A CAVLC-BASED VIDEO WATERMARKING SCHEME FOR H.264/AVC CODEC

SHINFENG D. LIN¹, CHIH-YAO CHUANG¹ AND MEI-JUAN CHEN²

¹Department of Computer Science and Information Engineering
²Department of Electrical Engineering
National Dong Hwa University
No. 1, Sec. 2, Da Hsueh Rd., Shoufeng, Hualien 97401, Taiwan
{david; cmj}@mail.ndhu.edu.tw

Received May 2010; revised October 2010

ABSTRACT. Forensics for digital data has become more and more important, especially for video surveillance systems. Recently, H.264/AVC CODEC has been widely applied to video surveillance systems. In this paper, therefore, a novel video watermarking scheme for H.264/AVC CODEC is proposed. It embeds a watermark bit by adaptively truncating the last nonzero quantized AC coefficients in a 4 × 4 luminance block. The existence of truncated coefficient has least perceptual degradation and least influence on successive prediction coding. Thus, the scheme could constrain the perceptual degradation, which is caused by embedding watermark, within an ideal degree. Experimental results show that the proposed scheme does not increase but even reduces the bit rate after embedding watermarks in general cases. Besides, the accuracy of extracted watermark bits after the re-encoding attack could be maintained over 80%. Moreover, since the scheme is low-complexity and implemented in the encoder, it can be easily adopted by real-time video applications.

Keywords: Video watermarking, H.264/AVC, Digital authentication

1. Introduction. H.264/AVC is the state-of-the-art video coding standard. This new standard provides a better coding efficiency than others. More and more digital video applications are adopting H.264/AVC codec, such as broadcast TV, video surveillance system, digital video camera and video conference. The same as digital data, H.264/AVC video sequence is prone to mass copy and tampering on the Internet. Therefore, copyright protection for it has become more and more important, especially for video surveillance systems.

Video watermarking is an adequate approach to achieve digital authentication and copyright protection for video data. Owing to the unique features of video coding, video watermarking is very different from image watermarking. The compressed video data have little capacity for data hiding in the code space. This is because video coding standards compress video sequences with much higher compression ratio than image compression techniques. On the other hand, the bit rate of video coding is very sensitive to the fluctuation between coefficients, especially in entropy coding. Therefore, the application of remarkable data hiding and image watermarking schemes [1-3] to video watermarking schemes would not be appropriate. In video encoder, if one bit is embedded in the frequency domain or motion field, the bit’s increasing usually will not be only one bit. Because the embedded bit would increase the size of current code space, it must destroy the reconstructed data for successive coding. Therefore, the successive coding has to employ more bits to compensate for the distortion additionally. Furthermore, low-complexity and
real-time detection are also necessary for video watermarking [4]. Thus, video watermarking is more difficult than image watermarking, especially for the high compression H.264 standard.

Different video watermarking techniques have been developed for different video coding standards. In the past decade, many video watermarking techniques have been proposed for MPEG CODEC. For example, Wolfgang et al. [5] employed the visual models, just noticeable difference (JND), to select embedding coefficients based on frequency sensitivity, luminance masking and contrast masking. Zhang et al. [6] proposed a video watermarking technique to hide copyright information in larger value motion vectors. Simitopoulos et al. [7] introduced a watermarking technique that embedded watermark bits directly into the MPEG stream. Barni and Bartolini [8] proposed an MPEG-4 watermarking technique with emphasis on watermarking in video objects. These methods [5-8] are appropriate watermarking solutions for MPEG and MPEG-4.

However, literature of MPEG watermarking technique cannot be applied to H.264 because of the difference between video coding standards. In H.264/AVC, most watermarking techniques embed the watermark in intra frames [9-14], because the existence of intra frames is more important than inter frames. On the other hand, inter frames have less capacity for data hiding due to higher compression ratio by motion compensation. Noorkami and Mersereau [9] embedded a watermark bit into the selected quantized coefficients, which is selected by a public key extracted from the feature of the macro-block. Tian et al. [10] improved the above method [9] by only selecting coefficients which are located within $4 \times 4$ intra mode. Then, the selected coefficients were modified according to CAVLC [15] coding efficiency in H.264/AVC. Noorkami and Mersereau [11] embedded watermark bits in I frame by measuring the motion intensity to avoid motion areas. Gong and Lu [12] employed a texture-masking-based perceptual model to adaptively determine the embedding strength of each block. Lu et al. [13] used the block parity and index modulation to select the quantized AC coefficient for embedding watermark bits. Chen et al. [14] embedded watermark bits according to block Sub-band Index and Coefficient Modulation. These methods [13,14] require plenty of extra computation for selecting embedded coefficients in the embedding procedure.

