ADAPTIVE REAL-TIME WATERMARKING USING BLOCK CLASSIFICATION FOR H.264 COMPRESSED DOMAIN

Yin Zhang, Zengxiang Lu and Haiming Lu
Tsinghua National Laboratory for Information Science and Technology
Research Institute of Information Technology, Tsinghua University, Beijing 100084, China

Keywords: H.264 compressed video, adaptive watermarking, block classification, real-time, drift compensation.

Abstract: Focusing on the problem that watermark can cause visible image distortions in some plain areas, an adaptive watermarking algorithm for H.264 is proposed. To embed watermark quickly, we directly operate in the DCT domain. For high imperceptibility, we classify the blocks based on Human Visual System (HVS). However, most current classification methods are not suitable for H.264 compressed domain because the DCT coefficients residue cannot reflect the texture activity accurately due to the employment of intra-prediction. A new block classification method is applied, in which we make restriction during the encoding process so that the 4×4 blocks can be classified into plain, edge and texture blocks according to the intra-prediction mode and quantized integer DCT coefficients in compressed domain. It is effective in block classification and realizes good adaptive performance for watermarking. Drift compensation is also accomplished in our watermarking algorithm. The experimental results demonstrate our watermarking method can achieve both large capacity and good image imperceptibility. Additionally, the method is simple and appropriate for real-time applications.

1 INTRODUCTION

Because of its high compression efficiency, H.264, the next generation video coding standard (Richardson, 2003), is expected to replace MPEG-2 and H.263 in many current applications. Therefore, copyright protection and authentication of this kind of video become an important issue. Digital watermarking has been proposed to address these needs. At present, there are many watermarking schemes for current popular standards such as MPEG-1 and MPEG-2, but only a few for H.264. Besides, as many new features are introduced to H.264, a large number of previous video watermarking algorithms cannot be applied directly, so development of new algorithms is required to fit this new standard.

Video watermarking algorithms can be divided according to the embedding domain, such as spatial domain and compressed domain. If we choose spatial domain, we should first decode the video and then re-encode it. It is impractical as it takes a lot of time. Thus, designing low complexity video watermarking algorithms to embed the watermark in the H.264 compressed domain is attractive.

Noorkami and Mersereau proposed a compressed-domain video watermarking method for H.264 (Noorkami and Mersereau, 2005). The watermark is embedded in the quantized AC coefficients of I frames. The embedding locations within a macroblock are determined by a key extracted from the macroblock itself. In (Sakazawa et al., 2006), the authors directly embedded a watermark into a bitstream generated by an H.264 video encoder and proposed a compensation method to solve drift problem. These embedding methods are compressed-domain based and have low complexity. However, they do not adapt to local image characteristics, so the watermark capacity is limited.

A more image-dependent scheme could provide high capacity as well as imperceptibility. In (Noorkami and Mersereau, 2007), a framework for robust watermarking of H.264-encoded video is presented. The authors employ Watson's human visual model to increase the payload and add robustness while limiting visual distortion. But the watermarking algorithm embeds the watermark in the H.264 encoder. The perceptual model finds the location of coefficients using the original video frame. So it cannot be performed in real time.

Zhang Y., Lu Z. and Lu H. (2008).
ADAPTIVE REAL-TIME WATERMARKING USING BLOCK CLASSIFICATION FOR H.264 COMPRESSED DOMAIN.
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 308-313
DOI: 10.5220/0001935703080313
Copyright © SciTePress

Another kind of adaptive watermarking schemes exploits the HVS characteristics to classify the image into several classes based on several parameters (texture, edge, luminance, chroma, etc.). As human sensitivity to error is different in different classes, different watermark strength is embedded. So the classification rule is the key technique in adaptive watermarking algorithms.

Some block classification methods in DCT compressed domain have been proposed. The AC energy of the DCT coefficients of a block is a common measure of the local block texture activity. By calculating the energy of AC coefficients, we can determine whether the block is smooth or not. In (Tong and Venetsanopoulos, 1998), a more complicated model for JPEG application based on block classification and texture masking is presented. However, these methods cannot be applied directly to H.264 video. With the employment of intra-prediction, the DCT coefficients residue cannot reflect the texture activity accurately.

The goal of this paper is to present an adaptive watermarking algorithm for H.264 compressed domain. If a compressed video bitstream is to be watermarked, it has to be decoded to some extent. Our algorithm is designed to operate directly in the DCT domain to make it suitable for real-time application. Then how to achieve block classification according to H.264 quantized DCT residues becomes the first task.

