ISSN: 0976-9102(ONLINE)
DOI: 10.21917/ijivp.2014.0124

AN EFFICIENT WATERMARKING SCHEME FOR H.264/AVC COMPRESSED VIDEO

Dawen Xu
School of Electronics and Information Engineering, Ningbo University of Technology, China
E-mail: xdw@nbut.cn

Abstract

Since H.264/AVC is the most widely-deployed video coding standard and has gained dominance, the necessity of copyright protection and authentication that are appropriate for this standard is unquestionable. According to H.264/AVC specific codec architecture, an efficient watermarking scheme for H.264/AVC video is proposed. The watermark information is embedded into quantized residual coefficients by slightly modulating the coefficients with specific symbol encoding, instead of directly adding the watermark to the quantized coefficients. It is not necessary to fully decode H.264/AVC compressed stream both in the embedding and extracting processes. Experimental results show that the proposed scheme can preserve high imperceptibility while achieving enough robustness against various attacks such as requantization, transcoding, AWGN, brightness and contrast adjustment.

Keywords:

H.264/Advanced Video Coding (AVC), Video Watermarking, Intra Prediction, Robustness

1. INTRODUCTION

H.264/AVC has become one of the most commonly practiced video compression standard since 2003. It achieves a significant improvement in compression performance as compared to the previous generation of video compression standard. As a result, H.264/AVC addresses a wide range of applications [1], such as video streaming (e.g., YouTube), Blu-ray video disc, surveillance camera, etc. Given the dominant application of H.264/AVC, copyright protection, authenticity and integrity verification for H.264/AVC video has becoming an important research topic nowadays. Video watermarking technology provides an effective solution for such problem by embedding the watermark information within the host video [1, 2]. The embedded watermark can be detected or extracted later from the host video to detect act of tampering with the content, or to prove ownership.

However, higher coding efficiency is likely to increase the difficulty of watermarking. Over the years, several watermarking techniques on H.264/AVC are proposed and realized at various stages in the compression pipeline, i.e., intra-prediction stage, quantized transform coefficients stage, and entropy coding stage. Noorkami and Mersereau [3] employed a human visual model adapted for a 4 × 4 DCT block to increase the payload and robustness. The algorithm can achieve high robustness, but its computational complexity is relatively high. Zhang et.al [4] proposed a robust video watermarking scheme of H.264/AVC using a grayscale watermark pattern. The pattern insertion process enhanced the robustness, but the computational complexity of the watermark pre-processing makes it a challenge to implement the watermark embedding efficiently. Later, Mansouri et.al [5] exploited the syntactic elements of the video stream for watermark embedding. These algorithms utilize the quantized or un-quantized integer transform coefficients for watermark embedding.

Besides DCT coefficients, motion vector (MV) is also used for watermark embedding in many schemes (referred to as MV-based watermarking schemes) [6]. As described in [7], the advanced JND (Just Noticeable Difference) model is hard to be applied to an MV-based watermarking scheme. As a result, it is hard to achieve a satisfied trade-off between the robustness and imperceptibility for most existing MV-based watermarking schemes. In addition to DCT-based and MV-based watermarking schemes, some other algorithms exploit new features of H.264/AVC (such as context-adaptive entropy coding, intra prediction mode and the reference index) for watermarking [8] – [10]. But these algorithms are not robust against some common attacks. Considerable more effort is needed to provide robust watermarking technique for H.264/AVC.

We note that the existing watermarking algorithms prefer to embed the watermark information into DCT coefficients, not only for H.264/AVC, but also for other coding standards such as MPEG-2/4 [11]. The goal of this paper is to provide an efficient and robust H.264/AVC compressed domain watermarking. To achieve this goal, we exploited the re-compression architecture of H.264/AVC and characteristics of residual coefficients. The watermark information is embedded in quantized residual coefficients by modifying positive and negative signs. This modification is based on the fact that the coefficients will be changed after re-encoding, while the distribution of positive and negative coefficients remains relatively stable. The remainder of the paper is organized as follows. In section 2, watermark embedding and extraction scheme is described. In section 3, experimental results and performance analysis are presented aiming at demonstrating the validity of the proposed scheme. Finally, the conclusion is drawn in section 4.

2. PROPOSED WATERMARKING SCHEME

2.1 SPECIAL PROBLEMS OF WATERMARKING IN INTRA-FRAMES OF H.264/AVC

Unlike some previous video coding standards (namely JPEG and MPEG-2), in H.264/AVC standard, intra prediction is employed to exploit the spatial correlation by using neighboring reconstructed pixels to predict the current block. Only the resulting prediction residue is transformed using an integer transform, and the transform coefficients are quantized and encoded using entropy coding methods. Traditional methods that embed watermarking information into DCT coefficients cannot be applied directly to H.264/AVC standard. The reason is that the quantized residual coefficients will be changed after re-compression. The typical re-compression architecture is depicted in Fig.1. We only concern Intra-frames of H.264/AVC in this paper.

