FAST ENCODING ALGORITHMS FOR MPEG-4 TWINVQ AUDIO TOOL

Naoki Iwakami, Takehiro Moriya, Akio Jin, Takeshi Mori, and Kazuaki Chikira

NTT Laboratories
3-9-11 Midori-cho Mosashino-shi Tokyo 180 Japan

ABSTRACT

The ISO/IEC MPEG-4 Audio standard includes the TwinVQ encoding tool. This tool is suitable for low-bit-rate general audio coding, but drawback is the computational complexity of the encoder. To develop a faster TwinVQ encoder, new fast vector quantization algorithms — area localized pre-selection and hit zone masking — are introduced. These algorithms exploit pre- and main-selection procedure scheme of the conjugate structure vector quantization which is used in the TwinVQ. The improvement is evaluated by measuring the encoding speed and the sound quality of reproduction.

1. INTRODUCTION

The ISO/IEC MPEG-4 Audio standard [1] includes a number of tools with a variety of functions, enabling a universal data format for broadcasting, movie, and multimedia applications. Transform-domain weighted interleave vector quantization (TwinVQ) [2]–[4], which is one of the MPEG-4 Audio tools, is designed for low-bit-rate general audio coding, but drawback is the computational complexity of the encoder.

In this paper we report on reducing the computational complexity of the TwinVQ encoder. The basic structure of the TwinVQ algorithm is first overviewed. The computational load of the encoder is then profiled. Then fast VQ encoding algorithms are proposed. Finally, performance improvement is evaluated.

2. OVERVIEW OF THE TWINVQ ALGORITHM

TwinVQ is a transform coder that uses modified discrete cosine transformation (MDCT) [5], as illustrated in Figure 1. The MDCT coefficients are normalized by the preprocessing module before being sent to the interleave vector quantization (VQ) module. The VQ encoding is weighted by the perceptual model to improve the quality of the reproduction.

At the interleave VQ module, the input vector is divided into sub-vectors by using the interleave and division technique [2], and all sub-vectors are encoded separately by elementary VQ modules.

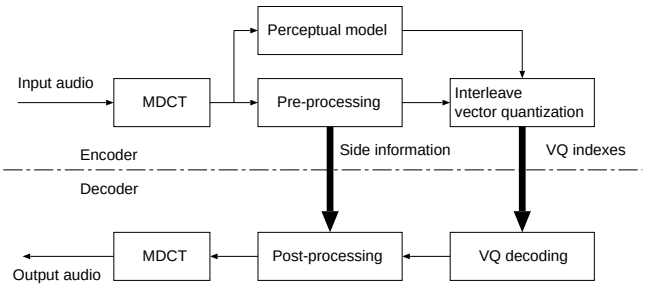


Fig. 1. Basic structure of TwinVQ

3. COMPUTATIONAL LOAD PROFILING

To identify the bottleneck of the TwinVQ encoding speed, the computational load was profiled. The encoder ran at 16 kbit/s for a 32-kHz sampling monaural audio signal input.

The measurement results are listed in Table 1. This table shows that the largest part of the computational load is the VQ procedure. New, faster VQ algorithms are thus required to speed up the TwinVQ encoder.

Table 1. Profile of computational load

module	load percentage
VQ	52.8 %
Pre-processing	16.6 %
Perceptual model	9.9 %
MDCT	6.9 %
Other	13.8 %

4. FAST VQ ENCODING ALGORITHMS

4.1. Algorithm overview

As mentioned in Section 2, the interleave VQ module is divided into elementary VQ modules. Each elementary VQ

module uses a conjugate-structure vector quantization (CSVQ) scheme [6], as shown in Figure 2. This type of VQ uses two codebooks to make a combined reproduction. The codebook size, the number of code vectors in the codebook, is 32 for each one.

In CSVQ encoding, one code vector is chosen from each codebook. The chosen pair of code vectors should be the best pair, giving the minimum quantization distortion. The most straightforward method of searching for the best pair is to calculate the distortion measure for all possible pairs. However, this method is not practical in terms of computational complexity because it requires a large number of distortion measure calculations ($32^2 = 1024$). For this reason, the CSVQ encoding algorithm includes separate pre- and main-selection procedures, as shown in Figure 3.

During pre-selection, a fixed number of candidate code vectors are chosen from a codebook. The number of candidates is less than the codebook size. The candidates chosen are those most likely to produce the best code vector. The pre-selection procedure is done twice for two codebooks.

