PAPER

Vector Quantization of Speech Spectral Parameters Using Statistics of Static and Dynamic Features

Kazuhito KOISHIDA[†*], **Keiichi TOKUDA**[††], **Takashi MASUKO**[†], ***and*** **Takao KOBAYASHI**[†], ***Regular Members***

SUMMARY This paper proposes a vector quantization scheme which makes it possible to consider the dynamics of input vectors. In the proposed scheme, a linear transformation is applied to the consecutive input vectors and the resulting vector is quantized with a distortion measure defined by the statistics. At the decoder side, the output vector sequence is determined using the statistics associated with the transmitted indices in such a way that a likelihood is maximized. To solve the maximization problem, a computationally efficient algorithm is derived. The performance of the proposed method is evaluated in LSP parameter quantization. It is found that the LSP trajectories and the corresponding spectra change quite smoothly in the proposed method. It is also shown that the use of the proposed method results in a significant improvement of subjective quality.

key words: *vector quantization, dynamic features, spectral quantization, LSP parameters*

1. Introduction

Efficient quantization of the spectral envelopes is a major concern in low bit rate coding of speech. A widely accepted method for evaluating the quantization performance is to measure the spectral distortion (SD) in each frame and then compare average SD and the number of outliers [1]. However, these values are not always indicative of the perceived distortion. One reason for this situation is that the evolution of the spectrum is not considered. In fact, a number of studies have demonstrated the importance of spectral dynamics [2]–[4]. A common technique used in these methods is to incorporate constraints for controlling the dynamics of the spectral parameters. Kleijn and Hagen introduced a constraint into the distortion measure of the codebook search in the encoder [2]. The basic idea is to penalize a large rate of change in the quantized spectral parameters compared to the original. In [3], Knagenhjelm and Kleijn presented a decoding scheme which smoothes the trajectory of the parameters under the constraint that the reconstructed parameters fall within the Voronoi regions associated with the transmitted quantization index. Samuelsson et al. [4] used above constraints for both the encoder and decoder.

In this paper, we present an alternative approach to spectral quantization in which spectral dynamics is taken into consideration [5], [6]. The proposed vector-quantization (VQ) scheme works as follows: in the encoder side, a linear transformation is applied to the consecutive input vectors. The linear transformation used in this paper is designed so that the transformed vector consists of static and dynamic features of the input vectors. The transformed vector is then quantized with a distortion measure defined by the statistics. In the decoder side, using the statistics associated with the transmitted indices, the output vector sequence is determined in such a way that a likelihood is maximized. To solve the maximization problem, a computationally efficient algorithm is derived. The performance of the proposed method is evaluated in LSP parameter quantization. It will be shown that the proposed method can generate the smoothly varying spectra and, as a result, improve the subjective quality.

This paper is organized as follows. The proposed VQ scheme is described in Sect. 2. In Sect. 3, we derive a time-recursive algorithm for computing the output vectors. Section 4 provides the experimental results. In Sect. 5, several aspects of the proposed scheme are discussed. Finally, conclusions are given in Sect. 6.

2. VQ Using Statistics of Linear Transform of Consecutive Input Vectors

2.1 Preliminaries

A K-dimensional input vector at time t is denoted by $\boldsymbol{x}_t = [x_t(1), x_t(2), \cdots, x_t(K)]'$ where the superscript $'$ indicates matrix transpose. Let us consider a vector $\boldsymbol{X}_t$ which consists of consecutive input vectors around t (from time $t - L_-$ to $t + L_+$):

$$\boldsymbol{X}_t = \left[\ \boldsymbol{x}'_{t-L_-}, \cdots, \boldsymbol{x}'_{t-1}, \boldsymbol{x}'_t, \boldsymbol{x}'_{t+1}, \cdots, \boldsymbol{x}'_{t+L_+}\ \right]'. \quad (1)$$

Using an N-by-S matrix $\boldsymbol{w}$ ($S = (L_- + L_+ + 1)K$) whose row vectors are independent, we define a vector $\boldsymbol{z}_t$ as a linear transformation of $\boldsymbol{X}_t$, i.e.,

$$\boldsymbol{z}_t = \boldsymbol{w}\boldsymbol{X}_t \quad (2)$$

Manuscript received April 17, 2000.

Manuscript revised February 28, 2001.

[†] The authors are with the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama-shi, 226–8502 Japan.

[††] The author is with the Department of Computer Science, Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.

[*] Presently, with Signal Compression Laboratory, University of California, Santa Barbara, CA 93106-9560, USA.

where the dimension of $\boldsymbol{z}_t$, N, must satisfy $N > K$. The matrix $\boldsymbol{w}$ can be chosen according to the characteristics of the input vectors. As an example, let the vector $\boldsymbol{z}_t$ include the current vector and its time derivative which is approximated by the first-order difference:

$$\boldsymbol{z}_t = [\boldsymbol{x}_t', \Delta\boldsymbol{x}_t']' \tag{3}$$

where

$$\Delta\boldsymbol{x}_t = g\,(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) \tag{4}$$

and g is a weighting coefficient. This can be accomplished by setting $L_- = 1, L_+ = 0$ and

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{0}_{K\times K} & \boldsymbol{I}_{K\times K} \\ -g\boldsymbol{I}_{K\times K} & g\boldsymbol{I}_{K\times K} \end{bmatrix} \tag{5}$$

where $\boldsymbol{I}_{K\times K}$ and $\boldsymbol{0}_{K\times K}$ are the K-by-K identity matrix and K-by-K null matrix, respectively.

In the proposed method, the vector $\boldsymbol{z}_t$ is quantized at the encoder and its index is transmitted to the decoder.