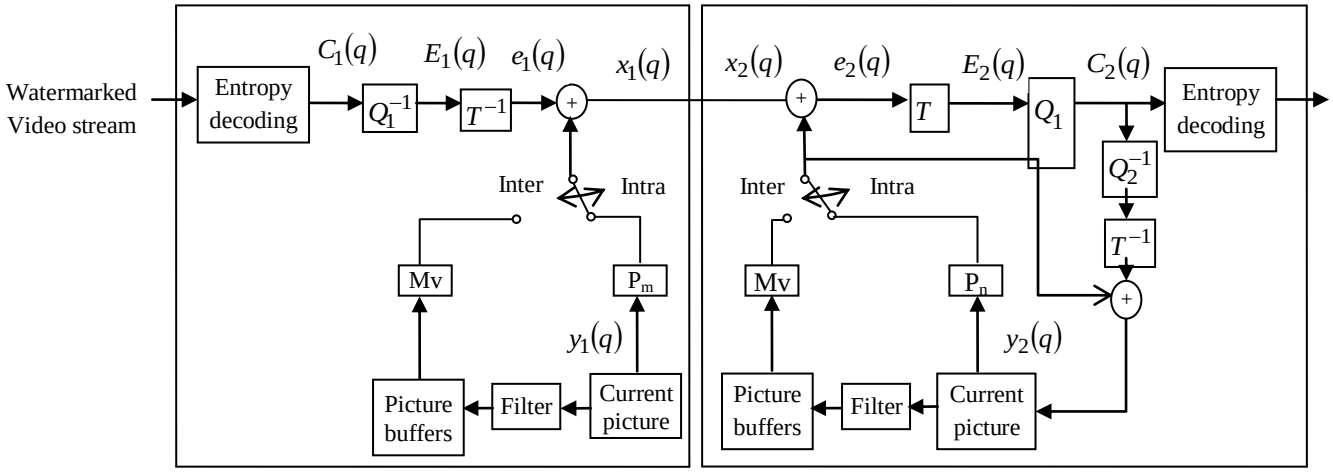


Fig.1. The typical re-compression architecture

The decoded pixels $x_1(q)$ at decoder site and the corresponding prediction error $e_2(q)$ at encoder site are expressed respectively in the following equations.

$$x_1(q) = P_m(y_1(q)) + e_1(q) \tag{1}$$

$$e_2(q) = x_2(q) - P_n(y_2(q)) \tag{2}$$

where, $P_m(.)$ and $P_n(.)$ denote the intra prediction operator with prediction modes m and n, $y_1(q)$ and $y_2(q)$ are reference blocks used for intra-prediction at decoder and encoder site. Since $x_1(q) = x_2(q)$, Eq.(2) can be rewritten as:

$$e_2(q) = P_m(y_1(q)) + e_1(q) - P_n(y_2(q)) \tag{3}$$

Suppose $C_1(q)$ and $C_2(q)$ denote input and output quantized coefficients respectively.

$$C_2(q) = Q_2[T(e_2(q))] \tag{4}$$

where, $T(.)$ and $Q_2[.]$denote the transform and quantization operators. Eq.(3) can then be substituted into Eq.(4):

$$\begin{aligned} C_2(q) &= Q_2\left[T\left(P_m(y_1(q)) + e_1(q) - P_n(y_2(q))\right)\right] \\ &= Q_2\left[T(e_1(q)) + T\left(P_m(y_1(q)) - P_n(y_2(q))\right)\right] \\ &= Q_2\left[T\left(T^{-1}\left(Q_1^{-1}[C_1(q)]\right)\right) + T\begin{pmatrix} P_m(y_1(q)) \\ -P_n(y_2(q)) \end{pmatrix}\right] \end{aligned} \tag{5}$$

where, $T^{-1}(.)$ and $Q^{-1}[\ ]$ denote the inverse transform and de-quantization operators respectively. According to Eq.(5), $C_1(q)$ is not identical to $C_2(q)$ which is caused by two factors. Firstly, in H.264/AVC, $T^{-1}(.)$ and $Q^{-1}[\ ]$ are not the exact inverse operation of $T(.)$ and $Q_2[\ ]$ because of rounding or truncation in integer arithmetic. In order to verify truncation process, one example is given in Fig.2. The quantized transform coefficients after the first compression are shown in Fig.2(a). The data are taken from one 4 × 4 sub-block in Hall sequence. The results after performing inverse quantization and inverse transform are given in Fig.2(b). The prediction reference pixels are listed in Fig.2(c). Next, the reconstructed pixel values can be obtained by combing Fig.2(b) and Fig.2(c), as shown in Fig.2(d). Obviously, overflow occurs to two pixel values which fall above the maximum possible value 255. The truncated results are presented in Fig.2(e). Supposing coding parameters remain unchanged during the recompression, the quantized transform coefficients after the recompression are given in Fig.2(f). As can be seen, the quantized transform coefficients have been changed after the recompression.

$$\begin{bmatrix} -2 & 2 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 2 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$ (a)

$$\begin{bmatrix} -1 & -1 & -1 & -1 \\ -12 & -2 & -14 & -36 \\ -6 & 1 & -17 & -42 \\ 11 & 6 & -6 & -11 \end{bmatrix}$$ (b)

$$\begin{bmatrix} 237 & 237 & 237 & 237 \\ 232 & 232 & 232 & 232 \\ 253 & 253 & 253 & 253 \\ 250 & 250 & 250 & 250 \end{bmatrix}$$ (c)

