Apply Semi-Fragile Watermarking to Authentication of Compressed Video Data

Da-Jinn Wang[1], Tsong-Yi Chen[2,*], Thou-Ho (Chao-Ho) Chen[2], Chien-Hua Huang[2], and Chung-Yih Lee[2]

[1] Department of Information Management, National Kaohsiung Marine University,
Kaohsiung 811, Taiwan, ROC

[2] Department of Electronic Engineering, National Kaohsiung University of Applied Sciences,
Kaohsiung 807, Taiwan, ROC
chentso@cc.kuas.edu.tw

Received 11 April 2006; Revised 26 May 2006 ; Accepted 12 June 2006

Abstract. This paper proposes an effective technique, which can detect malicious manipulations under video lossy compressing data (e.g. H.263) and still-image lossy compressing data (e.g. JPEG). A block-classification strategy is used to divide DCT-blocks into the flat-blocks and the normal-blocks. Simple features of the both blocks are embedded invisibly. For later authentication, the watermarked frame is put back to the compressed bit-streams. The proposed method can detect and locate alterations of the tampered frame whether it is stored again into original bit-streams or not. This goal is for detecting tamper area and surviving most video lossy compression. Experimental results show that the proposed technique can detect various tampered areas and hence provides an effective image authentication for lossy compressed image and video in DVR (Digital Video Recorder) System.

Keywords: watermarking, H.263, JPEG, image authentication, compressed video

1 Introduction

In the past, several techniques and concepts based on data hiding or steganography have been designed as a means for tamper detection in digital images and for image authentication - fragile watermarks, semi-fragile watermarks, and self-embedding [1]-[13]. One class of authentication watermarks is a form of semi-fragile watermarks. Such watermarks are marginally robust and are less sensitive to pixel modifications. Thus, it is possible to use them for quantifying the degree of tamper and distinguish simple LSB shuffling from malicious changes, such as feature adding and removal.

Lin and Chang [0]-[0] present a relation-based watermarking techniques for image authentication to extract a digital signature by using the invariant relation existing between any two DCT coefficients which were located at the same position of two different 8x8 blocks for making the image authentication system to be tolerable to JPEG compression. They dedicated themselves to exploring the operation in a JPEG-based system. Lu, et al. [0] propose to use the "structure" of an image as a digital signature. In their proposed scheme, the structure of an image's content is composed of a number of parent-child pairs located at the multiple scales in the wavelet domain. It is expected to be robust against content-preserving manipulations and fragile against content-changing manipulations.

More and more applications for tamper detection include authentication of digital data for courtroom evidence and copyright protection in DVR System. The reason is that the captured frames from DVR System could be malicious tampered when they spread over internet or database. In this paper, we propose an effective technique, which can detect malicious manipulations under lossy compressed video data (e.g. H.263) and lossy compressed still-image data (e.g. JPEG).

2 Proposed Data Embedding Scheme

The proposed watermarking process is based on 8x8-block DCT transform. The watermarking process performs on the Y component (luminance) of YCbCr color model. The basic process described briefly in the following.

At first, we hide the watermark into the chosen non-zero quantized AC coefficient (we called *NQAC*), denoted as $NQAC_i$, by backward zigzag-scan in each block. We set a threshold τ to divide the input frame into normal-

* Correspondence author

blocks and flat-blocks. If the number of *NQAC* of the block is more than τ, the block is named a normal-block, the other one is called a flat-block. The following subsections will illustrate how the watermark embedded into these two types of block.

2.1 Normal-Block Embedding

In this paper, we also call the normal-blocks embedding as *Self-Embedding*. To consider about the speed of tamper detection, we can embed feature bit "1" into extra space as a key, defined to *c*. We call the feature bit as *block classification bit*.