The paper is organized as follows. In Section 2, an improved block classification algorithm is introduced. In Section 3, we propose a simple watermarking method based on block classification. Experimental results are provided in Section 4 to show the performance of the proposed algorithm. Finally, conclusions are drawn in Section 5.

2 PROPOSED BLOCK CLASSIFICATION METHOD

In our prior work (Zhang et al., 2008), we have noticed the limitation of previous block classification methods encountered in H.264 and proposed a new method. Here we extend this method and further improve classification.

Our target is to get better block classification result from the DCT coefficients in H.264 bitstream. Because of the employment of intra-prediction, the DCT coefficients in the H.264 bitstream is in fact the prediction residue signal, which cannot reflect the texture activity accurately. However, for DC prediction mode, all samples of the current 4×4 block are predicted by the mean of the adjacent samples, and this will not affect the distribution of AC coefficients. The DCT coefficients residue is still available for the classification rules. So we propose a prediction mode restriction method to overcome this drawback.

The experiments shows that the texture of a block is uniform when its variance is very small, and the encoder will most probably choose mode 0~2 as the best prediction mode for blocks with a uniform texture; on the contrary, mode 3~8 will be selected as the best when the block has a complex texture (Wang et al., 2007). Based on this result, we give our approach as follows.

1) Restriction during the encoding process

a. Transform every 4×4 block into integer DCT domain

b. Calculate AC coefficients energy E_{AC}

$$E_{AC} = \sum_{i=0}^{3}\sum_{j=0}^{3} X_{i,j}^2 - X_{0,0}^2 \quad (1)$$

c. Set the prediction mode

$$\begin{cases} \text{determine the best mode using RDO} & \text{if } E_{AC} > T1 \\ \text{use DC mode only} & \text{if } E_{AC} \le T1 \end{cases}$$

If the energy value is larger than the threshold $T1$, the encoder determines the best mode using RDO (Rate Distortion Optimization) as usual. If the energy value is small, which means the predicted blocks with different prediction modes are almost the same, we then force the encoder to use DC mode. This restriction will neither degrade the picture quality nor increase the bit rate markedly like restriction of DC mode only, and is convenient for block classification in the next step.

2) *Block classification*

a. Obtain the intra prediction mode and the DCT coefficients residue of current block

b. If mode is DC, calculate AC coefficients residue energy E_R

c. The block is classified by

$$\begin{cases} \text{non-smooth area} & \text{if mode} \ne \text{DC or} \\ & \quad (\text{mode=DC and } E_R > T2) \\ \text{smooth area} & \text{if mode=DC and } E_R \le T2 \end{cases}$$

where $T2$ is another threshold and usually much smaller than $T1$.

For H.264 video encoded with our restriction, we can detect the block type directly in compressed domain. If the prediction mode is not DC mode, obviously the preceding AC energy value is large; we consider it as non-smooth block. If the prediction mode is DC mode, we cannot conclude whether the preceding AC energy value is large or not, and then

we calculate the AC residues energy for further classification because the distribution of AC coefficients are not destroyed in DC mode.

3) *Improved block classification*

In this way we can find non-smooth area with prediction mode and the AC coefficients energy calculation. With a little change of the method above, we can differentiate between the edge and texture blocks. We discover that directional prediction fits the edge areas nicely but for texture blocks prediction blocks are not so similar. That's to say, texture blocks will have a high AC energy even after intra prediction. So step b) and c) can be modified as:

b. Calculate AC coefficients residue energy E_R for all blocks

c. The block is classified by

$$\begin{cases} if\ E_R > T3 & \text{texture;} \\ else\ if\ \text{mode=DC} & \text{flat;} \\ else & \text{edge.} \end{cases}$$

where $T3$ is another threshold and maybe not equal to $T2$.

With only AC energy we cannot tell texture from edge. But here we utilize the information provided by H.264 intra coding, i.e. the encoder typically selects the prediction mode for each block that minimizes the residues. So after intra prediction edge blocks probably have lower AC residues energy. Then if E_R is large, no matter whether the prediction mode is DC or not, it is identified as a texture block; otherwise we should consider the prediction mode.

3 H.264 WATERMARKING ALGORITHM

When H.264 bitstream is to be watermarked, it is required to be decoded backwards to some extent. Considering the real-time requirement of video watermarking, the processes of the watermarking embedding and detecting are directly performed in the DCT domain to avoid some computationally expensive operations (i.e. DCT, inverse DCT). Thus only inverse entropy coding and run-level decoding must be executed. A block diagram of our method is illustrated in Fig. 1.

