A NEW SEGMENT QUANTIZER FOR LINE SPECTRAL FREQUENCIES USING LEMPEL-ZIV ALGORITHM

Minoru KOHATA[1], *Motoyuki SUZUKI*[2], *and Shozo MAKINO*[2]

Chiba Institute of Technology, Japan. kohata@net.it-chiba.ac.jp [1]
Graduate School of Engineering, Tohoku University, Japan. [2]

ABSTRACT

This paper presents a new segment quantization method, which is based upon Lempel-Ziv algorithm, and it is applied to quantize LSF segments of speech signal. The proposed quantizer is compared with other practical coders such as the G.729 and the MELP, by substituting those original LSF quantizers with the proposed segment quantizer. As a result, at the same cepstral distortion, nine bits per frame were saved compared with the scalar quantizer of the MELP (25 bits/frame), and three bits per frame were saved compared with the MA prediction VQ of the G.729 (18bits/frame), respectively.

1. INTRODUCTION

A low bit-rate coder below 2400 bps is still necessary to develop even if the bandwidth is becoming broad. In such a coder, it is quite difficult to preserve quality, and it is also difficult to allocate sufficient bits to the spectral parameters such as Line Spectral Frequencies (LSF). The basic quantization method of the spectral parameters is scalar quantization. The scalar quantization of ten-dimensional LSF requires about 30 to 40 bits per frame to obtain transparent quality. It is obvious that this rate is a burden to low bit-rate coders.

Then, a vector quantizer (VQ) was proposed to use for speech coding, which was able to reduce the bit-rate drastically. Meanwhile, we can observe temporal dependency between the spectral parameters of the nearby frames, since the short-time speech spectrum evolves slowly with time. Therefore, some VQ schemes, which utilize the temporal redundancy to reduce bit-rate further, were proposed. Those schemes are termed memory VQ or inter-frame VQ. As a current result of the investigation of the memory VQ, a fourth-order moving average (MA) predictive VQ (PVQ) included in the ITU-T Rec. G.729 can quantize a ten-dimensional LSF vector at 18 bit/frame with spectral distortion around 1.0dB.

Another simple but convenient spectrum quantization, which belongs to memory VQ, is the matrix quantization (MQ), or the segment quantization (SQ). The SQ is a simple extension of VQ to the temporal domain, and the same codebook-training algorithm as with the VQ can be used, such as the generalized Lloyd (GL) algorithm. But its performance is mostly impractical, because an efficient segmentation method for these quantizations has not been found so far, and this also makes codebook training of SQ or MQ difficult.

In this paper, a new segment quantizer is proposed based on the Lempel-Ziv (LZ) coding algorithm, which is one of the universal coding methods for discrete symbols. The method proposed in this paper reduces redundant information which lies in inter frames of LSF vectors by using a segmentation scheme, which is similar to the incremental parsing of the LZ78 [1] algorithm. The proposed method is applied to two well-known speech coders, the FS-MELP and the ITU-T Rec. G.729 CS-ACELP. Then, the speech synthesized by these coder's internal quantizers and by the proposed method are compared.

2. MODIFICATION OF LEMPEL-ZIV ALGORITHM

The Lempel-Ziv (LZ) coding method [1] is one of the universal coding algorithms that use a dictionary. In the LZ coding process, input discrete symbols are decomposed by matching them with a formerly decomposed sequence of symbols.

The LZ78 coding method, which decomposes input sequence with a segmentation scheme called "incremental parsing," is adopted in the proposed segment quantizer. The original Lempel-Ziv coding methods, including the LZ78, can be applied to only a discrete information source; therefore they cannot be directly applied to continuous information such as LSF vectors. Then, the incremental parsing in LZ78 must be modified so that it can decompose a sequence of LSF vectors.

The decomposition rule of the modified incremental parsing is as follows [6].

i) A segment in the input sequence of LSF vector is denoted by x_n^m. Where n and m represents the start and end time of the segment, respectively. The start time of the j-th segment can be denoted by $n(j)$.

ii) The first vector of the input LSF sequence x_1^1 is stored in the dictionary D, which has only the null sequence, before this step. ($D = \{\phi\}$)

$$x_1^1 \to D \tag{1}$$

iii) Modified incremental parsing: The j-th segment $x_{n(j)}^{n(j+1)-1}$, which comes from the j-th decomposition, should satisfy one the following two conditions. Here a threshold value, TH, is introduced, and $dist(\cdot)$ is defined as a distance measure.

