Codestream Length-Invariant Partial Scrambler Using DCT Signs for Motion JPEG Movies

Keijiro KUROIWA, Masaaki FUJIYOSHI, and Hitoshi KIYA
Department of Information and Communication Systems Engineering
Tokyo Metropolitan University
Hachioji 192–0397, Japan
E-mail: {kuroiwa@isys., fujiyoshi@, kiya@}eei.metro-u.ac.jp

***Abstract*— This paper proposes a partial scrambling method for Motion JPEG movies that inverts bits representing the positive and negative sign of particular AC coefficients in a codestream to scramble the codestream. Utilizing the relation between motion and the positive and negative sign of coefficients, moving object detection is not required in this method. Moreover, the proposed method completely descrambles codestreams to the original by simple inversion of bits without any knowledge on the position and shape of moving objects. The codestream domain processing in this method serves low processing time and invariant coding efficiency. Simulation results show the effectiveness of the proposed method.**

I. INTRODUCTION

Nowadays, monitoring systems have been widely used from secure areas in government and companies to public space in downtown to protect human lives, valuable items, and information. Cameras, in particular, mounted in public space, take not only accidents and/or crimes but also ordinary social living by people. To protect privacy of ordinary people, partial scrambling is one effective and practical approach. In addition, partial scrambled scenes grab observer's attention. Conventional partial scrambling techniques are classified into two categories. Methods in which moving objects are scrambled [1]–[3] and methods in which not moving objects but several units are scrambled [4]–[6]. This paper proposes a novel method classified to the former.

Conventional methods [1]–[3] have several disadvantages. Scrambling in the spatial domain [1], [2] requires decoding and encoding of movies before and after scrambling, it consumes much time and generally degrades the coding efficiency. Whereas the conventional method [3] taking the structure of codestreams account into keeps the length of codestreams as well as methods not for moving objects [4]–[6]. All three methods detects moving objects independently from scrambling [1]–[3], and the knowledge on the position and shape of moving objects is required to descramble codestreams.

This paper proposes a novel scrambling method for Motion JPEG movies taken by a fixed camera. The proposed method scrambles codestreams with no moving object detection. In definitive, this method inverts bits that indicates moving objects in codestream domain, i.e., without any decoding process. This codestream domain processing saves the processing time and keeps the codestream length. All the proposed method has to do to descramble is inversion of bits once again, and this simple process completely restores the original codestream.

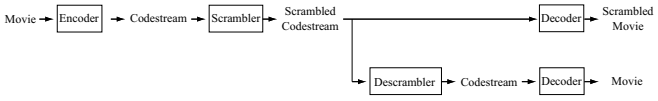


Fig. 1: Partial scrambling system.

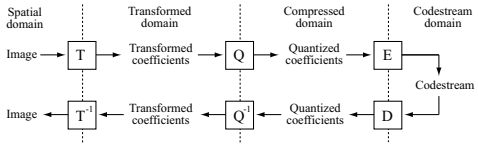


Fig. 2: Transform image coding (T: transformation, Q: quantization, E: entropy encoding, T^{-1}: inverse transformation, Q^{-1}: inverse quantization, D: entropy decoding).

II. BACKGROUND

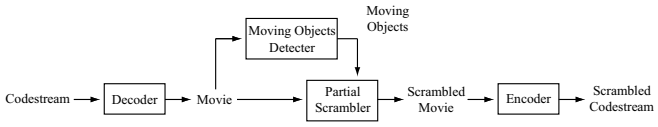
Let us consider the system shown in Fig. 1. In this system, a movie taken by a fixed camera is encoded by an arbitrary transform image compression technology of which the general block diagram is shown in Fig. 2. Moving objects in each frame are scrambled, then, a scrambled codestream is generated. By decoding a scrambled codestream, the movie is obtained but is partially scrambled. The system, however, is able to descramble a scrambled codestream by a request, and descrambled codestream gives descrambled movie. This paper focuses scrambling and descrambling methods in the system.