2.2 Encoding Process

The codebook C is modeled as a family of Gaussian probability density functions $P_i(\cdot)$ such that each cell is represented by an N-by-1 mean vector $\boldsymbol{m}_i$ and N-by-N covariance matrix $\boldsymbol{U}_i$:

$$C = \{P_i(\cdot) \mid 1 \le i \le I\}, \quad (I : \text{codebook size}) \tag{6}$$

where

$$P_i(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z};\ \boldsymbol{m}_i, \boldsymbol{U}_i) \tag{7}$$

and $\mathcal{N}$ is given by

$$\begin{aligned}&\mathcal{N}(\boldsymbol{z};\ \boldsymbol{m}_i, \boldsymbol{U}_i)\\ &\quad = \frac{1}{(2\pi)^{\frac{N}{2}}|\boldsymbol{U}_i|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{m}_i)'\boldsymbol{U}_i^{-1}(\boldsymbol{z}-\boldsymbol{m}_i)\right].\end{aligned} \tag{8}$$

The optimum index at time t, i_t, is found according to

$$i_t = \arg\min_{1\le i\le I} d(\boldsymbol{z}_t, \boldsymbol{m}_i) \tag{9}$$

where $d(\boldsymbol{z}_t, \boldsymbol{m}_i)$ is the distortion measure between $\boldsymbol{z}_t$ and $\boldsymbol{m}_i$ defined by

$$\begin{aligned} d(\boldsymbol{z}_t, \boldsymbol{m}_i) &= -\log P_i(\boldsymbol{z}_t)\\ &= -\log \mathcal{N}(\boldsymbol{z}_t;\ \boldsymbol{m}_i, \boldsymbol{U}_i)\\ &= \frac{1}{2}(\boldsymbol{z}_t-\boldsymbol{m}_i)'\boldsymbol{U}_i^{-1}(\boldsymbol{z}_t-\boldsymbol{m}_i)\\ &\quad +\frac{1}{2}\log|\boldsymbol{U}_i| + \frac{N}{2}\log 2\pi. \end{aligned} \tag{10}$$

Instead of Eq. (10), a simplified one such as the Mahalanobis distance or the Euclidean distance is also applicable.

The criterion for designing the codebook C is to let overall distortion $E[d(\boldsymbol{z}_t, \boldsymbol{m}_{i_t})]$ be minimized over all $P_i(\cdot)$, where $E[\cdot]$ denotes the expectation. In order to design the codebook, the conventional algorithm, such as the EM algorithm and the LBG algorithm, can be used.

2.3 Decoding Process

For a given index sequence i_t $(t = 1, \cdots, T)$, the output sequence $\hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}_1', \hat{\boldsymbol{x}}_2', \cdots, \hat{\boldsymbol{x}}_T']'$ is determined in such a way that the following likelihood is maximized with respect to $\hat{\boldsymbol{X}}$:

$$P(\hat{\boldsymbol{Z}}|C) = \prod_{t=1}^{T} P_{i_t}(\hat{\boldsymbol{z}}_t) \tag{11}$$

where

$$\hat{\boldsymbol{Z}} = [\hat{\boldsymbol{z}}_1', \hat{\boldsymbol{z}}_2', \cdots, \hat{\boldsymbol{z}}_T']' \tag{12}$$

and

$$\hat{\boldsymbol{z}}_t = \boldsymbol{w}\hat{\boldsymbol{X}}_t \tag{13}$$

$$\hat{\boldsymbol{X}}_t = [\hat{\boldsymbol{x}}_{t-L_-}', \cdots, \hat{\boldsymbol{x}}_t', \cdots, \hat{\boldsymbol{x}}_{t+L_+}']'. \tag{14}$$

The maximization of Eq. (11) is equivalent to the maximization of

$$\begin{aligned}\log P(\hat{\boldsymbol{Z}}|C) &= \log\sum_{t=1}^{T} P_{i_t}(\hat{\boldsymbol{z}}_t)\\ &= -\frac{1}{2}\sum_{t=1}^{T}(\hat{\boldsymbol{z}}_t-\boldsymbol{m}_{i_t})'\boldsymbol{U}_{i_t}^{-1}(\hat{\boldsymbol{z}}_t-\boldsymbol{m}_{i_t})\\ &\quad -\frac{1}{2}\sum_{t=1}^{T}\log|\boldsymbol{U}_{i_t}| - \frac{NT}{2}\log 2\pi \end{aligned} \tag{15}$$

with respect to $\hat{\boldsymbol{X}}$. The first term of Eq. (15) can be written in matrix form:

$$\begin{aligned}\varepsilon(\hat{\boldsymbol{X}}|C) &= \sum_{t=1}^{T}(\hat{\boldsymbol{z}}_t-\boldsymbol{m}_{i_t})'\boldsymbol{U}_{i_t}^{-1}(\hat{\boldsymbol{z}}_t-\boldsymbol{m}_{i_t})\\ &= (\boldsymbol{W}\hat{\boldsymbol{X}}-\boldsymbol{M})'\boldsymbol{U}^{-1}(\boldsymbol{W}\hat{\boldsymbol{X}}-\boldsymbol{M}) \end{aligned} \tag{16}$$

where

$$\boldsymbol{M} = [\boldsymbol{m}_{i_1}', \boldsymbol{m}_{i_2}', \cdots, \boldsymbol{m}_{i_T}']' \tag{17}$$

$$\boldsymbol{U} = \mathrm{diag}[\boldsymbol{U}_{i_1}, \boldsymbol{U}_{i_2}, \cdots, \boldsymbol{U}_{i_T}] \tag{18}$$

$$\boldsymbol{W} = [\boldsymbol{w}_1', \boldsymbol{w}_2', \cdots, \boldsymbol{w}_T']' \tag{19}$$

and

$$\boldsymbol{w}_t = \left[\boldsymbol{0}_{N\times(t-L_--1)K},\ \boldsymbol{w},\ \boldsymbol{0}_{N\times(T-t-L_+)K}\right]. \tag{20}$$

The maximization of Eq. (15) leads to the minimization of Eq. (16) with respect to $\hat{\boldsymbol{X}}$. Taking the derivative of Eq. (16) with respect to $\hat{\boldsymbol{X}}$ and setting the result equal to zero, we can determine the optimal vector sequence

$\hat{\boldsymbol{X}}$ by solving

$$\boldsymbol{R}\hat{\boldsymbol{X}} = \boldsymbol{r} \tag{21}$$

where

$$\boldsymbol{R} = \boldsymbol{W}'\boldsymbol{U}^{-1}\boldsymbol{W} \tag{22}$$

$$\boldsymbol{r} = \boldsymbol{W}'\boldsymbol{U}^{-1}\boldsymbol{M}. \tag{23}$$

Since $\boldsymbol{R}$ is a TK-by-TK matrix, direct solution of Eq. (21) requires $O(T^3K^3)$ operations. By utilizing the fact that $\boldsymbol{R}$ is a band matrix with $(L_-+L_++1)K$ bandwidth, the complexity of $O(T(L_-+L_++1)^2K^3)$ is obtained using an efficient technique such as the Cholesky decomposition.

