A Video Watermarking Algorithm of H.264/AVC for Content Authentication

Weiwei ZHANG, Ru ZHANG, Xianyi LIU, Chunhua WU, Xinxin NIU
Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China
National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China
Email: weiweizh@hqu.edu.cn, zhangru@bupt.edu.cn, liujy@bupt.edu.cn, wchadv@gmail.com, xxniu@bupt.edu.cn

Abstract—Since video products compressed by H.264/AVC become increasing popular, issues of authentication for H.264/AVC-based video will be very important. In this paper, a content-based authentication semi-fragile watermarking scheme for H.264/AVC video is proposed. The content-based authentication code is generated using the invariance of the relation between DCT DC coefficients of the two 4×4 sub-blocks. The authentication code is then embedded into the DCT coefficients in diagonal positions in I frames. Spatial tampering can be located by comparing the extracted watermarking and the content-based authentication code. The experimental results show that the proposed scheme can distinguish the malicious spatial tampering from the common signal processing. The tampered areas can also be located.

Index Terms—H.264/AVC, content authentication, semi-fragile, watermarking, multimedia security

I. INTRODUCTION

In recent years, digital videos achieve more and more applications, such as DVD, VCD, video conference, video-on-demand, etc. With the rapid development of multimedia processing technologies, digital videos can be easily tampered, altered or forged by unauthorized users with video editing tools. Under these circumstances, authenticity and integrity verification of digital video becomes an important research topic nowadays [1]. Video watermarking technology provides useful solution for such problems by embedding the watermark information behind a cover [2,3,4,5].

Because most digital videos are stored and distributed in compressed format in the internet, compressed-domain video watermarking is especially attractive [1]. H.264 [6] is the most recent video coding standard from the ITU-T Video Coding Experts Group and the ISO/IEC Moving Pictures Experts Group. It is expected that it will become one of the most popular video coding standard for broadcast on wireless channels and internet media [7]. However, difference from other image/video compression techniques, it is not easy to find a successful watermarking scheme for H.264/AVC [8].

In recent years, a few watermarking algorithms for H.264/AVC based video authentication have been proposed in the open literature. Pröfock et al. [9] proposed a fragile, blind and erasable watermarking algorithm. In their algorithm, watermark is embedded into some skipped macroblocks of the H.264. The algorithm achieves low video quality degradations and low data rate, but it has low watermark payload. Zhang et al. [10] proposed a new scheme which makes an accurate usage of the tree-structured motion compensation, motion estimation and lagrangian optimization of the standard. The watermark is embedded based on the best mode decision strategy in the sense that if undergone any spatial and temporal attacks, the scheme can detect the tampering by the sensitive mode change. Kim et al. [11] proposed a video authentication scheme which inserts a watermark bit on the motion vectors for inter-coded macroblocks or on the mode number for intra-coded macroblocks. The scheme has high watermark payload with small image quality and compression power degradation.

Those above proposed digital video watermarking algorithm [9-11] for authentication based on H.264/AVC are sensitive to even any little modification. However, most digital videos are stored and distributed in compressed format. It is difficult to distinguish common video processing from malicious tampering with fragile watermarking. Semi-fragile watermarking allows acceptable content-preserving manipulations such as H.264/AVC compression, Gaussian low-pass filtering, median filtering, and salt and peppers noise attacks, while it can detect content-altering malicious manipulations such as removal, addition, and modification of objects.

Semi-fragile watermark based on H.264/AVC had been studied in recent years. Chen et al. [12] proposed a video authentication system, watermark authentication code is used Block Sub-band Index and Coefficient Modulation to embed in the quantized AC coefficient of I frame. Their system can locate the tampered locations. However, the method requires extra computation to employ Index/Coefficient Modulation. Xu et al. [1] proposed a semi-fragile watermark for H.264/AVC algorithm which embeds watermark into the DCT coefficients in diagonal positions using a modulation method. The algorithm can detect both spatial and...
temporal tampering. Su et al. [13] proposed a semi-fragile video watermarking, in their scheme, the watermark signals, which represent the serial numbers of video segments, are embedded into nonzero quantization indices of frames to achieve both the effectiveness of watermarking and the compact data size. However, their scheme can just locate the edited segments in the tampered video. Besides, the method needs to compute the Just Noticeable Difference (JND) which is time consuming.

