Fragile Video Watermarking Technique by Motion Field Embedding with Rate-distortion Minimization

Tien-Ying Kuo
Dept of Electrical Engineering, National Taipei University of Technology, Taiwan, R.O.C.
tykuo@ee.ntut.edu.tw

Yi-Chung Lo
Dept of Electrical Engineering, National Taipei University of Technology, Taiwan, R.O.C.
yclo@image.ee.ntut.edu.tw

Chen-I Lin
Dept of Electrical Engineering, National Taipei University of Technology, Taiwan, R.O.C.
cylin@image.ee.ntut.edu.tw

Abstract

In this paper, we proposed a fragile video watermarking technique to authenticate the H.264 video content. Watermark information is embedded in motion vectors which have fragile nature. The proposed method finds out the best locations of motion vectors to embed the information to achieve the fragility, where are based on the statistical analysis of the motion vector by the H.264/AVC Rate-Distortion cost function. This scheme does not need the original video sequence for watermark detection. Experimental results show that the proposed watermarking technique can keep the perceptual quality at best effort and still has good fragility.

1. Introduction

With recent fast developing in digital technology, the interest and the need in the copyright protection and the authentication of digital multimedia data, such as text, audio, images and video, becomes more and more important. A secure authentication system is useful in showing that no tampering has occurred during situations where the credibility of a multimedia data may be questioned. Fragile watermarks are designed to detect slight changes to the watermarked multimedia data [1]. The primary focus of fragile watermarking systems is to authenticate and detect tampering. To achieve this goal, several desirable features have been identified in [2].

The fragile watermarking system focused on the authentication of MPEG video is proposed by Yin and Yu [5]. The feature of video frame is extracted and time code is used. Both of them are encrypted with private key and then are treated as watermark to embed in the quantized AC coefficients. Their disadvantage is the localization accuracy will reduce as the size of block increases, and the adopted RSA algorithm also increases the computation complexity a lot.

The H.264 standard is the latest video codec developed by the Joint Video Team (JVT) [3], which introduces several new coding tools to improve on the rate-distortion performance to the past coding standards, such as MPEG-x and H.263. Thus, issues of authentication that is appropriate for this standard become very important. There are not many research works dealing with authenticating the H.264/AVC standard. We shall focus on the authentication of H.264 video.

This paper describes a fragile watermarking scheme of H.264/AVC video format, working during encoding process. The watermark is embedded in motion vector by exploiting its fragile nature. Based on the statistical analysis of the motion vector using the H.264/AVC Rate-Distortion cost function, we can find out the best locations of motion vectors to embed the extracted feature of the previous video frame, to achieve the goal of fragility. One merit of our scheme is no need of original video sequence for watermark detection.

This paper is organized as follows. Section 2 presents the proposed approach. Simulation results are given in Section 3. Finally, Section 4 concludes the paper.

2. Proposed approach

Compared with the fixed block size motion compensation in traditional video codec, the variable block size motion compensation in H.264/AVC can not only precisely represent the true motion activity in a macroblock, but also provide more positions for watermark embedding. In the inter frame coding, the video encoder compresses the temporal redundancies between frames by motion compensation with motion vector estimation. We insert the watermark into original motion vectors. The new prediction error, i.e.
residue, is then computed for the best prediction mode. Finally, the new motion vectors together with new prediction errors are encoded into compressed bitstreams. Therefore, we can extract the watermark without decompressing the video.

2.1. Feature extraction

The localization for fragile watermark can be used to describe the spatial location of the tampering. To achieve a good and fast tamping detection, the localization feature should be well designed as below:

Step 1. Splitting the targeted video frame into several rectangular groups: For example, we used 9 rectangular groups per frame in QCIF format (176 × 144 pixels) as shown in Fig. 1(a).

Step 2. Dividing group into subgroups: For each rectangular group, we break it up into 3 subgroups, which consist of several macroblocks in a row. In general, since the interested object is usually concentrated in the central region of frame, we take 3 macroblocks as one subgroup if locating in the central region and 4 macroblocks in left or right subgroup.