3. Time-Recursive Algorithm for Computing Output Vectors

For spectral quantization in speech coding, the foregoing algorithm has several disadvantages listed below:

- A large value of T causes a large coding delay as well as a high computational complexity, which is generally unacceptable in most speech-coding applications.
- Since the output vectors are independently obtained every T-frame, the continuity of the output vectors between neighboring frames often breaks by a small value of T.

It is therefore desirable to determine the output vectors time-recursively with a small frame-delay. This can be achieved by a time-recursive algorithm derived in this section.

Before showing the time-recursive algorithm, we describe a recursive algorithm which is an alternative method to solve Eq. (21). The recursive algorithm is then developed into the time-recursive algorithm.

3.1 Recursive Algorithm

Let us consider replacing the elements of $\boldsymbol{W}$, $\boldsymbol{w}$, with $\overline{\boldsymbol{w}}$ from time t to T, and this yields the following matrix $\overline{\boldsymbol{W}}^{(t-1)}$:

$$\overline{\boldsymbol{W}}^{(t-1)} = \left[\boldsymbol{w}'_1, \cdots, \boldsymbol{w}'_{t-1}, \overline{\boldsymbol{w}}'_t, \overline{\boldsymbol{w}}'_{t+1}, \cdots, \overline{\boldsymbol{w}}'_T\right]' \tag{24}$$

where

$$\overline{\boldsymbol{w}}_t = [\boldsymbol{0}_{N\times(t-L_--1)K},\ \overline{\boldsymbol{w}},\ \boldsymbol{0}_{N\times(T-t-L_+)K}]. \tag{25}$$

In this case, the set of equations corresponding to Eq. (21) can be written as

$$\overline{\boldsymbol{R}}^{(t-1)}\overline{\boldsymbol{X}}^{(t-1)} = \overline{\boldsymbol{r}}^{(t-1)} \tag{26}$$

where

$$\overline{\boldsymbol{R}}^{(t-1)} = \overline{\boldsymbol{W}}'^{(t-1)}\boldsymbol{U}^{-1}\overline{\boldsymbol{W}}^{(t-1)} \tag{27}$$

$$\overline{\boldsymbol{r}}^{(t-1)} = \overline{\boldsymbol{W}}'^{(t-1)}\boldsymbol{U}^{-1}\boldsymbol{M} \tag{28}$$

$$\overline{\boldsymbol{X}}^{(t-1)} = \left[\overline{\boldsymbol{x}}'^{(t-1)}_1, \cdots, \overline{\boldsymbol{x}}'^{(t-1)}_t, \cdots, \overline{\boldsymbol{x}}'^{(t-1)}_T\right]' \tag{29}$$

The matrix $\overline{\boldsymbol{w}}$ is defined so that $\overline{\boldsymbol{R}}^{(0)}$ is a K-by-K block diagonal matrix. For the transform matrix $\boldsymbol{w}$ in Eq. (5), it is reasonable to choose

$$\overline{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{0}_{K\times K} & \boldsymbol{I}_{K\times K} \\ \boldsymbol{0}_{K\times K} & \boldsymbol{0}_{K\times K} \end{bmatrix}. \tag{30}$$

Since $\overline{\boldsymbol{R}}^{(0)}$ is block diagonal, each element of $\overline{\boldsymbol{X}}^{(0)}$, $\overline{\boldsymbol{x}}'^{(0)}_t$, can be obtained independently with $O(K^3)$ operations ($O(K)$ for diagonal covariance matrix).

Substituting $\boldsymbol{w}_t$ for $\overline{\boldsymbol{w}}_t$ in Eq. (24), we can express the set of equations corresponding to Eq. (21) as follows:

$$\overline{\boldsymbol{R}}^{(t)}\overline{\boldsymbol{X}}^{(t)} = \overline{\boldsymbol{r}}^{(t)} \tag{31}$$

where

$$\overline{\boldsymbol{W}}^{(t)} = \left[\boldsymbol{w}'_1, \cdots, \boldsymbol{w}'_{t-1}, \boldsymbol{w}'_t, \overline{\boldsymbol{w}}'_{t+1}, \cdots, \overline{\boldsymbol{w}}'_T\right]'. \tag{32}$$

Using the above equations, we get the following relations:

$$\overline{\boldsymbol{R}}^{(t)} = \overline{\boldsymbol{R}}^{(t-1)} + \boldsymbol{v}'_t\boldsymbol{U}^{-1}_{i_t}\boldsymbol{v}_t \tag{33}$$

$$\overline{\boldsymbol{r}}^{(t)} = \overline{\boldsymbol{r}}^{(t-1)} + \boldsymbol{v}'_t\boldsymbol{U}^{-1}_{i_t}\boldsymbol{m}_{i_t} \tag{34}$$

$$\boldsymbol{v}_t = \boldsymbol{w}_t - \overline{\boldsymbol{w}}_t. \tag{35}$$

It can be seen that the above relations are similar to the time update recursion of the set of equations for the RLS adaptive filtering [7]. On the analogy of the derivation of the RLS algorithm, i.e., the application of the matrix inversion lemma, we can derive a recursive algorithm for obtaining $\overline{\boldsymbol{X}}^{(t)}$ from $\overline{\boldsymbol{X}}^{(t-1)}$ recursively. The solution of Eq. (21), $\hat{\boldsymbol{X}}$, is obtained by setting $\hat{\boldsymbol{X}} = \overline{\boldsymbol{X}}^{(T)}$.

