not consider this property. Given the minimum value, ε, of dLSFs, Kim and Oh [6] re-estimated the event vectors from the lowest to the highest order, replaced $\mathbf{a}_{i-1,k}$ and $\mathbf{a}_{i,k}$ by $\hat{\mathbf{a}}_{i-1,k}$ and $\hat{\mathbf{a}}_{i,k} = \hat{\mathbf{a}}_{i-1,k} + \varepsilon$, respectively, whenever $\mathbf{a}_{i-1,k} + \varepsilon > \mathbf{a}_{i,k}$. Considering the increment of error E, where $E = \sum_{n=1}^{N} \| \mathbf{y}(n) - \hat{\mathbf{y}}(n) \|^2$, caused by this change, they determined $\hat{\mathbf{a}}_{i-1,k}$ as

$$\hat{\mathbf{a}}_{i-1,k} = \frac{\mathbf{a}_{i-1,k} + \mathbf{a}_{i,k} - \varepsilon}{2} \tag{8}$$

However, this routine still does not assure the LSF ordering property for $\mathbf{a}_k$ since $\hat{\mathbf{a}}_{i-1,k} < \mathbf{a}_{i-1,k}$ and there is no guarantee that $\mathbf{a}_{1,k} > 0$ or $\mathbf{a}_{P,k} < \pi$. We propose an improved algorithm to deal with this problem.

Firstly, a more general routine for changing J components ($1 \le J \le P - i + 1$): $\mathbf{a}_{i,k}, \mathbf{a}_{i+1,k}, \cdots, \mathbf{a}_{i+J-1,k}$ to $\hat{\mathbf{a}}_{i,k}$, $\hat{\mathbf{a}}_{i+1,k} = \hat{\mathbf{a}}_{i,k} + \varepsilon, \cdots, \hat{\mathbf{a}}_{i+J-1,k} = \hat{\mathbf{a}}_{i,k} + (J-1)\varepsilon$, respectively, is established. Consider that the increment of error E caused by this change is

$$\boldsymbol{\Delta} = \sum_{l=0}^{J-1} \left[\mathbf{a}_{i+l,k} - (\hat{\mathbf{a}}_{i,k} + l\varepsilon)\right]^2 \sum_n \boldsymbol{\phi}_k(n)^2$$

and $\hat{\mathbf{a}}_{i,k} \ge \mathbf{a}_{i-1,k} + \varepsilon$, $\hat{\mathbf{a}}_{i,k}$ should be determined as follows to minimize Δ:

$$\hat{\mathbf{a}}_{i,k} = \begin{cases} \mathbf{a}_{i-1,k} + \varepsilon, & \text{if } \tilde{\mathbf{a}}_{i,k} < \mathbf{a}_{i-1,k} + \varepsilon \\ \tilde{\mathbf{a}}_{i,k}, & \text{otherwise} \end{cases} \tag{9}$$

where

$$\tilde{\mathbf{a}}_{i,k} = \frac{\sum_{l=0}^{J-1} \mathbf{a}_{i+l,k}}{J} - \frac{(J-1)\varepsilon}{2} \tag{10}$$

In the sequel, an algorithm for normalizing an event vector $\mathbf{a}_k$ is developed. In order to assure that $\mathbf{a}_{1,k} > 0$ and $\mathbf{a}_{P,k} < \pi$, we add zero and π to $\mathbf{a}_k$ so that $\mathbf{a}_k = [0, \mathbf{a}_{1,k}, \ldots, \mathbf{a}_{P,k}, \pi]^T$. Zero and π are denoted as $\mathbf{a}_{0,k}$ and $\mathbf{a}_{P+1,k}$ for simplicity. Note that $\mathbf{a}_{0,k}$ and $\mathbf{a}_{P+1,k}$ cannot be changed during the normalization. The whole algorithm is depicted in Fig. 5 and described as follows:

Step 1: initialize $i \leftarrow 0$.

Step 2: if $i < P$ and $\mathbf{a}_{i,k} + \varepsilon \le \mathbf{a}_{i+1,k}$, set $i \leftarrow i+1$. Repeat this step until $i = P$ or $\mathbf{a}_{i,k} + \varepsilon > \mathbf{a}_{i+1,k}$. If $i = P$, go to step 6.

