DATA EMBEDDING IN MPEG-1/AUDIO LAYER II COMPRESSED DOMAIN USING SIDE INFORMATION

Akihiro Matsuoka, Kiyoshi Tanaka*, Akio Yoneyama** and Yasuyuki Nakajima***

* Faculty of Engineering, Shinshu University, Japan
** KDDI R&D Laboratories Inc, Japan

ABSTRACT

In this work, we propose a data embedding scheme in MPEG-1/Audio Layer II compressed domain. Data embedding is conducted every AAU by using side information (location of sub-band allocated audio signal) as a data carrier. In general, non-zero signals concentrates in low and middle frequency bands. Therefore we utilize sub-bands that are not allocated audio signal in high frequency bands to embed information. The proposed scheme can increase payload while achieving rewritable (reversible) data embedding by choosing appropriate parameter. We verify the basic performance of our scheme through computer simulation by using some voice and music signals.

1. INTRODUCTION

Recently, data hiding techniques [1]-[3] have been widely investigated for various media such as still images, moving pictures, documents, audio, 3D polygonal data, and so on. The information embedded can be used for various purposes such as copyright protection, forgery detection, authentication, copy control, annotation, indexing, and so on. In this work, we focus on audio data hiding. Although many audio data hiding methods were proposed in a recent decade [4]-[7], there are few proposals for annotation and indexing purpose for audio signals. Our motivation here is to propose a method to embed information in MPEG-1/Audio Layer II compressed domain for streaming applications together with MPEG video. In this work, we design embedding method to gain payload as much as possible in rewritable (reversible) form for annotation or indexing purpose [8], while keeping complete compatibility to general audio decoder. In order to accomplish this goal, we pay attention to a statistical property of audio signals in AAU (Audio Access Unit: 1152 samples) that non-zero signals concentrates in low and middle frequency bands and the case allocated no audio signal increases in high frequency bands. We utilize sub-bands that are not allocated audio signal in high frequency bands to embed information. Consequently, the proposed scheme can increase payload (amount of embedded information) while achieving rewritable (reversible) data embedding by choosing appropriate parameter. We verify the basic performance of our scheme through computer simulation by using some voice and music signals.

2. MPEG-1/AUDIO

MPEG-1/Audio contains three kinds of coding algorithm called Layer I, Layer II, and Layer III. Higher layer algorithm achieves high audio quality and high compression ratio, but requires larger hardware configuration. In this work, we focus on Layer II algorithm, which is often used in MPEG-2 encoders for PC, HDD recorder, and DVD. **Fig.1** shows the block diagram of the coding architecture of MPEG-1/Audio Layer II. The input PCM signals linearly quantized (16 bits) are decomposed into 32 pieces of frequency component in sub-band filter bank, which is implemented by PFB (Polyphase Filter Bank) with 512 taps. A scale factor is calculated for each sub-band to normalize the signals obtained. The calculation of scale factor is basically conducted every 384 samples (12 samples × 32 sub-bands). However, three factors for 1152 samples in Layer II (384 samples × 3 blocks) are replaced into a single value for each sub-band. The masking level is also calculated to achieve dynamical bit allocation for each sub-band signal through the psycho-acoustic model based on the results by FFT for input signals. Each sub-band signal subtracted a masking level is quantized into a bit numbers dynamically determined. Finally, the quantized signals and side information (scale factors and allocated bits) for all sub-bands are combined (formatted) into a single bitstream with header information. Our data embedding scheme realized in the process of side-information coding in **Fig.1**.

On the other hand, in the decoder, almost inverse operation in the encoder is performed to playback the encoded audio signals. A bitstream is decomposed into header information, side information and quantized sub-band signals. The quantized sub-band signals are de-quantized by their allocation bits included in side-information, and synthesized in the filter bank to reconstruct audio signals.