A summary of the recursive algorithm is presented in Table 1. In the table, the matrix $\overline{\boldsymbol{P}}^{(t)}$ represents the inversion of $\overline{\boldsymbol{R}}^{(t)}$. Since almost all the elements of the vector $\boldsymbol{v}_t$ equal to zero, Eq. (A.5) dominates the computational complexity, which is $O(T^2K^3)$. If $\boldsymbol{U}_{i_t}$ is diagonal, it will be reduced to $O(T^2K)$.

Table 1 Summary of recursive algorithm.

$$\boldsymbol{\pi} = \overline{\boldsymbol{P}}^{(t-1)}\boldsymbol{v}'_t \tag{A.1}$$

$$\boldsymbol{\nu} = \boldsymbol{v}_t\boldsymbol{\pi} \tag{A.2}$$

$$\boldsymbol{k} = \boldsymbol{\pi}\left(\boldsymbol{I}_{N\times N} + \boldsymbol{U}^{-1}_{i_t}\boldsymbol{\nu}\right)^{-1} \tag{A.3}$$

$$\overline{\boldsymbol{X}}^{(t)} = \overline{\boldsymbol{X}}^{(t-1)} + \boldsymbol{k}\left(\boldsymbol{U}^{-1}_{i_t}\boldsymbol{m}_{i_t} - \boldsymbol{v}_t\overline{\boldsymbol{X}}^{(t-1)}\right) \tag{A.4}$$

$$\overline{\boldsymbol{P}}^{(t)} = \overline{\boldsymbol{P}}^{(t-1)} - \boldsymbol{k}\boldsymbol{U}^{-1}_{i_t}\boldsymbol{\pi}' \tag{A.5}$$

Table 2 Summary of time-recursive algorithm.

$$\boldsymbol{Q} = \boldsymbol{J}_a \overline{\boldsymbol{P}}_L^{(t-1)} \boldsymbol{J}_a' + \widetilde{\boldsymbol{P}}_L^{(t)} \tag{B.1}$$

$$\boldsymbol{Y} = \boldsymbol{J}_a \overline{\boldsymbol{X}}_L^{(t-1)} + \widetilde{\boldsymbol{X}}_L^{(t)} \tag{B.2}$$

$$\boldsymbol{\pi} = \boldsymbol{Q} \boldsymbol{v}_L' \tag{B.3}$$

$$\boldsymbol{\nu} = \boldsymbol{v}_L \boldsymbol{\pi} \tag{B.4}$$

$$\boldsymbol{k} = \boldsymbol{\pi} \left(\boldsymbol{I}_{N\times N} + \boldsymbol{U}_{i_t}^{-1} \boldsymbol{\nu} \right)^{-1} \tag{B.5}$$

$$\overline{\boldsymbol{X}}_L^{(t)} = \boldsymbol{Y} + \boldsymbol{k} \left(\boldsymbol{U}_{i_t}^{-1} \boldsymbol{m}_{i_t} - \boldsymbol{v}_L \boldsymbol{Y} \right) \tag{B.6}$$

$$\overline{\boldsymbol{P}}_L^{(t)} = \boldsymbol{Q} - \boldsymbol{k} \boldsymbol{U}_{i_t}^{-1} \boldsymbol{\pi}' \tag{B.7}$$

3.2 Time-Recursive Algorithm

The time-recursive algorithm is derived by introducing the sliding window concept of the RLS algorithm into the recursive algorithm. The algorithm is capable of updating the matrix $\overline{\boldsymbol{P}}^{(t)}$ and vector $\overline{\boldsymbol{X}}^{(t)}$ based on the information of a finite number of frames. We note that, while the recursive algorithm provides the optimal output vectors for Eq. (21), the output vectors obtained with the time-recursive algorithm are sub-optimal.

Table 2 provides a summary of the time-recursive algorithm. In the algorithm, the matrix $\overline{\boldsymbol{P}}_L^{(t)}$ and the vector $\overline{\boldsymbol{X}}_L^{(t)}$ are defined as a KL-by-KL matrix and KL-by-1 vector, respectively, where L is the number of frames required for updating them. Eqs. (B.1) and (B.2) are key parts of the algorithm, in which old components are discarded using matrix $\boldsymbol{J}_a$:

$$\boldsymbol{J}_a = \underbrace{\begin{bmatrix} \boldsymbol{0}_{K\times K} & \boldsymbol{I}_{K\times K} & \boldsymbol{0}_{K\times K} & \cdots & \boldsymbol{0}_{K\times K} \\ \boldsymbol{0}_{K\times K} & \boldsymbol{0}_{K\times K} & \boldsymbol{I}_{K\times K} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \boldsymbol{0}_{K\times K} \\ \vdots & & \ddots & \ddots & \boldsymbol{I}_{K\times K} \\ \boldsymbol{0}_{K\times K} & \cdots & \cdots & \boldsymbol{0}_{K\times K} & \boldsymbol{0}_{K\times K} \end{bmatrix}}_{KL} \tag{36}$$

and new components are then added. In the table, $\widetilde{\boldsymbol{P}}_L^{(t)}$, $\widetilde{\boldsymbol{X}}_L^{(t)}$ and $\boldsymbol{v}_L$ are defined by

$$\widetilde{\boldsymbol{P}}_L^{(t)} = \mathrm{diag}\Big[\underbrace{\boldsymbol{0}_{K\times K}, \cdots, \boldsymbol{0}_{K\times K}}_{(L-1)K}, \underbrace{\widetilde{\boldsymbol{R}}_t^{-1}}_{K} \Big] \tag{37}$$

