A SCRAMBLING METHOD FOR MOTION JPEG VIDEOS ENABLING MOVING OBJECTS DETECTION FROM SCRAMBLED VIDEOS

Masaaki FUJIYOSHI, Keijiro KUROIWA, 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 scrambling method for Motion JPEG (MJ) videos and moving objects (MOs) detection from scrambled videos. In the proposed method, both scrambling and MOs detection utilize the property of the positive and negative sign of discrete cosine transformed (DCT) coefficients. Since a DCT sign is encoded separately from its corresponding magnitude, the sign is processed without decoding a MJ video. This feature makes the proposed method fast. Since this method only inverts some of DCT sign bits to scramble videos, the codestream length never change by scrambling. By this scrambling manner, the scrambling strength is easily controlled by the bit inversion ratio. Moreover, this method holds the relation of DCT signs between two frames to be able to detect MOs from scrambled videos. The proposed method can further scramble the detected MOs. In addition, descrambling never require either the position or the shape of the objects to be descrambled.

Index Terms— Privacy, Surveillance, Random codes, AC coefficients

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, encipherment or scrambling are effective and practical approach [1–5]. Meanwhile, video surveillance systems or observers are usually required to recognize or detect moving objects (MOs) from the scenes taken by a camera [3–5]. It is, however, quite difficult for the conventional detection methods to detect MOs from scrambled videos.

This paper proposes a method that scrambles Motion JPEG (MJ) coded videos and also detects MOs from scrambled videos, for video surveillance systems with a stationary camera. By using the scrambling algorithm that never change the characteristics used for the MOs detection, the proposed method detects MOs from not only the original but also scrambled videos. This method directly scrambles codestreams so that the coding efficiency never change. Since this method can re-detect MOs from a scrambled video, the method can descramble MOs without reference such as the shape and position of MOs. In addition, by this re-detection feature, the method can further scramble the MOs for security.

2. BACKGROUND

This section shows the system description and mentions the codestream structure of MJ.

2.1. System Description

The proposed method has two feature, i.e., scrambling of MJ videos and MOs detection from a scrambled video. The assumptions and requirements of the proposed method are summarized below.

1. System
   (a) A stationary camera takes videos.
   (b) A video is coded by MJ.
2. Scrambling
   (a) MJ coded videos are scrambled.
   (b) A scrambled codestream is losslessly descrambled to the original codestream.
   (c) Scrambling never change the length of a codestream.
   (d) Descrambling never require reference information like the shape and position of the target entity to be descrambled.
3. MOs detection
   (a) MOs are directly detected from a scrambled video.
   (b) A scrambled video is never decoded.
   (c) The background frame is never used.

2.2. Motion JPEG

In this paper, the coding technology used in the system is assumed to be MJ. Since MJ encodes each frame of a video by JPEG [6], JPEG encoding algorithm is briefly described in this section.

In JPEG encoding, pixel values in an original image is shifted and divided into 8×8-sized non-overlapped blocks, referred to as discrete cosine transform (DCT) blocks in this paper. The two dimensional DCT, then, is applied to each block to produce one DC and 63 AC transformed coefficients. All the coefficients are scalar quantized according to a quantization table scaled by Q-factor. Finally, the quantized coefficients are coded by an entropy coder with a Huffman table. The AC coefficients are directly encoded, whereas the difference between two consecutive DCT blocks is encoded for DC coefficients.

An entropy code consists of a Huffman code and an additional bits. The Huffman code represents the amplitude, whereas the positive and negative sign is represented by the most significant bit of the additional bits. Thus, a modification of an amplitude often changes the codestream length, whereas the sign modification never change the length. For DC coefficients, the effect by the sign modification spreads over multiple DCT blocks, because the difference between two consecutive DCT blocks is encoded.

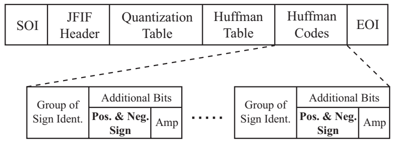


Fig. 1. JPEG codestream [6].