During main-selection, all possible pairs of candidates from the two codebooks are combined, and their distortion measures are calculated. The pair giving the minimum distortion measure is chosen as the CSVQ encoding output. This main-selection procedure is similar to the straightforward method mentioned above when the number of candidates equals the codebook size. However, the calculation load for the distortion measure is reduced as the number of candidates is decreased.

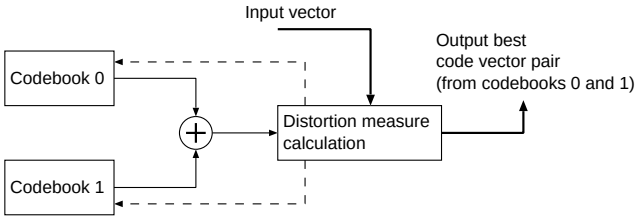


Fig. 2. Conjugate-structure VQ

4.2. Fast pre-selection algorithm

The measurement described in Section 3 shows that the computational load of CSVQ remains large even with the pre/main-selection procedure scheme. A fast pre-selection algorithm is thus proposed.

Figure 4(a) shows the conventional pre-selection method for eight candidates. In this method, the entire codebook is searched for candidates. The candidates are stored in a buffer. When updating the buffer, the insertion point of the new candidate is searched for based on the $O(\log(n))$ of the searching algorithm. Then the buffer data are shifted to create a vacancy at the insertion point. These two procedures

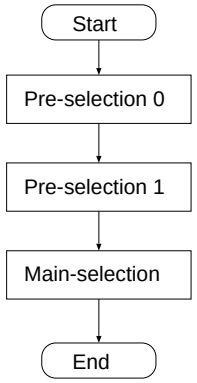
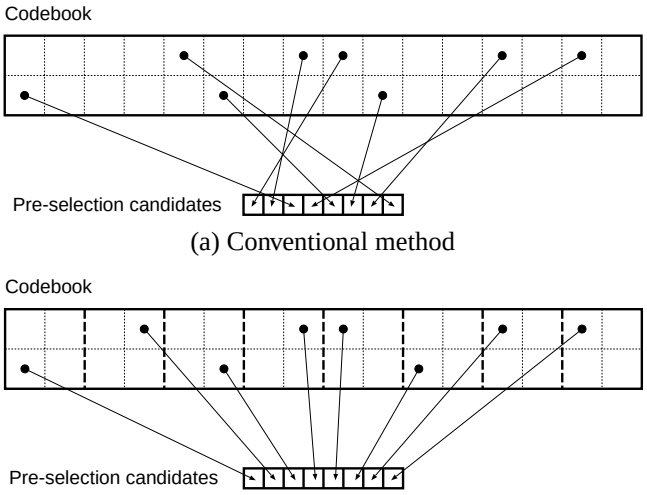


Fig. 3. CSVQ algorithm flow

are included in the computational load for pre-selection.

The proposed method, area-localized pre-selection, avoids this computation problem. In this method, the codebook is divided first. The number of the pieces is the same as the number of candidates. One candidate is chosen from each piece. This method does not require replacing data in the buffer nor changing the order in the buffer, so the computational load is smaller than that for the conventional method.



(a) Conventional method

(b) Area-localized method

Fig. 4. Pre-selection methods

4.3. Fast main-selection algorithm

In the main-selection procedure, all possible pairs of pre-selection candidates are evaluated by the distortion measure, based on the following equation:

$$d^2 = \|\vec{V_0} + \vec{V_1} - \vec{T}\|^2, \quad (1)$$

where d^2 is the square of the distortion measure, $\vec{V_0}$ and $\vec{V_1}$ are candidate code vectors from codebooks 0 and 1, respectively, and $\vec{T}$ is the target vector. The weighting by the perceptual model is omitted for simplification. Iteratively calculating equation (1) results in a high computational complexity, so the process can be sped up effectively if some of the distortion calculations are skipped, as in Figure 5(b), rather than testing all possible pairs, as in Figure 5(a).