$$\widetilde{\boldsymbol{X}}_L^{(t)} = \Big[\underbrace{0, \cdots, 0}_{(L-1)K}, \underbrace{\left(\widetilde{\boldsymbol{R}}_t^{-1} \widetilde{\boldsymbol{r}}_t \right)'}_{K} \Big]' \tag{38}$$

$$\boldsymbol{v}_L = \boldsymbol{w}_L - \overline{\boldsymbol{w}}_L \tag{39}$$

where

$$\widetilde{\boldsymbol{R}}_t = \boldsymbol{J}_b \overline{\boldsymbol{w}}_L' \boldsymbol{U}_{i_t}^{-1} \overline{\boldsymbol{w}}_L \boldsymbol{J}_b' \tag{40}$$

$$\widetilde{\boldsymbol{r}}_t = \boldsymbol{J}_b \overline{\boldsymbol{w}}_L' \boldsymbol{U}_{i_t}^{-1} \boldsymbol{m}_{i_t} \tag{41}$$

$$\overline{\boldsymbol{w}}_L = \Big[\underbrace{\boldsymbol{0}_{N\times K}, \cdots, \boldsymbol{0}_{N\times K}}_{(L-L_- -L_+)K}, \underbrace{\overline{\boldsymbol{w}}}_{(L_- +L_+ +1)K} \Big] \tag{42}$$

$$\boldsymbol{w}_L = \Big[\underbrace{\boldsymbol{0}_{N\times K}, \cdots, \boldsymbol{0}_{N\times K}}_{(L-L_- -L_+)K}, \underbrace{\boldsymbol{w}}_{(L_- +L_+ +1)K} \Big] \tag{43}$$

and

$$\boldsymbol{J}_b = \Big[\underbrace{\boldsymbol{0}_{K\times K}, \cdots, \boldsymbol{0}_{K\times K}}_{(L-1)K}, \underbrace{\boldsymbol{I}_{K\times K}}_{K} \Big] \tag{44}$$

It is obvious that the Eqs. (B.3) through (B.7) correspond with the Eqs. (A.1) through (A.5) in Table 1, respectively.

The value of L is determined by

$$L = \max(D_{dec}, L_-) + 1 \tag{45}$$

where D_{dec} is the length of frame delay allowed in the decoder.

The output vector at time t, $\hat{\boldsymbol{x}}_t$ is obtained as

$$\hat{\boldsymbol{x}}_t = \overline{\boldsymbol{x}}_{t-D_{dec}}^{(t)} \tag{46}$$

where $\overline{\boldsymbol{x}}_{t-D_{dec}}^{(t)}$ is a vector contained in $\overline{\boldsymbol{X}}_L^{(t)}$

$$\overline{\boldsymbol{X}}_L^{(t)} = \left[\overline{\boldsymbol{x}}'^{(t)}_{t-(L-1)}, \cdots, \overline{\boldsymbol{x}}'^{(t)}_{t} \right]'. \tag{47}$$

Note that, for $D_{dec} \geq L_-$, the output vector is always chosen from the first column-vector in $\overline{\boldsymbol{X}}_L^{(t)}$.

The time-recursive algorithm has a computational complexity of $O(L^2K^3)$ ($O(L^2K)$ for diagonal covariance). Since almost all the elements of $\boldsymbol{J}_a, \boldsymbol{J}_b$ and $\boldsymbol{v}_L$ are equal to zero, Eq. (B.7) still dominates the computational complexity of the algorithm.

Finally, we mention the frame delay of the proposed scheme. The overall frame delay D is given by

$$D = D_{enc} + D_{dec} \tag{48}$$

where D_{enc} is the length of the frame delay at the encoder side. It is noted that D_{enc} satisfies

$$D_{enc} = L_+ \tag{49}$$

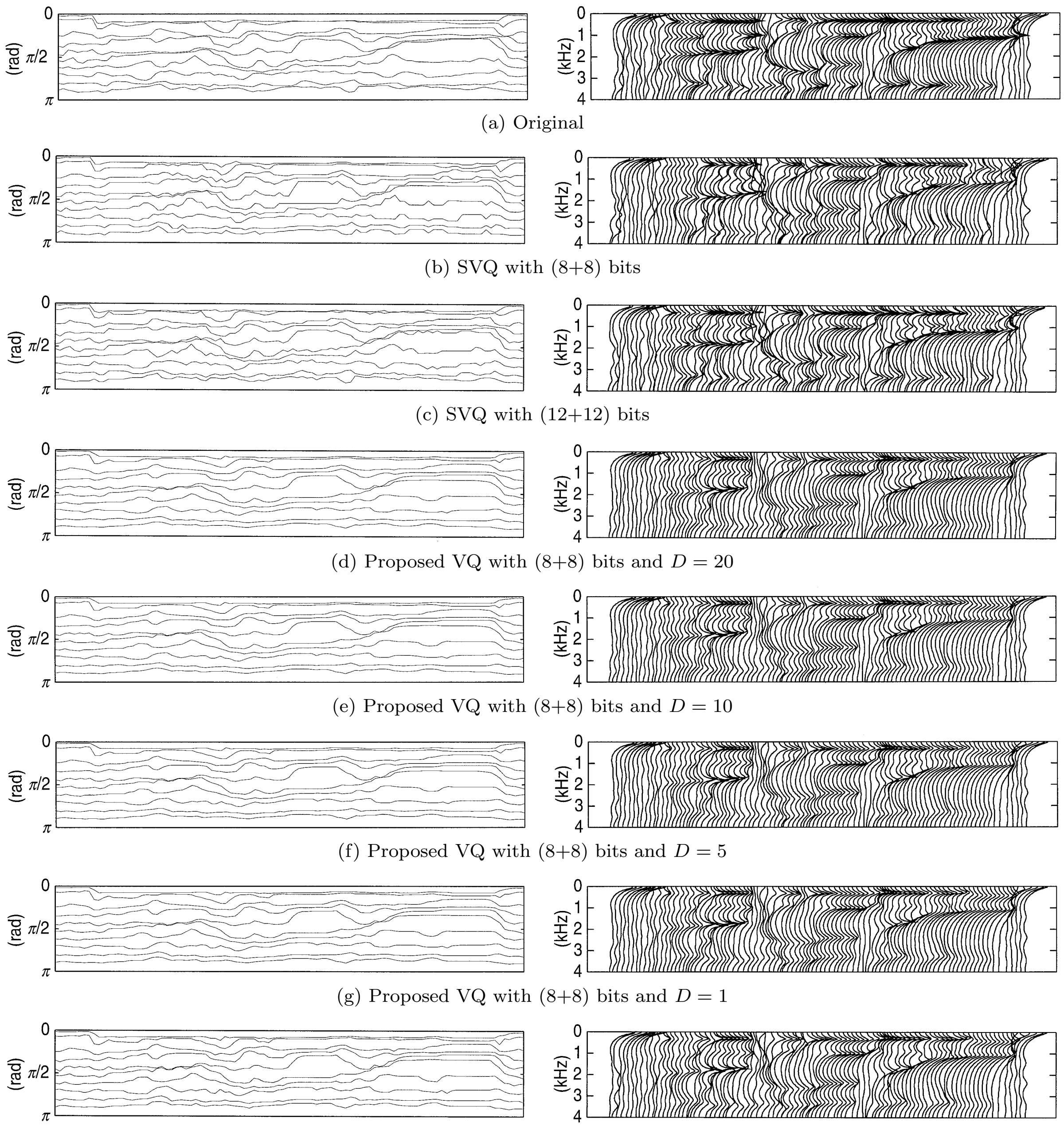
since the input vectors up to time $(t + L_+)$ are needed to compute the vector $\boldsymbol{z}_t$.