3. PROPOSED DATA EMBEDDING SCHEME

3.1. Statistical Property of Audio Signals in AAU

Here we observe in detail the statistics of bit allocation for each sub-band in AAU. **Fig.2** shows the average bits allocated in each sub-band, where we used an encoded music "Pops" (sampling frequency: $F_s = 44.1$[kHz] and coding rate: $C_R = 192$[kbps]). From this figure, we can see that

1-4244-0367-7/06/$20.00 ©2006 IEEE

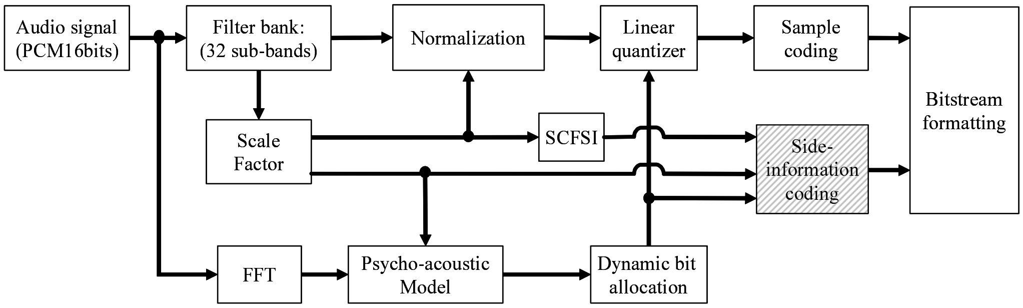


Fig. 1. Coding architecture of MPEG-1/Audio Layer II

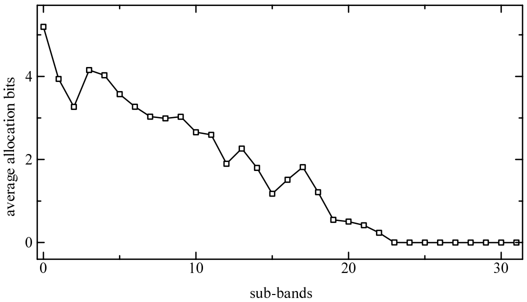


Fig. 2. Average bits allocated in each sub-band (“Pops”, $F_s = 44.1$[kHz], $C_R = 192$[kbps])

greater part of bits are allocated to low and middle frequency bands, and the number of bits allocated in higher frequency bands are significantly decreased as the location of sub-band approaches to the last sub-band P_{sb}. In this coding condition, almost no information is allocated in high frequency bands (after 20th sub-band). **Table 1** shows a typical example of allocated bits for all sub-bands for an AAU of “Pops” in the same coding condition, where available maximum bits are included as well. We can see that non-zero signals appeared under the 15th sub-band, and no signal is allocated hereafter (from the 16th to the final one P_{sb}). We utilize this statistical property on bit allocation in AAU to embed information in the following sections.

3.2. Basic Scheme

First we explain the basic scheme to embed a single bit into an AAU by using **Fig.3**.

Embedding Procedure

Step1: Find two sub-bands having non-zero signal from the last sub-band P_{sb} in descendant order. Denote the first sub-band and the second one as P_e and P_b, respectively. Also, denote corresponding two non-zero signals as V_e and V_b, respectively.

Step2: Determine the location (threshold) T to detect an embedded bit by

$$T = \lfloor \frac{P_b + P_{sb} + 1}{2} \rfloor, \tag{1}$$

where $\lfloor x \rfloor$ means rounding off operation to a nearest integer.

Step3: Determine a pair of sub-bands (P_e, P'_e) so that the distance D between P_e and P_b is equivalent to D' between P'_e and T as shown in **Fig.3**.

$$\begin{cases} D = P_e - P_b,\ P'_e = T + D\ (P_e \leq T) \\ D' = P_e - T,\ P'_e = P_e,\ P_e = P_b + D'\ (P_e > T) \end{cases} \tag{2}$$

Step4: Compare two maximum bits allowable, M_e and M'_e at P_e and P'_e, and set

$$\begin{cases} sub = 1\ (M_e = M'_e) \\ sub = 2\ (M_e \neq M'_e). \end{cases} \tag{3}$$

In this step, if $V_e > 2^{M'_e} - sub$ as shown in **Fig.4**, skip embedding and go to the next AAU. If $P'_e > P_{sb}$, also skip embedding and go to the next AAU. Otherwise, move to *Step5* to embed a single bit.

Step5: Extract a single bit $w_k (k = 1, 2, \cdots)$ from a message W, and embed it as follows:

$$\begin{cases} \text{(i) If } w_k = 0, \text{set } V_e \text{ at } P_e \text{ and } 0 \text{ at } P'_e. \\ \text{(ii) If } w_k = 1, \text{set } V_e \text{ at } P'_e \text{ and } 0 \text{ at } P_e. \end{cases}$$

That is, a single bit means which side contains non-zero signal V_e when we separate sub-bands by using T as a boundary in our scheme.

Step6: Encode bits allocated to each sub-band. Note that in case that the original audio signal exists at P'_e and no signal at P_e, the original signal V'_e is lost by data embedding because we assume that $V_e \neq 0$ and $V'_e = 0$.

