A Compressed Video Watermarking Scheme with Temporal Synchronization

CHEN Chao, GAO Tie-gang, LI Li-zong
College of Software, Nankai University, Tianjin, China, 300071
E-mail: chchcomputer@gmail.com

Abstract
As an effective technology for copyright protection, digital watermarking has been applied in videos. Temporal synchronization is an important aspect for video watermarking. If synchronization is lost, the detector may fail in extracting the correct watermark. This paper proposes a temporal synchronous compressed MPEG-4 video watermarking scheme. This scheme utilizes DCT direct current coefficients to divide video into several scenes. Watermarks are embedded according to features of scenes. Bit rate control is introduced to ensure that watermark embedding does not increase the size of video. Drift compensation is employed to counteract the visual distortion in P-frames and B-frames caused by watermark embedding. Visual masking is exploited to enhance the invisibility of watermark. Experimental results show that the scheme has satisfactory ability in resisting the temporal synchronization attacks of frame deleting and frame inserting, therefore achieving well robustness.

Keywords: MPEG-4; video watermarking; temporal synchronization

1. Introduction

Due to the rapid development of the Internet, more and more digital medium are stored and transmitted on the Internet. The visual quality of a coped digital media is the same as the original media. This makes pirating very easy. Digital watermarking is an effective technique for protecting digital copyright[1,2]. Its basic idea is embedding imperceptible watermarking signal into the original digital media, which is considered as host signal. The watermarking process should not tamper with the audiovisual quality, which means that watermarking must have imperceptibility. After the watermarked media has undergone some operations or processing, the watermarking should still exist in the media and can be detected or extracted. So it means that watermarking should have robustness. When copyright dissension happens, the owner of the copyright can prove and maintenance his or her rights and interests, using the watermark extracted from the media.

In the very early years, watermarking has been extensively investigated for still images. And now it is also used in other fields such as video. As videos are always stored and transmitted in compressed form, large numbers of researches concentrate on the compressed video, especially MPEG-2 and MPEG-4 video. In conventional video watermarking schemes, a compressed video stream is first decompressed into standard video, a watermark signal is then embedded into the video signal, and finally, the watermarked video is recompressed. This technique requires fully decompressing and recompressing the video stream, and takes a lot of computer processing time. Obviously, this technique does not meet the demand of the Internet stream media, which emphasis on the characteristic of real time. Besides, this technique does not consider the compression standard at all. When the watermarked video is compressed later, the...
robustness of watermark is not ensured. To achieve good
effect in practice, the ideal fashion is embedding the
watermark directly into the compressed domain. In the
compressed domain watermarking schemes, the original
compressed video is partially decoded. Then, the partially
decoded bit stream is modified in order to embed the
watermark signal. Finally, the watermarked bit stream is
compressed to form watermarked video.

According to the location watermark embedded into,
MPEG video watermarking schemes can be divided into
different types. The first step to decode a MPEG video is to
get the VLC(Variable Length Code). Embedding
watermark into VLC can achieve lower computational
complexity. In [3], Langelaar and Lagendijk proposed a
scheme in which watermark is added by modifying the
\((\text{run,level})\) code generated by a VLC. First, \((\text{run,level})\)
codes are divided into many groups, where each group
contains codes of same length and the \textit{level} difference
in each group is exactly one. During watermark
embedding, a \((\text{run,level})\) pair is either left unchanged or
replaced, depending on the embedded watermark bit. This
method is basically a least significant bit (LSB) approach,
quality degradation of a video after embedding can be
almost negligible. In [4], a MB(macro block) based video
watermarking scheme in VLC domain was proposed by
Lu, Chen and Fan. For each MB in a frame, the average
value of \textit{level} generated by VCLs within this MB is
calculated. \(u(i)\) is the resulting average value of \(MB(i)\).
Arrange all average values as a list, from \(u(1)\) to \(u(n)\),
where \(n\) is the total quantity of MB within a frame.
Calculate \(u'(i)\) as the average value of the listed elements in a fixed-length sliding window whose center is
\(u(i)\). Through adding a single watermark bit into all the
VCLs within \(MB(i)\), the sign of the difference between
\(u(i)\) and \(u'(i)\), i.e. \(\text{sgn}(u(i)−u'(i))\) is made equal to
the watermark bit value. When watermark detecting, the
watermark can be obtained by recalculating every
\(\text{sgn}(u(i)−u'(i))\).