4. Experiments

In this section, we apply the proposed scheme with the time-recursive algorithm to LSP parameter quantization and investigate its performance.

4.1 Experimental Conditions

Speech signals sampled at 8-kHz are used for training



(a) Original

(b) SVQ with (8+8) bits

(c) SVQ with (12+12) bits

(d) Proposed VQ with (8+8) bits and $D = 20$

(e) Proposed VQ with (8+8) bits and $D = 10$

(f) Proposed VQ with (8+8) bits and $D = 5$

(g) Proposed VQ with (8+8) bits and $D = 1$

(h) Proposed VQ with (8+8) bits and $D = 0$

Fig. 1 LSP trajectories (left side) and corresponding spectra (right side).

and evaluation. A 10th-order LP analysis is performed every 10-ms using a 32-ms Hamming window. The LP coefficients are then transformed into the LSP parameters. The 10th-dimensional LSP vector is split into two parts prior quantization: $\boldsymbol{x}_t^{(l)}$ and $\boldsymbol{x}_t^{(h)}$ which include the first five and the remaining five LSP parameters, respectively.

4.1.1 Conventional Method

For comparison purpose, split VQ (SVQ) is used to quantize the vector $\boldsymbol{x}_t^{(l)}$ and $\boldsymbol{x}_t^{(h)}$ with the weighted Euclidean distance [8]. The codebooks are trained by the LBG algorithm.

4.1.2 Proposed Method

In the proposed method, the same transform matrix $\boldsymbol{w}$ as Eq. (5) is applied to the input vectors, where $L_- = 1$ and $L_+ = 0$. The weighting parameter of g in the matrix is set to $1/2$ by informal listening tests. This value gave higher performance than other values of g in the informal tests. The transformed vectors $\boldsymbol{z}_t^{(l)}$ and $\boldsymbol{z}_t^{(h)}$ are quantized with the Euclidean distance. Note that, in this case, the dimension of the quantized vectors, N, equals ten. It is also noted that, since $D_{enc} = L_{+} = 0$, the frame delay is caused only by the decoder, i.e., $D = D_{dec}$.

The codebooks for the proposed scheme are generated in the following procedure. The training LSP vectors are transformed by the matrix $\boldsymbol{w}$. Using the transformed vectors, the LBG algorithm is performed to generate the mean vectors of the codebook. After the transformed vectors are encoded by the Euclidean distance, the diagonal covariance matrix for each cell is calculated from the mean vector and the vectors assigned to the cell.