Step 3: if $i = 0$, set $i \leftarrow 1$ and $j \leftarrow 1$ since $\mathbf{a}_{0,k}$ could not be changed; if not, set $j \leftarrow 2$.

Step 4: change $\mathbf{a}_{i,k}, \ldots, \mathbf{a}_{i+j-1,k}$ to $\hat{\mathbf{a}}_{i,k}, \ldots, \hat{\mathbf{a}}_{i+j-1,k}$ using Eq. (9). If $i + j - 1 = P$, go to step 6.

Step 5: if $\mathbf{a}_{i+j-1,k} + \varepsilon > \mathbf{a}_{i+j,k}$, restore $\mathbf{a}_k$ from the previous step, set $j \leftarrow j+1$, and go back to step 4; if not, set $i \leftarrow i + j$. Go back to step 2 if $i < P$.

Step 6: if $\mathbf{a}_{P,k} + \varepsilon \le \mathbf{a}_{P+1,k}$, $\mathbf{a}_k$ has been normalized; if not, restore i and the corresponding value of vector $\mathbf{a}_k$ from the previous step, set $j \leftarrow P - i + 1$ and go back to step 4.

At step 6, it is of interest to notice that if i is the last component of a modified segment, i is then set to the beginning of that segment. In particular, if $i = 0$, vector $\mathbf{a}_k$ is set as $[0, \pi - P\varepsilon, \pi - (P-1)\varepsilon, \ldots, \pi]^T$. However, in practice this case almost never occurs.

In the result, when the locations of events n_k, where $k = 1, \ldots, K$, are known and the corresponding event vectors are initialized with the samples of the LSF

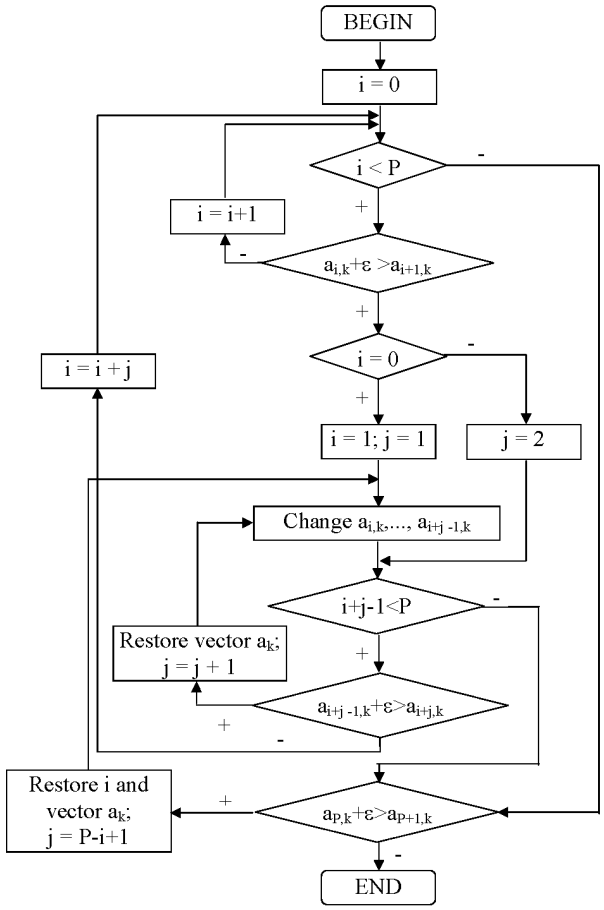

Fig. 5 Block diagram of the improved algorithm for normalizing event vectors.

vector trajectory $\mathbf{y}(n_k)$, we can calculate proper event functions and event vectors iteratively using Eqs. (5), (7), and (9). Here, we suggest using the local minima of the following spectral feature transition rate (SFTR) based on LSF parameters as the initial locations of events [6], [8].

$$\text{SFTR}: \quad s(n) = \sum_{i=1}^{P} c_i(n)^2, \quad 1 \le n \le N \tag{11}$$

where

$$c_i(n) = \frac{\sum_{m=-M}^{M} m\mathbf{y}_i(n+m)}{\sum_{m=-M}^{M} m^2}, \quad 1 \le i \le P \tag{12}$$