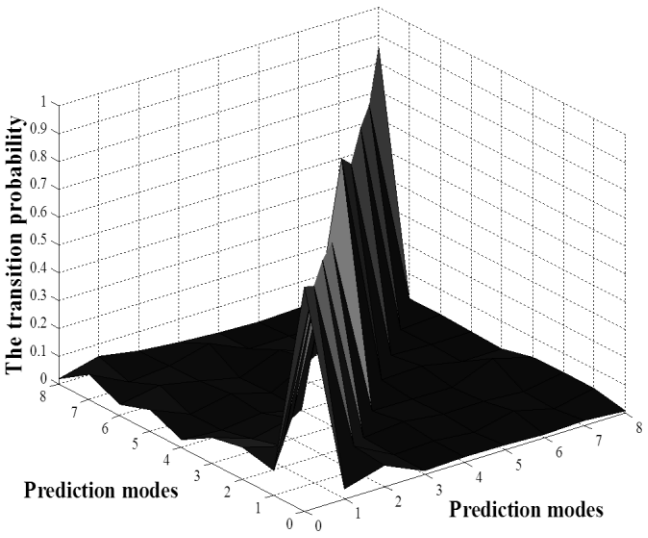
$$\begin{bmatrix} 236 & 236 & 236 & 236 \\ 220 & 230 & 218 & 196 \\ 247 & 254 & 236 & 211 \\ 261 & 256 & 244 & 239 \end{bmatrix}$$ (d)

$$\begin{bmatrix} 236 & 236 & 236 & 236 \\ 220 & 230 & 218 & 196 \\ 247 & 254 & 236 & 211 \\ 255 & 255 & 244 & 239 \end{bmatrix}$$ (e)

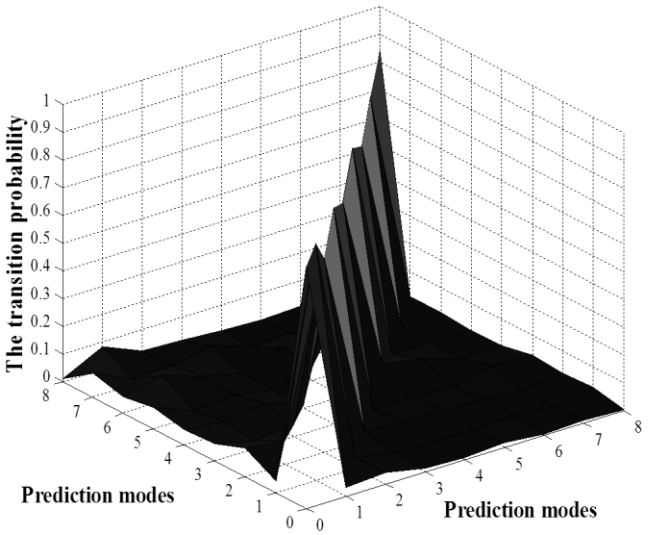
$$\begin{bmatrix} -2 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$ (f)

Fig.2. An example of truncation process

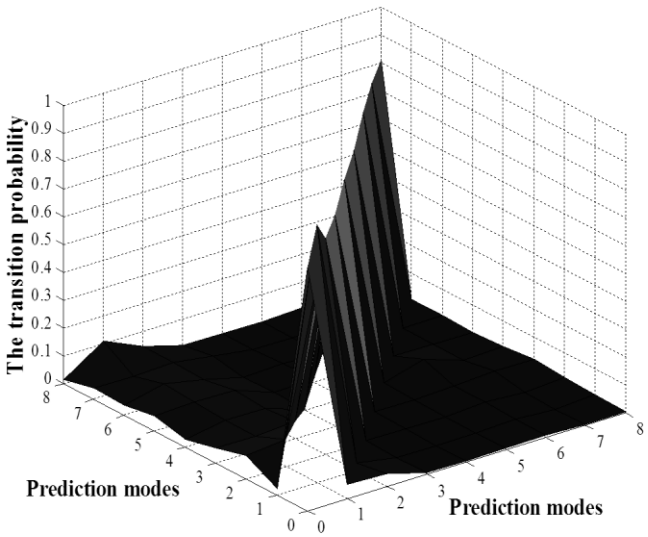
Secondly, the reference blocks used for intra-prediction at the 2nd compression may be different from those at the 1st compression, which may change the prediction modes and the values of the residual coefficients [12]. According to H.264/AVC standard, the Intra_4 × 4 is based on predicting each 4 × 4 luminance block separately and is suited for encoding parts of a picture with significant detail. Nine prediction modes are supported in the Intra_4 × 4. After re-encoding, prediction mode transition will occur in some 4 × 4 sub-blocks. The statistical results of six standard video sequences are given in Fig.4. As can be seen, after re-encoding, the transition probability is relatively large among those modes which have similar prediction directions, for example, horizontal modes (i.e., 1, 6, 8), or vertical modes (i.e., 0, 3, 5, 7).



