CODESTREAM DOMAIN SCRAMBLING OF MOVING OBJECTS BASED ON DCT SIGN-ONLY CORRELATION FOR MOTION JPEG MOVIES

Keijiro KUROIWA, Masaaki FUJIYOSHI, and Hitoshi KIYA

Dept. of Information and Communication Systems Engineering, Tokyo Metropolitan University
6–6 Asahigaoka, Hino-shi, Tokyo 191–0065, Japan

ABSTRACT

This paper proposes a moving objects scrambling method for Motion JPEG movies that consists of two parts: a moving objects detection based on DCT sign-only correlation (DSOC) and a partial scrambling. DSOC is the correlation of the positive and negative signs of DCT coefficients. Utilizing the codestream structure of Motion JPEG movies and the relation between motion and DSOC, the proposed moving objects detection is achieved in codestream domain. The proposed scrambling is also achieved in codestream domain by inverting signs of DCT coefficients that each sign is independently encoded as one bit. The codestream domain processing in this method serves low processing time and keeps coding efficiency. Moreover, the proposed method completely descrambles scrambled codestreams to the original without any knowledge on the position and shape of moving objects. Simulation results show the effectiveness of the proposed method.

Index Terms— Privacy, Surveillance, Object detection, Discrete cosine transforms, Correlation, Image coding

1. INTRODUCTION

Nowadays, video surveillance systems have been widely used in secure areas by 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 lives by people. To protect privacy of ordinary people, partial scrambling is one effective and practical approach [1–5]. 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, 5]. This paper proposes a novel method classified to the former.

Conventional methods scramble moving objects [1–3] have several disadvantages. Scrambling in the spatial domain [1, 2] requires decoding and encoding of movies before and after scrambling, it, thus, consumes much time and generally degrades the coding efficiency. To descramble scrambled movies, the knowledge on the position and shape of moving objects is required [2, 3].

This paper proposes a novel scrambling method for Motion JPEG movies taken by fixed cameras. The proposed method detects and scrambles moving objects in codestream domain, i.e., without any decoding process. This codestream domain processing saves the processing time and keeps the codestream length. The descrambling is achieved with the same way as the scrambling and requires no knowledge on the moving objects. This simple process completely restores the original codestream.

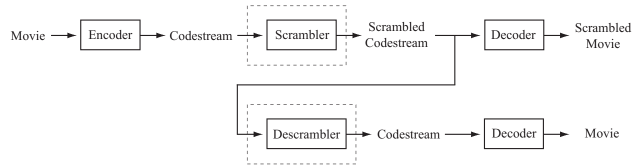


Fig. 1. System scrambles moving objects in videos.

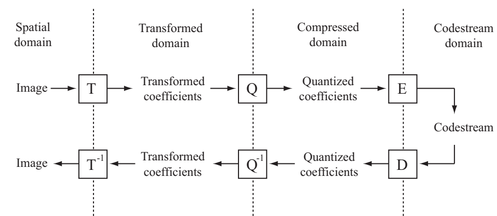


Fig. 2. Transform image coding (T: transformation, Q: quantization, E: entropy encoding, T^{-1}: inverse transformation, Q^{-1}: inverse quantization, D: entropy decoding).

2. BACKGROUND

This section shows the system description, summarizes related works, and mentions the algorithm and the codestream structure of Motion JPEG.

2.1. System Description

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. Hereafter, scrambling and descrambling algorithms in the system are focused.

2.2. Related Works

Several partial scrambling methods have been proposed [1–3]. Figures 3, 4, and 5 show the scrambling and descrambling algorithms in these works.

Figure 3 shows the scrambling algorithm of the conventional irreversible method [1]. Since this method detects and scrambles moving objects in the spatial domain, codestreams have to be decoded before scrambling. A scrambled movie, of course, is encoded again after scrambling. Moreover, this scrambling method is irreversible

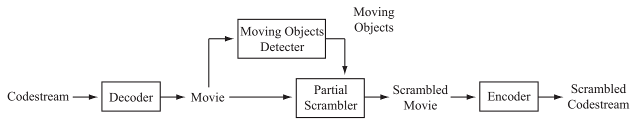
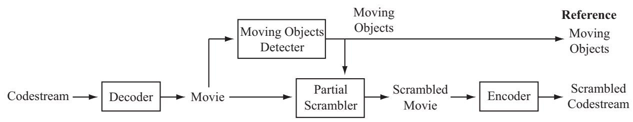
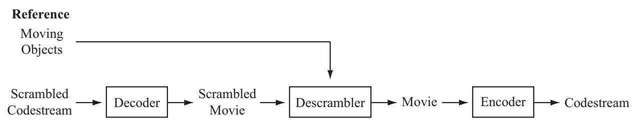


Fig. 3. Scrambling algorithm in conventional irreversible methods [1].