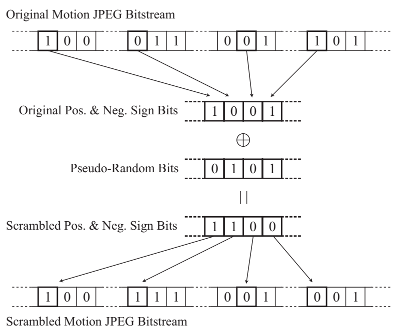


Fig. 2. The principle of the proposed scrambling process.

Figure 1 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 the quantization table for scalar quantization and the Huffman table for entropy encoding. Then, the entropy-coded DCT coefficients are put. The end of image (EOI) marker follows the last byte of a codestream.

3. PROPOSED METHOD

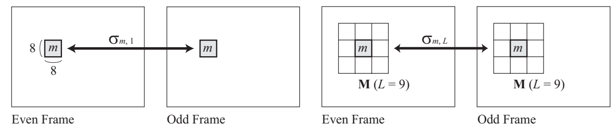
The proposed method consists of two processes; the video scrambling and the moving objects detection. This section describes each of them and introduces two application examples of the proposed method for video surveillance systems.

3.1. Scrambling of Video

This method scrambles MJ coded videos by utilizing DCT signs. Figure 2 shows the principle of the method. For the i-th frame that consists of $N_1 \times N_2$ pixels, the following algorithm is applied.

1. With a pseudo random bit (PRB) generator and seed r_i, an $N_1 N_2$-length PRB sequence consisting of one with probability p_i and zero with probability $(1 - p_i)$ is generated.
2. Acquire the DCT sign bits from the MJ codestream. It complements the sign of zero coefficients with zeros so that the length of the DCT sign bit sequence becomes $N_1 N_2$.
3. Bitwise XOR is applied on each bit of the DCT sign bit sequence and its corresponding bit in the PRB sequence.
4. Replace the DCT sign bits in the MJ codestream with the corresponding bits in the XOR-ed DCT sign bit sequence.

The DCT sign bits are randomly inverted by XOR with the PRB sequence, so the decoded frame is scrambled. This scrambling never change the length of a codestream. The scrambling strength is controllable by one's probability p_i. The proposed method leaves DC sign bits not modified, because an inversion of the DC sign bits affects over many DCT blocks.



(a) $\sigma_m = \sigma_{m,1}$. (b) $\sigma_{m,L}$.

Fig. 3. Moving objects detection.

For the j-th frame where $j \neq i$, the PRB sequence is generated with seed r_j and one's probability p_j where

$$r_j = r_i, \qquad p_j = p_i, \tag{1}$$

so that MOs detection is capable. Hereafter, for conciseness, the seed given to a PRB generator and the one's probability are represented by r and p, respectively.

The scrambling algorithm described above is re-applied to scrambled videos for descrambling. By XOR with the PRB sequence generated with seed r and one's probability p that are identical with those of the scrambling, the randomly inverted sign bits return to those original state.

3.2. Moving Objects Detection

This method detects MOs in each DCT block by using sign of AC coefficients of two adjacent frames. Here, MOs detection between a set of even and odd frames is considered. MOs are detected in the m-th DCT block by the following measure, i.e.,

$$\sigma_m = \frac{\sum_{n=1}^{63} s_0(m,n)s_1(m,n)}{\sum_{n=1}^{63} |s_0(m,n)s_1(m,n)|}, \tag{2}$$

where $s_0(m,n)$ is the sign of the n-th AC coefficient in the m-th DCT block of an even frame and $s_1(m,n)$ represents that of the corresponding odd frame. Sign $s_i(m,n)$ is given by

$$s_i(m,n) = \begin{cases} -1, & c_i(m,n) < 0 \\ 0, & c_i(m,n) = 0 \\ 1, & c_i(m,n) > 0 \end{cases}, \tag{3}$$

where $c_i(m,n)$ is the corresponding AC coefficient. The denominator of Eq. (2) is the number of positions in which coefficients of two adjacent frames are non-zero, and the numerator is the sum of values given by the following rule on corresponding coefficients of two adjacent frames; 1: coefficients are non-zero and the same, 0: either or both are zero, and -1: coefficients are non-zero and different. If all pairs of corresponding coefficients in two adjacent frames are non- and the same, σ_m gets its maximum value, i.e., one. Several works have formulated that σ_m measures the similarity between two images, as the special form of DCT sign phase correlation [7].