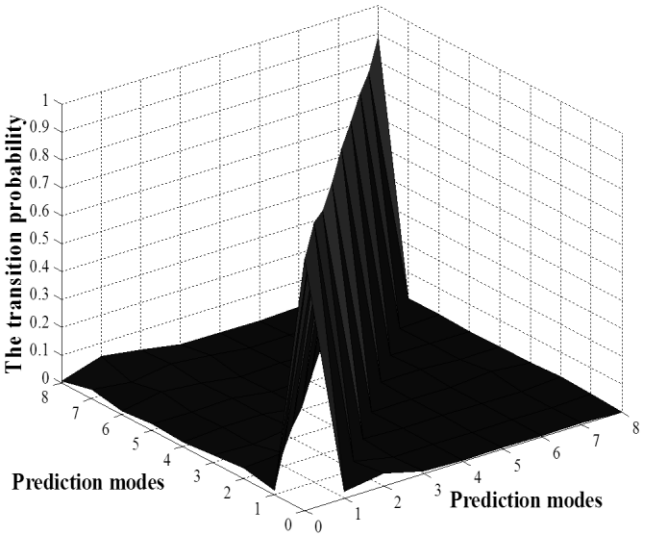
(a). Foreman



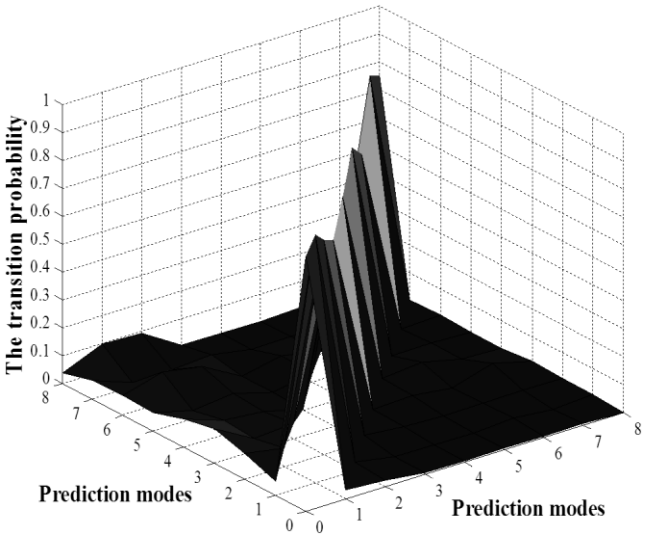
(b). Carphone



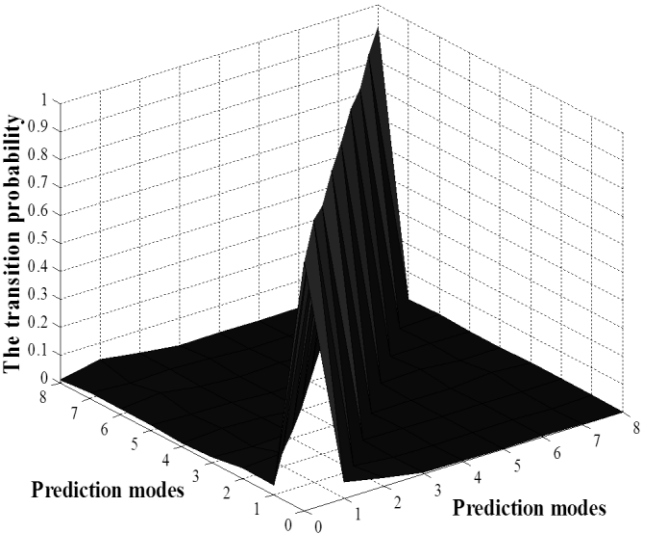
(c). Container



(d). Mobile



(e). News



(f). Stefan

Fig.3. The transition probability of nine Intra_4 × 4 prediction modes

2.2 WATERMARK EMBEDDING

According to the analysis in section 2.1, a watermarking process using an additive or multiplicative operation in intra-frames inherently retains high possibility to lose some of the embedded data during the extraction process. Although the DCT coefficients will change after re-encoding, but the statistical analysis shows that the sign of the coefficients remain relatively stable. In this paper, watermark embedding is performed by modulating DCT coefficients in a novel way. In general, DCT coefficient watermarking requires a partial decoding, i.e., entropy decoding of the coefficients and afterwards the watermark information is embedded by modifying the coefficients. The coefficients in diagonal position are stable than the others [3]. Thus, one coefficient in diagonal position (i.e., location 7, 8, 9, and 10 in Fig.4) in a 4 × 4 DCT block is selected to embed the watermark information. How to choose the specific embedding location among 7, 8, 9, and 10 is the key issue to be resolved. In the proposed scheme, the rate-distortion costs are calculated respectively in four cases. The final location is selected by minimizing a Lagrangian function [13].

1	2	6	7
3	5	8	13
4	9	12	14
10	11	15	16

Fig.4. Zig-Zag scan for 4 × 4 luminance block

A binary image is used as a watermark and it can be rearranged to a sequence of binary values as a bit stream. Watermark embedding procedure is described as follows:

Step 1: The interleaving technique is applied to eliminate the relativity of neighboring elements and improve the robustness of embedding watermark. Then, for the sake of security, the interleaved sequence S_{in} is encrypted by chaotic sequence to form the watermark signal W.

$$W = E(S_{in}, S_{encrypt}) = \{w_1, w_2, \cdots, w_m\} \tag{6}$$

where, $E(.)$ is the encryption operation, $S_{encrypt}$ is binary chaotic sequence generated by using logistic map with an initial value K_1.