The above-mentioned methods [9-14] result in bit rate increasing and PSNR degression. In addition, due to selecting embedded blocks based on the features of current macro-block, the consistent volume of watermark bits for different sequences or different frames could not be maintained. This paper proposes an appropriate video watermarking scheme in H.264/AVC encoder. The proposed scheme embeds a watermark bit by adaptively varying the number of nonzero quantized AC coefficients in a $4 \times 4$ luminance block. It not only effectively restrains the bit rate increasing but also maintains very slight distortion from watermark embedding. This is because the proposed scheme only modifies the coefficient which existence or not has least influence on perceptual quality. Moreover, the drift error will not occur since the proposed scheme is implemented in the appropriate processing point of H.264 encoder. Therefore, the reconstructed frames of the encoder and decoder are fully consistent.

This paper is organized as follows: in Section 2, the related work is described; Section 3 presents the proposed video watermarking scheme in the luminance component of intra frames; Simulation results are demonstrated in Section 4; and finally, Section 5 draws the conclusions.
The first scenario can perform the partial decompression to select the embedded block, according to its mode [10] or the relative difference of the DC coefficients [9]. Therefore, the advantage of the first scenario is that it can minimize the distortion of embedded blocks, which is induced by watermark bits. However, a crucial problem of the first scenario is the drift error, which refers to the drift accumulations caused by watermark among different blocks during intra or inter predictions [16]. Moreover, another disadvantage is that the first scenario cannot be adopted in real-time encoder system, such as video conference.

Unlike the first scenario, the drift error will not occur in the second scenario. This is due to the fact that the distortion induced by the watermark will be estimated by its successive prediction in the encoder. Therefore, it is inevitable that bit rate increasing in the second scenario is more significant than in the first scenario. Furthermore, the second scenario does not select the appropriate or particular mode for embedding procedure, because the mode decision is ultimately determined by RDO (Rate-Distortion Optimization) after reconstructing data process. If the embedding procedure is implemented after RDO, the error drift will occur due to the embedding data was not referred in reconstructing procedure. In addition, the bit rate increasing of the second scenario is usually more than that of the first scenario. This is because the bit rate increasing in the second scenario is induced by the watermark embedding together with estimating the drift in the successive prediction.

Noorkami and Mersereau [9] adopted the first scenario. First, it has to partially decompress the encoded bitstream. Then, a watermark bit is embedded into the selected quantized coefficient within the macro-block. This selection is determined by the public key, extracted from the feature of each macro-block. This scheme embeds a watermark bit by adjusting the number of nonzero quantized coefficients either from even to odd or from odd to even. After the embedding procedure, the modified coefficients are partially encoded into the original bitstream. Meanwhile, this scheme will result in drift error while decoding the watermarked bitstream.

Tian et al. [10] supported a CAVLC-based blind watermarking scheme in H.264/AVC. They improved Noorkami’s technique [9] by employing the concept of CAVLC [11] coding efficiency. This scheme selects blocks belonging to $4 \times 4$ intra mode to embed the watermark. Then, the watermark data is only embedded into both the last non-zero and the non-trailing quantized AC coefficient of the selected $4 \times 4$ block. The selected coefficient is modified according to the odd-even relation between this coefficient and the watermark bit. Additionally, the improved scheme keeps the selected coefficient from being modified to ‘0’, because such modification will influence four components of the entropy encoding, CAVLC [15]: $coeff\_token$, $trailing\_ones\_sign\_flag$, $total\_zeros$ and $run\_before$. The advantage of Tian’s scheme is that it just influences one component of the entropy encoding, i.e., $level$. Therefore, this scheme could control the bit rate increasing, resulting from embedding the watermark, within 0.1%.

