LETTER Special Section on Digital Signal Processing

Scrambling of MPEG Video by Exchanging Motion Vectors

Ayuko TAKAGI[†a)] ***and*** **Hitoshi KIYA**[††], ***Members***

SUMMARY A method of scrambling MPEG video by exchanging the motion vector (MV) in the MPEG bitstream is proposed. It deals directly with the MPEG bitstream and exclusive MPEG encoders are unnecessary. The size of the scrambled bitstream does not increase and image quality is maintained after descrambling. Moreover, the structure of the MPEG bitstream is maintained and can be decoded with a standard MPEG video decoder. We demonstrate the effectiveness of this method through simulation results that reveal unchanged image quality and size of bitstreams.
key words: *MPEG video, image scrambling, motion vector*

1. Introduction

In recent years, the distribution of digital images as a commercial product has been increasing due to improved computer performance, and the broadband transmission network. Since digital data can be reproduced without degrading quality, the protection of copyright has become an important issue. The techniques used to protect copyright of digital images can generally be classified into three types: (a) digital watermarking [1], [2], (b) data encryption of the entire image, and (c) image scrambling (partial encryption) [3]–[6]. The method proposed in this paper can be classified under (c), i.e., image scrambling.

Digital watermarking embeds the copyright information in the image and deters the illegal use. However, as watermarked images nearly have the same quality as the original, it is difficult to prevent illegal use and reproduction beforehand. Encrypting the entire image, on the other hand, is the strongest protection for image information, even though the content of the image cannot be identified without decryption and there are restrictions in treating and referencing the image data. Moreover, image data, specially video, are large and encrypting the entire image requires a great deal of computation.

Image scrambling is done by partially encrypting the image data. Since encryption is done to selected data in the entire image, little computation is required for encryption. Also, by selecting the data to be encrypted, some features of the original image can still be recognized, and this is called scrambling. Image scrambling is expected to be an effective method of protecting copyright in the image data circulated in commercial use. The, image provider first scrambles the image data and degrades the quality to a degree where only the outline of content can be confirmed. Then, the scrambled image is distributed through a medium, such as networks and CD-ROMs. Meanwhile, users download the scrambled image and confirm the outline. If they want to see the entire image, they obtain the decryption key from the provider and descramble the image.

In this paper, a method of scrambling MPEG video is proposed. There are two ways of scrambling image data. The first is by directly scrambling the original image and the second is based on image compression, which is our method. Here, scrambling is done through motion vectors and does not need the use of an exclusive MPEG encoder. Also, MPEG bitstream data do not increase, and the original image can be obtained after descrambling. Scrambled MPEG bitstreams can be decoded with a standard MPEG decoder without descrambling. In our proposed method, Intra frames are not modified and can be used for database or image search [7]. Also, the proposed method maintaines the standard MPEG bitstream structure, it can be combined with other scrambling method for MPEG bitstream.

2. Scrambling with Motion Vectors

There are two requirements for scrambling MPEG video and distributing it as a commercial product. The first is :

(i) Preserve the MPEG bitstream structure.
To distribute MPEG video as a commercial product, the scrambled bitstream must be decoded with a standard MPEG decoder. If exclusive decoders are needed, it will disrupt the distribution of the product.
The second is:

(ii) No increase in bitstream size.
There should be no increase in bitstream size after the scrambling process as this is a disadvantage in distributing the MPEG video, especially in dealing with media with limited capacities such as CD-ROMs. Furthermore, if the bitstream is increased with scrambling process, image quality is sacrificed in maintaining the same size.

This paper discusses a method of directly scrambling an MPEG bitstream. There are several methods that can be used to directly encrypt MPEG video [3], [6]. When encrypting MPEG video, modifications are done either to the DCT coefficients [3] or the motion vectors [6]. The