Step 2: A 4 × 4 luminance block b^p ($1 \le p \le 16$) used for watermark embedding in each macroblock is selected according to another key K_2. Besides, the positions of macroblocks which containing watermark information should also be stored and sent to the decoder Suppose $posit.m_j^p$ and $negat.m_j^p$ denote the numbers of positive and negative coefficients in diagonal position in the selected 4 × 4 luminance block of the j^{th} macroblock. Then, the parity bit can be calculated as:

$$posit.par_j^p = \begin{cases} 1 & if \ \mathrm{mod}(posit.m_j^p, 2) == 1 \\ 0 & if \ \mathrm{mod}(posit.m_j^p, 2) == 0 \end{cases} \tag{7}$$

$$negat.par_j^p = \begin{cases} 1 & if \ \mathrm{mod}(negat.m_j^p, 2) == 1 \\ 0 & if \ \mathrm{mod}(negat.m_j^p, 2) == 0 \end{cases} \tag{8}$$

where, mod(•) is the modulo operation. In order to describe our embedding algorithm more clearly, we define the notation $par.pos\_neg_j^p$ as following:

$$par.pos\_neg_j^p = posit.par_j^p \oplus negat.par_j^p \tag{9}$$

where, $\oplus$ is XOR (exclusive OR) operation.

Step 3: Suppose $B\_Quant.coeff_j^{p,k}$ and $A\_Quant.coeff_j^{p,k}$ $(7 \le k \le 10)$ denote the coefficients in diagonal position before and after quantization, respectively. Watermark embedding is performed by modulating $A\_Quant.coeff_j^{p,k}$ as follows:

Case 1: If $(w_k == 0)$ && $(par.pos\_neg_j^p == 0)$, the original coefficient value $A\_Quant.coeff_j^{p,k}$ keeps unchanged.

$$\overline{A\_Quant.coeff}_j^{p,k} = A\_Quant.coeff_j^{p,k} \tag{10}$$

Case 2: If $(w_k == 1)$ && $(par.pos\_neg_j^p == 1)$, the original coefficient value $A\_Quant.coeff_j^{p,k}$ also keeps unchanged.

$$\overline{A\_Quant.coeff}_j^{p,k} = A\_Quant.coeff_j^{p,k} \tag{11}$$

Case 3: If $(w_k == 0)$ && $(par.pos\_neg_j^p == 1)$, modify the original coefficient value $A\_Quant.coeff_j^{p,k}$ as follows.

$$\overline{A\_Quant.coeff_j^{p,k}} = \begin{cases} \mathrm{sgn}(B\_Quant.coeff_j^{p,k}) & if A\_Quant.coeff_j^{p,k} == 0 \\ 0 & if A\_Quant.coeff_j^{p,k} != 0 \end{cases} \tag{12}$$

Case 4: If $(w_k == 1)$ && $(par.pos\_neg_j^p == 0)$, modify the original coefficient value $A\_Quant.coeff_j^{p,k}$ according to the following equation.

$$\overline{A\_Quant.coeff}_j^{p,k} = \begin{cases} \mathrm{sgn}(B\_Quant.coeff_j^{p,k}) & if \ A\_Quant.coeff_j^{p,k} == 0 \\ 0 & if \ A\_Quant.coeff_j^{p,k} != 0 \end{cases} \tag{13}$$

where, $\overline{A\_Quant.coeff}_j^{p,k}$ is the modified coefficient, $sgn(x)$ is defined as

$$\mathrm{sgn}(x) = \begin{cases} +1 & if \ x > 0 \\ -1 & if \ x < 0 \end{cases} \tag{14}$$

In order to reduce the video quality degradation caused by watermark embedding, the signs of the coefficients before quantization are introduced. The transform coefficients of the prediction residue below the corresponding quantization step size are quantized to zeros. If a positive or negative constant (e.g., +1/-1) is used to replace the zero-coefficient directly for watermark embedding, the influence on the video quality should

be larger than the method in Eq.(12) and Eq.(13). For example, suppose the coefficient before quantization (i.e., $B\_Quant.coeff_j^{p,k}$) is negative, and its corresponding quantized value (i.e., $A\_Quant.coeff_j^{p,k}$) is zero. Apparently, the distortion caused by replacing zero with -1 will be lower than that with +1.

2.3 WATERMARK EXTRACTING

Since watermark information is embedded in quantized residual coefficients, inverse quantization, inverse DCT, and full decoding are not performed on the watermarked bit-streams. On the contrary, entropy decoding is only required to obtain quantized residual coefficients. In addition, watermark extraction does not require access to the original host video. The watermark signal is extracted from the watermarked H.264/AVC video through the following steps:

Step 1: The watermarked 4 × 4 luminance block in each macroblock can be determined using key K_2. Calculate the numbers of positive and negative coefficients in diagonal position in the watermarked 4 × 4 luminance block, and denoted as $\overline{\overline{posit.m_j^p}}$ and $\overline{\overline{negat.m_j^p}}$ respectively.

Step 2: Calculate the parity bit:

$$\overline{\overline{posit.par_j^p}} = \begin{cases} 1 & if \ \mathrm{mod}\left(\overline{\overline{posit.m_j^p}}, 2\right) == 1 \\ 0 & if \ \mathrm{mod}\left(\overline{\overline{posit.m_j^p}}, 2\right) == 0 \end{cases} \quad (15)$$

$$\overline{\overline{negat.par_j^p}} = \begin{cases} 1 & if \ \mathrm{mod}\left(\overline{\overline{negat.m_j^p}}, 2\right) == 1 \\ 0 & if \ \mathrm{mod}\left(\overline{\overline{negat.m_j^p}}, 2\right) == 0 \end{cases} \quad (16)$$

Step 3: Watermark $\overline{w_k}$ can be extracted as follows:

$$\overline{w_k} = \begin{cases} 0 & if \ \overline{\overline{posit.par_j^p}} == 0 \ \&\& \ \overline{\overline{negat.par_j^p}} == 0 \\ 1 & if \ \overline{\overline{posit.par_j^p}} == 1 \ \&\& \ \overline{\overline{negat.par_j^p}} == 0 \\ 1 & if \ \overline{\overline{posit.par_j^p}} == 0 \ \&\& \ \overline{\overline{negat.par_j^p}} == 1 \\ 0 & if \ \overline{\overline{posit.par_j^p}} == 1 \ \&\& \ \overline{\overline{negat.par_j^p}} == 1 \end{cases} \quad (17)$$

Step 4: Generate the same binary chaotic sequence $S_{encrypt}$ which used during the embedding process with an initial value K_1. The extracted watermark sequence $\overline{W} = \{\overline{w_1}, \overline{w_2}, \cdots, \overline{w_m}\}$ should be decrypted and de-interleaved to get the watermark information.

3. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed watermarking scheme has been integrated into the H.264/AVC JM-8.6 reference software [14]. Four standard video sequences (i.e., News, Carphone, Hall and Container) in QCIF format (176 × 144) at a rate 30 frames/sec are used for our simulation. Some important configuration parameters of the reference software are given in Table.1. The remaining parameters retain their default values. The watermark to be embedded is a binary image with size 94 × 30, as shown in Fig.5(a).

Table.1. Configuration parameters of the JM

Parameter	Configuration
Profile	Baseline
Sequence type	"IPPPP"
Frames To Be Encoded	100
Frame Rate	30 fps
Intra Period	5
QP(Quantization Parameter)	28
Symbol Mode	0:CAVLC
RD Optimization	1:on

(a). Original watermark (b). Extracted watermark

Fig.5. Watermark image

In the experiments, no visible artifacts can be observed in all of the test video sequences. Fig.6 shows the average PSNR and bit rate comparison results. The test sequence is encoded with fixed quantization parameters *QP* = [24, 26, 28, 30, 32]. Simulation results show that the bit rate increase caused by watermarking is about 2.36%, 2.05%, 2.42%, 2.21% for the sequence Carphone, News, Container and Hall encoded with *QP* = 28.

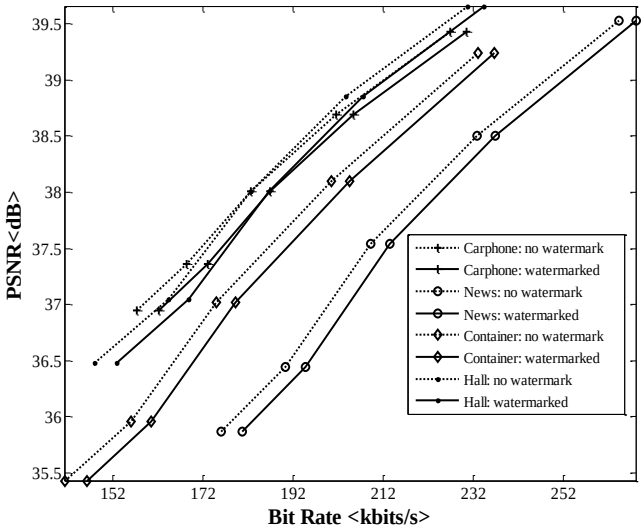
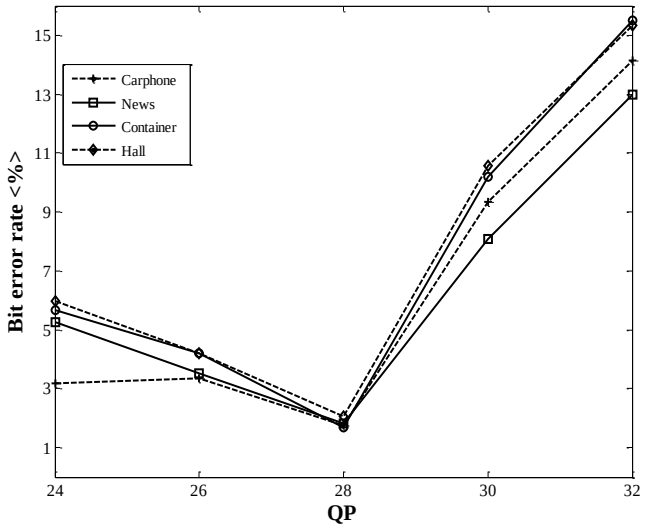


Fig.6. PSNR and bit rate comparison results

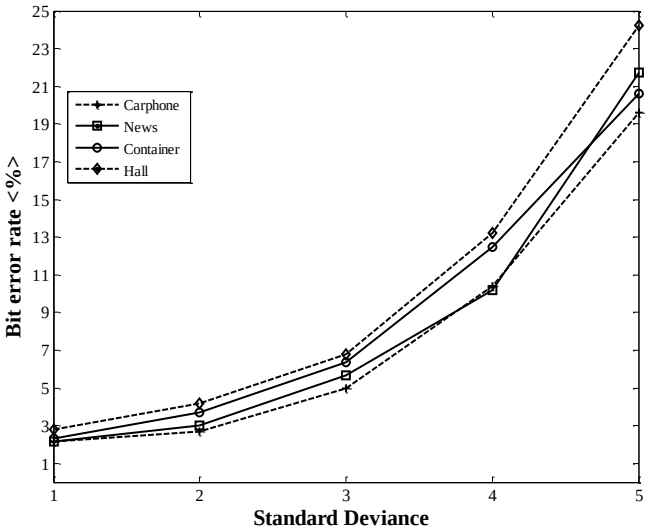
The watermark signature can be extracted completely without any attack, as shown in Fig.5(b). To test the robustness of the watermark algorithm, several attacks, including requantization transcoding, additive white Gaussian noise (AWGN), brightness and contrast adjustment are performed. Generally, the robustness is measured by bit error rate (BER) between the extracted watermark and the original watermark, as illustrated in Fig.7.