$$\begin{aligned} &if \quad dist(x_{n(j)}^{n(j+1)-2}, x_{n(r)}^{n(r+1)-1}) < TH \qquad then \\ &\quad x_{n(j)}^{n(j+1)-1} \cong s_r a_j \qquad (2) \\ &else \\ &\quad x_{n(j)}^{n(j+1)-1} = \phi \ a_j \end{aligned}$$

where a_j represents the last LSF vector in the segment $x_{n(j)}^{n(j+1)-1}$. Eq. (2) means that the segment $x_{n(j)}^{n(j+1)-1}$ is decomposed if the distance between $x_{n(j)}^{n(j+1)-2}$ and s_r is lower than TH, where s_r is one of the segments in the current dictionary (codebook) D nearest to the $x_{n(j)}^{n(j+1)-2}$.

iv) The new segment $x_{n(j)}^{n(j+1)-1}$ is added to the dictionary in the case of "else" in Eq.(2).

$$x_{n(j)}^{n(j+1)-1} \to D \tag{3}$$

Otherwise, the code segment is modified by calculating the temporal centroid of the segments. The temporal centroid is calculated by Eq. (4).

$$s_r = \frac{c_r \cdot s_r + x_{n(j)}^{n(j+1)-2}}{c_r + 1} \quad (1 \le r \le M) \tag{4}$$

where M and c_r are the codebook size and the number of segments that have satisfied Eq. (2) for s_r till the time $n(j)-1$, respectively.

v) Terminate the process if the input LSF sequence runs out, otherwise go to step iii).

As a result, the input sequence is decomposed into indices, which correspond to segments in the dictionary, and each code segments are trained by Eq. (4).

3. LZ-SEGMENT QUANTIZER

Applying the improved incremental parsing directly to a segment quantizer is not practical, because its codebook training process is insufficient to achieve the best rate-distortion characteristics. In order to improve the rate-distortion characteristics, quantization and centroid calculation are carried out alternatively as in the generalized Lloyd algorithm [4]. In this work, at first, the initialization of the codebook is carried out by the improved incremental parsing introduced in the last section. Then, the following quantization and centroid calculation by the generalized Lloyd algorithm are iterated till the distortion converges. That is, the modified incremental parsing is used only for building an initial codebook as shown in Fig. 1. The segment quantizer, of which codebook is designed with this method is named LZ Segment Quantizer LZSQ).

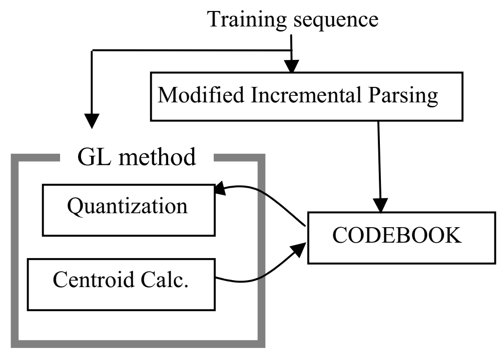

Fig.1 Codebook training for the LZ Segment quantization.

In the centroid calculation, weighted square error is used as a distance measure. It is more convenient than cepstral or spectral distortion when split quantization must be used. As a centroid calculation method for weighted square error, Eq. (5) is used.

$$\hat{s}_{ij}^r = \frac{\overline{w_{ij} x_{ij}}}{\overline{w_{ij}}} \qquad (1 \le i \le n, \quad 1 \le j \le p) \tag{5}$$

Where s_{ij}^r is j-th LSF component in the i-th frame in the r-th code segment whose length is n. p is the dimension of the LSF vector. x_{ij} and w_{ij} represents j-th LSF component in the i-th frame in the input LSF segment quantized to s^r, and the corresponding weighting coefficients, respectively.

There is another problem named "the curse of dimensionality", which is more imminent in segment quantization than in vector quantization, because a segment can be regarded as a high dimensional vector.

There are two choices to solve this problem, one is to adopt the split segment quantization (Split-SQ) and the other is to adopt the multi stage segment quantization (MS-SQ). We have compared these two methods experimentally, using the MS-SQ with two stages of the LZSQ, and the Split-SQ with three LSZQ of which dimensions are (3:3:4). As results, the MS-SQ cannot reduce the cepstral distortion lower than about 1.3 dB, while the Split-SQ can reduce it to 0.8 dB or lower.

Then, the results of the experiments for the Split-SQ only will be shown in the next section. This Split-SQ method, which uses the LZSQ to quantize each split vector, is termed Split-LZSQ.