4.2 LSP Trajectories, Spectra and Spectral Steps

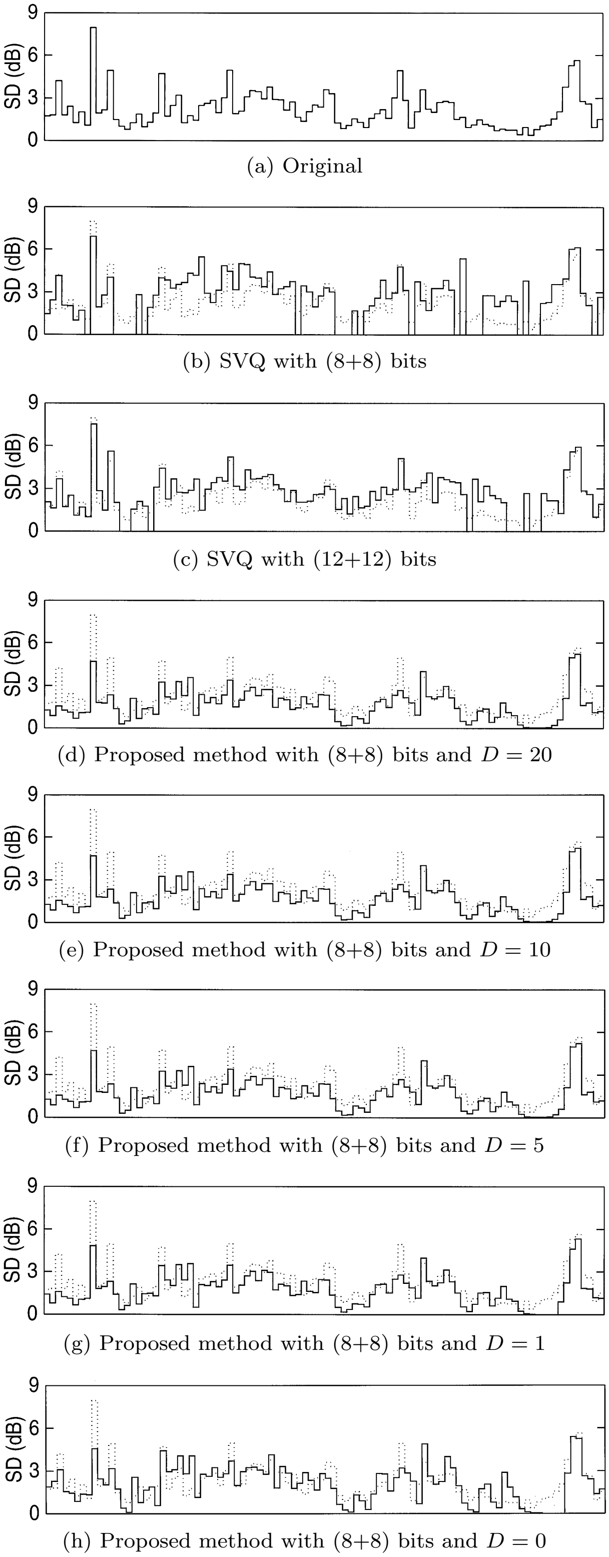
Figure 1 shows the trajectories of the LSP parameters and the corresponding spectra. In the figure, (A+B) indicates that A and B bits are allocated to the first and second splits, respectively. It is seen that the trajectroies of the conventional quantizers move stepwise, especially for lower bit rate. On the other hand, the trajectories are significantly smoother in the proposed method; accordingly the spectra of the proposed quantizers change smoothly. It seems that, in the proposed scheme, the system with a larger delay generates smoother spectra.

Figure 2 shows the spectral steps before and after quantization to verify the spectral smoothness. The spectral steps are obtained from the same speech as Fig. 1 and calculated from the RMS log-spectral difference between adjacent frames. It is shown from the figure that, in the regions where the original spectral steps change slowly, rapid changes occur in the steps of the SVQ. This is due to the fact that the same index is often used in successive frames. On the other hand, the spectral steps of the proposed method can be seen to mimic the steps of the original, especially in the steady-state regions.

4.3 Listening Test

A DMOS test was conducted to evaluate the subjective performance of the proposed method. Six persons listened with headphone to four sentences uttered by two male and two female speakers. The reference signal is the original speech, and the test signals are the speech obtained with the seven quantizers in Fig. 1. The test speech was generated in the following way. The LP-residual was computed using the unquantized LP coefficients. The test speech was reconstructed by exciting the quantized synthesis filter with the residual. The synthesis filter used the LP coefficients which are converted from the quantized LSP parameters. The LSP parameters were not interpolated, i.e., they were updated every 10-msec frame.

Figure 3 shows the test result including DMOS values and confidence intervals (95%). It is clear that, if the frame delay is allowed, the proposed quantizers with (8+8) bits achieve a significant improvement over the



(a) Original

(b) SVQ with (8+8) bits

(c) SVQ with (12+12) bits

(d) Proposed method with (8+8) bits and $D = 20$

(e) Proposed method with (8+8) bits and $D = 10$

(f) Proposed method with (8+8) bits and $D = 5$

(g) Proposed method with (8+8) bits and $D = 1$

(h) Proposed method with (8+8) bits and $D = 0$

Fig. 2 Spectral steps of adjacent frames. The dotted lines correspond to the original spectral steps.

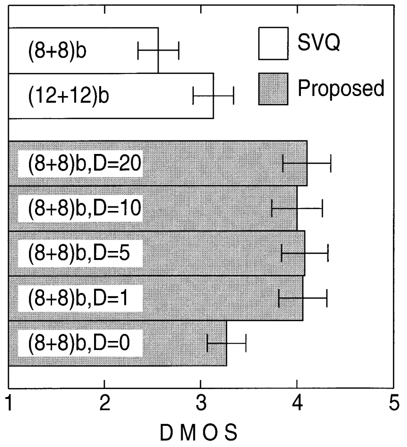


Fig. 3 Results of subjective test.

SVQ with (12+12) bits. The proposed quantizer with no delay is found to slightly outperform the SVQ with (12+12) bits. We have observed from listening that the reconstructed speech is quite smooth and natural in the proposed scheme, while discontinuity is annoying in the conventional SVQ.

Another interesting result is that there is no significant difference in subjective performance among the proposed quantizers with frame delay ($D > 0$). This result suggests that the smooth spectra can be obtained with a small delay and reasonable complexity (e.g., $O(K)$ for $D = 1$).

5. Discussions

Although the performance of the recursive algorithm was not measured in the listening test, it can be considered to be the same as the performance of the time-recursive algorithm with $D = 20$. This is because the difference between the outputs of the recursive and time-recursive algorithms is negligible for a large value of D. For example, if we choose $D = 20$, the average and maximum spectral distortions are 1.1×10^{-4} dB and 5.4×10^{-2} dB, respectively, for the sentences used in the listening tests, where the parameter T of the recursive algorithm is set to be the total number of frames in one sentence.

From the test results, the proposed scheme with $D = 0$ outperforms the SVQ with (8+8) bits. One reason is to take the delta parameters into consideration to determine the best index in the encoder. Another reason is the time-recursive algorithm. While a larger value of D gives a solution closer to the optimal one, the algorithm still provides a reasonably good approximation of the optimal output vectors even if $D = 0$. In this case, the time-recursive algorithm determines the output vectors using the statistics of the static and dynamic parameters of the current and previous frames (The number of frames in the sliding window, L in Eq. (45), is 2 for $D = 0$).