We extract other feature bits, CF_a, from quantized DC coefficient to embed into the chosen quantized AC coefficients. To emphasize the security of embedded feature bits, a fast one-dimensional pseudorandom number generating approach is used to shuffle the feature bits to disperse their energy relationship as feature bits f_i via exclusive-or operation, where *i* denotes *authentication strength*. A degree of authentication according to *authentication strength*, but more authentication bits will make more degrade of image quality, vice versa. This is a trade-off between the quality of video and robustness of tamper detection. We embed the feature bits into LSB of the chosen nonzero quantized AC coefficients a_i by zigzag-scan.

The reason is that when the quantized AC coefficients are chosen to embed data, the artifacts caused by the replaced LSBs in the quantized coefficients will appear mostly in the sharp features in the image frames, like edges, lines, noise, etc., and will mostly be imperceptible. The quantized AC coefficients will be altered by data embedding as follows:

$$a_i^{'} = \begin{cases} sign(a_i) * a_i, & if\ \ Bit_0(|\ a_i\ |) = f_i \\ sign(a_i) * AF(a_i), & if\ \ Bit_0(|\ a_i\ |) \neq f_i \end{cases} . \qquad \textbf{(1)}$$

The bit_0 means the least significant bit of one DCT coefficient, and "+1" or "-1" will be selected according to the sign of a_i. An *adjustment function*, *AF*, has two major features. In the first feature, the quantized AC coefficient "1" will alter to "0" while the embedding feature bit *fi* is "0". This will generate a data-extracting fault due to the absence of quantized coefficients. The second feature is to transform the quantized AC coefficient 2 or -2 into 1 or -1 while the embedding feature bit is "1". This is an additional adjustment due to the result of the first feature. Hence, the *adjustment function* is shown as follow:

$$AF(a_i) \Rightarrow \begin{cases} Bit_1(|\ a_i\ |) = f_i \oplus 1 & ,if\ |\ a_i\ | = 1 \\ Bit_1(|\ a_i\ |) = f_i \oplus 1 & ,if\ |\ a_i\ | = 2 \\ Bit_0(|\ a_i\ |) = f_i \end{cases} . \qquad \textbf{(2)}$$

For example, according to the result of *adjustment function*, the quantized AC coefficient "1" is "1", "-2" is "-1", "3" is "3", "-4" is "-5" while embedding feature bit is "1". Other quantized AC coefficient "1" is "2", "-2" is "-2", "3" is "2", "4" is "4" while embedding feature bit is "0".

12	13	13	12	11	98	94	95
12	13	13	13	12	11	10	97
13	13	13	14	13	12	10	95
13	12	13	13	14	13	11	92
12	12	12	12	13	12	10	93
12	12	12	12	12	12	11	10
13	13	12	12	11	11	12	12
13	13	13	12	11	11	13	14

(a)

13	13	13	12	11	97	91	90
13	13	13	13	11	10	97	92
13	13	13	13	12	11	10	95
13	13	13	13	13	12	10	96
13	12	12	13	13	12	11	10
13	12	12	12	12	12	11	11
13	13	13	12	12	12	12	12
13	13	13	13	12	12	13	14

(b)

Fig. 1. (a) 8x8 pixel-block, number of non-zero AC coefficients = 5 (by H.263, QP = 7), (b) watermarking to (a), authentication strength is 5 (worst-case)

2.2 Reduction of Clipping-Errors

The quantized AC coefficient could be corrupted while enforcing normalization in spatial domain. The clipping error may be occurred due to that all pixels must be normalized to [0,255] in spatial domain. This is a *false alarm* for tamper detection. Find out maximum clipping error, and calculate its basis image from the Formula 3. The basis image with a scaling factor as a weight-mask (see Formula 5). Using the sign of the weight-mask to map the

chosen quantized AC coefficient. Then, the maximum clipping error would not rise any more. For example, as the Figure 2 shown, the pixel of coordinates (1,7) is 253 that into the maximum clipping error (269) will occur, and it will affect the chosen quantized AC coefficient. The basis image *B(x,y;u,v)* is as

$$B(x,y;u,v) = \cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \quad . \tag{3}$$

from IDCT (inverse discrete cosine trans- form):

$$f(x,y) = \sum_{x=0}^{N}\sum_{y=0}^{N}\alpha(u)\alpha(v)C(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \quad . \tag{4}$$