As shown in Fig.7(a), after requantization transcoding with different *QP*, the watermark can be extracted effectively. When the watermarked videos are transcoded with *QP* of 32, bit error rates are increased, but still below 15%. Next, we consider the

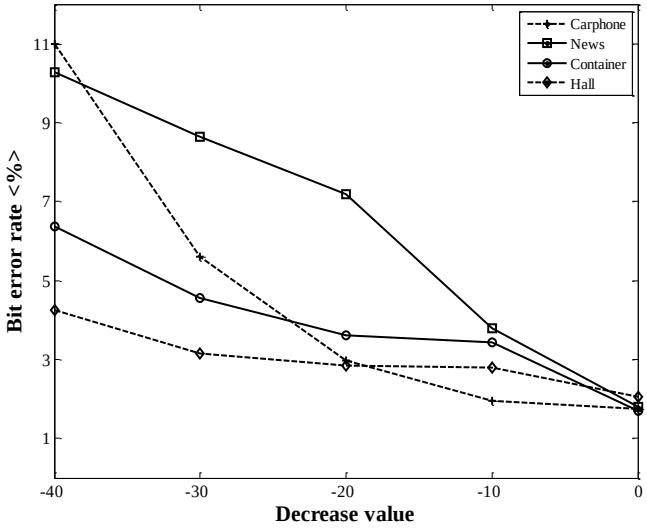
effect of additive white Gaussian noise. We add white noise of mean zero and standard deviance σ to each frame of the video sequence. The simulation results show that the watermark can still survive AWGN with Standard Deviation = 5 at least, as depicted in Fig.7(b). Finally, we examine the robustness of the proposed algorithm to brightness and contrast adjustment attacks. Studying Fig.7(c) ~ Fig.7(d) shows that the watermark can be extracted with low BER after brightness increase or decrease, and the maximum BER is below 13%. After contrast increase or decrease, the algorithm still achieves the desired performance as shown in Fig.7(e) ~ Fig.7(f). The above results show that the proposed method can resist common signal processing attacks.