In this paper, we propose a semi-fragile watermarking scheme for H.264/AVC video authentication. Our scheme requires little extra computation. The content-based authentication code is generated according to the invariance of the relation between DCT DC coefficients. Then, the authentication code is embedded into the DCT coefficients in diagonal positions in I frames. The tampered areas can be located by comparing the extracted watermarking and the content-based authentication code.

II. THE PROPOSED SCHEME

A. Content-based watermark generation

In content-based authentication watermarking, extract robust content feature is very important. Lin et al. [14, 15] found and proved that after repetitive JPEG or MPEG-2 compression, the magnitude relationship between the two DCT coefficients at the same position in separate blocks of an image remains invariance. However, unlike JPEG or MPEG-2 compression, only the resulting prediction residue is transformed using an integer transform in the H.264/AVC encoding process. The invariant features in JPEG or MPEG-2 cannot be applied directly to H.264/AVC video [1].

Inspired by reference [14, 15], after repetitive experiments, we found that most of the relationship between DCT DC coefficients at the same position in the two 4 × 4 sub-blocks remains invariance after H.264/AVC compression. Taking sequence “foreman” (cif 352 × 288) as example, there are total 6336 sub-blocks (4 × 4) in the first frame. After H.264 recompression, the number of the changing magnitude relationship after H.264 recompression.

In H.264/AVC, each frame consists of macroblocks (16 × 16) which may be divided into sub-block partition for motion prediction. The prediction residuals will be processed by an 4×4 integer Discrete Cosine Transform. And each macroblock contains 16 sub-blocks (4 × 4). These sub-blocks are scanned according to the predetermined order, as illustrated in Fig.1.

Taking sequence “foreman” (cif 352 × 288) as example, after H.264/AVC recompression, DCT DC coefficients of the sixteen 4×4 sub-blocks in the third macroblock in the first I frame is shown as equation (1)

\[
\begin{bmatrix}
DC_{i_1} & DC_{i_2} & DC_{i_3} & DC_{i_4} \\
DC_{i_5} & DC_{i_6} & DC_{i_7} & DC_{i_8} \\
DC_{i_9} & DC_{i_{10}} & DC_{i_{11}} & DC_{i_{12}} \\
DC_{i_{13}} & DC_{i_{14}} & DC_{i_{15}} & DC_{i_{16}}
\end{bmatrix}
= \begin{bmatrix}
896 & 896 & 896 & 896 \\
896 & 896 & 896 & 832.5 \\
896 & 816 & 682.25 & 659.5 \\
671.5 & 689 & 756 & 839
\end{bmatrix}
\]  

(1)

Transform it into one-dimensional array:

\[
[DC_{i_1}, DC_{i_2}, DC_{i_3}, DC_{i_4}, DC_{i_5}, DC_{i_6}, DC_{i_7}, DC_{i_8}, DC_{i_9}, DC_{i_{10}}, DC_{i_{11}}, DC_{i_{12}}, DC_{i_{13}}, DC_{i_{14}}, DC_{i_{15}}, DC_{i_{16}}]
= [896, 896, 896, 896, 896, 896, 896, 832.5, 896, 816, 682.25, 659.5, 671.5, 689, 756, 839]
\]

The content-based authentication code is defined as:

\[w_n = \begin{cases} 
1, & \text{if } DC_i \geq DC_{i+4} \\
0, & \text{else}
\end{cases}
\]  

(2)

Where \(w_n\) is the authentication code, \(DC_i\) is the DCT DC coefficients in a 4×4 sub-blocks, \(k\) is the position of a 4×4 sub-blocks in a macroblock.

The third macroblock authentication code can be generated as following:

\[\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}
\]  

(3)

After H.264 recompression, the DCT DC coefficients of the sixteen 4×4 sub-blocks in the third macroblock in the first I frame is shown as follows:

\[
\begin{bmatrix}
DC_{i_1} & DC_{i_2} & DC_{i_3} & DC_{i_4} \\
DC_{i_5} & DC_{i_6} & DC_{i_7} & DC_{i_8} \\
DC_{i_9} & DC_{i_{10}} & DC_{i_{11}} & DC_{i_{12}} \\
DC_{i_{13}} & DC_{i_{14}} & DC_{i_{15}} & DC_{i_{16}}
\end{bmatrix}
= \begin{bmatrix}
896 & 896 & 896 & 896 \\
896 & 896 & 896 & 832.5 \\
896 & 816 & 682.25 & 659.5 \\
679.5 & 681 & 754.5 & 838.75
\end{bmatrix}
\]  

(4)