The basis image with a scaling factor $\alpha(u)\alpha(v)$:

$$\alpha(u)\alpha(v)B(x,y;u,v) \quad . \tag{5}$$

18	16	16	17	17	18	22	25
12	12	13	19	23	24	25	25
45	80	11	19	24	24	25	25
55	56	10	20	24	23	23	24
72	49	11	19	22	23	24	25
85	75	14	17	19	24	25	13
88	69	10	12	13	16	16	13
66	75	94	66	68	80	90	74

(a)

18	16	16	17	16	18	22	25
12	12	13	19	23	24	26	26
44	80	11	19	24	24	25	24
54	54	10	19	24	23	23	24
71	47	10	19	22	23	24	25
84	75	14	17	19	24	25	23
87	68	10	12	13	16	16	13
64	74	93	84	66	79	89	72

(b)

Fig. 2. (a) 8x8 pixel-block, (b) applying H.263 non-linear quantizer on (a)

0.13	-0.17	0.16	-0.15	0.12	-0.1	0.07	-0.03
0.15	-0.2	0.19	-0.17	0.15	-0.12	0.08	-0.04
0.07	-0.09	0.09	-0.08	0.07	-0.05	0.04	-0.02
-0.03	0.05	-0.05	0.04	-0.03	0.03	-0.02	0.01
-0.12	0.17	-0.16	0.15	-0.12	0.1	-0.07	0.03
-0.17	0.24	-0.23	0.2	-0.17	0.14	-0.09	0.05
-0.16	0.23	-0.21	0.19	0.16	0.13	-0.09	0.05
-0.1	0.14	-0.13	0.12	-0.1	0.08	-0.05	0.03

(a)

16	-28	-5	1	2	2	0	0
17	-5	1	0	2	-1	-1	0
-8	1	6	-1	-2	0	0	0
7	0	3	0	0	1	1	0
-4	3	1	-1	1	0	0	0
-1	1	0	0	0	0	0	0
0	0	0	-1	-1	0	0	0
-1	0	0	0	0	0	0	0

(b)

Fig. 3. (a) weight-mask for pixel (1,7), (b) the selected embedding position, by setting authentication strength to 5

The Formula 5 is the weight-mask, which can map the 8x8 coefficient from DCT domain to spatial domain. As the Figure 3 shown, we must alter the sign of the five chosen embedded coefficients by the weight-mask. Therefore, the pixel (269) will not rise any more. It is useful for revertible capability of the chosen quantized coefficient.

Consequently, we can preprocess above two steps so that the chosen quantized AC coefficient can map back for the normal-block embedding.

2.3 Flat-Block Embedding

In order to get a best trade-off between the robust of authentication and the capacity of authentication-key, we embed only a few authentication-bits into the flat-blocks. The detail algorithm of flat-block embedding is described as follows:

Step 0: Get a block classification bit *c*. Set *c*=0 and embed *c* into extra space as a key.

Step 1: Obtain the first watermark-bit set, CF_0, from bit_5, bit_6, and bit_7 of quantized DC coefficient. Embed CF_0 to extra space. Here, bit_7 is the most significant bit of quantized DC coefficient.

Step 2: Obtain the second watermark-bit set, CF_1, from the previous pseudo-random number.

Step 3: Replace the bit_0, bit_1 of quantized DC coefficient by CF_0.

Step 4: Go to step 0, until each block is handled.

In the proposed flat-block embedding algorithm, we embed one bit as classification key, to extra space and five watermark bits to authenticate tampering blocks. It's very useful for decreasing strength of the authentication key and maintaining quality of the frame.

Review the proposed embedding algorithm, we classify each 8x8 DCT blocks into normal-block or flat-block, depend on the number of nonzero quantized AC coefficients of each block. That is, a block whose number of NQAC is more than τ is called a normal-block. The threshold τ is also called the authentication-strength. Thus, the feature-bits for embedding will always suit with the number of nonzero AC coefficients. Moreover, the classification key makes a fast authenticating process. This will be efficient to authenticate on video frames.

