A Scrambling Technique Embedding Soundtracks into Videos for Streaming Media

Xuefei Li, Yasutoshi Miura, Seok Kang, Yuji Sakamoto
Graduate School of Information Science and Technology, Hokkaido University
N14, W9, Kita-ku, Sapporo, 060-0814, Japan
Email: lee@ist.hokudai.ac.jp

Abstract—**In these years, streaming media have become very popular. In order to protect the legal copyright of streaming media, DRM (Digital Right Management) techniques are utilized widely. However, encryption, the most representative DRM technique, is difficult to install into real-time communication systems. In addition, existing DRM techniques for real-time communication are not supported by several widely-used web browsers. Scrambling, as a substitute to encryption, is suitable for real-time communication. In this paper, we propose a scrambling technique embedding soundtracks into videos. In the proposed technique, we embed audio data into quantized discrete cosine transform (qDCT) coefficients of the video data and generate scrambling at the same time during the video encoding process. After transmission, the streaming media data can be recovered during decoding process. We conducted experiments to investigate proper parameters to make the technique effective. Using the determined parameters, we conducted verification experiments to prove the effectiveness of the proposed technique.**

Keywords—scrambling; data hiding; DCT; streaming media; real-time communication

I. Introduction

Nowadays, the Internet has significantly changed our lifestyle. We focus on streaming, as it is a popular technique that allows people to watch live broadcasting and convene video conferences smoothly on the Internet.

The majority of streaming media is video and sound data. Most streaming media are copyrighted. In order to protect legal copyright of video data, DRM techniques are utilized widely. As the most representative DRM technique, encryption has been studied actively. Several encryption techniques for videos have been proposed[1]. In such techniques, video data are encrypted and decrypted integrally to protect the copyright of them. In other words, every part of the encrypted data is needed for decryption. However, streaming media are generally divided into packets to be transmitted. In general streaming system using UDP (User Datagram Protocol) based RTP (Real-time Transport Protocol)[2], real-time communication is highly guaranteed but packet loss is usual especially in situations where the communication path is bad. Consequently, the fore-mentioned techniques are unsuitable for streaming media because of the risk of losing all video data.

As one of the DRM techniques that can be applied to streaming media, video scrambling has been focused these years[3]. It is obvious that streaming media almost all have sound with them. However, it is extremely easy to extract sound from streaming media. Consequently, conventional scrambling methods are not robust against sound copyright infringement problems such as illegal copying and re-uploading of sound.

To solve those problems, a partial scrambling technique embedding different types of contents, which has been proved to be effective, was proposed[4]. In the technique, audio data are embedded into qDCT coefficients of the video data and partial scrambling is generated at the same time during the MPEG-4 encoding process. However, the parameters used were determined somewhat subjectively, the investigation of objectively proper parameters was required. In this paper, we conduct experiments to investigate objectively proper parameters in order to improve the technique. Using the improved parameters, we conduct verification experiments to prove that the proposed technique is effective. we also conduct experiments to verify the affinity of the proposed technique with existing streaming systems.

II. Proposed Techniques

A. Overview

There are plenty of potential targets in MPEG-4 videos to embed data, such as DCT coefficients, histogram data and motion vectors. However, DCT coefficients provide a large and stable container to embed data into it. MPEG-4 video data consist of plural GOPs(Groups of Pictures), each GOP usually contains one I(Intra) frame, several P(Predictive) frames and several B(Bidirectionally Predictive) frames. Each frame is divided into plenty of 16×16 MBs(Macro Blocks). Each MB generally includes 4 luminance blocks which are also called Y-blocks and 2 chrominance blocks. Commonly, changes in luminance are more evident and identifiable for people to recognize than changes in chrominace. In MPEG-4 codec, MBs are roughly classified into intra type and inter type automatically. According to the research by Nemoto[5], as inter-MBs use intra-MBs as the referenced data to realize the inter frame prediction, the changes in intra-MBs will be propagated to inter-MBs. Therefore, we choose intra Y-blocks as the targets of the proposed process. Because the changes in DCT coefficients before quantization have a risk of disappearing during the quantization procedure, we will perform the proposed processes to qDCT coefficients.