During the watermarking process, first, H.264 compressed data are analyzed to distinguish a picture frame, and the blocks in I-frames are extracted from the bitstream by entropy decoding (CALVC or CABAC). Then the masking matrix is generated based on block classification, and the DCT coefficients in the selected blocks are modified according to the embedding information. After the modification, DCT coefficients are entropy encoded to form an H.264 compliant bitstream. The details of embedding process are described as follows.

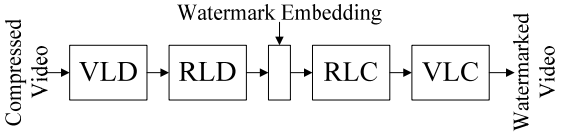


Figure 1: The block diagram of the proposed watermarking embedding procedure.

3.1 Watermark Embedding

At first, the binary sequence is mapped into a bipolar sequence and then modulated by the Spread-Spectrum technology to obtain the watermark sequence for embedding, denoted by Ws.

We use the masking matrix to choose the blocks that are suitable for embedding watermark. The masking matrix is obtained by block classification. If the *(i,j)* block is classified as a smooth area, then M_{ij}=0; otherwise M_{ij}=1, where M_{ij} denotes the masking value of the *(i,j)* block. Only M_{ij}=1 is a candidate for embedding the watermark.

We insert watermark bits by altering the quantized DC coefficients of luminance blocks within I-frames. We choose DC coefficient because it is more stable than ACs; moreover, as altering DC do not change AC energy, the block classification will not be affected. The *X(0,0)* is replaced by the watermarked coefficient *X*(0,0)*,

$$if\ Ws_i = 0, X_i^* = \begin{cases} X_i + wm_i & if\ X_i \bmod 2 = 1 \\ X_i & if\ X_i \bmod 2 = 0 \end{cases}$$

$$if\ Ws_i = 1, X_i^* = \begin{cases} X_i & if\ X_i \bmod 2 = 1 \\ X_i + wm_i & if\ X_i \bmod 2 = 0 \end{cases} \quad (2)$$

$$wm_i = wm_{i-1} * (-1), wm_i \in \{-1, 1\}$$

Where X_i and X_i* respectively denote the DC coefficient of original and watermarked image in block Bi, and Bi refer to blocks with M_{ij}=1.

For H.264 low bit-rates video stream, the quantization step size is large, so 1 quantization level will affect the image quality a lot. We only alter DC coefficient by 1 to change the parity.

We notice that the DC coefficients are added and subtracted alternately, but there is only subtracting in (Noorkami and Mersereau, 2005). As the intra-prediction scheme in H.264 may cause the error propagation due to the embedding, our method can make compensation for the drift and will not introduce significant degradation.

3.2 Watermark Detection

Blind watermark extraction is performed after entropy decoding without the original video. The major steps are described as follows.

1. Block classification is used to select all the possible blocks from all 4*4 blocks.
2. The watermark bit is determined by

$$Ws_i^* = \begin{cases} 1 & \text{if } X_i^* \bmod 2 = 1 \\ 0 & \text{if } X_i^* \bmod 2 = 0 \end{cases} \quad (3)$$

4 EXPERIMENTAL RESULTS

In our experiments, the proposed method is implemented in H.264 reference software version JM11.0. The standard video sequence *Foreman* (QCIF, 176×144) is used for simulation. All experiments are running on Core Duo 2 2.2GHz.

4.1 Block Classification

We have tested the proposed encoding restriction method with RD performance (Zhang et al., 2008). The result demonstrates our algorithm is effective in block classification without affecting compression efficiency too much. The block classification result based on the energy of AC coefficients after 4×4 Integer DCT transform without intra-prediction is used as the reference result. The results of our method are compared with it to measure the performance. The experiment shows that our method can make good judgement and our results are almost the same as the reference.

However, those results are only the classification of smooth and non-smooth areas. Results of the further classification are given in Fig. 2, where black represents texture, grey represents edge and white represents flat.

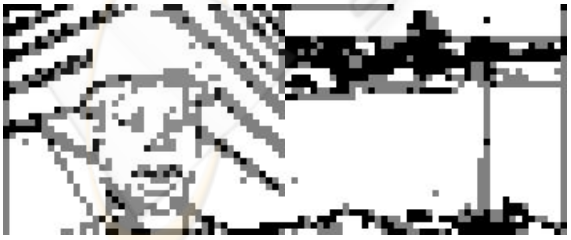


Figure 2: Further block classification result.