After VLC decoding, the next step forward is to get
the DCT coefficients of each MB by inversed
quantization. Many excellent watermarking schemes in
DCT domain were proposed. In [5], Langelaar and
Lagendijk proposed a differential energy watermarking
approach, which is transplanted from the research of still
image watermarking. It is based on selectively discarding
high frequency DCT coefficients in the compressed data
stream. Watermark is embedded into I-frames. The \(8\times8\)
pixels blocks of the video frame are first pseudo randomly
shuffled. The obtained shuffled frame is then split into \(n\)
\(8\times8\) blocks. One bit is embedded into each one of those
blocks by introducing an energy difference between the
high frequency DCT-coefficients of the top half of the
block (region A) and the bottom half (region B). The
value of the embedded bit is encoded as the sign of the
energy difference \(D\) between the two regions A and B.
Finally the shuffling is inversed in order to obtain the
watermarked frame. On the detector side, the energy
difference \(D\) is computed again and the embedded bit is
determined according to the sign of the difference \(D\). In
[6], Hartung and Girod proposed a watermarking scheme
aiming at MPEG-2 video, but it can be easily improved to
be used on MPEG-4 video stream. The watermark signal
is first transformed by \(8\times8\) DCT, and then embedded
into the DCT coefficients of the host video signal. To
enhance the length and redundancy of the watermark
signal, the technique of spread spectrum is used.
Watermark bits embedded in an I-frame may injure the
visual quality of the following P-frames, so a
drift-compensation signal is added into the P-frames to
obtain good visual quality of the watermarked video. The
scheme embeds watermark bit into a VLC only if it won’t
increase the length of the VLC. By this way, the bit-rate
of the video is not increased after watermark embedding.
The whole video signal is viewed as a linear signal which
consists of samples of frames within the video. The
watermark signal is denoted as:

\[
a(i) \in \{1,-1\}, i \in N.
\]

This sequence is spread by a chip-rate \(Cr\) according
to the following equation:

\[
h(i) = a(j)j \times Cr \leq i < (j+1) \times Cr; i \in N.
\]
The obtained sequence is then amplified locally by an adjustable factor \( \lambda(i), i \in N \) and modulated by a pseudo-random binary sequence \( p(i) \in \{1,-1\}, i \in N \). Finally, the spread spectrum watermark is added to the line scanned video signal; which gives the watermarked video signal. The overall embedding process is consequently described by the following equation:

\[
v'(i) = v(i) + v(i) = v(i) + \lambda(i)p(i), i \in N. \tag{3}
\]

The adjustable factor \( \lambda(i) \) may be tuned according to local properties of the video signal, e.g. spatial and temporal masking of the HVS, or kept constant depending on the targeted application. On the detector side, recovery is easily accomplished with a simple correlation. The watermarked video signal is multiplied by the pseudorandom noise \( p(i) \in \{-1,1\} \) used for embedding and summed over the window for each embedded bit. The correlation sum for the \( j \) th bit is given by the following equation:

\[
s(j) = \sum_{i=j}^{j+C-1} p(i)v'(i) \tag{4}
\]

The hidden bit is then directly given by the sign of correlation sum:

\[
b'(j) = \text{sgn}(s(j)) \tag{5}
\]

Finally, the correlation between the detected watermark and original watermark is calculated. If correlation is greater than a predefined threshold, the watermark is detected correctly. This scheme offers a foundational framework for many other video watermarking schemes.

MV (motion vector) is an important part in the prediction coding of the MPEG standard and there are lots of MVs in compressed video stream. Some watermarking schemes exploiting MV are proposed. The earlier MV watermarking schemes often embed watermark just by modifying MVs lightly. In [7], the scheme proposed by Zhu and Jiang embeds watermark by changing the parity of both x-component and y-component of a MV. Watermarking is only embedded into MVs whose x and y components are both even integer. If watermark bit is 0, x-component is changed to an odd integer; if watermark bit is 1, y-component is changed to an odd integer. At the same time, other three kinds of MVs are modified so that x-component and y-component of a single MV have the same parity. On the detector side, watermarking is extracted only from the MVs whose x-component and y-component have different parity. This scheme is simple, only modifies MV. But modifying MVs will introduce visual error when decoding. In [8], a more complex MV watermarking scheme is proposed by Liu and Liang. This scheme is based on the concept of mapping in essential. First, build a relation set which looks like \{relation a, relation b, relation c, ...\} and a mapping which maps (relation x, watermark bit i) to another relation y. The frame is divided into high texture area and low texture area. The MVs in high texture area are only modified lightly while the MVs in low texture area can be modified greatly. The prediction error is recalculated after modifying the MVs, which brings better visual quality.