3. **The Proposed Scheme.** The proposed video watermarking scheme, belonging to the second scenario, is implemented in the H.264/AVC encoder. Watermark bits are embedded in all intra frames of one video sequence. Without performing a large amount of extra computation, this scheme embeds watermark bits by adaptively truncating the last nonzero quantized AC coefficient within a $4 \times 4$ block in the luminance component. The truncated coefficient is the most insignificant one, its existence or not will not make too much influence. The remarkable contribution of this scheme is to attain more low-complexity in embedding procedure and to restrain the bit rate increasing. This scheme is appropriate for the real-time system, such as video conference and video surveillance systems. It can also maintain perceptual quality after embedding watermarks.
The proposed video watermarking scheme within the H.264/AVC encoder is illustrated in Figure 1 with dotlines. It is implemented after the transform (integer DCT) and quantization which processing unit is 4 x 4 block-based. Then, the entropy coding and reconstructed procedure (inverse quantization and inverse transform) will receive the watermarked data in the meantime. Thus, the drift error will not occur in the proposed scheme. On the other hand, the selection procedure is based on a predetermined 4-bits secret key. The secret key selects one block within each macro-block for watermark embedding procedure, and it is constant throughout the entire encoding process. In other words, the proposed scheme does not select beneficial embeddee based on the particular intra mode or human visual model. This is because specific selections cause the non-consistent embeddable capacity of each frame. And it might increase the computation complexity. In the proposed scheme, however, the embeddable capacity for each frame is consistent and the complexity of the selection procedure is low.

The details of watermark embedding and extraction procedures are described as follows.

3.1. **Watermark embedding.** Followed by the procedure of the integer DCT and that of quantization, the number of nonzero AC coefficients in the selected block \( k \) (Block_ID), \( NumAC_k \), is counted. Next, the last nonzero AC coefficient is marked as \( AC_n \), where \( n \) is the order of this AC coefficient by Zig-Zag scan. Figure 2 shows an example from selecting block to finding operators (\( NumAC_k \) and \( AC_n \)) out. Then, the watermark is embedded in the block as Formula (1).

\[
AC_n = \begin{cases} 
AC_n & \text{if } \text{LSB}(NumAC_k) == W_i \\
0 & \text{otherwise}
\end{cases}
\]  

whether high frequency coefficients exist or not makes little distortion to the block. Therefore, Formula (1) demonstrates the proposed scheme embeds the watermark bit by truncating the last non-zero coefficient in the reordered series, while \( \text{LSB}(NumAC_k) \) and \( W_i \) are inconsistent. Then, the number of nonzero AC coefficients, \( NumAC_k \), is subtracted by one. For instance, if \( NumAC_k \) is odd and \( W_i \) is ‘0’, the last nonzero AC coefficient will be modified to ‘0’.

\[\text{Figure 1. The proposed watermark embedding scheme in H.264/AVC encoder}\]
However, Formula (1) could not be applied to a case: all 15 quantized AC coefficients in a luminance block are zero, such as AZBs (all-zero DCT coefficients blocks) [17]. This case is quite common in low bit rate video applications or sequence with more smooth textures. Meanwhile, Formula (2) will be adopted instead of Formula (1) in this case.

$$AC_0 = \begin{cases} AC_0 & \text{if } W_i == 0 \\ 1 & \text{otherwise} \end{cases}$$  \hspace{1cm} (2)

Formula (2) demonstrates the proposed scheme embeds the watermark bit by modifying $AC_0$ from ‘0’ to ‘1’, while $LSB(NumAC_k)$ and $W_i$ are inconsistent. Both Formulas are essential for the proposed scheme.

![Block Selection and the Determination of Operators](image)

**Figure 2.** Block selection and the determination of operators

### 3.2. Watermark detection

As shown in Figure 3, watermark detection of the technique is performed between entropy decoding and inverse quantization. First, the authenticator has to be informed the secret key, which can locate the $4 \times 4$ embedded block $k$. Second, the decoder counts the number of nonzero quantized AC in the embedded block $k$. If it is odd number, the watermark bit is ‘1’, otherwise it is ‘0’. In other words, the watermark bit can be extracted by making odd-even relation of the number, which means how many non-zero AC coefficients in the embedded block. The equation of watermark extraction is shown below.

$$W_i' = \begin{cases} 0 & \text{if } NumAC_k' \text{ is even} \\ 1 & \text{if } NumAC_k' \text{ is odd} \end{cases}$$  \hspace{1cm} (3)