Step 3. Generating authentication data: As the H.264/AVC compression standard using a DCT transform of size 4 × 4 as shown in Fig. 1(b), we have 16 blocks inside an H.264 macroblock. We can randomly select one coefficient from each DCT block via a private key. After selection, for each macroblock, we can get sixteen coefficient values. An Exclusive-Or (XOR) operation is then performed on the six-teen coefficient values within one macroblock, it generates one coefficient value. Finally, a subgroup coefficient is generated as the watermark for each subgroup by performing XOR again on previous XOR-operated macroblock coefficients within the subgroup.

Exploiting the motion vectors to carry the watermark has been proposed in [6] and [7]. Their problem is that they select embedding location by intuition without reasoning. In contrast, we propose to select the embedding locations by statistics using the BDPSNR (Bjontegaard Delta PSNR) [8], which is derived from JVT group to evaluate the average PSNR differences over all bit rates between the RD curves of two algorithms.

In H.264, the encoder does not directly transmit motion vector to the decoder. Instead, it sends the difference value of motion vector (DeltaMV) between MV and its predicted value estimated from neighboring blocks. Let |ΔMV| and |MV| represent the length of the DeltaMV and MV respectively. The length is defined as the absolute sum of x and y components (MVx and MVy). We first categorize MV into 7 classes according to their length. And we then perform watermarking embedding on MV (by Section 2.3 method) for each class separately to measure their PSNR degradation and bit rate increase, the BDPSNR is then calculated and plotted in Fig. 2, which shows two video sequences as other sequences having the similar trend. In Fig. 2, we observe some classes of MVs, i.e., |MV| is from 1 to 10, are more suitable to embed because they produce little BDPSNR degradation. Therefore, we will pick up the MVs within the class of length 1 to 10, and select the final embedded location from them by a private key and frame number.

2.2. Embedding position selection

During the embedding process, the value of MV (or MVD) is changed and offset by the watermark. So we have to find a criterion to get the best motion vector offset by minimizing the Lagrangian R-D cost function of H.264/AVC:

$$J = \min \{D(S, MV) + \lambda \cdot R(MVD)\}$$

Where \(\lambda\) denotes the predetermined Lagrange multiplier, and \(D\) and \(R\) represent the distortion and consumed bits for encoding the current mode, respectively. H.264 uses (1) to choose the best mode and the best motion vector of the mode.

2.3. Watermark embedding

Fig. 2. Selecting embedding positions by using the BDPSNR statistics

During the embedding process, the value of MV (or MVD) is changed and offset by the watermark. So we have to find a criterion to get the best motion vector offset by minimizing the Lagrangian R-D cost function of H.264/AVC:

$$J = \min \{D(S, MV) + \lambda \cdot R(MVD)\}$$

Where \(\lambda\) denotes the predetermined Lagrange multiplier, and \(D\) and \(R\) represent the distortion and consumed bits for encoding the current mode, respectively. H.264 uses (1) to choose the best mode and the best motion vector of the mode.
Once the best vector has been obtained and its length is qualified to embed (as described in Section 2.2), we can embed the watermark (generated in Section 2.1) into it, we proposed the watermarking algorithm as follows:

\[
\begin{align*}
&\text{if } ((MV_x\%2) \text{ XOR } (MV_y\%2)) \neq WM \\
&\quad (MV'_x, MV'_y) = (MV_x + 1, MV_y) ; \\
&\quad \text{or } (MV'_x, MV'_y) = (MV_x - 1, MV_y) ; \\
&\quad \text{or } (MV'_x, MV'_y) = (MV_x, MV_y + 1) ; \\
&\quad \text{or } (MV'_x, MV'_y) = (MV_x, MV_y - 1) ; \\
&\text{else} \\
&\quad MV'_x = MV_x ; \\
&\quad MV'_y = MV_y ; 
\end{align*}
\]