Detection Procedure

Step1: Find P_e, P_b and V_e similar to the embedding procedure.

Step2: Determine the location (threshold) T to detect the embedded bit by Eq.(1).

Step3: Determine a pair of sub-bands (P_e, P'_e) by Eq.(2). Calculate sub by Eq.(3), and if $V_e > 2^{M'_e} - sub$, skip detection and go to the next AAU. If $P'_e > P_{sb}$, also skip detection and go to the next AAU. Otherwise, move to *Step4* to detect a single bit.

Step4: Detect a single bit $w_k (k = 1, 2, \cdots)$ as follows:

$$\begin{cases} \text{(i) If } P_e \leq T,\ w_k = 0. \\ \text{(ii) If } P_e > T,\ w_k = 1. \end{cases}$$

Step5: In case of (ii), set V_e at P_e and 0 at P'_e to reconstruct the original code. This procedure is repeated for all AAU included in an audio file, and all bits detected $w_k (k = 1, 2, \cdots)$ are reconstructed as an entire message W.

Table 1. A typical example of bit allocation for all sub-bands

sub-band	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
maximum bits	15	15	15	15	15	15	15	15	15	15	15	7	7	7	7	7
allocated bits	3	3	3	3	3	2	2	2	1	2	1	1	0	1	0	1
sub-band	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
maximum bits	7	7	7	7	7	7	7	3	3	3	3	0	0	0	0	0
allocated bits	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

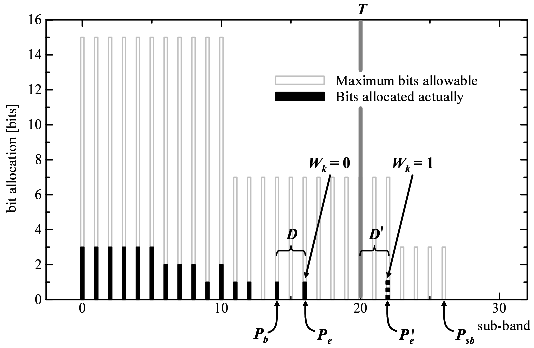


Fig. 3. The location of sub-bands found and threshold used for data embedding

3.3. Extension to Embed Multiple Bits

Here we extend the basic scheme so that we can embed multiple (m) bits in an AAU. In order to accomplish this, we find $m + 1$ pieces of non-zero audio signals P_b and $P_{ei}(i = 1, 2, \cdots, m)$ from the last sub-band P_{sb} in descendant order. Then we create m pieces of sub-band pair (P_{ei}, P'_{ei}) in inverse order $(i = m, m - 1, \cdots, 1)$ satisfying that the distances D_i between P_{ei} and P_b are equivalent to D'_i between P'_{ei} and T obtained by Eq.(1). Depending on the value of a single bit $w_k(k = 1, 2, \cdots)$, we set V_{ei} at P_{ei} or P'_{ei} similar to *Step5* in the embedding procedure in **3.1**. The conditions to skip embedding are also similar to the basic scheme. It is obvious that we can increase payload as we increment m. However, it may introduce the case where P'_{ei} collides with P_{ej} $(i \neq j)$ as shown in **Fig.5**. In this example, the location of P'_{e3} is exactly the same as the one of P_{e1}. Therefore we can not move V_{e3} to P'_{e3}, and thus we must skip embedding. Note that such collision may happen in some pairs of (P_{ei}, P'_{ei}). But it can be detected, and therefore data detection process is not affected.