### 4. Experimental Results

The proposed scheme was implemented in the H.264/AVC reference software JM15.0 [18]. Many standard benchmark video sequences, including two sizes: QCIF (176 × 144) and CIF (352 × 288), were employed: mobile, Stefan, flower, Foreman, etc. In our simulations, 150 frames for each sequence are encoded to the watermarked bitstream. Main parameter settings of the JM15.0 encoder are listed as follows:

- **Profile/Level IDC**: 66,44 (Baseline)
- **Sequence Type**: IPPP (QP: I 28, P 28)
- **Frame Rate**: 30fps
- **Period of I Frame**: 15
- **Transform 8 × 8 Mode**: 0
- **Search Mode**: Fast Full Search (Default)
- **Rate Control**: 0 (Default)
- **RD Optimization**: 1 (Default)
In experiments, each 16 × 16 macro-block in I frames carries one watermark bit. Therefore, there are 990 bits embedded to a QCIF sequence and 3960 bits embedded to a CIF sequence. Average bit rate increasing and the PSNR degradation of watermarked sequences (150 frames) are listed in Table 1. As shown in Table 1, the proposed video watermarking scheme leads to a slight distortion for watermarked sequences. This is because this scheme adaptively truncates the most insignificant coefficient for minimizing the influence, resulted from a watermark bit embedding in a 4 × 4 block. And the distortion induced by watermark bits embedding will be estimated by the successive prediction. Therefore, drift error will not occur.

**Table 1.** The bit rate increasing and PSNR degradation of watermarked sequences

<table>
<thead>
<tr>
<th>Resolution Measurement Sequence</th>
<th>QCIF</th>
<th></th>
<th>CIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Bit Rate</td>
<td>△ Bit Rate</td>
<td>△ PSNR</td>
<td>Original Bit Rate</td>
</tr>
<tr>
<td>mobile</td>
<td>507.61</td>
<td>-0.146%</td>
<td>-0.02</td>
</tr>
<tr>
<td>Stefan</td>
<td>414.21</td>
<td>-0.041%</td>
<td>-0.02</td>
</tr>
<tr>
<td>flower</td>
<td>347.04</td>
<td>-0.167%</td>
<td>-0.02</td>
</tr>
<tr>
<td>tempeste</td>
<td>325.42</td>
<td>-0.058%</td>
<td>-0.01</td>
</tr>
<tr>
<td>table</td>
<td>206.30</td>
<td>-0.150%</td>
<td>-0.02</td>
</tr>
<tr>
<td>Foreman</td>
<td>169.33</td>
<td>-0.443%</td>
<td>-0.02</td>
</tr>
<tr>
<td>news</td>
<td>115.91</td>
<td>0.293%</td>
<td>-0.05</td>
</tr>
<tr>
<td>Hall</td>
<td>91.76</td>
<td>-0.174%</td>
<td>-0.02</td>
</tr>
</tbody>
</table>

As for bit rate increasing, which is shown in the gray-background column of Table 1, the average increase of the sequence size in most cases is minus. That means the bit rate after embedding watermark actually decreases in most cases. The appearance of bit rate decreasing is wildly different from other video watermarking schemes. There are two reasons that lead to this result: (1) The number of nonzero quantized AC coefficients is subtracted by one. As to the CAVLC [15] coding efficiency, this will shorten the encoded bits of the embedded block. (2) Because the truncated coefficient is the least important one, there will be almost no extra distortion. The successive coding requires few bits to estimate and compensate the distortion induced by embedding watermark bits. However, when the embedded 4 × 4 block is located in the smooth region, encoded bits may increase a little. Because the block’s quantized AC coefficients would be all zero, the embedding procedure need to adopt Formula (2). Then, plenty of additional
information is oppositely required to entropy encode the modified AC₀, from ‘0’ to ‘1’, such as `trailing_ones_sign_flag`, `total_zeros` and `run_before`. Therefore, sequences with more smooth textures (news and Hall, etc.) or higher resolution (CIF) are inferior to restraining bit rate increasing.