3 Proposed Tamper Detection Scheme

When we suspect that the protected frame is tampered by other users, we must confirm whether the frame had been malicious tampered or not. According to the proposed watermark-embedding algorithm, we also classify each 8x8 DCT blocks into normal-block or flat-block. For these two types of block, using different method to detect if a block is malicious tampered. The algorithm of proposed tamper detection scheme is shown as Figure 4, and the detail approach is described as follows:

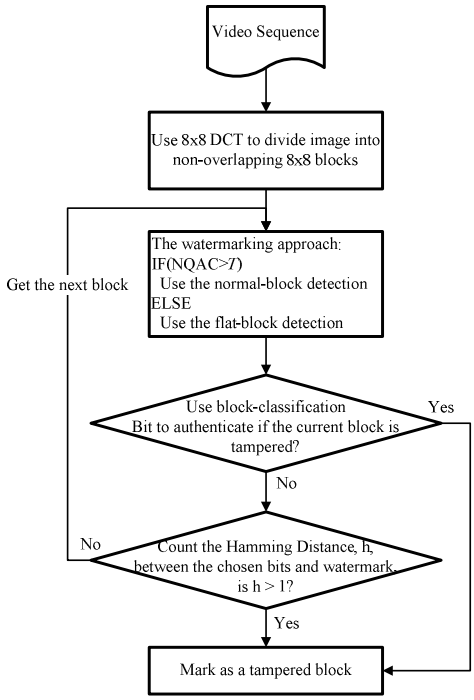


Fig. 4. The proposed algorithm of block-based video tamper detection

Step 0: Using block-classification strategy to classify 8x8 DCT blocks into normal-block or flat-block.

Step 1: Get the block classification bit, *c*, from extra space, and compare with the defined type of the current 8x8 DCT block. If the result is not match, then we can call it a tampered block. This step will speed up authentication process.

Step 2: For a normal-block, extract the feature bits from the quantized DC coefficient as CF_0, and the bit_0 and bit_1 from the embedded AC coefficients as CF_1. For a flat-block, we extract the four kinds of feature-bits sets. First, extract the feature bits from extra space as CF_0. Second, extract the bit_5, bit_6 and bit_7 from the quantized DC coefficient as CF_1. Thirdly, extract the pseudo-random number from extra space as CF_2. At last, we extract the bit_0 and bit_1 from the quantized DCT coefficient as CF_3.

Step 3: Compare the CF_0 with CF_1, and CF_2 with CF_3 by counting the Hamming-Distance, τ.

Step 4: If τ>0, then we call the current block is a malicious tampered one.

Step 5: Go to step 1 until each block has been confirmed.

4 Experimental Results

The results are experiment by an IBM- compatible PC with Pentium Ⅳ 2.4 GHz CPU, 256 MB RAM, and the platform Windows XP. The image processing software is Ulead PhotoImpact 7.0. The digital video format of

DVR System is Square pixel SIF (NTSC) (320 * 240). In this paper, the sample frames are shown as Figure 5, and the PSNR for each test frame is shown as Figure 6.

(a) (b) (c) (d) (e)