2. MPEG-4 Video Compression

As the most advanced video compression standard at present, MPEG-4 is widely used. To take advantage of temporal redundancy, MPEG-4 standard includes three kinds of frames: 1) intra picture frames (I-frames); 2) forward-predicted frames (P-frames); 3) bidirectional-predicted frames (B-frames). I-frames are coded without reference to other frames. P-frame apps motion prediction by referencing an I-frame or P-frame in front of it, motion vector points to the block in the referenced frame. B-frame applies motion prediction, referencing a frame in front of it and (or) a frame behind it. Each of the two referenced frames may be I-frame or P-frame. Macro block (MB) in video stream is represented as a \( 16 \times 16 \) sample area. Each MB contains
six $8 \times 8$ blocks, four for luminance and two for chrominance. A block of I-frame contains simply values of luminance or chrominance of its own. A block of P-frame or B-frame contains the difference between the values of itself and the referenced block. This process is called motion compensation. Each frame is divided into MBs. Coding process of each block includes DCT, quantization, run-level coding and entropy coding in order. The resulting video stream consists of entropy codes, motion vectors and control information about the structure of video and characteristics of coding.

The structure of frame sequence, coding process and decoding process of MPEG-4 video can be described as following figures:

![Figure 1. MPEG-4 frame sequence](image1)

![Figure 2. MPEG-4 encoding process](image2)

![Figure 3. MPEG-4 decoding process](image3)

3. Temporal Synchronization

A video can be seen as a sequence of single frames, which have temporal redundancy. So temporal synchronization is required for a robust video watermarking scheme. The fundamental idea is that watermark should be adaptively embedded according to the temporal property of the frame it embedded into. If the same watermark is embedded into all frames, it is very easy to remove the watermark by frame averaging.

An approach for synchronization is the construction of templates in the watermarked signal. A template is a pattern which describes the coordinates of the embedded watermark. The detector use templates to search for synchronization. There are mainly two kinds of template methods. The first method is constructing template signal as an auxiliary signal into the watermarked signal [9]. Disadvantage is that the template signal may increase the distortion. The second method is to apply constrains on the structure of the watermark to generate the synchronization pattern [10]. Disadvantage is that the resulting structure of watermark may reduce the capacity and security of the watermark.

It is pointed out that the better approach for synchronization is to utilize features of the original video signal [11]. This viewpoint is assimilated by the scheme proposed in this paper which uses scene features to obtain synchronization.

4. Watermarking Scheme

This paper proposes a compressed MPEG-4 video watermarking scheme with temporal synchronization. The synchronization is based on the conception of scene. A scene is a series of temporal continual and perceptually similar frames. For the purpose of temporal synchronization, same watermark is embedded into frames of a single scene while different watermark are embedded into perceptually different scenes. Each frame in a scene is made up of samples. A sample is denoted by luminance and chrominance. So switch of scenes can be apperceived by tracking the changes of both luminance and chrominance of frames. A frame contains luminance blocks and chrominance blocks. The energy of a block is
focused on the DC DCT coefficient. And the high frequency AC coefficients are almost 0 or close to 0. Therefore DC coefficients of all luminance and chrominance blocks within a frame reflect the main feature of the frame. The detailed description is as following:

Suppose that each frame contains \(N_i\) luminance blocks and \(N_c\) chrominance blocks. Assume \(F_i\) is a frame, \(Lum_i\) is the aggregate of all luminance DC coefficients of \(F_i\):

\[
Lum_i = \{d_{c,n} | 1 \leq n \leq N_i\}
\]

\(Chr_i\) is the aggregate of all chrominance DC coefficients of \(F_i\):

\[
Chr_i = \{c_{dc,n} | 1 \leq n \leq N_c\}
\]

Suppose \(F_j\) is another frame different from \(F_i\), \(D\) denotes the difference between \(F_i\) and \(F_j\):

\[
D = w_1D_1 + w_2D_2
\]

\[
D_1 = \sum_{n=1}^{N_i}(d_{c,n} - d_{c,j,n})^2
\]

\[
D_2 = \sum_{n=1}^{N_c}(c_{dc,n} - c_{dc,j,n})^2
\]

\(w_1\) and \(w_2\) is weight factor of luminance and chrominance respectively.