(a) Scrambling algorithm.



(b) Descrambling algorithm.

Fig. 4. Conventional methods in spatial domain [2].

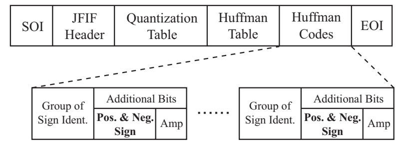
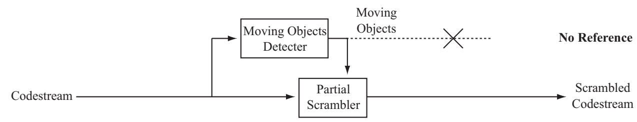
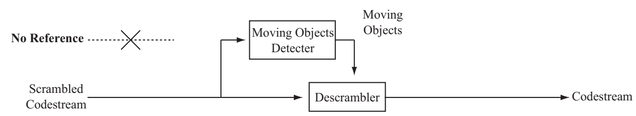


Fig. 6. JPEG Codestream.



(a) Scramble algorithm.



(b) Descramble algorithm.

Fig. 7. Scrambling and descrambling in the proposed method.

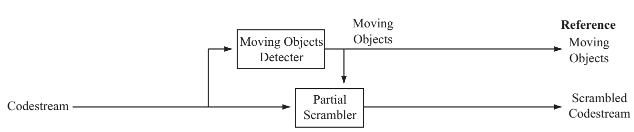
method, i.e., it is not able to descramble a scrambled movie to its original state.

Figures 4 (a) and (b) show the conventional scrambling and descrambling algorithms in spatial domain [2], respectively. Because this method also detects and scrambles moving objects in the spatial domain, decoding and encoding must be applied to codestreams before and after scrambling, respectively. Decoding and encoding are 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. The knowledge on the position and shape of moving objects is required to descramble scrambled movies.

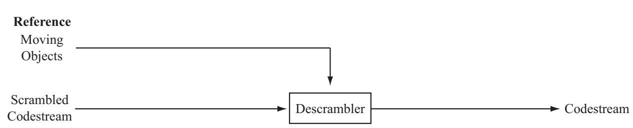
Figures 5 (a) and (b) show the conventional scrambling and descrambling algorithms in codestream domain [3], respectively. Since this method scrambles and descrambles in the codestream domain, neither decoding to the spatial domain nor encoding from the spatial domain is required to scramble and descramble codestreams. Moving object detector also runs in codestream domain, but the knowledge on moving objects is still needed in descrambling.

2.3. 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 [6], JPEG encoding algorithm is briefly described in this section.



(a) Scrambling algorithm.



(b) Descrambling algorithm [3].

Fig. 5. Conventional method in codestream domain [3].

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. 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 sorted by the zigzag order are coded by an entropy encoder, Huffman or the arithmetic.

Figure 6 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 such as 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.

3. PROPOSED METHOD

In this section, a method for scrambling and descrambling of moving objects in Motion JPEG movies is proposed. Figure 7 show the proposed scrambling and descrambling algorithms. The proposed method detects and scrambles moving objects in codestream domain. The knowledge of moving objects is not required to descramble scrambled movies.

3.1. Notations and Terminologies

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

- **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 $0 \le i \le I-1$.
- M represents the number of constituent 8×8 blocks (called DCT blocks) in a frame. This M relates to the size of a frame.
- $X_i(m,n)$ represents the n-th AC coefficient in the m-th block in frame X_i, where $0 \le m \le M-1$ and $0 \le n \le 62$.
- $\mathrm{sgn}\left(X_i(m,n)\right)$ represents the positive and negative sign of $X_i(m,n)$. It derived by the following equation:

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

- $s_j(m,n)$ is the n-th pseudo random number in the m-th block in the j-th matrix, where $s_j(m,n) \in \{1,-1\}$ and $0 \le j \le I/2$.
- r is the inversion rate and is defined as $\nu_j(m)/63$, where $\nu_j(m)$ represents the number of $s_j(m,n)$ whose value is -1 in the m-th block of the j-th matrix.