It is expected that the mean number of times a code segment is invoked for quantization in the LZSQ has a relation to its segment length; the longer code segments are used relatively infrequently. Then, the Huffman code is applied to allocate binary code to the code segments. According to the experimental results, which will be shown in the next section, the Huffman coding can reduce the total bits of the Split-LZSQ by about two bits per frame, compared with not using the Huffman code.

4. PERFORMANCE EVALUATION

The proposed segment quantizer is evaluated by experiments in this section. Our goal is to reduce the distortion of the Split-LZSQ to a value compatible with those of coders in practical use. For example, the MELP of the federal standard[2] and the ITU-T Recommendation G.729[3], have LSF quantizers operating at 25 bit/frame and 18 bit/frame, respectively, whereas its bit-rate is lower than those of such quantizers.

Here the schemes proposed in the last section are applied to the LZSQ to decrease the distortion. First, the generalized Lloyd method is applied with weighting coefficients for LSF. In order to compare the result with the MELP or the G.729, the same weighting coefficients for these codecs are used [2][3].

Secondly, the quantization scheme with Split-SQ is evaluated. In splitting a segment, the number of the sub-segments should be fixed, and their dimensions must be arranged. We carried out several experiments to find the best combination of the number of the sub-segments and their dimensions. As results of the experiments, letting the number of sub-segments be three, and setting those dimensions as (3:3:4) was found to be the best combination. Under these conditions, the performance of the Split-LZSQ was evaluated, varying the threshold TH_k, and codebook size M_k, where k means the split segment number (k =1,2,3). It would be best to fix these parameters theoretically, but we have no valid theoretical analysis of the Split-LZSQ for the LSF quantization, so these two parameters were sampled and combined, and the experiments were carried out for each of them. If there are C_k combinations of (TH_k, M_k), which are applied to the Split-LZSQ quantizer for the k -th sub-segments, then the number of the total combinations becomes $C_1 \cdot C_2 \cdot C_3$.

Thirdly, Huffman coding is applied to the output of Split-LZSQ, by the reason described in the last section.

Finally, the Split-LZSQ codebooks are designed for the MELP and the G.729, and the segment quantization is carried out using each of them. The conditions of the experiments are described in Table I.

Table I Experimental conditions for Split-LZSQ

Sampling frequency		8 kHz
LSF order		10 (3:3:4)
Frame period	MELP	22.5ms
	G.729	10ms
Speech data	training	2502 s
	test	985 s
Codebook size $M_{1,2,3}$		2^{14}, 2^{15}, 2^{16}
$TH_{1,2,3}$		0.005 - 0.035

Figure 2 shows the result for the MELP. Each of the plotted points corresponds to one of the combinations of $\prod_{k=1,2,3}(TH_k, M_k)$. (Only 300 points are plotted, which correspond to the conditions, where the product of distortion and rate are smaller than the other combinations.) In Fig. 2, the rate and the distortion by the quantizer in the MELP are also shown. The proposed Split-LZSQ saves about 9 bit/frame compared with the MELP, assuming that it operates at the same distortion as the MELP.

Figure 3 shows the result for the G.729, which uses the MA-PVQ as the quantizer. The rate and the distortion for the MA-PVQ are also shown in this figure. This result shows that the Split-LZSQ can outperform the MA-PVQ of the G.729 by about 3 bit/frame, at a cepstral distortion of 1.05 dB. It is reported that the theoretical lower boundary of the rate of a first-order AR-PVQ is around 16 bit/frame if the spectral distortion is kept within 1.0 dB [5]. Thus, the result for the Split-LZSQ of 15 bit/frame is close to the theoretical limit of a first-order AR-PVQ, though the result for the Split-LZSQ is an experimental result.

Table II shows the evaluation in spectral distortion, where the condition of the Split-LZSQ is chosen so that its distortion is approximately the same with that of the G.729 or the MELP. Here the outlier (>2.0dB) is also measured, and the Split-LZSQ achieved better performance than the G.729.

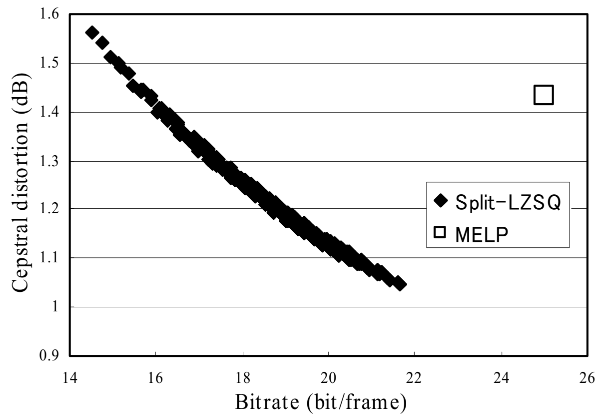

Fig. 2 Rate-distortion characteristics by Split-LZSQ for MELP coder.