The window size, $2M$, of SFTR analysis is the only parameter that effects the initial number and locations of events. In addition, a new event is inserted where the initial reconstruction error $e(n) = \| \mathbf{y}(n) - \hat{\mathbf{y}}(n) \|^2$ has a local maximum larger than a certain threshold θ as considered in [6]. The way of segmenting input vectors

Table 1 Percentage number of invalid LSF event vectors and well-shaped event functions for RTD and MRTD methods. The speech data set consists of 250 utterances spoken by 10 speakers (5 male and 5 female) of the ATR Japanese speech database.

Method	% invalid LSF event vectors	% well-shaped event functions
RTD	0.08%	88%
MRTD	0%	100%

Table 2 Event rate, average LSD, and percentage number of outlier frames for RTD and MRTD methods. The speech data set consists of 250 utterances spoken by 10 speakers (5 male and 5 female) of the ATR Japanese speech database.

Method	Event rate	Avg. LSD	2–4 dB	> 4 dB
RTD	20.16 events/sec	1.563 dB	22.97%	0.96%
MRTD	20.16 events/sec	1.568 dB	23.15%	0.98%

for online analysis presented in [6] is also adopted in the MRTD method.

2.3 Performance Evaluation

A set of 250 sentences of the ATR Japanese speech database were selected as the speech data. This speech data set consists of about 20 minutes of speech spoken by 10 speakers (5 male & 5 female) re-sampled at 8 kHz sampling frequency. 10th order LSF parameters were calculated using a LPC analysis window of 30 ms at 10 ms frame intervals, and TD analyzed using the original RTD and the MRTD methods in turn. Here, $2M = 4$, $\theta = 0.2$, and $\varepsilon = 0.01$ were empirically chosen as suitable values for the window size of SFTR analysis, the event insertion threshold, and the minimum dLSF, respectively.

Table 1 gives the summary of invalid LSF event vectors and well-shaped event functions obtained from the MRTD and RTD methods for the above speech data set. Results indicate that the drawbacks of RTD method described in Sects. 2.1 and 2.2 have been overcome in the proposed MRTD method.

Log spectral distortion (LSD) measure [11], [12] was used to evaluate the interpolation performance of the proposed MRTD algorithm in comparison with the original RTD. The LSD evaluated is that between the original LSF parameters, $\mathbf{y}(n)$, and the reconstructed LSF parameters, $\hat{\mathbf{y}}(n)$. Table 2 gives the summary of spectral distortion results obtained from the RTD and MRTD methods for the speech data set mentioned above. Results indicate slightly better performance in the case of RTD over MRTD.

Shortly speaking, the drawbacks of RTD method in terms of invalid LSF event vectors and ill-shaped event functions can be solved with a negligible increase in spectral distortion. Note that LSD was calculated for the interpolation step only, i.e. before quantization.

Figure 6 shows the plot of event functions obtained from the MRTD method for an example of a Female/Japanese speech utterance.

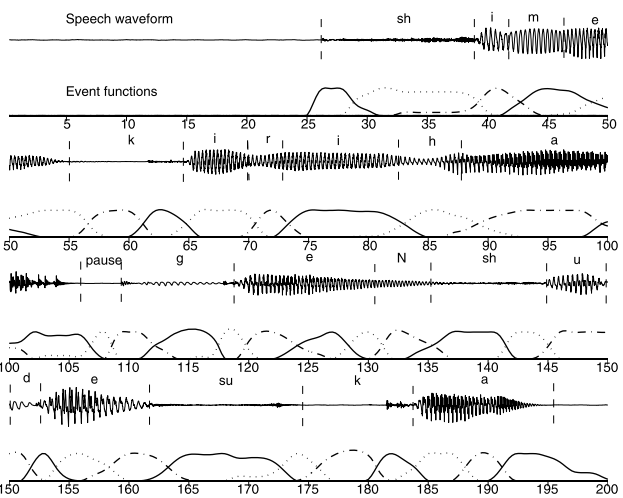

Fig. 6 Plot of the event functions obtained from MRTD for the Female/Japanese speech utterance "*shimekiri ha geNshu desu ka.*" The speech waveform is also shown together with the phonetic transcription for reference. The numerals indicate the frame numbers.