After embedding procedure, if the watermarked sequence does not suffer any attack, the embedded watermarks can be extracted perfectly. On the Internet, people usually re-encode the downloaded video data in order to reduce the size of video bitstream, even tamper the downloaded video data then re-encode to be re-encoded bitstream, before widely distributing it. Therefore, this paper re-encodes the watermarked video data. The measurement of resistance is the recovery rate. It means the average accuracy rate of extracting watermark bits from the re-encoded bitstream, which size has been less than original watermarked bitstream. In our re-encoding evaluation, watermarked sequences are re-encoded by the original JM15.0 encoder with the same parameter settings. The recovery rate after re-encoding watermarked sequences is listed in Table 2. According to the observation, the watermark bits can be extracted with 80% accuracy from re-encoded bitstream, when its data size is reduced to 87% (CIF, Hall). In general cases, the recovery rate from re-encoding watermarked sequences could be maintained over 80%.

**Table 2.** The watermark recovery rate after H.264 re-encoding attack

<table>
<thead>
<tr>
<th>Resolution Measure-</th>
<th>QCIF</th>
<th>CIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>Data Size</td>
<td>Avg.</td>
</tr>
<tr>
<td></td>
<td>of Bitstream</td>
<td>Recovery Rate</td>
</tr>
<tr>
<td>mobile</td>
<td>-2.87%</td>
<td>90.81%</td>
</tr>
<tr>
<td>Stefan</td>
<td>-3.55%</td>
<td>90.40%</td>
</tr>
<tr>
<td>flower</td>
<td>-3.21%</td>
<td>83.03%</td>
</tr>
<tr>
<td>tempete</td>
<td>-4.64%</td>
<td>88.48%</td>
</tr>
<tr>
<td>table</td>
<td>-9.24%</td>
<td>80.20%</td>
</tr>
<tr>
<td>Foreman</td>
<td>-7.99%</td>
<td>83.54%</td>
</tr>
<tr>
<td>news</td>
<td>-5.91%</td>
<td>79.39%</td>
</tr>
<tr>
<td>Hall</td>
<td>-5.83%</td>
<td>80.91%</td>
</tr>
</tbody>
</table>

Noorkami and Mersereau [9] and Tian et al. [10] embedded 22 ~ 85 bits in one I frame. Because Noorkami and Mersereau [9] selected the embedded coefficient by several bits (Kₚᵢ) and Tian et al. [10] only selected the block of 4 × 4 intra mode to embed, these methods could not embed the same watermark in different sequences. The proposed scheme, however, does not select the beneficial mode or the macro-block to embed. For each macro-block, one bit has to be embedded. Therefore, the proposed scheme embeds more watermark bits than those methods. Comparisons of the bit rate increasing and recovery rate with above-mentioned methods [9,10] is shown in Table 3. It is worth noticing that even the proposed scheme embeds more bits, our scheme still outperforms other methods in general cases.

5. **Conclusions.** A low-complexity video watermarking scheme in H.264/AVC encoder has been proposed. Watermark bits are embedded in quantized 4 × 4 luminance blocks by adaptively varying the number of nonzero AC coefficients in embedded blocks. The scheme not only effectively restrains the bit rate increasing but also maintains very slight distortion from watermark embedding. If an authenticator intends to extract watermarks, watermark bits can be extracted by counting the number of nonzero quantized AC coefficients in embedded blocks. Simulation results show that the proposed scheme performs...
Table 3. Comparisons of the bit rate increasing after watermark embedding and watermark recovery rate after re-encoding

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Embedded Watermark Bits (per 1 frame)</td>
<td>Re-encoding Recovery Rate</td>
<td>Embedded Watermark Bits (per 1 frame)</td>
</tr>
<tr>
<td>mobile</td>
<td>85</td>
<td>0.23%</td>
<td>85</td>
</tr>
<tr>
<td>tempete</td>
<td>81</td>
<td>0.44%</td>
<td>83</td>
</tr>
<tr>
<td>CarPhone</td>
<td>44</td>
<td>0.80%</td>
<td>58</td>
</tr>
<tr>
<td>mother</td>
<td>42</td>
<td>0.69%</td>
<td>68</td>
</tr>
<tr>
<td>table</td>
<td>38</td>
<td>0.31%</td>
<td>62</td>
</tr>
<tr>
<td>Claire</td>
<td>22</td>
<td>0.44%</td>
<td>83</td>
</tr>
</tbody>
</table>

excellent in terms of restraining the bit rate increasing and perceptual degradation, especially for sequences with more textures. Therefore, the proposed scheme provides an adequate mechanism for copyright protection and digital authentication for H.264/AVC applications.

REFERENCES