Fig. 5. The sample frames

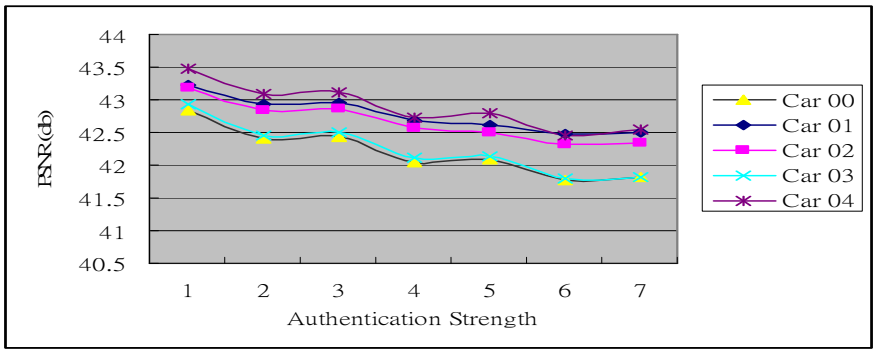
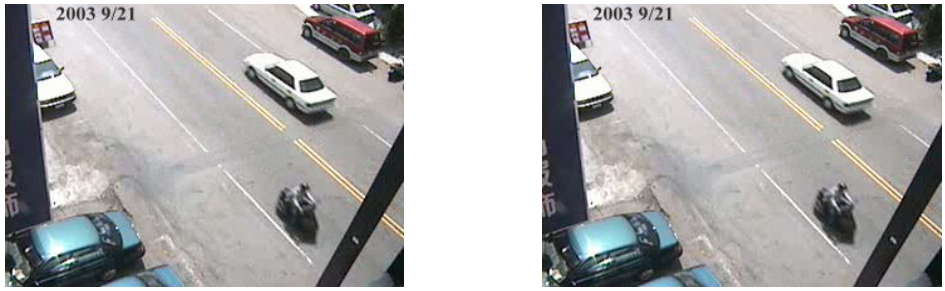
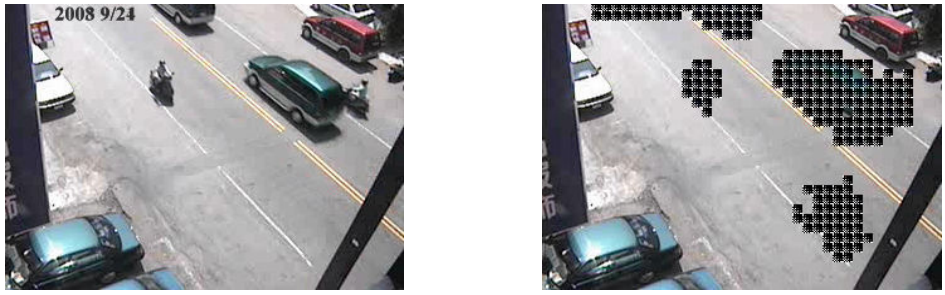


Fig. 6. The PSNR value of each sample frames. The *authentication strength* is set to 1 to 7, and QP is set to 7



(a) (b)

Fig. 7. (a) Original authentication frame, by H.263, QP=7, bit-streams=12169 Bytes, (b) Watermarked frame, PSNR=42.5db and bit-streams=12378 Bytes



(a) (b)

Fig. 8. (a) Illegal tampered frame, (b) The detected tamper area by the proposed approach

The experimental results are discussed in two aspects. One is the quality of a frame after data embedding and extra space spending. Another is tamper detection on the authentication frames. First we show the experimental results of a frame quality after data embedding, PSNR = 42.5 db, and the extra space spending is composed of the difference of original bit-streams and water- marked bit-streams, and the key. As Figure 8 shown, number of the normal-blocks is 544, and the extra bit spending of the normal-blocks needs one bit. Number of flat-blocks is

656, and the extra bit spending of the flat-blocks needs three bits. the extra space spending is (12378 – 12169) + (3 * 656 + 544) / 8 = 523 bytes.

The key of extra space spending has two cases: the best-case is that all quantized DCT-blocks of the captured frame are normal-blocks (320 * 240 / (8 * 8) = 150 bytes), and the worst-case is that all quantized DCT-blocks of the captured frame are flat-blocks (320 * 240 / (8 * 8) * 3 = 450 bytes). The region of extra space spending is (diff + 150 bytes) ~ (diff + 450 bytes). The diff denotes the difference between original bit-streams and watermarked bit-streams.