3. Coding Speech at Very Low Rates Based on STRAIGHT Using MRTD

3.1 Overview of the Proposed Speech Coding Method

As shown earlier, the speech in TD is no longer represented by a vector updated frame by frame, but instead by the continuous trajectory of a vector. The trajectory is decomposed into a set of phoneme-like events, i.e. a series of temporally overlapping event functions and a corresponding series of event vectors. Since the updating rate of events is much less than the frame rate, TD has been considered for efficient coding of spectral parameters [1], [3], [6], [8]–[10], [13].

STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum) has been proposed by Kawahara et al., which is a high quality vocoder type algorithm [5]. STRAIGHT can decompose a speech waveform into a spectral envelope, i.e. spectrogram, F0 (fundamental frequency) information, and noise ratios. Those parameters and the maximum value of amplitude are required for synthesizing speech. The spectrogram derived from STRAIGHT is very smooth thanks to a time-frequency interpolation procedure. It follows that the LSF parameters extracted from the spectrogram are correlated among frames, and thus the corresponding LSF vector trajectory is smooth also. It is not the case of normal LPC analyses, where LSF parameters are extracted independently on a frame-by-frame basis.

To make STRAIGHT applicable to low rate speech coding, the bit rate required to represent the spectral envelope must be minimized. Since the spectral en-

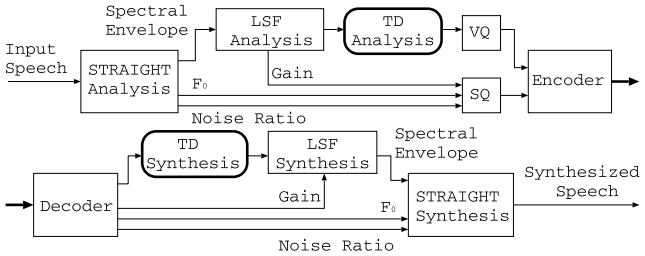

Fig. 7 Proposed speech encoder and decoder block diagrams (top: endcoder, bottom: decoder).

velope can be further analyzed into spectral parameters and gain information, TD can be incorporated with STRAIGHT to create high quality speech coders working at low bit rates.

In this section, we introduce a method for low rate speech coding based on STRAIGHT using MRTD. The encoder and decoder block diagrams are shown in Fig. 7 and a detailed description of the proposed speech coding method is shown in the subsections followed.

3.2 Derivation of LSF Parameters

The amplitude spectrum $X[m]$, where $0 \leq m \leq \frac{M}{2}$ with M is the number of samples in the frequency domain, obtained from STRAIGHT analysis is transformed to the power spectrum using Eq. (13).

$$S[m] = |\, X[m] \,|^2, \quad 0 \leq m \leq \frac{M}{2} \tag{13}$$

The ith autocorrelation coefficient, $R[i]$, is then calculated using the inverse Fourier transform of the power spectrum as follows.

$$R[i] = \frac{1}{M} \sum_{m=0}^{M-1} S[m] \exp\left\{ j \frac{2\pi m i}{M} \right\} \tag{14}$$

where $S[m] = S[M-m]$ and $0 \leq i \leq M-1$. Assume that the speech samples can be estimated by a Pth order all-pole model, where $0 < P < M$, the reconstruction error is calculated as given in Eq. (15).

$$P_L = R[0] - \sum_{l=1}^{P} a_l^P R[l] \tag{15}$$

where $\{a_l^P\}$, $l = 1, 2 \cdots P$, are the corresponding linear predictive coding (LPC) coefficients. P_L hereafter is referred to as gain. By minimizing P_L with respect to a_l^P, where $l = 1, 2 \cdots P$, a_l^Ps could be evaluated. They are then transformed to the LSF parameters.

3.3 Determination of LSFs' Order

3.3.1 Spectral Distortion vs. LSFs' Order

Log spectral distortion (LSD) [11], [12] measure was also used as the objective measure of performance to determine the suitable order of LSFs. A set of 112 phoneme balanced sentences uttered by speaker MMY of the ATR Japanese speech database were used as the speech data. This speech data set were re-sampled at 8 kHz sampling frequency and then STRAIGHT analyzed. In the following, the spectral envelopes obtained from STRAIGHT analysis were transformed to LSF parameters of orders 14, 18, 22, 26, 30, and 34 using the procedure described in Sect. 3.2. Finally, the resulting LSF parameters were MRTD analyzed.