3.2. Moving Objects Detection

This algorithm detects moving objects by the following steps.

1. Set $j := 0$.
2. Set $m := 0$.
3. For the m-th block in frames X_{2j} and X_{2j+1}, it estimates similarity $d_j(m)$:

$$d_j(m) = \frac{\sum_{n=0}^{62} \mathrm{sgn}\left(X_{2j}(m,n)\right) \mathrm{sgn}\left(X_{2j+1}(m,n)\right)}{\sum_{n=0}^{62} \left|\mathrm{sgn}\left(X_{2j}(m,n)\right) \mathrm{sgn}\left(X_{2j+1}(m,n)\right)\right|}, \quad (2)$$

   where, $-1 \le d_j(m) \le 1$.
4. If $d_j(m) < t$, where t represents the user-defined threshold, it is determined that motions occur between m-th block of frames X_{2j} and X_{2j+1}.
5. Set $m := m+1$. Continue to Step 3 unless $m = M$.
6. Set $j := j+1$. Continue to Step 2 unless $j = I/2$.

This detection algorithm is based on the relation between motion and DSOC. As mentioned in Sect. 2.1, this paper assumes that the video surveillance system uses a fixed mounted camera, i.e., frames without motions are the identical excepting for luminance (DC) level. Since the positive and negative sign of corresponding AC coefficients between different images are different [7,8], corresponding AC coefficients in the blocks containing moving objects in the assumed system have different signs between consecutive frames. Therefore, if similarity $d_j(m)$ is small, it implies that corresponding DCT block of adjacent two frames are different. Then, it is determined that moving objects exist in the block.

3.3. Scrambling Algorithm

The m-th block of consecutive frames in which motions are detected by the algorithm described in Sect. 3.2 are scrambled by the following steps.

1. Set $n := 0$.
2. Scrambled AC coefficients, $X'_{2j}(m,n)$ and $X'_{2j+1}(m,n)$, are generated by

$$X'_{2j}(m,n) = s_j(m,n) X_{2j}(m,n), \quad (3)$$

$$X'_{2j+1}(m,n) = s_j(m,n) X_{2j+1}(m,n). \quad (4)$$

3. Set $n := n+1$. Continue to Step 2 unless $n = 63$

This algorithm scrambles a DCT block by random inversions of positive and negative signs of AC coefficients. This sign inversion of a JPEG codestream is the same as scrambling of a JPEG image in the DCT domain [3]. The inversion rate, r, is proportional to the strength of scrambling.

3.4. Descrambling Algorithm

To descramble scrambled movies, the moving object detection algorithm in Sect. 3.2 and the scrambling algorithm in Sect. 3.3 are applied in the same way as scrambling. Threshold value t and pseudo random numbers $s_j(m,n)$ are identical with those of the scrambling. Since $s_j(m,n) \in \{1,-1\}$, re-scrambling of scrambled coefficient results in

$$s_j(m,n) X'_{2j}(m,n) = s_j(m,n) s_j(m,n) X_{2j}(m,n) = X_{2j}(m,n), \quad (5)$$

i.e., recovering the original coefficient.

3.5. Features of Proposed Method

3.5.1. Codestream Domain Processing

The proposed moving objects detection and scrambling use positive and negative signs of AC coefficients in a JPEG codestream. As shown in Fig. 6, the positive and negative signs are encoded independently from those corresponding magnitude in a JPEG codestream. 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, the codestream domain processing serves the ability of keeping the length of a scrambled codestream as the same as its original state, i.e., the length of the original codestream before scrambling. The detail is described in the next section.

3.5.2. Codestream Length Invariant

This scrambling method inverts the positive and negative signs using the pseudo random number $s_j(m,n)$. That is, the proposed method inverts the sign bits in a JPEG codestream to scramble moving objects. Since the inversion of bits does not affect any other bits in a JPEG codestream, the length of codestreams neither increase nor decrease. It is noteworthy that this feature essentially realized by the codestream domain processing.

3.5.3. Reference-Free Descrambling

To descramble scrambled movies, the proposed method firstly detects moving objects by Eq. (2) from the scrambled movie. Since

Table 1. Simulation conditions

Input Movie	240×320 pixels, 8 bits/pixel, gray scale
	15 frames/sec, 250 frames
Q-factor	50

(a) The 141st frame, X_{141}. (b) The 142nd frame, X_{142}.