On the encoding side, we first separate the input raw video with soundtrack into video data and audio data. We

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

then encode the audio data into aac or mp3 format. Then we perform the normal encoding process to turn the video data into MPEG-4 format until we obtain the qDCT coefficients of the intra Y-blocks. We embed 8 bits of audio data into AC coefficients, and perform partial scrambling process. As embedding method, we use "Twice-plus", and we perform sign inversion to the AC coefficients and shuffling to the DC coefficients in order to generate partial scrambling. All the 3 processes will be explained later. After those 3 processes, the remaining encoding process will be performed.

On the decoding side, we perform the normal decoding process of MPEG-4 format until we obtain the qDCT coefficients of the intra Y-blocks. We restore the DC and AC coefficients, and extract the embedded audio data. After that, we finish the remaining decoding process. Finally, the restored video and audio data can be combined for watching.

B. Twice-plus

In twice-plus method, we first double the value of one target AC coefficient, then add one bit of audio data to it.

We represent the 63 AC coefficients of one Y-block in zigzag scanning order as $A_k (k \in \{1, 2, ..., 63\})$. Then we set eight mutually different parameters for data embedding which are $k_1, k_2, ..., k_8 (1 \leq k_1, k_2, ..., k_8 \leq 63)$ as the embedding positions. For $k = k_1, k_2, ..., k_8$, we embed one bit of sound data S_{in} into an AC coefficient A using Eq. (1).

$$A^* = 2 \times A + S_{in} \tag{1}$$

A^* stands for the newly calculated AC coefficient to replace A. Since S_{in} is one bit of data, it is either 0 or 1. Therefore, A^* is even when $S_{in} = 0$ and A^* is odd when $S_{in} = 1$. Hence we can obtain the extracted one bit of sound data S_{out} according to the parity of A^* using Eq. (2),

$$S_{out} = A^* \bmod 2 \tag{2}$$

where mod stands for modulo operation and the recovered AC coefficient A^{**} using Eq. (3).

$$A^{**} = (A^* - S_{out}) \div 2 \tag{3}$$

It is obvious that $S_{in} = S_{out}$ and $A = A^{**}$ according to Eqs. (1) to (3). Therefore, we can say that the proposed embedding process is reversible. An example of the twice-plus method is shown in Fig. 1.

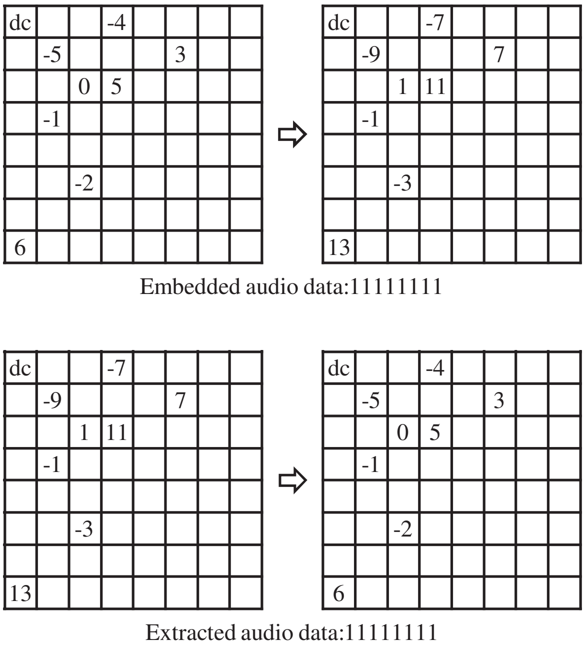
C. Sign Inversion

For AC coefficients, we use a common method i.e. sign inversion to generate scrambling because sign inversion has little influence on file-size and is easy to control.