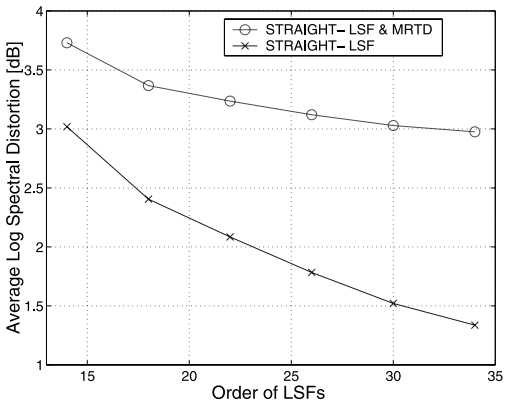

Fig. 8 Spectral distortion vs. the order of LSFs.

The spectral distortion results obtained from STRAIGHT analysis and LSF transformation, abbreviated as STRAIGHT-LSF, from STRAIGHT analysis, LSF transformation, and MRTD analysis, abbreviated as STRAIGHT-LSF & MRTD, are shown in Fig. 8. Note that these results were obtained before the quantization step. The horizontal and vertical axes indicate the order of LSFs and the average log spectral distortion, respectively. Results show that a considerable reduction of the spectral distortion for STRAIGHT-LSF & MRTD is not achieved when the order of LSFs exceeds the 22nd order.

3.3.2 Quality of Synthesized Speech vs. LSFs' Order

We used the Scheffe's method of paired comparison [14] to subjectively evaluate the quality of the synthesized speech as a function of the LSFs' order. Six graduate students known to have normal hearing ability were recruited for the listening experiment. Each listener was asked to make one of the following statements for each ordered pair of stimuli (i, j).

(−2) Distortion of i is much larger than that of j.
(−1) Distortion of i is slightly larger than that of j.
(0) Distortion of i is equivalent to that of j.
(1) Distortion of j is slightly larger than that of i.
(2) Distortion of j is much larger than that of i.

Two phoneme balanced sentences uttered by speaker MMY of the ATR Japanese speech database were re-sampled at 8 kHz sampling frequency, and then ana-

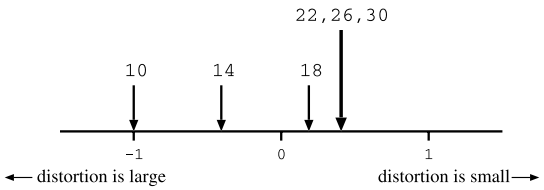

Fig. 9 Speech quality vs. the order of LSFs.

lyzed by STRAIGHT. The resulting spectral envelopes were transformed to LSF parameters of orders 10, 14, 18, 22, 26, and 30. In the following, the LSF parameters were analyzed into event vectors and event functions using the MRTD method. Those event vectors and event functions were combined to reconstruct LSF parameters used for synthesizing stimuli.

Figure 9 shows the results of the listening experiment. In this figure, the positions on the horizontal axis indicate the relative distances of stimuli. Here, the positive values mean that the distortion is small whilst the negative values indicate the high distortion. The number on each arrow corresponds to the order of LSFs. Results also show that an increase of distortion is not easily realized when the order of LSFs exceeds the 22nd order.

For the above reasons, the 22nd LSF parameters were used in the proposed speech coding method.

3.4 MRTD Based VQ of LSF Parameters

The reason for interpolating the vector trajectory of LSF parameters by using TD is that the updating rate of events is much less than the frame rate, and both event vectors and event functions can be quantized efficiently. In other words, the LSF parameters can be quantized efficiently by transforming them into the event sequences first, and then quantizing event vectors and event functions.