Each scene is defined dynamically in the watermark embedding process, which means that one or more frames are added into a scene one by one. Every time a new frame added into a scene, the feature of this scene is recalculated. The updated feature is used to decide whether the next frame belong to this scene or a new scene.

Suppose \(S\) is a scene and until now there have been \(m\) frames belonging to it, that is:

\[
S = \{F_1,F_2,...F_m\}
\]

So at present, feature of \(S\) is denoted as \(E_s\):

\[
E_s = \{ldc_{s,1},...,ldc_{s,N_i},cdc_{s,1},...,cdc_{s,N_c}\}
\]

\(ldc_{s,j}\) is the average value of luminance DC coefficients representing \(block(j)\) of each of the \(m\) frames within scene \(S\), \(cdc_{s,j}\) is the average value of chrominance DC coefficients representing \(block(j)\) of each of the \(m\) frames within scene \(S\). So a feature can be considered as an integrative reflection of the scene.

Suppose the next frame is \(F\). Calculate difference \(D_f\) between \(F\) and \(E_s\). If \(D_f\) is not greater than predefined threshold \(T\), frame \(F\) is added into scene \(S\). In this case, \(S\) is updated as:

\[
S = S \cup \{F\} = \{F_1,F_2,...,F_m\} \cup \{F\}
\]

Otherwise, if \(D_f\) is greater than threshold \(T\), frame \(F\) is considered as the first frame of the next scene.

As can be seen from above, a crucial thing is that a mapping relation between different scene features and a series of different watermarks should be built. When watermark detecting, this mapping relation is used to decide which watermark a scene should contain.

Two or more scenes may be embedded with same watermark if they are very similar. Using same watermark for similar scenes is a natural notion of resisting temporal desynchronization. For each scene \(S\) starting from the third scene (note that the first and second scenes are different of course), compare the first frame \(F\) of this scene with the features of all previous scenes(not including the directly preceding scene). If difference between \(F\) and feature \(E'\) of certain previous scene \(S'\) is less than threshold \(T\), the same watermark as scene \(S'\) is embedded into all frames in scene \(S\) beginning with \(F\). Otherwise a watermark never used before should be created and embedded into all frames in scene \(S\).
4.1 Watermark Constructing

Basic component of a watermark consists of a bit string $W, W(i) = \{1, -1\}, i \in N$. Each bit is 1 or -1. It is strongly suggested that the watermark string $W$ is obtained by encrypted the copyright information issued by authentic third-part institution. The copyright information contains time stamp. This time stamp is used to resist multi-watermark attack in which the attacker embeds his own watermark into the watermarked signal. When a new watermark is needed for a certain scene, it is constructed by encrypting the basic watermark string $W$ with a new secret key $K_i$.

$$W_e = Encrypt(W, K_i)$$  \hspace{1cm} (14)

Each watermark is created using an unique secret key. Therefore, the mapping from scene features to watermarks equal to mapping from scene features to secret keys.

Then watermark $W_e$ is spread-spectrum modulated, using a similar method as [6]. An extended m-sequence $M$ is used in modulation [12].

4.2 Watermark Embedding

Watermark data is embedded into some of luminance DCT coefficients and can be detected and extracted from decompressed video. Note that if watermark is embedded into MV or VLC, watermark detection can not be implemented on the decompressed video where the MV and VLC no longer exist.

A series of scenes are identified according to the analysis of scene features. Each unique scene feature is associated with a corresponding watermark.

For the importance of DC coefficients in this scheme, watermark is not added in them. According to the frequency characteristic of DCT coefficients, AC coefficients in high frequency are almost 0. Therefore the watermark is embedded into middle frequency AC coefficients of luminance blocks.