Transform it into one-dimensional array:

\[
[DC_{i_1}, DC_{i_2}, DC_{i_3}, DC_{i_4}, DC_{i_5}, DC_{i_6}, DC_{i_7}, DC_{i_8}, DC_{i_9}, DC_{i_{10}}, DC_{i_{11}}, DC_{i_{12}}, DC_{i_{13}}, DC_{i_{14}}, DC_{i_{15}}, DC_{i_{16}}]
= [896, 896, 896, 896, 896, 896, 896, 832.5, 896, 816, 682.25, 659.5, 671.5, 681, 756, 839]
\]

The third macroblock authentication code is generated as following:

\[\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}
\]  

(5)

According to Equation (3) and (5), After H.264 recompression, one of the authentication code is changed, for example, “0” in Equation (3) is changed into “1” in Equation (5). The reason of changing is that “689” in
Equation (1) is changed into “681” in Equation (4), which changes the magnitude relationship between the current DC coefficient and its adjacent.

The reason why the magnitude relationship between integer DCT DC coefficients of the two adjacent sub-blocks is changed is that the adjacent pixels of natural images have certain correlation. Because of the correlation, some of the integer DCT DC coefficients of the two adjacent sub-blocks are very closely even in the same equation, which causes the robustness of the authentication code low.

In order to improve the robustness of the authentication code, the correlation of some of the integer DCT DC coefficients must be removed. Taking the first frame of sequence “foreman” (cif 352×288) as example, there are 6336 integer DCT DC coefficients, we scramble the positions of these coefficients according to the following rule:

\[
[DC, DC_{11}, DC_{12}, DC_{21}, DC_{22}, \ldots, DC_{11}, DC_{12}, DC_{21}, DC_{22}]
\]

After scrambling the positions of the coefficients, the difference of the adjacent integer DCT DC coefficients increases. So it improves the robustness of the authentication code.

Adopting the scrambling method, after H.264 recompression, the number of the changing magnitude relationship between integer DCT DC coefficients of the two adjacent sub-blocks of the sequence “foreman” is 81. It accounts for about 1.28%.

Table 1 shows the number of the changing magnitude relationship between integer DCT DC coefficients of the two adjacent sub-blocks of the four test sequences. N1 denotes the number of the changing magnitude relationship without scrambling, N2 denotes after scrambling.

**TABLE 1.** THE NUMBER OF THE CHANGING MAGNITUDE RELATIONSHIP OF THE TWO ADJACENT SUB-BLOCKS WITHOUT SCRAMBLING AND AFTER SCRAMBLING

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Without scrambling (N1)</th>
<th>Scrambling (N2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akiyo</td>
<td>605</td>
<td>57</td>
</tr>
<tr>
<td>Bus</td>
<td>379</td>
<td>99</td>
</tr>
<tr>
<td>Traffic</td>
<td>545</td>
<td>50</td>
</tr>
<tr>
<td>Coastguard</td>
<td>476</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 1 shows that after scrambling the position of the integer DCT DC coefficients, the method can improve the robustness of the authentication code obviously.

**B. Watermark embedding**

The proposed watermark embedding scheme in H.264/AVC encoder is as following. The authentication code is embedded into the DCT coefficients in diagonal positions of the high frequency quantized coefficients. If the authentication code is “1”, let the high frequency coefficients be zero, else let them higher than zero. The watermark is embedded in the block as follows:

\[
X'_{level} = \begin{cases} 
X_{level}, & \text{if } X_{level} > 1 \text{ and } w_n = 1; \\
X_{level} + 2, & \text{if } X_{level} \leq 1 \text{ and } w_n = 1; \\
0, & \text{if } w_n = 0 
\end{cases}
\]

Where \(w_n\) is the authentication code, \(X_{level}\) is quantized DCT coefficients, \(X'_{level}\) is the watermarked DCT coefficients.

**C. Watermark extraction**

Watermark extraction is very simple. First, the decoder selects the embedded sub-block. Then the algorithm extracts the watermark as follows:

\[
w_n' = \begin{cases} 
0, & \text{if } X_{level} = 0 \\
1, & \text{if } X_{level} > 0 
\end{cases}
\]

Where \(w_n'\) is the extracted watermark, \(X_{level}\) is the watermarked DCT coefficients.

**D. Video authentication**

In the decoder side, the authentication codes can be extracted from the current I frame, and the watermarks are extracted from the next I frame. The algorithm can detect the tampering areas by mismatch between the authentication codes with watermark.