3.4.1 VQ of Event Vectors

Since the event vectors obtained from the MRTD method are valid LSF parameter vectors [9], they can be quantized by usual quantization methods for the LSF parameters. Here, the Split-VQ method [12] was adopted. Due to the distribution of LSFs, the event vectors were divided into three subvectors of dimensions 7, 7, 8 and each subvector was quantized independently. We assigned 8 or 9 bits to each subvector, which resulted in the number of bits allocated to one event vector was 24 or 27, respectively.

3.4.2 VQ of Event Functions

In the case of event functions, normalizing event functions is necessary to fix the dimension of the event function vector space. Notice that only quantizing $\phi_k(n)$ in the interval $[n_k; n_{k+1}]$ is enough to reconstruct the whole event function $\phi_k(n)$. Moreover, $\phi_k(n)$ always starts from one and goes down to zero in that interval, and the type of decrease (after normalizing the length of $\phi_k(n)$) can be vector quantized. Therefore, an event function $\phi_k(n)$ can be quantized by its length $L(k) = n_{k+1} - n_k$ and shape in $[n_k + 1; n_{k+1} - 1]$. In this work, 10 equidistant samples were taken from each event function for length-normalization and then vector quantized by a 7-bit codebook. Considering that all intervals between two consecutive event locations are less than 256 frames long (note that the frame period used in STRAIGHT analysis is 1 ms long), we used 8 bits for quantizing the length of each event function.

3.5 Coding Excitation Parameters

3.5.1 Coding F0 Parameters

For encoding F0 information, the lengths of voiced and unvoiced segments were quantized by using SQ first, with an average bit rate of 36 bps. In the following, linear interpolation was used within the unvoiced segments to form a continuous F0 contour. The continuous F0 contour was re-sampled at 28 ms intervals, and then quantized by a 5-bit logarithmic quantizer.

In the decoder, F0 values were reconstructed from the quantized samples using the linear interpolation. In the sequel, F0 values of unvoiced intervals were set to zero. The root mean square (RMS) F0 error was found to be about 3.7 Hz for the speech data set used in Sect. 3.3.1.

3.5.2 Coding Gain Parameters

The gain contour was re-sampled at 20 ms intervals. Logarithmic quantization was performed using 6 bits for each sampled value. The quantized samples and the spline interpolation were used in the decoder to form the reconstructed gain contour. The RMS gain error was found to be about 4.6 dB for the speech data set used in Sect. 3.3.1.

3.5.3 Coding Noise Ratio Parameters

The noise ratio parameters were estimated from the noise ratio targets and the event functions as follows.

$$\hat{i}(n) = \sum_{k=1}^{K} i_k \phi_k(n), \quad 1 \le n \le N \tag{16}$$

where $\hat{i}(n)$ and i_k are the reconstructed noise ratio parameter for the nth frame and the kth noise ratio target, respectively. The noise ratio targets were determined by minimizing the sum squared error, E_i, between the original and the interpolated noise ratio parameters.

Table 3 Bit allocation for the proposed speech coders.

Parameter	Proposed Coder 1	Proposed Coder 2
Event vector	24 bits (8+8+8)	27 bits (9+9+9)
Event function	7 bits	7 bits
Event location	8 bits	8 bits
Noise ratio target	5 bits	5 bits
Subtotal A (sum × event rate)	660 bps	705 bps
F0	215 bps	215 bps
Gain	300 bps	300 bps
Maximum amplitude of input speech	5 bps	5 bps
Subtotal B	520 bps	520 bps
Total (A+B)	1180 bps	1225 bps

$$E_i = \sum_{n=1}^{N} \left(i(n) - \hat{i}(n) \right)^2 = \sum_{n=1}^{N} \left(i(n) - \sum_{k=1}^{K} i_k \phi_k(n) \right)^2 \tag{17}$$

where $i(n)$ is the original noise ratio parameter for the nth frame. The noise ratio targets were quantized by using SQ with 5 bits. The RMS noise ratio error was found to be about 0.1 for the speech data set used in Sect. 3.3.1.

3.6 Bit Allocation

The bit allocation for the proposed speech coding method is shown in Table 3. The average number of events per second, i.e. the event rate, was set as 15 events/sec. We allocated 8 bits and 9 bits to each subvector of the event vectors, which resulted in 1.18 kbps and 1.23 kbps speech coders, respectively.