Where MV'_x and MV'_y denote the watermarked MV_x and MV_y; WM is the binary bit to be embedded. If the above equation is satisable, then MV is not changed. Otherwise, we have to modify and offset the horizontal component MV_x or the vertical component MV_y of motion vector in rate-distortion sense. For example in Fig. 3, the best quarter-pixel motion vector before watermarking is determined in (1) as in location 7. During the motion search, we also record the best half-pixel motion vector (location B). When we embed the WM, if (2) is satisfied, the embedded motion vector is not changed (still in location 7). Otherwise, we have to modify the motion vector from location 7 to one of locations B, 1, 3, 6 and 8 as selected by the one with minimal value of (1) with the replacement of MV to MV'.

\[
\begin{align*}
&MV'_x = MV_x + 1 ; \\
&MV'_y = MV_y ; 
\end{align*}
\]

Where MV'_x and MV'_y denote the watermarked MV_x and MV_y; WM is the binary bit to be embedded. If the above equation is satisable, then MV is not changed. Otherwise, we have to modify and offset the horizontal component MV_x or the vertical component MV_y of motion vector in rate-distortion sense. For example in Fig. 3, the best quarter-pixel motion vector before watermarking is determined in (1) as in location 7. During the motion search, we also record the best half-pixel motion vector (location B). When we embed the WM, if (2) is satisfied, the embedded motion vector is not changed (still in location 7). Otherwise, we have to modify the motion vector from location 7 to one of locations B, 1, 3, 6 and 8 as selected by the one with minimal value of (1) with the replacement of MV to MV'.

\[
\begin{align*}
&MV'_x = MV_x + 1 ; \\
&MV'_y = MV_y ; 
\end{align*}
\]

Where MV'_x and MV'_y denote the watermarked MV_x and MV_y; WM is the binary bit to be embedded. If the above equation is satisable, then MV is not changed. Otherwise, we have to modify and offset the horizontal component MV_x or the vertical component MV_y of motion vector in rate-distortion sense. For example in Fig. 3, the best quarter-pixel motion vector before watermarking is determined in (1) as in location 7. During the motion search, we also record the best half-pixel motion vector (location B). When we embed the WM, if (2) is satisfied, the embedded motion vector is not changed (still in location 7). Otherwise, we have to modify the motion vector from location 7 to one of locations B, 1, 3, 6 and 8 as selected by the one with minimal value of (1) with the replacement of MV to MV'.

\[
\begin{align*}
&MV'_x = MV_x + 1 ; \\
&MV'_y = MV_y ; 
\end{align*}
\]

Where MV'_x and MV'_y denote the watermarked MV_x and MV_y; WM is the binary bit to be embedded. If the above equation is satisable, then MV is not changed. Otherwise, we have to modify and offset the horizontal component MV_x or the vertical component MV_y of motion vector in rate-distortion sense. For example in Fig. 3, the best quarter-pixel motion vector before watermarking is determined in (1) as in location 7. During the motion search, we also record the best half-pixel motion vector (location B). When we embed the WM, if (2) is satisfied, the embedded motion vector is not changed (still in location 7). Otherwise, we have to modify the motion vector from location 7 to one of locations B, 1, 3, 6 and 8 as selected by the one with minimal value of (1) with the replacement of MV to MV'.

\[
\begin{align*}
&MV'_x = MV_x + 1 ; \\
&MV'_y = MV_y ; 
\end{align*}
\]

Where MV'_x and MV'_y denote the watermarked MV_x and MV_y; WM is the binary bit to be embedded. If the above equation is satisable, then MV is not changed. Otherwise, we have to modify and offset the horizontal component MV_x or the vertical component MV_y of motion vector in rate-distortion sense. For example in Fig. 3, the best quarter-pixel motion vector before watermarking is determined in (1) as in location 7. During the motion search, we also record the best half-pixel motion vector (location B). When we embed the WM, if (2) is satisfied, the embedded motion vector is not changed (still in location 7). Otherwise, we have to modify the motion vector from location 7 to one of locations B, 1, 3, 6 and 8 as selected by the one with minimal value of (1) with the replacement of MV to MV'.