We first set a parameter $N(1 \leq N \leq 63)$, where N is the number of the AC coefficients to perform sign inversion, and then set N mutually different parameters $i_1, i_2, ..., i_N (1 \leq i_1, i_2, ..., i_N \leq 63)$ as the sign inversion positions in zigzag scanning order. For $i = i_1, i_2, ..., i_N$, the sign of an AC coefficient A is inverted using Eq. (4).

$$A^* = -A \tag{4}$$

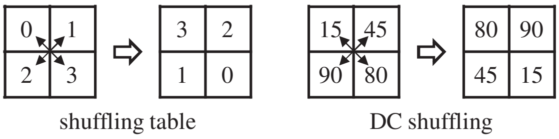
A^* stands for the newly calculated AC coefficient to replace A. We can restore the AC coefficient using Eq. (5).



Embedded audio data:11111111

Extracted audio data:11111111

Fig. 1. An example of the twice-plus method



shuffling table

DC shuffling

Fig. 2. An example of the DC shuffling method

$$A^{**} = -A^* \tag{5}$$

It is obvious that $A = A^{**}$ according to Eqs. (4) and (5). Therefore, we can say that the proposed AC coefficients sign inversion process is reversible.

D. DC Shuffling

For DC coefficients in one MB, we perform a process called DC shuffling. We shuffle the four DC coefficients in the four Y-blocks of one intra MB according to a shuffling table. Shuffling table consists of four integer parameters $T_0, T_1, T_2, T_3 (0 \leq T_0, T_1, T_2, T_3 \leq 3)$.

When performing DC shuffling in one MB, given the four DC coefficients D_0, D_1, D_2, D_3, we obtain the shuffled DC coefficients ${D_{T_0}}^*, {D_{T_1}}^*, {D_{T_2}}^*, {D_{T_3}}^*$ using Eq. (6) where i equals to 1, 2, 3 and 4.

$${D_{T_i}}^* = D_i \tag{6}$$

We could then obtain the four restored DC coefficients ${D_0}^{**}, {D_1}^{**}, {D_2}^{**}, {D_3}^{**}$ using Eq. (7) where i equals to 1, 2, 3 and 4.

$${D_i}^{**} = {D_{T_i}}^* \tag{7}$$

With Eqs. (6) and (7), we could conclude that D_0,

TABLE I. Effectiveness verification parameters

(a) AC inversion parameters

Stage	N	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
Strong	8	1	2	3	4	5	6	7	8
Medium	8	5	6	7	8	9	10	11	12
Weak	8	9	10	11	12	13	14	15	16

(b) DC shuffling parameters

Stage	T_0	T_1	T_2	T_3
Strong	3	2	1	0
Medium	0	2	1	3
Weak	0	1	2	3

TABLE II. Details of the experimental input videos

Duration	10 seconds
Video Resolution	1280 × 720
Video Frame Rate	25fps
Video Format	yuv
Soundtrack Format	flac

D_1, D_2, D_3 and $D_0{}^{**}$, $D_1{}^{**}$, $D_2{}^{**}$, $D_3{}^{**}$ are equal correspondingly. Thus, we can say that the proposed DC shuffling process is also reversible. An example of the DC shuffling method is shown in Fig. 2.

E. Qualitatively Adjustable Degradation Degrees

In the proposed technique, three methods are utilized during the encoding process to MPEG-4 format. In the three methods, we set plenty of parameters to control the proposed processes. The parameters are $k_1, k_2, ..., k_8$ for data embedding, $N, i_1, i_2, ..., i_N$ for AC sign inversion and T_0, T_1, T_2, T_3 for DC shuffling.

In each of those methods, changes in parameters could significantly change the degradation degree of the output videos. In other words, it is possible to adjust the degradation degree by adjusting the parameters appositely. In order to qualitatively adjust the degradation degree, we conducted several experiments to verify the most suitable parameter sets considering video quality and file-size.

The results showed that the proper embedding parameters are ($k_1 = 54, k_2 = 55, k_3 = 56, k_4 = 58, k_5 = 60, k_6 = 61, k_7 = 62, k_8 = 63$) for all, and the proper AC inversion and the DC shuffling parameters are shown in Table I.