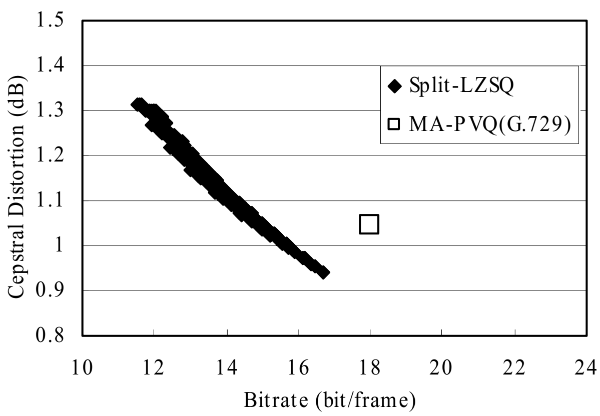

Fig. 3 Rate-distortion characteristics by Split-LZSQ for ITU G.729 coder.

Table II Spectral distortion and outlier (>2.0dB).

	G.729	Split-LZSQ	MELP	Split-LZSQ
SD (dB)	1.30	1.22	1.64	1.66
Outlier (%)	7.49	3.70	19.55	19.18
CD (dB)	1.04	1.04	1.43	1.42
bit-rate(bpf)	18	15.00	25	15.88

We carried out subjective listening tests for the synthesized speech by the Split-LZSQ, as the final evaluation of the proposed method. As the type of subjective test, the A-B comparison test was adopted, where the Split-LZSQ was set to make the same cepstral distortion as that of the MELP or the G.729. The experimental conditions are shown in Table III. If the results show that the subjective quality for the two quantizers are the same or that of the Split-LZSQ is better than the other, then we can evaluate the bit rate saved by the Split-LZSQ in Fig. 2 and Fig. 3. Table IV shows the results, and the preference score of the Split-LZSQ seems to be almost the same as that of the MELP or the G.729. There was no subjective quality degradation caused by the segment-wise quantization, which might not be observed through objective measures such as cepstral distortion.

Table III Experimental conditions of the subjective tests.

Language	Japanese
Database	ASJ continuous speech database
Talker	8 (male:4, female:4)
Condition	Clean
Listener	10

Table IV Comparison with original quantizer at the same cepstral distortion by preference tests.

(a) MELP

Quantizer	Preference score (%)
MELP	47.5
Split-LZSQ	52.5

(b) G.729

Quantizer	Preference score (%)
G.729	47.5
Split-LZSQ	52.5

5. CONCLUSION

A new segment quantizer for LSF quantization at very low bit rates is presented. By modifying incremental parsing of the LZ78 algorithm, an efficient segmentation can be realized. The experimental results for the Split-LSZQ show that its rate-distortion characteristics can attain cepstral distortion of 1 dB at a bit-rate of 15 bit/frame. It can quantize LSF with the same distortion as the MELP or the G.729, saving 9 bit/frame for the MELP, and 3 bit/frame for the G.729. Theoretical analysis of the LZSQ would be our further work.

ACKNOWLEDGEMENT

This work was supported in part by a grant from SCAT (Support Center for Advanced Telecommunications Technology Research).

REFERENCES

[1] J. Ziv and A. Lempel, "Coding Theorems for Individual Sequences," IEEE Trans. Inform. Theory, vol. IT-24, no, 4, pp.405-412, 1978.

[2] A. V. McCree, K. Truong, E. B. George, T. B. Barnwell and V. Viswanathan, "A 2.4kbit/s MLP coder candidate for the new U. S. federal standard," in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, pp.200-203, 1996.

[3] Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP), ITU-T Recommendation G.729, 1996.

[4] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Communications, vol. COM-28, pp.84-95, 1980.

[5] J. Samuelsson and P. Hedelin, "Recursive coding of spectrum parameters," IEEE Trans. Speech Audio Processing, vol.9, no.5, pp.492-503, Oct. 2001.

[6] M.Kohata, I.Mitsuya, M.Suzuki, S.Makino, "Efficient segment quantization of LSP coefficients for very low bit speech coding", Proc. International Conference on Spoken Language Processing 2000, vol.3, pp.826-829,(2000)