\[
\begin{align*}
&MV'_x = MV_x + 1 ; \\
&MV'_y = MV_y ; 
\end{align*}
\]

Where MV'_x and MV'_y denote the watermarked MV_x and MV_y; WM is the binary bit to be embedded. If the above equation is satisable, then MV is not changed. Otherwise, we have to modify and offset the horizontal component MV_x or the vertical component MV_y of motion vector in rate-distortion sense. For example in Fig. 3, the best quarter-pixel motion vector before watermarking is determined in (1) as in location 7. During the motion search, we also record the best half-pixel motion vector (location B). When we embed the WM, if (2) is satisfied, the embedded motion vector is not changed (still in location 7). Otherwise, we have to modify the motion vector from location 7 to one of locations B, 1, 3, 6 and 8 as selected by the one with minimal value of (1) with the replacement of MV to MV'.

\[
\begin{align*}
&MV'_x = MV_x + 1 ; \\
&MV'_y = MV_y ; 
\end{align*}
\]

Where MV'_x and MV'_y denote the watermarked MV_x and MV_y; WM is the binary bit to be embedded. If the above equation is satisable, then MV is not changed. Otherwise, we have to modify and offset the horizontal component MV_x or the vertical component MV_y of motion vector in rate-distortion sense. For example in Fig. 3, the best quarter-pixel motion vector before watermarking is determined in (1) as in location 7. During the motion search, we also record the best half-pixel motion vector (location B). When we embed the WM, if (2) is satisfied, the embedded motion vector is not changed (still in location 7). Otherwise, we have to modify the motion vector from location 7 to one of locations B, 1, 3, 6 and 8 as selected by the one with minimal value of (1) with the replacement of MV to MV'.

\[
\begin{align*}
&MV'_x = MV_x + 1 ; \\
&MV'_y = MV_y ; 
\end{align*}
\]

Where MV'_x and MV'_y denote the watermarked MV_x and MV_y; WM is the binary bit to be embedded. If the above equation is satisable, then MV is not changed. Otherwise, we have to modify and offset the horizontal component MV_x or the vertical component MV_y of motion vector in rate-distortion sense. For example in Fig. 3, the best quarter-pixel motion vector before watermarking is determined in (1) as in location 7. During the motion search, we also record the best half-pixel motion vector (location B). When we embed the WM, if (2) is satisfied, the embedded motion vector is not changed (still in location 7). Otherwise, we have to modify the motion vector from location 7 to one of locations B, 1, 3, 6 and 8 as selected by the one with minimal value of (1) with the replacement of MV to MV'.

\[
\begin{align*}
&MV'_x = MV_x + 1 ; \\
&MV'_y = MV_y ; 
\end{align*}
\]

Where MV'_x and MV'_y denote the watermarked MV_x and MV_y; WM is the binary bit to be embedded. If the above equation is satisable, then MV is not changed. Otherwise, we have to modify and offset the horizontal component MV_x or the vertical component MV_y of motion vector in rate-distortion sense. For example in Fig. 3, the best quarter-pixel motion vector before watermarking is determined in (1) as in location 7. During the motion search, we also record the best half-pixel motion vector (location B). When we embed the WM, if (2) is satisfied, the embedded motion vector is not changed (still in location 7). Otherwise, we have to modify the motion vector from location 7 to one of locations B, 1, 3, 6 and 8 as selected by the one with minimal value of (1) with the replacement of MV to MV'.
Table I. PSNR and Rate-distortion