4. EXPERIMENTAL RESULTS AND DISCUSSION

In our computer simulation, we used four kinds of audio data, (i) Male voice (ii) Female voice, (iii) Pops music, and (vi) Jazz music. Because of the limited space, we show the results only when the sampling frequency is fixed to $F_s = 44.1$[kHz] and the coding rate is changed in the range of $C_R = \{128, 192, 256, 320\}$[kbps] for (ii) and (iii). First, we show the payload (average bits embedded per AAU) in **Fig.6**. From this result, we can see that the payload for $C_R = \{128, 192\}$[kbps] linearly increases while the one for $C_R = \{256, 320\}$[kbps] saturates as we increment parameter m. This can be explained with **Fig.7** showing the ratio of skipped data embedding for all possible sub-band pairs (P_{ei}, P'_{ei}). From this result, the ratio of skipping noticeably increases as we increment m especially in case of higher coding rate. Next, we show the ratio of lost sub-bands by data embedding for all possible sub-band pairs in **Fig.8**. From this figure, it can be seen that the ratio of lost sub-bands is very small. In the case of female voice (**Fig.8(a)**), completely reversible data embedding (no lost sub-bands) has been achieved. Even in the case of pop music (**Fig.8(b)**), we can achieve rewritable (reversible) data embedding if we carefully choose parameter m. Finally, **Fig.9** shows some examples of the amplitude difference for data embedded one (**Fig.9(a)** and **(c)**), and for the reconstructed (i.e. after embedded data removal) one (**Fig.9(b)** and **(d)**) when compared with the original. These results are obtained for an AAU in (ii) and (iii) for $m = 10$ (maximum setting). We can see almost distortion free signals for both data embedded and reconstructed audio in the case of voice data, whereas very large difference can be found for data embedded audio in the case of music data due to the existence of high frequency components of music (**Fig.9(c)**). However, it can be also found that this distortion can be significantly removed

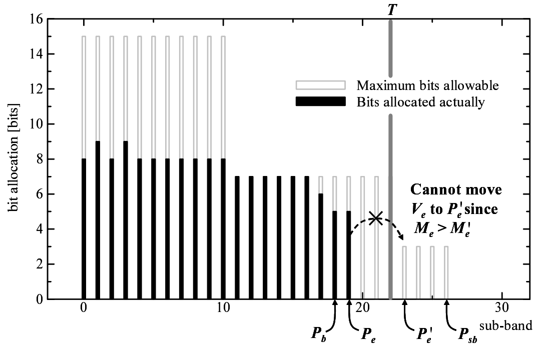


Fig. 4. Skip embedding because of maximum bits allowable

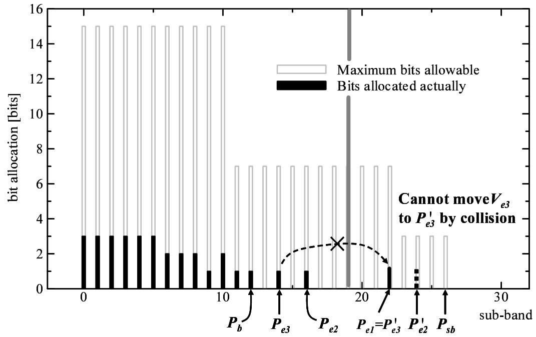
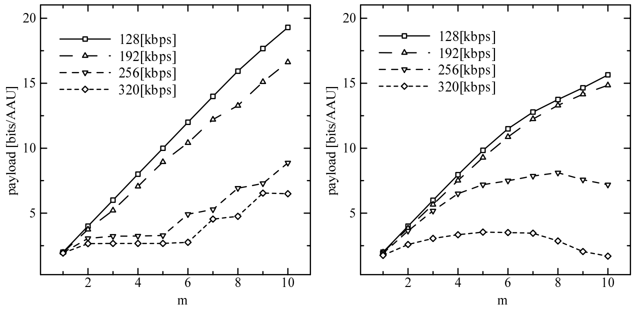
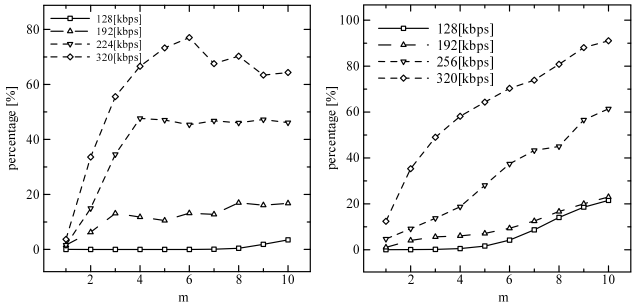


Fig. 5. Skip embedding because of "collision"



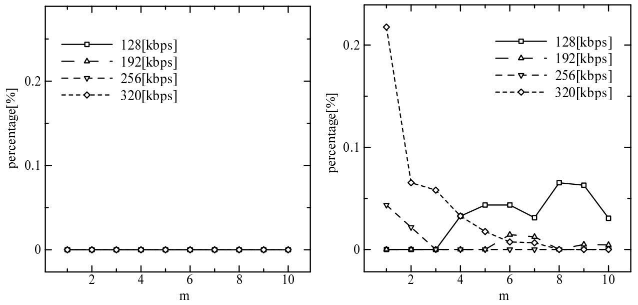
(a) Female voice (b) Pops music

Fig. 6. Payload for parameter m ($F_s = 44.1$[kHz])