3.7 Subjective Tests

In order to evaluate the performance of the proposed speech coding method, the quality of the reconstructed speech was compared to that of other low bit rate speech coders: the 4.8 kbps FS-1016 CELP and 2.4 kbps FS-1015 LPC-10E coders.

A listening experiment was carried out by using the Scheffe's method of paired comparison [14] similarly to that in Sect. 3.3.2. A set of 108 phoneme balanced sentences of the ATR Japanese speech database were selected as the training data for the proposed speech coders. Speakers were 3 male & 3 female reading each of sentences. These speech utterances were re-sampled at 8 kHz sampling frequency, and then STRAIGHT analyzed using the frame shift of 1 ms. 22nd order LSF transformation was performed and the resulting LSF parameters were MRTD analyzed. Two phoneme balanced sentences, which are out of training set, uttered by a male and a female were used as the testing data. Stimuli were synthesized by using the following coders: 4.8 kbps FS-1016 CELP, 2.4 kbps FS-1015 LPC-10E, proposed 1.18 kbps speech coder 1, and proposed 1.23 kbps speech coder 2. Also, four

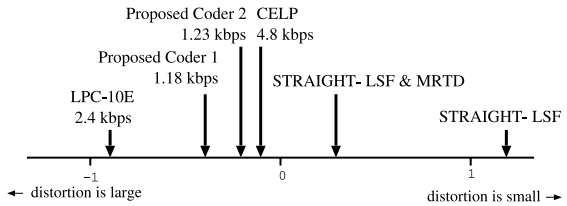

Fig. 10 Results of the listening experiment.

other stimuli were STRAIGHT synthesized using the speech parameters obtained from STRAIGHT-LSF and STRAIGHT-LSF & MRTD. The original and the reconstructed speech files are located at the following URL: http://www.jaist.ac.jp/∼chien/OF/

Results of the listening experiment are shown in Fig. 10. It can be seen from this figure that the quality of the reconstructed speech obtained from the proposed speech coder 2 is close to that of the 4.8 kbps FS-1016 CELP coder and is much better than that of the 2.4 kbps FS-1015 LPC-10E coder.

As shown previously, the reconstructed LSF parameters after RTD analyzed and synthesized may be invalid, which causes the reconstructed speech to be noisy as well as to have click tones. We therefore did not evaluate the performance of the method for speech coding using RTD.

4. Conclusion

We have presented a method of temporal decomposition, MRTD, for the LSF parameters. The additional constraint on the event functions in the second order TD model makes them monotonic during the transition from one event towards the next, from which the event functions can describe the temporal structure of speech more effectively. Also, this reduces the quantization error of event functions when vector quantized. The ordering property of LSFs has completely been ensured for the event vectors using the improved algorithm so that MRTD can be used for decomposing the LSF parameters.

We have also described a low rate speech coding method based on STRAIGHT using MRTD, where MRTD based VQ is used for encoding spectral information of speech. As a result, two low rate speech coders operating at rates around 1.2 kbps were produced. Although the quality of the reconstructed speech is little bit lower than that of the 4.8 kbps FS-1016 CELP coder according to the listening experiment, it is much better than that of the 2.4 kbps FS-1015 LPC-10E coder. However, the speech quality of the proposed speech coding method can be improved by increasing the event rate, which results in an increase in the bit-rate required for encoding speech.

It is necessary to evaluate other attributes of the proposed speech coding method: algorithmic delay, complexity, and noise robustness. In this work the

event rate was set as 15 events/sec, thus resulting in an average algorithmic delay of about 90 ms. We can add additional events, if necessary, to keep the algorithmic delay below 100 ms. Meanwhile, the computational cost and noise robustness of the proposed speech coding method depend mainly on STRAIGHT. This is because MRTD has significantly reduced the computational cost of TD by avoiding the use of the computationally costly singular value decomposition routine and the adaptive Gauss-Seidel iterations used in Atal's method. On the other hand, MRTD can be applied to analyzing any LSF vector trajectory. Currently, a real-time method for STRAIGHT based low rate speech coding using MRTD still remains for future research.

Acknowledgements

This work was supported by CREST (Core Research for Evolutional Science and Technology) of JST (Japan Science and Technology Corporation).