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Foreman</th>
<th>News</th>
<th>Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP 28</td>
<td>PSNR 35.79</td>
<td>36.70</td>
<td>30.65</td>
</tr>
<tr>
<td></td>
<td>BR 81.80</td>
<td>28.62</td>
<td>47.73</td>
</tr>
<tr>
<td>QP 36</td>
<td>PSNR 30.69</td>
<td>36.63</td>
<td>30.58</td>
</tr>
<tr>
<td></td>
<td>BR 88.90</td>
<td>29.84</td>
<td>54.58</td>
</tr>
<tr>
<td>Zhu [6]</td>
<td>PSNR 35.74</td>
<td>30.53</td>
<td>36.62</td>
</tr>
<tr>
<td></td>
<td>BR 87.90</td>
<td>29.84</td>
<td>54.58</td>
</tr>
<tr>
<td>Zhang [7]</td>
<td>PSNR 35.76</td>
<td>30.58</td>
<td>36.63</td>
</tr>
<tr>
<td></td>
<td>BR 87.34</td>
<td>29.56</td>
<td>52.13</td>
</tr>
<tr>
<td>Qiu [10]</td>
<td>PSNR 35.75</td>
<td>30.57</td>
<td>36.62</td>
</tr>
<tr>
<td></td>
<td>BR 86.93</td>
<td>29.45</td>
<td>51.95</td>
</tr>
<tr>
<td>Proposed WM</td>
<td>PSNR 35.76</td>
<td>30.60</td>
<td>36.68</td>
</tr>
<tr>
<td></td>
<td>BR 85.46</td>
<td>29.21</td>
<td>49.68</td>
</tr>
<tr>
<td>Sequences</td>
<td>Container</td>
<td>Silent</td>
<td>Paris</td>
</tr>
<tr>
<td>QP 28</td>
<td>PSNR 35.94</td>
<td>30.39</td>
<td>35.74</td>
</tr>
<tr>
<td></td>
<td>BR 27.68</td>
<td>8.42</td>
<td>63.62</td>
</tr>
<tr>
<td>Non-WM</td>
<td>PSNR 30.30</td>
<td>35.60</td>
<td>30.17</td>
</tr>
<tr>
<td></td>
<td>BR 36.88</td>
<td>9.65</td>
<td>70.58</td>
</tr>
<tr>
<td>Zhu [6]</td>
<td>PSNR 35.86</td>
<td>30.30</td>
<td>35.60</td>
</tr>
<tr>
<td></td>
<td>BR 36.88</td>
<td>9.65</td>
<td>70.58</td>
</tr>
<tr>
<td>Zhang [7]</td>
<td>PSNR 35.84</td>
<td>30.35</td>
<td>35.61</td>
</tr>
<tr>
<td></td>
<td>BR 36.36</td>
<td>9.65</td>
<td>70.17</td>
</tr>
<tr>
<td>Qiu [10]</td>
<td>PSNR 35.85</td>
<td>30.33</td>
<td>35.62</td>
</tr>
<tr>
<td></td>
<td>BR 35.15</td>
<td>9.70</td>
<td>69.80</td>
</tr>
<tr>
<td>Proposed WM</td>
<td>PSNR 35.85</td>
<td>30.35</td>
<td>35.64</td>
</tr>
<tr>
<td></td>
<td>BR 33.01</td>
<td>9.32</td>
<td>68.39</td>
</tr>
</tbody>
</table>

Fig. 5. Rate-distortion curves

Fig. 6. The watermarked and attached frames

We also tested the fragility of our watermarking algorithm by testing the case of transcoding. One result is presented in Fig. 6. As illustrated in Figs. 6 our algorithm can effectively identify the modification area, as indicated by the black rectangles. It shows our algorithm has a good locating capability for fragility test.

4. Conclusions

A fragile video watermarking via motion vector embedding has been proposed. We extract feature of the previous video frame and embed it on the motion vectors at the current video frame in optimizing the rate-distortion sense. Therefore, our algorithm can achieve the goal of fragile watermarking while keeping a better R-D performance than the literature works.

5. Acknowledgement

This work was supported by the National Science Council (Grant #96-2221-E-027-009-, 96-2422-H-194-002- ) projects.

6. References