A. Related Works

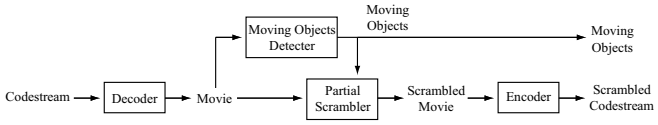
Several partial scrambling methods have been proposed [1]–[3]. Figure 3 shows the scrambling method in these works, and those corresponding descrambling methods are shown in Fig. 4.

Figure 3 (a) shows the scrambling of conventional method 1 [1]. Since this method scrambles movies in the spatial domain, codestreams have to be decoded before scrambling. A scrambled movie, off course, is encoded again after scrambling. In this method, moving object detection is applied in the spatial domain before scrambling. Moreover, This scrambling method is irreversible method, i.e., it is not able to descramble a scrambled movie to its original state.

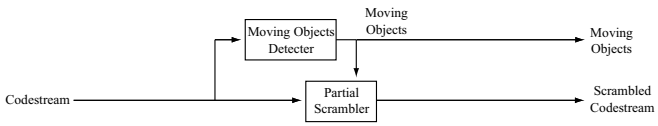
0-7803-9733-9/06/$20.00



(a) Conventional method 1 [1].

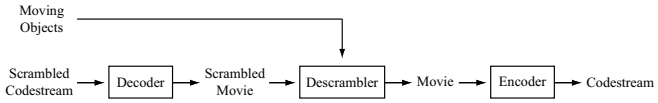


(b) Conventional method 2 [2].

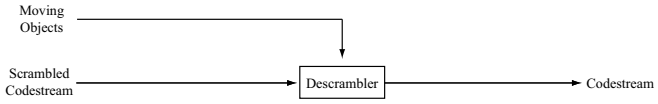


(c) Conventional method 3 [3].

Fig. 3: Conventional scrambling methods.



(a) Previous Method 2 [2].



(b) Previous Method 3 [3].

Fig. 4: Descrambling Methods.

Figures 3 (b) and 4 (a) show scrambling and descrambling in conventional method 2 [2], respectively. Because this method also scrambles movies in the spatial domain, decoding and encoding must be applied to codestreams before and after scrambling, respectively. Decoding is also required in descrambling. Additionally, scrambling in the spatial domain generally degrades the compression efficiency, i.e., it increases the length of a codestream in comparison with its original state. A moving object detection is applied to a movie in the spatial domain before scrambling. The knowledge on the position and shape of moving objects is required to descramble scrambled movies.

Figures 3 (c) and 4 (b) show the scrambling and descrambling in conventional method 3 [3], respectively. Since this method scrambles and descrambles in the compressed domain, neither decoding to the spatial domain nor encoding from the spatial domain is required to scramble and descramble codestreams. Moving object detector is also used in this method, and the knowledge on moving objects is still needed in descrambling.

B. Motion JPEG

In this paper, the coding technology used in the system is assumed to be Motion JPEG. Since Motion JPEG encodes each frame of video streams by JPEG [7], JPEG encoding algorithm is briefly described in this section.

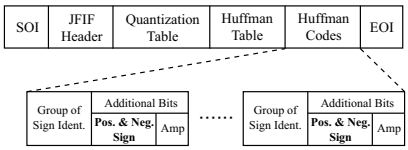


Fig. 5: JPEG Codestream.

In JPEG encoding, pixel values in an original image is shifted, and the shifted image is divided into non-overlapped blocks, referred to as DCT blocks in this paper, that each block consists of 8×8-pixels blocks. The two dimensional discrete cosine transformation (2D DCT), then, is applied to each block to produce one DC and 63 AC transformed coefficients. All the DCT coefficients are quantized according to a quantization table scaled by Q-factor. Finally, all the coefficients that are sorted by the zigzag order are coded by an entropy encoder, Huffman or the arithmetic.

Figure 5 shows the structure of a JPEG codestream that is generated from a grayscale image and with Huffman encoder. The start of image (SOI) marker is the head of a JPEG codestream. The JPEG File Interchange Format (JFIF) header contains information like the image size. The next two entities are tables that are used in encoding process and are required to decode the codestream. Huffman codeword trailer represent all the DCT coefficients in an image. The end of image (EOI) marker follows the last byte of a codestream.