Further robust MOs detection is served by taking into account that a MO may cover several DCT blocks. In this detection, the measure is defined by

$$\sigma_{m,L} = \frac{\sum_{m' \in \mathbf{M}} \left(\sum_{n=1}^{63} s_0(m',n)s_1(m',n) \right)}{\sum_{m' \in \mathbf{M}} \left(\sum_{n=1}^{63} |s_0(m',n)s_1(m',n)| \right)}, \tag{4}$$

where $\mathbf{M}$ is the set consisting of m-th DCT block and its surrounding DCT blocks and L is the number of DCT blocks in $\mathbf{M}$, c.f. Fig. 3.

That is,

$$\sigma_{m,1} = \sigma_m. \tag{5}$$

When $\sigma_{m,L} < T$ where T represents the user-defined threshold, it is decided that MOs are in the m-th DCT block.

Since the scrambling process described in Sect. 3.1 inverts DCT sign bits with one single PRB sequence for all frame pairs, the relation of DCT signs between adjacent frames is never changed. Thus, the proposed process detects MOs from scrambled frames as well as unscrambled. Though one single PRB sequence is used for all frames for its simplicity, either seed r or one's probability p can be changed between different frame pairs.

3.3. Application Examples for Video Surveillance Systems

The proposed method detects MOs from a scrambled video. This section describes two application examples for video surveillance systems that utilizes the above mentioned feature.

3.3.1. Double Scrambling of Moving Objects

This method makes the system more secure by further scrambling of MOs that are detected from whole scrambled frames. The double scrambling algorithm is as follows:

1. A video is scrambled by the process proposed in Sect. 3.1 with the PRB sequence generated with r_1 and p_1.
2. MOs are detected from the scrambled video by the process proposed in Sect. 3.2 with threshold T and number of DCT blocks L.
3. Detected MOs are scrambled with the PRB sequence generated with r_2 and p_2, where $r_1 \neq r_2$ and $p_1 \neq p_2$.

The descrambling process of this application is as follows:

1. Double scrambled MOs are detected from the video by the process proposed in Sect. 3.2 with T and L.
2. Double scrambled MOs that are detected in the previous step are descrambled with the PRB sequence generated with r_2 and p_2.
3. Scrambled frames are descrambled with the PRB sequence generated with r_1 and p_1.

This double scrambling improves resistance to malicious descrambling and simultaneously varies the scrambling strength according to the part. Moreover, since MOs can be detected from a double scrambled video, descrambling never require reference information like the shape and position of MOs.

3.3.2. Partial Descrambling of Scrambled Video

This method can descramble only MOs from a scrambled video by detecting and re-scrambling of MOs. The process is as follows:

1. A video is scrambled with the PRB sequence generated with r_1 and p_1.
2. MOs are detected from the scrambled video with T and L.
3. Detected MOs are re-scrambled with the PRB sequence generated with r_1 and p_1.

Figure 4 shows the concept of this application. The process described above is for pattern 1 in Fig. 4 in which only MOs are descrambled. Pattern 2 in Fig. 4 in which only background is descrambled is achieved by re-scrambling the whole frame excepting for detected MOs. In addition, the descrambled video is served by re-scrambling the whole frame shown as pattern 3 in Fig. 4.

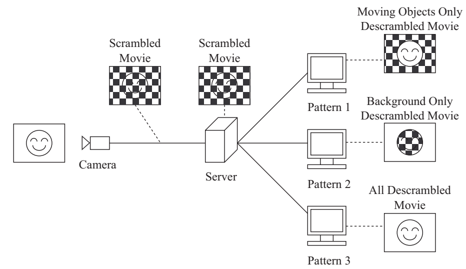


Fig. 4. Partial descrambling system.

Table 1. Conditions.

Video	240×320 pixels, grayscale, 8 bits/pixel 30 frames/sec, 200 frames
Q-factor	25, 50, and 100

3.4. Features

This section lists three main features of the proposed method.

3.4.1. Codestream Domain Processing

This method never require decoding and re-encoding of codestreams because of direct processing of codestreams. The method reduces procedure in comparison with spatial domain-processing methods [4, 5]. Moreover, the proposed method scrambles a video by inverting the sign of AC coefficients. As mentioned in Sect. 2.2, the sign is encoded as one bit independently from its corresponding magnitude, the sign bit inversion in the proposed method never change the codestream length, whereas spatial domain-processing methods [4, 5] generally degrade the coding efficiency.