(a) Female voice (b) Pops music

Fig. 7. Ratio of skiped data embedding



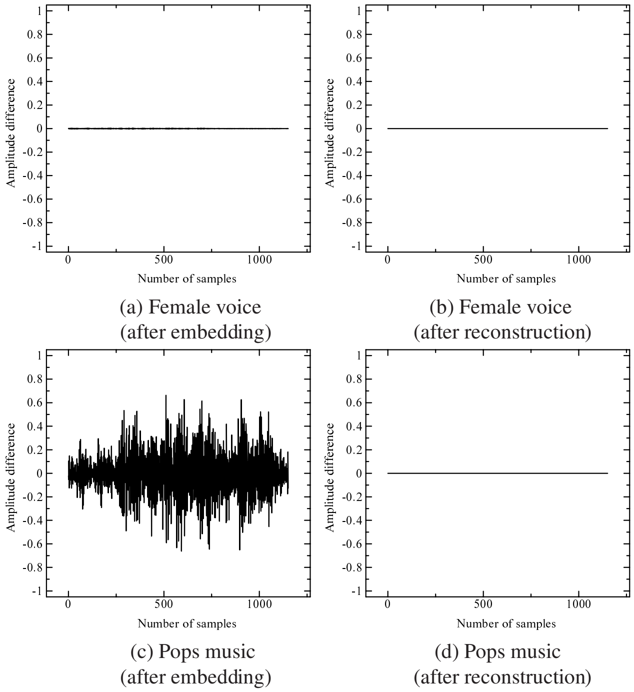
(a) Female voice (b) Pops music

Fig. 8. Ratio of lost sub-bands

after the removal of embedded data (**Fig.9(d)**). Due to the above observation, we can consider two scenarios to utilize our scheme: (i) contorl the payload so that a normal decoder can playback the data embeded MPEG audio without any noticeable degradation when compared with the original MPEG audio, (ii) provide two different types of service quality where the normal decoder can playback data embeded MPEG audio at medium quality and the premium decoder can playback it at high quality by reconstructing the original MPEG audio.

5. CONCLUSIONS

In this work, we have proposed a data embedding scheme in MPEG-1/Audio Layer II compressed domain, which uses the location of sub-bands allocated audio signal in side information. As a result, we can embed about 20[bits] per second (11.5[kbits] for $F_s = 44.1$[kHz] and $C_R = 128$[kbps]) for voice data while keeping rewritable (reversible) form (all original signals are reconstructed by removal of embedded data).



(a) Female voice
(after embedding)

(b) Female voice
(after reconstruction)

(c) Pops music
(after embedding)

(d) Pops music
(after reconstruction)

Fig. 9. Differential signals between original and processed signals

As future works, we should further improve this scheme especially for some sampling frequencies and coding rates that are relatively difficult to embed information. Also, we should investigate another embedding scheme in the process of sample coding in **Fig.1**.

6. REFERENCES

[1] K. Matsui, *Fundamentals of Digital Watermark*, Morikita Publishing, 1998 (in Japanese).

[2] S. Katzenbeisser, F. A. P. Petitcolas, *Information Hiding: Techniques for Steganography and Digital Watermarking*, Artech House, 2000.

[3] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann, 2002.

[4] H. Malik, S. Khokhar, A. Rashid, "Robust audio watermarking using frequency selective spread spectrum theory", *Proc. IEEE ICASSP2004*, vol.V, pp.385-388.

[5] A. N. Lemma, J. Aprea, W. Oomen, L. van de Kerkhof, "A temporal domain audio watermarking technique", *IEEE Trans. Signal Processing*, Vol 51, pp.1088-1097, 2003.

[6] Foo Say Wei, Ho Shuet Mun, Ng Ling Mei "Audio watermarking using time-frequency compression expansion", *Proc. IEEE ISCAS2004*, Vol.III, pp.201-204.

[7] Ching-Te Wang, Tung-Shou Chen, Wen-Hung Chao, "A new audio watermarking based on modified discrete cosine transform of MPEG/audio layer III", *Proc. IEEE Int'l Conf. on Networking, Sensing and Control*, Vol.II, pp.984-989.

[8] K. Nakajima, K. Tanaka, T. Matsuoka and Y. Nakajima, "Rewritable Data Embedding on MPEG Coded Data Domain", *Proc. IEEE ICME2005*, in CD-ROM.