III. Experiments and Results

A. Overview

We use four uncompressed videos, p.yuv, r.yuv, s.yuv, t.yuv[6], and one uncompressed soundtrack 70.flac as our experimental videos and soundtrack[7]. The details of the experimental input videos and soundtrack are shown in Table II. The video is encoded to MPEG-4 format and the soundtrack is encoded to aac format. The video quality parameter qscale is 8 and the sound bit rate is 192kbps. The first frames of the input videos are shown in Fig. 3.

The metrics used in the experiments are MSE (Mean Squared Error), PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity)[8].

Fig. 3. First frames of the experimental videos

TABLE III. Effectiveness verification results

(a) File-size increment[%]

Stage	p	r	s	t	avg.
Strong	-6.91	-8.99	-8.09	-3.93	-6.98
Medium	-6.60	-8.61	-8.13	-3.81	-6.79
Weak	-7.47	-9.65	-8.78	-4.24	-7.54

(b) PSNR[dB]

Stage	p	r	s	t	avg.
Strong	22.0	21.0	19.7	19.5	20.6
Medium	25.9	25.3	24.2	22.6	24.5
Weak	30.1	32.9	30.8	27.4	30.3

(c) SSIM

Stage	p	r	s	t	avg.
Strong	0.827	0.809	0.653	0.715	0.751
Medium	0.930	0.929	0.875	0.861	0.899
Weak	0.973	0.988	0.973	0.954	0.972

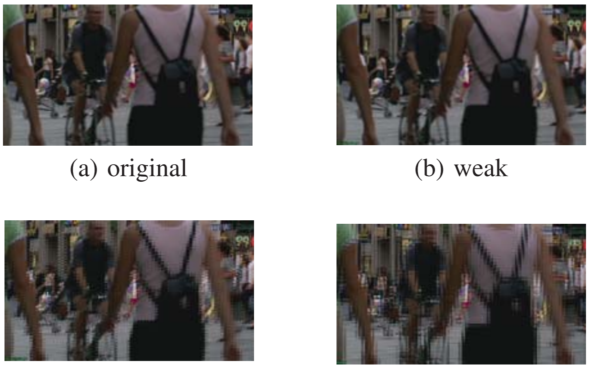
B. Effectiveness Verification Experiments

The results of the effectiveness verification experiments using the parameters combinations for “Strong”, “Medium” and “Weak” are shown in Tables III. We also used MSE to verify the reversibility of the proposed technique, the results were all 0 for any original video or soundtrack with the corresponding restored video or soundtrack. That indicated the reversibility of the proposed technique.

The results in Table III(a) show that all experimental videos become smaller in file-size after the proposed process. The results in Table III(b) and Table III(c) show that the proposed technique realizes adjustable degradation degrees for different situations - e.g. “Strong” for video conferences, “medium” for live broadcasting, and so on. As an example, the original video and the sample videos for p are shown in Fig. 4.

C. Affinity Verification Experiments

In order to verify the affinity of the proposed technique with existing streaming systems, we combine the proposed technique with the key-frame duplication technique, which is commonly used to suppress the influences of packet loss in various streaming systems. We use IEEE802.11a standard for transmission simulations[9]. The parameters are shown in Table IV[10]. We conducted simulation experiments for



(a) original

(b) weak

(c) medium

(d) strong

Fig. 4. Comparison of p videos

TABLE IV. MAIN PARAMETERS OF WIRELESS TRANSMISSION SIMULATION

Parameter	Value
Communication standard	IEEE802.11a
Modulation scheme	OFDM (64QAM)
No. of subcarriers	52 (64 FFT points)
OFDM signal length	4.0 [μs] (GI = 0.8μs)
Coding rate	3/4
Transmission rate	54 [Mbps]
Model of trans. medium	3-ray Rayleigh fading
Doppler frequency	0.001 [Hz]