A set of Huffman codewords consists of the indicator of codeword group and additional bits. The latter part is further divided into the positive and negative sign bit and the codeword for the amplitude of a DCT coefficient. Since this positive and negative sign is independent of other bits in a codestream, the signs are directly acquired from codestreams.

III. Proposed Method

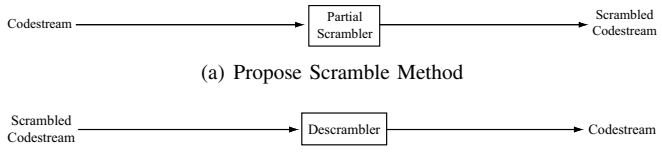
In this section, a method for scrambling and descrambling Motion JPEG movies is proposed. The proposed method satisfies the following.

1) No moving object detector
2) Codestream domain processing
3) Codestream length invariant
4) Descrambling without any reference
5) Reversible scrambling

A. Notations and Terminologies

Several notations and terminologies used in the following sections are listed here.

- $\mathbf{X}$ represents a Motion JPEG movie.
- I represents the number of frames in $\mathbf{X}$.
- X_i represents the i-th frame of $\mathbf{X}$, where $0 \leq i \leq I-1$.
- M represents the number of constituent 8×8 blocks in a frame, where $0 \leq m \leq M-1$. This M relates to a frame size.



(a) Propose Scramble Method

(b) Propose Descramble Method

Fig. 6: Scrambling and descrambling in the proposed method.

- $X_i(m,n)$ represents the n-th AC coefficient in the m-th block in frame X_i, where $0 \leq n \leq 62$.
- $\mathrm{sgn}\left(X_i(m,n)\right)$ represents the positive and negative sign of $X_i(m,n)$. It derived by the following equation:

$$\mathrm{sgn}(a) = \begin{cases} -1, & a < 0 \\ 0, & a = 0 \\ 1, & a > 0 \end{cases} \tag{1}$$

B. Partial Scrambling Algorithm

The proposed partial scrambling algorithm consists of the following steps.

1. Set $j := 0$.
2. Set $m := 0$.
3. Set $n := 0$.
4. If $X_{2j}(m,n) = 0$ or $X_{2j+1}(m,n) = 0$, proceed to Step 6.
5. $\mathrm{sgn}\left(X_{2j}(m,n)\right)$ is substituted by $\mathrm{sgn}\left(X_{2j+1}(m,n)\right)$, and vice versa. That is,

$$\mathrm{sgn}\left(X_{2j}(m,n)\right) := \mathrm{sgn}\left(X_{2j+1}(m,n)\right), \tag{2}$$
$$\mathrm{sgn}\left(X_{2j+1}(m,n)\right) := \mathrm{sgn}\left(X_{2j}(m,n)\right). \tag{3}$$

6. Set $n := n+1$. Continue to Step 4 unless $n = 63$.
7. Set $m := m+1$. Continue to Step 3 unless $m = M$.
8. Set $j := j+1$. Continue to step 2 unless $j = I/2$.

C. Descrambling Algorithm

Descrambling algorithm is the identical to the scrambling algorithm described above. That is, the scrambling algorithm also descrambles scrambled codestreams that are scrambled by itself.

D. Features of Proposed Method

Features of the proposed method are the following.

1) No Moving Object Detector

In a movie taken by a fixed camera, consecutive frames are identical or quite similar each other unless moving objects exist. Existence of motion objects makes corresponding parts between adjacent frames different. Since the positive and negative sign of corresponding DCT coefficients are different between different images [8], [9], corresponding DCT coefficients in the blocks containing motion objects have different signs between consecutive frames.