Manuscript received November 28, 2003.
Manuscript revised March 14, 2004.
Final manuscript received April 14, 2004.
[†]The author is with the Dept. of Informatics and Media Technology, Shohoku College, Atsugi-shi, 243–8501 Japan.
[††]The author is with the Dept. of Electrical Engineering, Tokyo Metropolitan University, Hachioji-shi, 192–0397 Japan.
a) E-mail: a-takagi@shohoku.ac.jp

scrambling effect on methods which modify the DCT coefficients are mainly caused by the change in I frames. The image degradation in I frames propagates on P and B frames. Therefore, if I frames are not scrambled, the scrambling effect on P and B frames are small. To scramble P and B frames effectively, we use information of the motion vector for scrambling. As modifications did not destroy the MPEG bitstream structure, condition (i) was satisfied. In [6], during the encoding procedure, the motion vectors are randomly modified and the modified vectors are used to generate the MPEG bitstream. This conventional method preserves the compression ratio of the original bitstream. However, the modification to motion vectors decreases the coding efficiency, and decreased coding efficiency increases bitstream size or degrades image quality. The variable length code for motion vector is prescribed based on its frequency of occurrence. Motion vectors in the MPEG system are coded with difference coding, and the difference between adjacent macro blocks is mostly small. To code motion vectors efficiently, fewer bits are assigned to small values and more bits are assigned to large values. In the conventional method, the motion vectors are modified randomly, and this causes large differences between adjacent motion vectors. This increases bitstream size. The motion vectors are also modified during the encoding procedure in the conventional method, and although target bitrate is maintained, the size of binary codes representing motion vectors is increased and image quality is degraded. To avoid these problems, proposed method uses the binary codes of motion vectors for scrambling to satisfy condition (ii) to maintain bitstream size.

3. Proposed Method

Here, we describe the method we propose for both scrambling and descrambling MPEG video using the binary codes of motion vectors. Figure 1 is a diagram outlining the method where the MPEG bitstream is directly scrambled/ddescrambled.

Step 1 Parse input MPEG bitstream
The input MPEG bitstream is parsed and the binary codes for the motion vector, and their positions are detected. Here, the number of motion vectors and their binary codes are defined as n and $MV(x)$, $(x = 0, 1, ..., n-1)$, respectively.

Step 2 Generate table $T(x)$ and $T^{-1}(x)$
Table $T(x)$ is used for scrambling and $T^{-1}(x)$ is used for descrambling. They are generated through a random sequence generator and encryption key k. $T(x)$ and $T^{-1}(x)$ must satisfy the equation

$$T^{-1}(T(x)) = x. \quad (1)$$

Step 3 Re-order motion vectors with generated tables.
The re-ordered motion vector codes are defined as $MV'(x)$. $MV'(x)$ is generated with the next equation using $T(x)$ and $T^{-1}(x)$. The equation

$$MV'(x) = MV(T(x)), \quad (2)$$

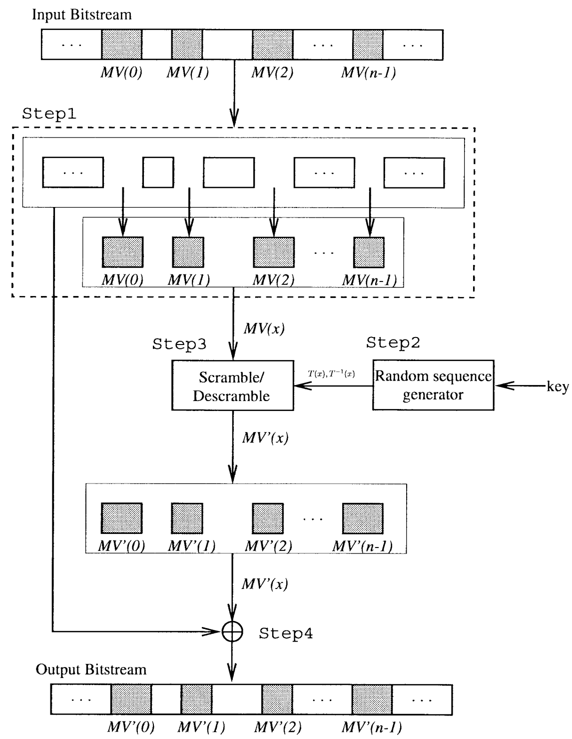

Fig. 1 Proposed method of scrambling and descrambling.

is used for scrambling and

$$MV'(x) = MV(T^{-1}(x)) \quad (3)$$

is used for descrambling.