To control the bit-rate, watermark embedding must abide to the constraint that the size of the whole video should not increase after watermarking. The control is conducted based on blocks. Once the size of a block being watermarked reaches its original size, no more bits will be added into this block.

To enhance the imperceptibility of watermark, brightness visual masking is applied when watermark embedding, taking into account the characteristic of HVS(Human Visual System)[13]. Biggish watermark energy is embedded into the biggish luminance coefficients and vice versa.

$$v_w(i) = v(i)(1 + \alpha \cdot w(i)) \hspace{1cm} (15)$$

When watermark signal is embedded into an I-frame or P-frame, it may injure visual quality of frames that use this I-frame or P-frame for motion compensation. In this scheme, a drift compensation mechanism proposed by [6] is employed. The drift compensator keeps track of the energy difference between samples of original reference frame and samples of watermarked reference frame in watermark embedding and counteracts this energy difference in motion compensation.

4.3 Watermark Detecting

In watermark detecting, all scenes are recognized by the same method used in watermark embedding. For each scene, watermarks are extracted from all frames in it by demodulating with the extended m-sequence $M$ used in modulation. The detected watermark $S$ is calculated by averaging the resulting watermarks. Then the scene feature $E$ is calculated and compared with the features which have been calculated and reserved in watermark embedding, to find out the most similar one. Assume $E'$ is the most similar scene feature with $E$ and $S'$ is the corresponding watermark of $E'$. Compare $S$ and $S'$ to calculate the ratio of correctly detected watermark bits to all watermark bits. Watermark is detected correctly if the correct ratio is greater than a certain threshold $Td$. 
5. Experimental Results

Experiments used three video sequences Foreman(FR), Akiyo(AK) and Coastguard(CG). Each of these three video sequences contains 300 frames. Figure 4 shows the result of watermarking.

Two main kinds of temporal synchronization attacks are frame deleting and frame inserting, which are implemented to test the temporal synchronization of this watermarking scheme. Experimental results in table 1 and table 2 show that the scheme has temporal synchronization. The theoretic explanation is that this scheme takes into account not only the single frames but also features of scene containing a number of frames, so deleting or inserting few frames does not affect the detecting result greatly.

![Figure 4. Watermark embedding](image)

Table 1. Relation of frame deleting and watermark accuracy

<table>
<thead>
<tr>
<th>Delete frame ratio</th>
<th>0%</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR watermark accuracy</td>
<td>98.33%</td>
<td>98.28%</td>
<td>98.26%</td>
<td>98.50%</td>
</tr>
<tr>
<td>AK watermark accuracy</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>CG watermark accuracy</td>
<td>93.23%</td>
<td>92.88%</td>
<td>92.48%</td>
<td>92.33%</td>
</tr>
</tbody>
</table>

Table 2. Relation of frame inserting and watermark accuracy

<table>
<thead>
<tr>
<th>Insert frame ratio</th>
<th>0%</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR watermark accuracy</td>
<td>98.33%</td>
<td>98.38%</td>
<td>98.39%</td>
<td>98.22%</td>
</tr>
<tr>
<td>AK watermark accuracy</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>CG watermark accuracy</td>
<td>93.23%</td>
<td>93.55%</td>
<td>93.85%</td>
<td>93.83%</td>
</tr>
</tbody>
</table>

6. Conclusion

Because of temporal redundancy in video, temporal synchronization is important for video watermarking. The best way to achieve temporal synchronization is to utilize features of the original video. This paper proposed a MPEG-4 compressed video watermarking scheme, which employs the DC coefficients of both luminance and chrominance blocks to figure the feature of each scene. Different watermarks are embedded into scenes with different features. Watermark is embedded in portions of luminance DCT coefficients. Spread-spectrum is exploited to increase the redundancy of watermark. In the process of watermark embedding, bit rate control, visual masking and drift compensation are all taken into account to improve the performance. Experimental results show that the scheme is resistant to attacks of frame deleting attack and frame inserting attack.
7. Acknowledgement

The research of this paper is supported by the Tianjin Natural Science Foundation of China under Grant No. 07JCZDJC06600 and the Nankai University Innovation Foundation.

Reference