Eqs. (2) and (3) in Step 5 are transformed into

$$\mathrm{sgn}\left(X_{2j}(m,n)\right) := -\mathrm{sgn}\left(X_{2j}(m,n)\right), \tag{4}$$
$$\mathrm{sgn}\left(X_{2j+1}(m,n)\right) := -\mathrm{sgn}\left(X_{2j+1}(m,n)\right), \tag{5}$$

if $\mathrm{sgn}\left(X_{2j}(m,n)\right) \neq \mathrm{sgn}\left(X_{2j+1}(m,n)\right)$, $X_{2j}(m,n) \neq 0$, and $X_{2j+1}(m,n) \neq 0$ are satisfied. Otherwise, they are transformed into

$$\mathrm{sgn}\left(X_{2j}(m,n)\right) := \mathrm{sgn}\left(X_{2j}(m,n)\right), \tag{6}$$
$$\mathrm{sgn}\left(X_{2j+1}(m,n)\right) := \mathrm{sgn}\left(X_{2j+1}(m,n)\right), \tag{7}$$

i.e., nothing is happened. This method, thus, deals with coefficients having different signs between adjacent frames that indicate existence of motion objects. Consequently, the proposed method requires no moving object detector to scramble moving objects.

2) Codestream Domain Processing

The proposed method exchanges the positive and negative sign of DCT coefficients in a frame with that in the consecutive frame to scramble moving objects, as shown in Eqs. (2) and (3) in Step 5. As shown in Fig. 5, the positive and negative signs are encoded independently from those corresponding magnitude. The positive and negative signs, thus, are directly obtained from a codestream without any decoding process.

Therefore, this method scrambles and descrambles movies in codestream domain rather than in the spatial [1], [2] or compressed [3] domains. This method, thus, requires neither decoding before scrambling nor encoding after scrambling. Descrambling scrambled movies also requires no decoding process. This feature saves the time.

In addition, codestream domain processing serves the ability of keeping the length of a scrambled codestream as the same as its original state, i.e., the codestream before scrambling. The detail is described in Sect. III-D.3.

3) Codestream Length Invariant

This method swaps the positive and negative signs between adjacent frames in codestream domain as described above. Since the positive and negative sign of a AC coefficient is represented by one bit as shown in Fig. 5, a bit is exchanged by the other to scramble codestreams. From the perspective with Eqs. (4), (5), (6), and (7), this method inverts the positive and negative sign if signs between consecutive frames are different each other. That is, the proposed method inverts the sign bit in codestream domain.

This scrambling process that swaps bits between codestreams or inverts a bit in a codestream does not affect any other bits in codestreams. Consequently, the length of codestreams neither increase nor decrease. It is noteworthy that this feature fundamentally realized by the codestream domain processing.

4) Descrambling without Any Reference

As described in Sect. III-D.1, moving objects are indicated by inversion of the sign of corresponding coefficients between adjacent frames. In the scrambling algorithm, Eqs. (2) and (3) in Step 5 swap $\mathrm{sgn}\left(X_{2j}(m,n)\right)$ and $\mathrm{sgn}\left(X_{2j+1}(m,n)\right)$, unless $X_{2j}(m,n) = 0$ or $X_{2j+1}(m,n) = 0$. This process stores inversion of the sign of corresponding coefficients.

TABLE I: Simulation conditions

Input Movie	480×720 pixels, 8 bits/pixel, gray scale 15 frames/sec, 250 frames
Q-factor	50

(a) The 240th frame, X_{240}.

(b) The 241st frame, X_{241}.

Fig. 7: An example of adjacent frames with moving objects.

The relation on sign of coefficients between consecutive frames also stores the position and shape of moving objects in itself. Therefore, this method requires no knowledge on the position and shape of moving objects in descrambling process.

5) Reversible scrambling

In the scrambling algorithm, Eqs. (2) and (3) in Step 5 exchange $\mathrm{sgn}\left(X_{2j}(m,n)\right)$ with $\mathrm{sgn}\left(X_{2j+1}(m,n)\right)$, unless $X_{2j}(m,n) = 0$ or $X_{2j+1}(m,n) = 0$. To descramble scrambled codestreams, all this method has to do, thus, is simple swapping them again.

Therefore, this method completely descrambles a scrambled codestream to the original codestream, differently from conventional method 1 [1] that does not have descramble ability and conventional method 2 [2] that sometimes reconstruct different movie because of re-encoding after scrambling.