The hit zone masking method is used to implement this skipping process. To understand this method, equation (1) is decomposed as

$$d^2 = \|(\vec{V_{0n}} + \vec{V_{0p}}) + (\vec{V_{1n}} + \vec{V_{1p}}) - \vec{T}\|^2, \qquad (2)$$

where $\vec{V_n}$ and $\vec{V_p}$ are normal and parallel to the target vector respectively, i.e.

$$\begin{aligned} \vec{V_n} + \vec{V_p} &= \vec{V} \\ \vec{V_n} &\perp \vec{T} \\ \vec{V_p} &\parallel \vec{T}. \end{aligned} \qquad (3)$$

Given that the inner product of normal vectors is 0, equation (2) can be simplified as

$$d^2 = \|\vec{V_{0n}} + \vec{V_{1n}}\|^2 + (v_{0p} + v_{1p} - t)^2, \qquad (4)$$

where $v_p = \|\vec{V_p}\|$ and $t = \|\vec{T}\|$. The first term of equation (4) is always positive, so the distortion measure is always greater than the second term:

$$d \geq |v_{0p} + v_{1p} - t|. \qquad (5)$$

The hit zone masking method exploits this characteristic. Figure 6 illustrates how this method works. In this figure, the current minimum distortion measure in the main-selection iteration loop is expressed by d_{min}. The area between the parallel hatched lines is called the "hit zone." If the sum of two candidate vectors is outside the hit zone, like vector V_{1b}, then the distortion measure calculation for this pair is skipped because there is no possibility for this pair to improve d_{min}. This hit-zone judgment can be determined by adding parallel vectors. The sum of the parallel vectors must be in the range between $t - d_{min}$ and $t + d_{min}$ to be in the hit zone. Parallel vectors can be added with low computational complexity because this is a scalar operation.

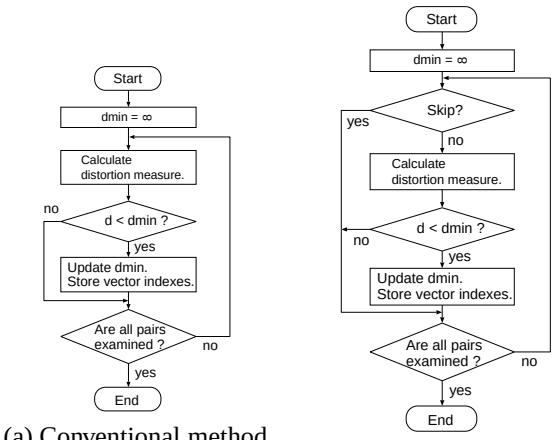
4.4. Common terms

The distortion measure for pre-selection is defined by

$$d_p^2 = \|2\vec{V} - \vec{T}\|^2 = 4\|\vec{V}\|^2 - 4\vec{V} \cdot \vec{T} + \|\vec{T}\|^2, \qquad (6)$$

where d_p is the distortion measure. On the other hand, the distortion measure for main-selection, described by equation (1), can be decomposed as

$$d^2 = \|\vec{V_0}\|^2 + \|\vec{V_1}\|^2 - 2\vec{V_0} \cdot \vec{T} - 2\vec{V_1} \cdot \vec{T} + 2\vec{V_0} \cdot \vec{V_1} + \|\vec{T}\|^2. \qquad (7)$$



(a) Conventional method (b) Skipping method

Fig. 5. Main-selection methods

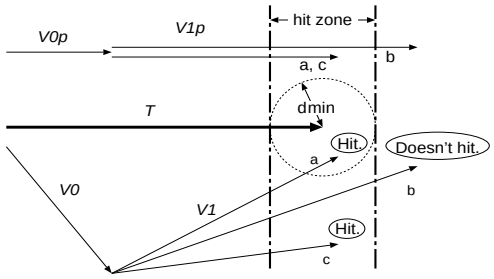


Fig. 6. Hit zone masking method

The first four terms of equation (7) are already calculated for equation (6) during pre-selection. Therefore, these terms can be used in common by both pre- and main-selection to reduce the computational complexity.

Furthermore, the last term of each equation, $\vec{T}$, does not need to be calculated, because it is constant throughout the code vector selection process.

5. EVALUATION

5.1. Encoding time

The encoding time of the new algorithm was compared with that of the original algorithm by using an Intel Pentium III 733-MHz CPU. The input was 16.8 seconds of monaural signal sampled at 32 kHz, and the coding bit rate was 16 kbit/s. The numbers of pre-selection candidates were 2, 4, 8, and 16. The original algorithm was from the MPEG-4 reference software.

Table 2 lists the results of the measurement. Ratios of input signal length to encoding time are also listed as encoding speeds. Improvement of encoding speed was achieved, especially with larger number of candidates. With 16 candidates, the encoding speed improved by over 75 percent.