The decoders in [3] and [4] also have an ability to control the dynamics of the output vectors. An advantage of the proposed scheme over those methods is the delta parameters. In the decoders of [3] and [4], the delta parameters have to be estimated from the output vectors, because they are unknown from the indices. On the other hand, dynamic features are available for the decoder of the proposed scheme, since they are transmitted together with the static features. In addition, the output vectors of the proposed scheme are reconstructed from statistics of both static and dynamic feature in such a way that the likelihood defined in Eq. (11) is maximized with respect to the output vectors. The time-recursive algorithm is one efficient way to compute output vectors based on the above criteria, and enables us to obtain the output vectors time-recursively.

As shown in Figs. 1 and 2, the proposed scheme at low bit rates is unable to follow the rapid changes in some cases such as transition frames; accordingly the intelligibility of the output speech may degrade. The reason is that, as the bit rate is lower, the codebooks contain fewer entries for describing rapid changes (because the number of frames with rapid changes is small in speech). However this situation will be improved at higher bit rates, since the codebooks accommodate more entries for representing rapid changes, e.g., some entries have large values of delta parameters.

In the experiments, we consistently used the parameters $L_{-} = 1, L_{+} = 0$ and transform matrix $\boldsymbol{w}$ defined in Eq. (5). The improved performance of the proposed method could be achieved by larger values of L_{-} and L_{+}. Further research is needed to find the best setting.

6. Conclusions

This paper presented a VQ scheme using statistics of linear transform of consecutive input vectors. The proposed scheme was applied to LSP parameter quantization, in which our focus was on controlling the dynamics of the quantized parameters. It was shown that the proposed method can generate smoothly varying spectra and improve the subjective quality of the synthesized speech. These results indicated that the proposed method has the ability to appropriately control the dynamics using statistical information.

One aspect of the proposed scheme was revealed in this paper. We believe that the proposed scheme is useful in other situations, such as image coding and waveform coding, by selecting the linear transformation function according to the input characteristics.

Acknowledgment

The authors would like to thank Prof. Satoshi Imai, Chiba Institute of Technology, for his valuable discus-

sions.

This work was supported in part by Research Fellowships of the Japan Society for the Promotion of Science for Young Scientists, and in part by Support for International Communication Research of the International Communication Foundation.

References

[1] K.K. Paliwal and W.B. Kleijn, "Quantization of LPC parameters," in Speech Coding and Synthesis, pp.433–466, Elsevier, 1995.
[2] W.B. Kleijn and R. Hagen, "On memoryless quantization in speech coding," IEEE Signal Processing Letters, vol.3, no.8, pp.228–230, Aug. 1996.
[3] H.P. Knagenhjelm and W.B. Kleijn, "Spectral dynamics is more important than spectral distortion," Proc. International Conference on Acoustics, Speech and Signal Processing, vol.1, pp.732–735, 1995.
[4] J. Samuelsson, J. Skoglund, and J. Lindén, "Controlling spectral dynamics in LPC quantization for perceptual enhancement," Proc. 31st Asilomar Conference on Signal, Systems and Computers, vol.2, pp.1066–1070, Nov. 1997.
[5] K. Tokuda, T. Masuko, T. Kobayashi, and S. Imai, "Quantization of vector sequence using statistics of neighboring input vectors," Proc. 3rd Joint Meeting of ASA and ASJ, pp.1067–1072, 1996.
[6] K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi, "Vector quantization of speech spectral parameters using statistics of dynamic features," Proc. International Conference on Speech Processing, pp.247–252, 1997.
[7] S. Haykin, Adaptive filter theory, Prentice-Hall, Englewood Cliffs, N.J., 1991.
[8] H. Ohmuro, K. Mano, and T. Moriya, "Vector-matrix quantization of LSP parameters," IEICE Technical Report, SP91-70, 1991.

Kazuhito Koishida received the B.E. degree in electrical and electronic engineering, and M.E., and Dr.Eng. degrees in intelligence science from Tokyo Institute of Technology, Tokyo, Japan, in 1994, 1995 and 1998, respectively. Since 1996, he has been a research fellow of the Japan Society for the Promotion of Science. He is currently a post-doctoral researcher with Signal Compression Laboratory, University of California, Santa Barbara. His current research interests include speech coding at medium and low bit rates and wideband speech/audio coding. He is a recipient of the 1998 TELECOM System Technology Prize for Student from the Telecommunications Advancement Foundation Award. He is a member of IEEE and ASJ.

Keiichi Tokuda was born in Nagoya, Japan, in 1960. He received the B.E. degree in electrical and electronic engineering from the Nagoya Institute of Technology, Nagoya, Japan, the M.E. and Dr.Eng. degrees in information processing from the Tokyo Institute of Technology, Tokyo, Japan, in 1984, 1986, and 1989, respectively. From 1989 to 1996 he was a Research Associate at the Department of Electronic and Electric Engineering, Tokyo Institute of Technology. Since 1996 he has been with the Department of Computer Science, Nagoya Institute of Technology as Associate Professor. His research interests include speech coding, speech synthesis and recognition and multimodal signal processing. He is a member of IEEE, ASJ and JSAI.

Takashi Masuko received the B.E. degree in computer science, and M.E. degrees in intelligence science from Tokyo Institute of Technology, Tokyo, Japan, in 1993 and 1995, respectively. In 1995, he joined the Precision and Intelligence Laboratory, Tokyo Institute of Technology as a Research Associate. He is currently a Research Associate of the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama, Japan. His research interests include speech synthesis, speech recognition, speech coding, and multimodal interface. He is a member of IEEE, ISCA and ASJ.

Takao Kobayashi received the B.E. degree in electrical engineering, the M.E. and Dr.Eng. degrees in information processing from Tokyo Institute of Technology, Tokyo, Japan, in 1977, 1979, and 1982, respectively. In 1982, he joined the Research Laboratory of Precision Machinery and Electronics, Tokyo Institute of Technology as a Research Associate. He became an Associate Professor at the same Laboratory in 1989. He is currently a Professor of the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama, Japan. His research interests include speech analysis and synthesis, speech coding, speech recognition, and multimodal interface. He is a member of IEEE, ISCA, IPSJ and ASJ.