IV. Simulations

The conditions are summarized in Table I. Two adjacent frames in the movie are shown in Fig. 7. These are the 240th and the 241st frames, i.e., X_{240} and X_{241}, in the input movie, respectively.

Figures 8 (a) and (b) are magnified view of Figs. 7 (b) and its corresponding scrambled frame, respectively. Those figures show that the proposed method scrambles moving objects properly.

Figure 9 shows that the number of the sign inversion in each DCT block between two frames as its luminance. The

(a) Clipping from Fig. 7 (b).

(b) Clipping from the scrambled X_{240}.

Fig. 8: Magnified view.

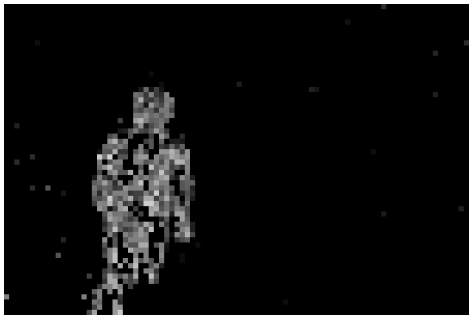


Fig. 9: The number of the inverted signs (The brightness corresponds to the number).

larger the number of the sign inversion occurs, the brighter the luminance of the DCT block becomes. The maximum number is 63, and the minimum number is 0. Thus, white indicates all 63 AC coefficients are inverted, whereas no AC coefficient is inverted in the block filled with black. It is found that blocks containing moving objects are bright, i.e., this method adequately scrambles moving objects.

It is noted that the proposed method completely descrambles the scrambled codestream to its original state.

V. Conclusion

This paper has proposed a partial scrambling method for Motion JPEG movies. This method achieves partial scrambling of movies by only swapping bits in codestream domain without moving object detection. Neither decoding nor encoding is required to scramble and descramble codestream in this method, the proposed method serves low processing time and invariant coding efficiency. Moreover, the proposed method completely descrambles scrambled codestreams without any reference information.

References

[1] I. Kitahara, K. Kogure, and N. Hagita, "Stealth Vision: A method for video capturing system with protecting privacy," *IEICE Tech. Rep.*, vol.103, no.738, pp.89–94, Mar. 2004.

[2] K. Yabuta, H. Kitazawa, and T. Tanaka, "A fixed monitoring camera image processing method satisfying both privacy protection and object recognition," *IEICE Tech. Rep.*, vol.105, no.29, pp.13–18, Apr. 2005.

[3] K. Kuroiwa, M. Fujiyoshi, and H. Kiya, "A moving object detection using POC and a scramble for Motion JPEG," *ITE Tech. Rep.*, vol.30, no.17, pp.43–46, Feb. 2006.

[4] A. Takagi, K. Nishikawa, and H. kiya, "Scalable scrambling for MPEG Video," in *Proc. Int. Workshop Advance Image Technol.*, pp.91–96, 2005.

[5] H. Kiya, S. Imaizumi, and O. Watanabe, "Partial-scrambling of JPEG2000 images without generating marker codes," *IEICE Trans. Inform. & Syst.*, vol.J86-D-II, pp.1628–1636, Nov. 2003.

[6] S. Imaizumi, O. Watanabe, M. Fujiyoshi, and H. Kiya, "Generalized hierarchical encryption of JPEG 2000 codestreams for access control," in *Proc. IEEE ICIP*, vol.II, pp.1094–1097, 2005.

[7] *Information technology — Digital compression and coding of continuous-tone still images — Requirements and guidelines*, ISO/IEC International Standard 10918-1, 1994.

[8] Fitri Arnia, I. Iizuka, M. Fujiyoshi, and H. Kiya, "Fast and robust identification methods for JPEG images with various compression ratios," in *Proc. IEEE ICASSP*, 2006.

[9] I. Ito, M. Fujiyoshi, and H. Kiya, "DCT sign-only image and its application to image matching," *ITE Tech. Rep.*, vol.30, no.33, pp.29–32, Jun. 2006.