The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] B.S. Atal, "Efficient coding of LPC parameters by temporal decomposition," Proc. ICASSP '83, pp.81–84, 1983.
[2] P.J. Dix and G. Bloothooft, "A breakpoint analysis procedure based on temporal decomposition," IEEE Trans. Speech Audio Process., vol.2, no.1, pp.9–17, 1994.
[3] S. Ghaemmaghami, M. Deriche, and B. Boashash, "Comparative study of different parameters for temporal decomposition based speech coding," Proc. ICASSP '97, pp.1703–1706, 1997.
[4] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals," J. Acoust. Soc. Am., vol.57, p.S35, April 1975.
[5] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," Speech Commun., vol.27, no.3-4, pp.187–207, 1999.
[6] S.J. Kim and Y.H. Oh, "Efficient quantization method for LSF parameters based on restricted temporal decomposition," Electron. Lett., vol.35, no.12, pp.962–964, 1999.
[7] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantiser design," IEEE Trans. Commun., vol.COM-28, no.1, pp.84–95, 1980.
[8] A.C.R. Nandasena and M. Akagi, "Spectral stability based event localizing temporal decomposition," Proc. ICASSP '98, pp.957–960, 1998.
[9] P.C. Nguyen and M. Akagi, "Improvement of the restricted temporal decomposition method for line spectral frequency parameters," Proc. ICASSP2002, pp.265–268, 2002.
[10] P.C. Nguyen, T. Ochi, and M. Akagi, "Coding speech at very low rates using STRAIGHT and temporal decomposition," Proc. ICSLP2002, pp.1849–1852, 2002.
[11] K.K. Paliwal, "Interpolation properties of linear prediction parametric representations," Proc. EuroSpeech '95, pp.1029–1032, 1995.
[12] K.K. Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," IEEE Trans. Speech Audio Process., vol.1, no.1, pp.3–14, 1993.
[13] C.H. Ritz and I.S. Burnett, "Temporal decomposition: A promising approach to low rate wideband speech compression," Proc. EuroSpeech2001, pp.2315–2318, 2001.
[14] H. Scheffe, "An analysis of variance for paired comparisons," J. Am. Stat. Assoc., vol.47, pp.381–400, 1952.
[15] Y. Shiraki and M. Honda, "Extraction of temporal pattern of spectral sequence based on minimum distortion criterion," Proc. 1991 Autumn Meet. Acoust. Soc. Jpn, pp.233–234, 1991.

Phu Chien Nguyen was born in Bacninh, Vietnam on April 18, 1974. He received the B.S. degree in Mathematics from Hanoi University of Pedagogy, the M.S. degree in Information Technology from Hanoi University of Science, Hanoi, Vietnam in 1994 and 2000, respectively. He is currently working towards the Ph.D. degree in Information Science at Japan Advanced Institute of Science and Technology (JAIST), Japan. His research interests include Speech Signal Processing, Pattern Recognition, and Natural Language Processing.

Takao Ochi was born in Fukuoka, Japan on August 21, 1976. He received the B.E. degree in Electrical Engineering from the National Institution for Academic Degrees, the M.E. degree in Information Science from JAIST in 2000 and 2002, respectively. He is currently with Matsushita Communication Industrial Co., Ltd.

Masato Akagi was born in Okayama, Japan on September 12, 1956. He received the B.E. degree in Electronic Engineering from Nagoya Institute of Technology in 1979, the M.E. and the Dr.Eng. degrees in Computer Science from Tokyo Institute of Technology in 1981 and 1984, respectively. In 1984, he joined the Electrical Communication Laboratories, Nippon Telegraph and Telephone Corporation (NTT). From 1986 to 1990, he worked at the ATR Auditory and Visual Perception Research Laboratories. Since 1992, he has been with the School of Information Science, JAIST, where he is currently a professor. During 1988, he joined the Research Laboratories of Electronics, MIT as a visiting researcher, and in 1983, he studied at the Institute of Phonetic Science, University of Amsterdam. His research interests include Speech Perception Mechanisms of Humans, and Speech Signal Processing. He received the IEICE Excellent Paper Award from the IEICE in 1987, and the Sato Prize for Outstanding Paper from the Acoustical Society of Japan in 1998.