Fig. 8. An example of adjacent frames with moving objects.

$$\begin{aligned} d'_j(m) &= \frac{\sum_{n=0}^{62} \operatorname{sgn}\left(X'_{2j}(m,n)\right) \operatorname{sgn}\left(X'_{2j+1}(m,n)\right)}{\sum_{n=0}^{62} \left|\operatorname{sgn}\left(X'_{2j}(m,n)\right) \operatorname{sgn}\left(X'_{2j+1}(m,n)\right)\right|} \\ &= \frac{\sum_{n=0}^{62} s_j(m,n) \operatorname{sgn}\left(X_{2j}(m,n)\right) s_j(m,n) \operatorname{sgn}\left(X_{2j+1}(m,n)\right)}{\sum_{n=0}^{62} \left|s_j(m,n) \operatorname{sgn}\left(X_{2j}(m,n)\right) s_j(m,n) \operatorname{sgn}\left(X_{2j+1}(m,n)\right)\right|} \\ &= d_j(m), \end{aligned} \tag{6}$$

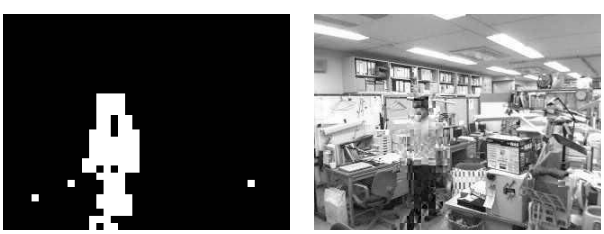
moving objects are detected in the descrambling phase as well as the scrambling phase, i.e., no knowledge on the size and shape of moving objects is required to descramble scrambled movies.

4. SIMULATION RESULTS

The simulation conditions are summarized in Table 1. An example of two adjacent frames in the movie is shown in Fig. 8. These are the 141st and 142nd frames, i.e., X_{141} and X_{142}, in the input movie, respectively.

An example of moving objects detection and partial scrambling by simulation using the prpropose method are shown in Fig. 9. Figure 9 (a) is the results of moving object detection in the 141st frame. The DCT blocks which are determined that it contains moving objects are filled with white. The threshold value t was 1. Figure 9 (b) is the scrambled 141st frame. Those figures show that the proposed method scrambled only moving objects properly.

Figures 10 (a) and (b) are magnified view of Figs. 8 (a) and 9 (b), respectively. Those figures show that the proposed method provided privacy protection to the human object.



(a) Moving objects in X_{141}. (b) Scrambled X_{141}, i.e., X'_{141}.

Fig. 9. Simulation results.

(a) X_{141}. (b) X'_{141}.

Fig. 10. Magnified view.

The length of JPEG codestream was invariant after scrambling. Moreover, the original movie $\mathbf{X}$ was obtained from scrambled movie $\mathbf{X}'$ using proposed method again without any reference.

Though the proposed method detects moving objects even an object slightly moves between adjacent frames as shown in Fig. 8, it cannot detect an object that is completely motionless. A further study is currently held for this problem by the authors, it, however, is not included in this paper.

5. CONCLUSIONS

This paper has proposed a partial scrambling method of moving objects in Motion JPEG movies. This method achieves detection and scrambling of moving objects in codestream domain. Thus, 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. Kogura, and N. Higita, “Stealth vision for protecting privacy,” in *Proc. IAPR ICPR*, 2004, pp.404–407.

[2] K. Yabuta, H. Kitazawa, and T. Tanaka, “A new concept of real-time security camera monitoring with privacy protection by masking moving objects,” in *Proc. IS&T/SPIE Electronic Imaging*, vol.6063, 2006.

[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, Feb. 2006, pp.43–46.

[4] H. Kiya, S. Imaizumi, and O. Watanabe, “Partial-scrambling of image encoded using JPEG2000 without generating marker codes,” in *Proc. IEEE ICIP*, 2003.

[5] D. Xie and C.-C.J. Kuo, “Multimedia data encryption via random rotation in partitioned bit streams,” in *Proc. IEEE ISCAS*, 2005, pp.5533–5536.

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

[7] 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.

[8] I. Ito and H. Kiya, “DCT sign-only correlation with application to image matching and the relationship with phase-only correlation,” in *Proc. IEEE ICASSP*, vol.I, 2007, pp.1237–1249.