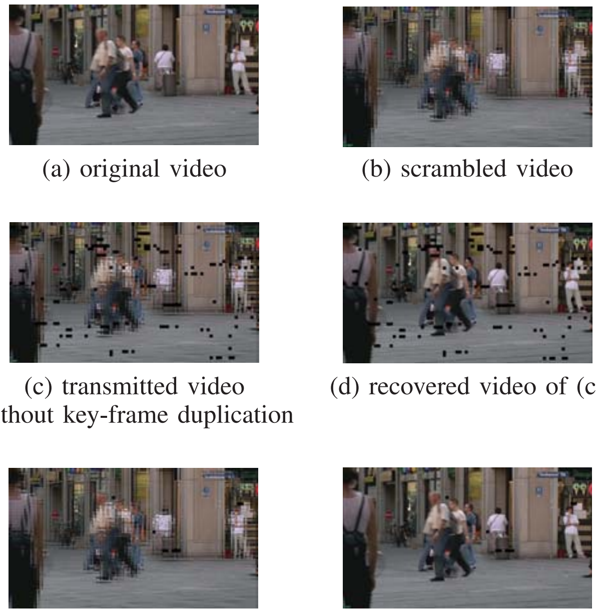
all experimental videos. The average BER (Bit Error Ratio) was 3.50%. An example of video p with "strong" intensity is shown in Fig. 5. Macro-blocks where errors occurred are represented as black squares.

As shown in Fig. 5, the proposed technique can be combined with key-frame duplication technique well, and the scrambling can be removed after transmission. In other words, the proposed technique can be considered as a feasible DRM solution suitable for existing streaming systems.

IV. CONCLUSION

In this paper, we proposed a scrambling technique embedding soundtracks into videos for streaming media. In the technique, we embedded audio data into qDCT coefficients of the video data and generate scrambling at the same time during the coding process to MPEG-4 format. The experimental results showed that the proposed technique is effective. We also conducted simulation experiments to verify that the proposed technique can be combined with existing streaming systems and can recover the scrambled video after transmission.

However, there is still much improvement that can be implemented into the proposed technique. In the future, we will conduct subjective verification experiments, try to implement quantitatively adjustable degradation degrees, and combine the proposed technique with other streaming systems to further verify the universal affinity of the proposed technique. If possible, we will perform the proposed technique to MPEG transport stream (MPEG-TS) to survey the feasibility.



(a) original video

(b) scrambled video

(c) transmitted video without key-frame duplication

(d) recovered video of (c)

(e) transmitted video with key-frame duplication

(f) recovered video of (e)

Fig. 5. An example of simulation result

REFERENCES

[1] O. J. Hanas, P. D. Toonder and F. Pennypacker, "An Addressable Satellite Encryption System for Preventing Signal Piracy, " IEEE Transactions on Consumer Electronics, Vol. **CE-27**, pp. 631–635, 1981.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications, " RFC 3550 (Standard), 2003.

[3] M. Takayama, K. Tanaka, A. Yoneyama, and Y. Nakajima, "A video scrambling scheme applicable to local region without data expansion, " Proc. IEEE on ICME, pp. 1349–1352, Toronto, Canada, 2006.

[4] X.Li, S.Kang, Y.Sakamoto, "A Partial Scrambling Technique with Adjustable Degradation Degrees Embedding Different Types of Contents", International Workshop on Advanced Image Technology 2016, 2C-5, 2016.

[5] Y. Nemoto, Y. Toyota, S. Sakazawa, L. Zhao and H. Yamamoto, "A STUDY ON VIDEO SCRAMBLING CONSIDERING INTER-FRAME PREDICTION, " IEICE technical report, Image engineering 105(500), pp. 207–212, 2006.

[6] https://media.xiph.org/video/derf

[7] https://tech.ebu.ch/publications/sqamcd

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity, " IEEE Trans. Image Process., Vol. **13**, No. 4, pp. 600–612, 2004.

[9] IEEE Std 802.11a-1999, "Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band, " 1999.

[10] K. Yamaguchi, R. Watanabe, A. Nozaki and Y. Sakamoto, "Influence of Transmission Error on Reconstructed-image Quality in Electro-holography, " The Journal of The Institute of Image Information and Television Engineers, Vol. **70**, No. 5, pp. J105–J113, 2016.