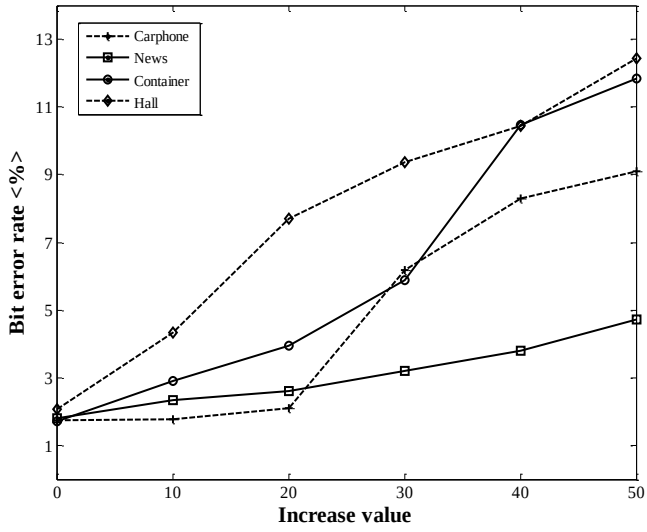
(a). Requantization transcoding



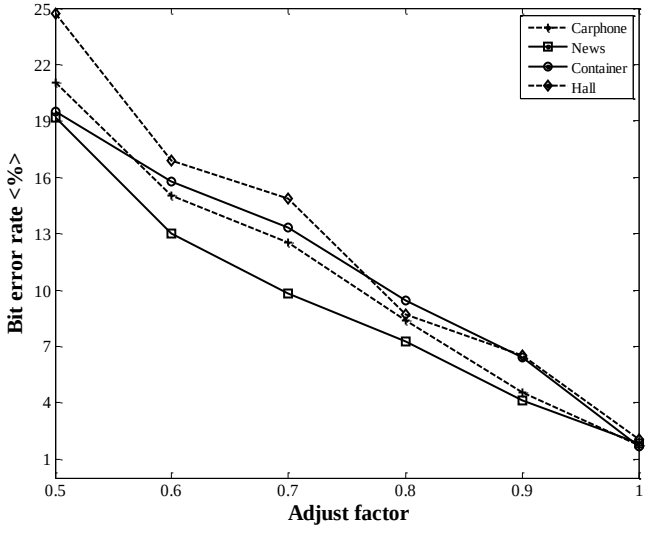
(b). AWGN



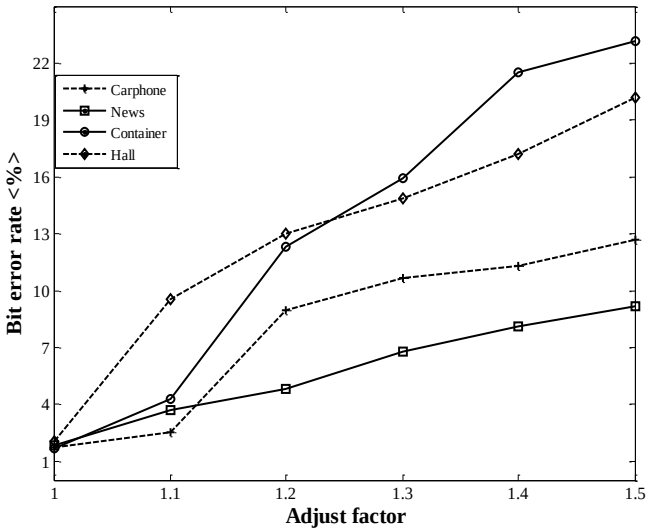
(c). Brightness decrease



(d). Brightness increase



(e). Contrast decrease



(f). Contrast increase

Fig.7. Watermark detection results after attacks

4. CONCLUSION

In this paper, a blind watermarking algorithm based on H.264/AVC codec is presented. In the embedding procedure, the watermark information is inserted by slightly modulating quantized residual coefficients in diagonal positions. The watermarked H.264/AVC video streams maintain good visual quality and slight increase in bit rate. The embedded watermark can be extracted blindly without the need of original video signals. The proposed algorithm avoids full decoding and re-encoding in both embedding and extracting phases. We also test the robustness of the proposed algorithm to several common signal processing attacks such as requantization transcoding, AWGN, brightness and contrast adjustment. Simulation results show that the algorithm is robust against these attacks. It is obvious that there is no perfect solution to achieve high robustness and security, low visual distortion and high payload simultaneously. The development of joint coding framework for compression, encryption, and watermarking [15], focusing on the H.264/AVC or the recently finalized High Efficiency Video Coding (HEVC) standard will be investigated as our future work.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (61301247), Zhejiang Provincial Natural Science Foundation of China (LY13F020013) and Ningbo Natural Science Foundation (2013A610059).

REFERENCES

[1] Yiqi Tew and Kok Sheik Wong, "An overview of Information Hiding in H.264/AVC Compressed Video", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 24, No. 2, pp. 305-319, 2014.

[2] Dawen Xu, Rangding Wang and Jicheng Wang, "A novel watermarking scheme for H.264/AVC video authentication", *Signal Processing: Image Communication*, Vol. 26, No. 6, pp. 267-279, 2011.

[3] M. Noorkami and R. M. Mersereau, "A framework for robust watermarking of H.264-encoded video with controllable detection performance", *IEEE Transactions on Information Forensics and Security*, Vol. 2, No. 1, pp. 14-23, 2007.

[4] Jing Zhang, A. T. S. Ho, Gang Qiu and P. Marziliano, "Robust video watermarking of H.264/AVC", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 54, No. 2, pp. 205-209, 2007.

[5] A. Mansouri, A. M. Aznaveh, Torkamani-Azar F and F. Kurugollu, "A Low Complexity Video Watermarking in H.264 Compressed Domain", *IEEE Transactions on Information Forensics and Security*, Vol. 5, No. 4, pp. 649-657, 2010.

[6] H. A. Aly, "Data hiding in motion vectors of compressed video based on their associated prediction error", *IEEE Transactions on Information Forensics and Security*, Vol. 6, No. 1, pp. 14-18, 2011.

[7] Jian Li, Hongmei Liu, Jiwu Huang and Yun Q. Shi, "Reference index-based H.264 video watermarking scheme", *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol. 8, No. 2S, pp. 1-22, 2012.

[8] Dawen Xu, Rangding Wang and Y. Q. Shi, "Data Hiding in Encrypted H.264/AVC Video Streams by Codeword Substitution", *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 4, pp. 596-606, 2014.

[9] Dawen Xu, Rangding Wang and Jicheng Wang, "Prediction mode modulated data-hiding algorithm for H.264/AVC", *Journal of Real-Time Image Processing*, Vol. 7, No. 4, pp. 205-214, 2012.

[10] Dawen Xu and Rangding Wang, "Watermarking in H.264/AVC Compressed Domain Using Exp-Golomb Code words Mapping", *Optical Engineering*, Vol. 50, No. 9, pp. 1-11, 2011.

[11] Yulin Wang and A. Pearmain, "Blind MPEG-2 video watermarking robust against geometric attacks: a set of approaches in DCT domain", *IEEE Transactions on Image Processing*, Vol. 15, No. 6, pp. 1536-1543, 2006.

[12] Dong Wook Kim, Young Geun Choi, Hwa Sung Kim, Ji Sang Yoo, Hyun Jun Choi and Young Ho Seo, "The Problems in Digital Watermarking into Intra-Frames of H.264/AVC", *Image and Vision Computing*, Vol. 28, No. 8, pp. 1220-1228, 2010.

[13] K. P. Lim, G. Sullivan and T. Wiegand, "Text description of joint model reference encoding methods and decoding concealment methods", *JVT of ISO/IEC MPEG and ITU-T VCEG Document*, JVT-O079, 2005.

[14] K. Suhring, "H.264/AVC Joint Model 8.6 (JM-8.6) Reference Software", Available at: http://iphome.hhi.de/suehring/html/.

[15] A. Boho, G. Van Wallendael, A. Dooms, J. De Cock, G. Braeckman, P. Schelkens, B. Preneel and R. Van de Walle, "End-To-End Security for Video Distribution: The Combination of Encryption, Watermarking, and Video Adaptation", *IEEE Signal Processing Magazine*, Vol. 30, No. 2, pp. 97-107, 2013.