3.4.2. Moving Objects Detection

Since the scrambling never change the characteristics for MOs detection, this method detects MOs from scrambled videos as well as unscrambled videos. This feature gives abilities to the method; scrambling of MOs after MOs detection, MOs detection after scrambling, and partial descrambling. Moreover, the proposed method detects MOs based on the difference between two adjacent frames, it naturally detects multiple MOs and is resistant to MOs occlusion. In addition, since the background image is not used in this method, either storing or updating of the background image is not required.

3.4.3. No-reference Descrambling

Since this method scrambles a video by an inversion of the sign bit of AC coefficients with a PRB sequence, re-inverting of the sing bits with the identical PRB sequence recovers the original state of sign bits. Thus, the method descrambles a scrambled video without loss and without coding efficiency degradation. As the method can detect MOs from a scrambled video, reference information such as the shape and position of MOs are not required. Whereas the conventional methods [3–5] require such reference.

4. EXPERIMENTAL RESULTS

Conditions are summarized in Table 1. Figure 5 shows the 180th and 181st frames of the video.

(a) Frame no.180. (b) Frame no.181.

Fig. 5. Frames from the video for evaluation.

(a) $p = 0.3$. (b) $p = 0.5$.

Fig. 6. Scrambled frames (Q-Factor = 50).

4.1. Scrambling

The images shown in Fig. 5 are encoded by MJ and then is scrambled by the proposed scrambling as shown in Fig. 6. It is hard to recognize the scene in frames. Ones' probability p controls the scrambling strength as shown in Figs 6 (a) and (b).

4.2. Moving Objects Detection

Figure 7 (a) shows the MOs detected by the proposed detection with number of DCT blocks $L = 1$ and threshold $T = 1$. MOs are detected among white-filled DCT blocks in which $\sigma_{m,L} < T$. Since the proposed MOs detection works for both unscrambled and scrambled videos, the results are the same for both videos. Figure 7 (b) shows the result with $L = 9$ and $T = 0.9$. The detection accuracy is improved by using the surrounding DCT blocks. From Fig 8, MOs are detected from videos with various compression ratios.

4.3. Partial Descrambling

Figure 9 (a) shows the frame in which MOs detected from the scrambled frame are descrambled. Whereas Fig. 9 (b) shows the frame in which the background is descrambled. The proposed scrambling can descramble a frame partially as shown in Fig. 9.

It is confirmed that the length of codestreams are not changed by scrambling, i.e., the coding efficiency hold in the proposed method.

(a) Number of DCT blocks $L = 1$ and threshold $T = 1$. (b) Number of DCT blocks $L = 9$ and threshold $T = 0.9$.

Fig. 7. Detected moving objects (Q-factor: 50). Moving objects are detected over white-filled DCT blocks.

(a) Q-Factor: 25. (b) Q-Factor: 100.

Fig. 8. Detected moving objects ($L = 9$ and $T = 0.9$).

(a) Moving objects. (b) Background.

Fig. 9. Partial descrambling (Q-factor: 50, $L = 9$ and $T = 0.9$).

5. CONCLUSIONS

This paper has proposed a scrambling method for MJ videos and MOs detection from scrambled videos. The proposed scrambling inverts bits of a codestream, so the coding efficiency is never changed. Since the proposed MOs detection detects MOs from both unscrambled and scrambled videos, security improvement by double scrambling of MOs and partial descrambling can be served. Descrambling never require reference information such as the shape and position of MOs. Moreover, all procedure in the proposed method are directly applied to codestreams, the proposed method is fast.

Computer vision-based cryptanalysis of the scrambled videos is a further work.

REFERENCES

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

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

[3] F. Dufaux and T. Ebrahimi, "Scrambling for video surveillance with privacy," in *Proc. IEEE Workshop PRIV*, 2006.

[4] I. Kitahara, K. Kogure, and H. Hagita, "Stealth vision for protecting privacy," in *Proc. IAPR ICPR*, 2004, pp.404–407.

[5] 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*, 2006.

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

[7] 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*, 2007, pp.1237–1240.