Step 4 Replace $MV(x)$ with $MV'(x)$.
Finally, $MV(x)$ is replaced with $MV'(x)$. In some cases, the bit lengths of $MV(x)$ and $MV'(x)$ are not the same, and errors occur at the slice headers. Therefore, if required, some additional '0' bits are added before the slice headers to construct a valid MPEG bitstream.

We scrambled/descrambled an MPEG bitstream directly with the same process and there was no need for a specific MPEG codec. The number of motion vectors n was decided arbitrarily. A large n requires a large memory to maintain bitstream information. The n could be fixed, or scrambling could be done in frames with various n. This information must be sent to the decoder with key k descrambling for appropriate. There was little increase in the bitstream size, and this was extremely small compared with the conventional method. The DCT coefficients, which contribute to image quality, were not modified and the original image was obtained by descrambling. As the proposed method only modified the order of the motion vector codes, the MPEG bitstream structure was maintained and could be decoded with a standard MPEG decoder. Since only motion vectors were modified, the effect of scrambling is small on video sequences with less motion. In those cases, the proposed method can be combined with other scrambling methods for MPEG bitstream, since the MPEG bitstream structure is maintained.

4. Simulation

We did simulation to test the effectiveness our method with MPEG software [8]. The video sequences and their sizes, and the target bitrate were as follows:

- news, silent (176 × 144, 128 [kbps])
- paris, tempete (352 × 288, 512 [kbps]])
- football (704 × 240, 1 [Mbps])
- mobile & calendar (720 × 576, 1.5 [Mbps]).

The GOP structure was IBBPBBPBBPBBI and we used 37 frames. The rate was 30 frames/sec.

We scrambled the MPEG bitstreams generated under the above conditions. Scrambling was done on each frame. Table 1 lists the average PSNR of P and B frames for both original and scrambled bitstreams. Table 2 lists the percentage of existing motion vectors. The percentage of existing motion vectors is the rate the actual number of motion vectors in the maximum number of motion vectors which may exist in each frame. Figure 2 shows the PSNR of each frame before and after the video sequence "paris," and the fourth frame (B picture) in Fig. 3 were scrambled. As we can see in Tables 1 and 2, the higher the percentage of existing motion vectors, the lower the image quality in the scrambled video. The motion vectors are modified, so that the more there are of these, the more modifications to MPEG bitstreams, increasing the degree of scrambling with our method. We can see from Fig. 2, that the image quality of B pictures is lower than that of P pictures in the scrambled bitstream. This means that B pictures are scrambled more than P pictures. This is because the estimates of motion for B pictures are bi-directional and they have more motion vectors than P pictures. As we can see from Fig. 2, I pictures are not modified.

Table 3 compares and shows the average PSNR from all frames between the conventional and proposed methods after descrambling. The image quality after descrambling degraded with the conventional method. The coding efficiency for motion vectors decreased for the conventional method, and as a result, the coding efficiency for the entire bitstream decreased. Table 3 confirms that our method

Table 1 Average PSNR of P and B pictures for original and scrambled bitstreams [dB].

sequence	PSNR [dB] original	PSNR [dB] scrambled
news	27.27	22.55 (−4.72)
silent	27.53	20.40 (−7.13)
paris	23.29	17.13 (−6.16)
tempete	22.04	15.31 (−6.73)
football	26.81	16.33 (−10.48)
mobile	30.69	19.14 (−11.55)

Table 2 Percentage of existing motion vectors [%].

sequence	percentage of MV[%]
news	13.21
silent	16.42
paris	26.05
tempete	36.51
football	42.35
mobile	40.68

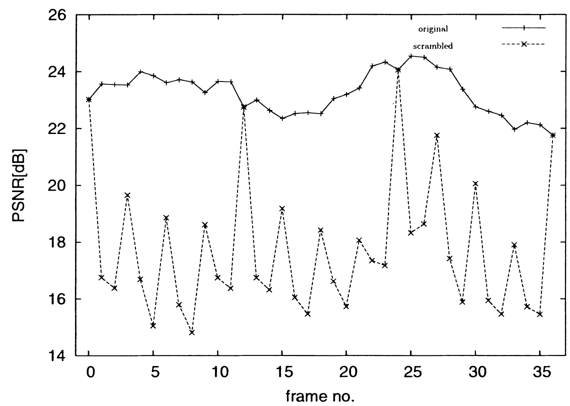

Fig. 2 PSNR comparison for each frame of "paris" between original and scrambled bitstreams [dB].