- **Authentication process**

Firstly, the difference of the authentication codes and extracted watermark is transform to a two dimension difference image.

\[
D_y = |w_y - w'_y|
\]

Where \(w_y\) is the authentication codes, \(w'_y\) is extracted watermark.

The difference image denotes the difference between the authentication codes and the extracted watermarks. If the pixel in the difference image is “0”, it denotes the authentication codes and the extracted watermarks, vice versa. So the areas are tampered or not can be justified according to the distribution of the “1” in the difference image.

Secondly, the dense and sparse points are defined as followed:

If a pixel “1” in the difference image has at least one “1” pixel in its eight neighborhood, and it is defined the dense point. Otherwise, it is defined the sparse point.

The malicious manipulation or content-based manipulation can be justified according to equation (9) and (10)

\[
r = \frac{N_d}{N_s + N_d}
\]

\[
\begin{cases} 
\text{content-based manipulation, if } r < T \\
\text{malicious manipulation, if } r \geq T 
\end{cases}
\]

Where \(N_d\) denotes the number of the sparse point, \(N_s\) is the number of the dense point. \(T\) is the threshold according to the experiment.

- **Tampered areas location**
If the manipulation is content-based, then the authentication passes, otherwise, locates the malicious manipulation areas.

If a pixel in the difference image is the dense point, then define the four neighborhoods as dense points. Then locate the areas of $4 \times 4$ sub-block corresponding to the dense point.

### III. EXPERIMENT RESULTS

The proposed digital video watermarking scheme is implemented in H.264/AVC JM-12.4 reference software. Six standard video sequences (Foreman, Stefan, Coastguard, Flower Garden, Container Ship, Traffic) in CIF format ($288 \times 352$) are used for our simulation. There are 100 frames of the test video to be encoded and decoded. The GOP size is set as 15 with one I frame followed by 14 P frames. The frame rate of the video is 30 fps. The value of QP is set to 20, there are 6336 bit watermarks to be embedded into one I frame.

#### A. Imperceptibility test

To evaluate the imperceptibility of the proposed scheme, the test sequence “Traffic” is shown in Figure 3.

![Original frame (the 6th frame)](image1)

![Watermarked frame (the 6th frame)](image2)

Fig.3. Original and watermarked frame (QP=26)

In the experiment, figure 3 (a) is the 6th original frame of test sequence “Traffic“, figure 3 (b) is the watermarked frame, as we can see, no visible artifacts can be observed in the Figure 3.

![PSNR degradation at frame rate 1M bit/s](image3)

![PSNR degradation at frame rate 768k bit/s](image4)

![PSNR degradation at frame rate 512k bit/s](image5)

Fig.4. PSNR degradation of six test video at difference frame rates

Figure 4 shows that at different frame rate, the six test video sequences have little PSNR degradation after watermark.

#### B. Robustness to common signal processing

Semi-fragile watermark should be robustness to common signal processing. Table 2 shows that the proposed watermark scheme can robust to some common signal processing.

<table>
<thead>
<tr>
<th></th>
<th>Gaussian noise (0,0.0005)</th>
<th>Salt and pepper noise 0.05</th>
<th>Gaussian low pass filter (5*5)</th>
<th>Contrast enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>($\rho^*$)</td>
<td>0.89</td>
<td>0.94</td>
<td>0.74</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Where $\rho^*$ is the ratio of difference between the authentication codes and the exacted watermark and the total number of the extracted watermark.

#### C. Tampering areas location

Figure 5 shows the proposed watermark scheme can detect and locate the malicious areas.

![The tampered frame (the 6th frame)](image6)

![Tampered area location](image7)

Fig.5. Tempered area location

### IV. CONCLUSIONS

A semi-fragile watermarking scheme for H.264/AVC is proposed in this paper to detect the spatial tampering. The robust video features extracted from video frame are used to form the authentication code. Then the authentication code is embedded into the DCT coefficients in diagonal positions in I frames. Spatial tampering can be located by comparing the extracted watermarking and the content-based authentication code. Experiment results show that the proposed semi-fragile watermarking scheme can justify the malicious manipulation and content-based manipulation.

### ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (61003284), National Natural Science Foundation of China (61170271), Beijing Municipal Natural Science Foundation(4122053), the Fundamental Research Funds for the Central Universities (BUPT2011RC0210), Press and publication of major science and technology research and development projects “Digital rights protection technology research and development project” (GXTC-CZ-1015004/09 , GXTC-CZ-1015004/15-1)

### REFERENCES