5 Conclusions

In this paper, we propose an effective technique, which can detect malicious manipulations on lossy compressed video data (e.g. H.263) and reduce the clipping-error problem. Although the quantized coefficients are more suitable than original coefficients for data embedding, it is noteworthy that the damaging problem of clipping errors could be caused by normalization in spatial domain. In this paper, we reduce the problem of clipping errors.

Besides, we cannot embed too much data into the flat (smooth) block due to the value and number of the non-zero quantized AC coefficient is small and less. So, we use a specific approach to embed the watermark into the flat-block. Further, we can try to recover the tampered frame according to the few authentication-key.

References

[1] Kwang-Fu Li, Tung-Shou Chen, and Seng-Cheng Wu, "Image tamper detection and recovery system based on discrete wavelet transformation," *Communications, Computers and signal Processing, IEEE Pacific Rim Conference on*, Vol.1, Aug. 2001, pp: 164-167.

[2] M.Wu, and B. Liu, "Watermarking for Image Authentication," *Proceedings of IEEE International Conference on Image Processing*, Oct. 4-7, 1998, Chicago, Illinois, USA, Vol. 2, pp. 437-441.

[3] Tae-Yun Chung, Min-Suk Hong, Young-Nam Oh, Dong-Ho Shin, and Sang-Hui Park, "Digital Watermarking for Copyright Protections of MPEG2 Compressed Video," *IEEE Trans. on Consumer Electronics*, Vol. 44, No. 3, Aug. 1998, pp. 895-901.

[4] C.Y. Lin, S.F. Chang, "SARI: Self-Authentication-and-Recovery Image watermarking system," *ACM Multimedia 2001 Workshops - 2001 Multimedia Conference*, 2001, pp. 628-629.

[5] C.Y. Lin, and S.F. Chang, "Semi-Fragile Watermarking for Authenticating JPEG Visual Content," *SPIE International Conf. on Security and Watermarking of Multimedia Contents II*, Vol. 3971, No.13, EI '00, San Jose, USA, Jan 2000.

[6] C.Y. Lin and S.F. Chang, "Issues and Solutions for Authenticating MPEG Video," *SPIE International Conf. on Security and Watermarking of Multimedia Contents*, Vol.3657, No.6, EI '99, San Jose, USA, Jan 1999.

[7] Chung-Shien Lu, and Hong-Yuan Mark Liao, "Structural Digital Signature for Image Authentication: An Incidental Distortion Resistant Scheme," *IEEE Transactions on Multimedia*, Vol. 5, No. 2, June 2003.

[8] L. M. Marvel, G. Hartwig, and C. G. Boncelet Jr., "Compression Compatible Fragile and Semi-Fragile Tamper Detection," *SPIE International Conf. on Security and Watermarking of Multimedia Contents II*, Vol. 3971, No. 12, EI '00, San Jose, USA, Jan 2000.

[9] E. J. Delp , C. I. Podilchuk, and E. T. Lin, "Detection of Image Alterations using Semi-Fragile Watermarks," *SPIE International Conf. on Security and Watermarking of Multimedia Contents II*, Vol. 3971, No. 14, EI '00, San Jose, USA, Jan 2000.

[10] N. D. Memon, P. Vora, and M. Yeung, "Distortion Bound Authentication Techniques," *SPIE International Conf. on*

Security and Watermarking of Multimedia Contents II, Vol. 3971, No. 15, EI '00, San Jose, USA, Jan 2000.

[11] K. Toyokawa, N. Morimoto, S. Tonegawa, K. Kamijo, and A. Koide, "Secure Digital Photograph Handling with Watermarking Technique in Insurance Claim Process," *SPIE International Conf. on Security and Watermarking of Multimedia Contents II*, Vol. 3971, No. 42, EI '00, San Jose, USA, Jan 2000.

[12] Jiri Fridrich, and Miroslav Goljan, "Images with self-correcting capabilities," *Proceedings of 1999 IEEE International Conference on Image Processing*, Vol.3, October 1999, pp.24 -28.

[13] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *in Proc. IEEE Int. Conf. on Image Processing*, Sep. 1996, Vol. 3, pp. 243-246.