(a) Original

(b) Scrambled

Fig. 3 Scrambled picture using proposed method ("paris" frame no. 4 (B picture)).

Table 3 Average PSNR from all frames after descrambling [dB].

sequence	original	after descrambling	
		conventional	proposed
news	27.21	26.21 (−1.00)	27.21 (0.00)
silent	27.46	26.78 (−0.68)	27.46 (0.00)
paris	23.25	22.84 (−0.41)	23.25 (0.00)
tempete	21.99	21.78 (−0.21)	21.99 (0.00)
football	26.95	26.41 (−0.54)	26.95 (0.00)
mobile	30.59	28.90 (−1.69)	30.59 (0.00)

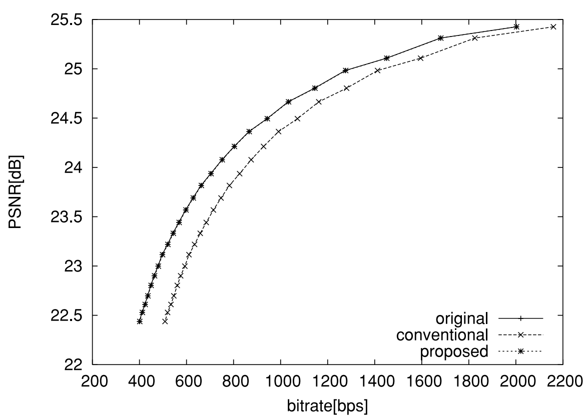

Fig. 4 Comparison of bitrate and PSNR for video sequence "paris."

restores the original image after descrambling.

Figure 4 compares bitrate and PSNR for the video sequence "paris." In the conventional method, coding efficiency is decreased, and a greater bitrate is needed to obtain the same image quality. In other words, image quality with the conventional method has degraded compared with ours at the same bitrate. There is a greater influence on image quality at lower bitrates. The bitstream size with the proposed method has increased, but this is negligible compared with the original bitstream.

5. Conclusion

This paper proposed a method of scrambling MPEG video using motion vectors. It re-orders the motion vector codes in the MPEG bitstream through a randomly generated table. The increase in the scrambled bitstream size is negligible, and the original image is obtained by descrambling. Scrambled MPEG bitstreams can be decoded with a standard MPEG decoder. For the future work, we would like to modify this method for scrambling I pictures.

References

[1] A.M. Alattar, E.T. Lin, and M.U. Celik, "Digital watermarking on low bit-rate advanced simple profile MPEG-4 compressed video," IEEE Trans. Circuits Syst. Video Technol., vol.13, no.8, pp.787–800, Aug. 2003.
[2] G.C. Langelaar, I. Setyawan, and R.L. Langendijk, "Watermarking digital image and video data," IEEE Signal Process. Mag., vol.17, no.5, pp.20–46, Sept. 2000.
[3] C. Shi and B. Bhargava, "A fast MPEG video encryption algorithm," Proc. 6th ACM International Conference on Multimedia, pp.81–88, 1998.
[4] H. Fujii and Y. Yamanaka, "Digital image scrambling method for information distribution," Trans. IPSJ, vol.38, pp.1945–1955, Oct. 1997.
[5] K. Ando, O. Watanabe, and H. Kiya, "Partial-scrambling on still images based on JPEG2000," Internationl Conference on Information, Communications, and Signal Processing, vol.2F2.5, Oct. 2001.
[6] J.-C. Yen and J.-I. Guo, "A new MPEG/encryption system and its VLSI architecture," Proc. International Symposium on Communications, pp.215–219, Nov. 1999.
[7] A. Takagi and H. Kiya, "Scrambling of MPEG video suitable for image search," Proc. 18th Picture Coding Symposum of Japan, vol.P.5-19, pp.99–100, Nov. 2003.
[8] MPEG Software Simulation Group, "MPEG-2 encoder/decoder version 1.2," [Online] Available www:http://www.mpeg.org/MPEG/MSSG/