Table 2. Comparison of encoding speed

num.	original		new		improve-
cand.	time	ratio	time	ratio	ment
2	5.11s	3.29	4.21s	3.99	21.3%
4	5.24s	3.21	4.27s	3.93	22.4%
8	5.91s	2.84	4.34s	3.87	36.3%
16	8.13s	2.07	4.62s	3.64	75.8%

Table 3. Profile of computational load for new encoder

module	load percentage
CSVQ	19.0 %
Pre-processing	27.9 %
Perceptual model	16.8 %
MDCT	11.5 %
Other	24.8 %

Table 3 shows the comptational load profile for the new encoder with 16 pre-selection candidates. Comparing with Table 1 shows that the new VQ module reqires a smaller part of the total computational load.

5.2. Quality of reproduction

Introducing fast algorithms increases the risk of degraded sound quality. To evaluate changes in sound quality, a listening test was carried out with 21 listeners. Both the new and the original encoder were tested for comparison. The encoding conditions were the same as for the test in Section 5.1. Seven input signals were used. Each signal was played once for the listeners. The signals were evaluated by the degradation category rating (DCR) method. In this method, listeners hear a reference sound followed by a test sound. They then mark a quality score according to the following guidelines:

5: Excellent
4: Good
3: Fair
2: Poor
1: Bad

Figure 7 shows the results of the test. The average scores for each signal are plotted on the graph with error bars indicating 95% confidence intervals. As the graph shows, there was no significant degradation of quality by using the new encoder.

6. CONCLUSION

We considered how to speed up the MPEG-4 TwinVQ Audio tool. Profiling its computational load showed that the largest part of the load was for the VQ module. The VQ encoding algorithm used by TwinVQ has a pre- and main-selection procedure flow. Two fast vector selection methods — area localized pre-selection and hit zone masking — were introduced, and common terms were used to calculate the distortion measures for both pre- and main-selection. Due to these algorithm improvements, the new encoder ran over 75 percent faster than the original MPEG-4 reference software. A subjective listening test was also done to verify that there was no significant degradation of reproduction quality.

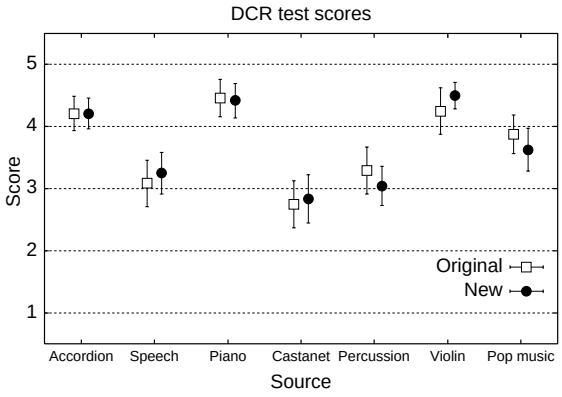


Fig. 7. Listening test results

7. REFERENCES

[1] ISO/IEC JTC1/SC29/WG11 MPEG, International Standard ISO/IEC 14496-3, *Generic Coding of Moving Pictures and Associated Audio: Audio*, 1999.

[2] N. Iwakami, T. Moriya, and S. Miki, "High-quality audio coding at less than 64 kbit/s by using transform-domain weighted interleave vector quantization (twinvq)," in *Proc. ICASSP'95*, 1995, pp. 937–940.

[3] N. Iwakami and T. Moriya, "Transform-domain weighted interleave vector quantization (twinvq)," in *AES 101st Convention preprint*, 1996, vol. 4377.

[4] J. Herre, E. Allamanche, K. Brandenburg, M. Dietz, B. Teichmann, B. Grill, A. Jin, T. Moriya, N. Iwakami, T. Norimatsu, M. Tsushima, and T. Ishikawa, "The integrated filterbank based scalable mpeg-4 audio coder," in *AES 105th Convention preprint*, 1998, vol. 4810.

[5] J. Princen, A. Johnson, and A. Bradley, "Adaptive transform coding incorporating time domain aliasing cancellation," *Speech Commun.*, vol. 6, pp. 299–308, 1982.

[6] T. Moriya, "Two-channel conjugate vector quantizer for noisy channel speech coding," *IEEE JSAC*, vol. 10, no. 5, pp. 866–874, 1992.