Intuitively, it can be seen that the classification result is correct, although may not be very accurate. In fact, accurate division of edge and texture is difficult since they are often mixed together. However, here the classification is appropriate for adaptive watermarking even if it is not accurate. Because in adaptive watermarking, the scaling factor varies according to the classification and usually has a relationship as below:

$$\alpha_{texture} > \alpha_{edge} > \alpha_{flat}$$

By using our method, the texture block has a high AC energy, i.e. the DCT coefficients have a large value, thus the effect of watermark embedding is smaller and we can embed a high strength. That's to say, even though an edge block is classified as texture by mistake and uses a large scaling factor accordingly, the watermark embedding will not affect the perceptual quality as the block has large coefficients.

The thresholds are determined according to real applications. Concerning adaptive watermarking, the reference block classification is performed in advance and a threshold T is chosen to fulfil the need for watermark capacity. The optimum $T1$ should be identical with T used for reference classification, in order that not only good judgement is made but also the number of non-plain blocks classified is larger than the watermark capacity. $T2$ can be set to be 5 based on our experiments, and $T3$ should be larger than $T2$ to find stronger texture, e.g. 10 in our experiments.

4.2 Watermarking

Our watermarking algorithm embeds the watermark in JM11.0 decoder, according to block classification result. As the algorithm used here is based on parity, the scaling factor is of no effect, so we only classify each 4×4 block into plain and non-plain blocks.

4.2.1 Image quality

First, the effect of watermark is evaluated both subjectively and objectively. Fig. 3 shows one frame of this sequence with and without the watermark, along with PSNR (dB). As shown in Fig. 3c, the watermark is invisible after embedding, and the quality of watermarked video is as well as the original video even though PSNR is not high. For comparison, Fig. 3d gives the result for non-adaptive watermarking, in which we embed 1 bit watermark every 3 blocks. Although the watermark bits are fewer, we can see visible artifacts in the "hat" area.

(a) Original (b) Embedding position

(c) Adaptive watermarked (627 bits, 29.77dB) (d) Non-adaptive watermarked (528 bits, 30.15dB)

Figure 3: Adaptive and non-adaptive watermarking of *Foreman*.

We notice that the watermarked frame in Fig. 3c is of better quality but with a lower PSNR. The PSNR metric does not take into account image properties such as flat and textured regions. According to our method, the watermark is embedded into textured regions and into edges, so the PSNR is inadequate to measure image quality in this case. In other examples, we only compare the perceived visual quality.

(a) Original (b) Embedding position

(c) Adaptive watermarked (548 bits) (d) Non-adaptive watermarked (528 bits)

Figure 4: Adaptive and non-adaptive watermarking of *Container*.

(a) Original (b) Embedding position

(c) Adaptive watermarked (615 bits) (d) Non-adaptive watermarked (528 bits)

Figure 5: Adaptive and non-adaptive watermarking of *Carphone*.

Watermarking of another two standard video sequences is also simulated. Fig. 4 and 5 shows the original frames, the block classification results, the watermarked frames with adaptive and non-adaptive watermarking. In the experiments, no visible artifacts can be observed for adaptive watermarking, but for non-adaptive watermarking, we can tell distortions both in the "sea" and in the "ceiling".

4.2.2 Capacity

Watermark capacity is compared to some other compression-domain methods (Noorkami and Mersereau, 2005; Sakazawa et al., 2006). The latter embeds 16 bits per frame, and the number varies from 22 to 81 in the former method. However, our method embeds several hundred watermark bits in one I frame, with good imperceptibility. We can change $T1$ and $T2$ in block classification method and accordingly adjust the total number of watermark bits.

4.2.3 Computational Complexity

Most previous embedding algorithm is integrated in H.264 encoder. That's to say, for a compressed H.264 bitstream, these methods should first decode the video sequence, then insert the watermark during encoding process. Obviously this make the computation complicated. The time cost is at least the sum of video decoding time and re-encoding time. However, the proposed method needs only entropy decoding and DCT coefficient alteration, so performs much faster.

The processing time is recorded in Table 1. In the simulation, *Foreman* (176×144, comprised of 50 I-frames) is tested as experimental samples. For simplicity, we only give the processing time of JM encoder and decoder.

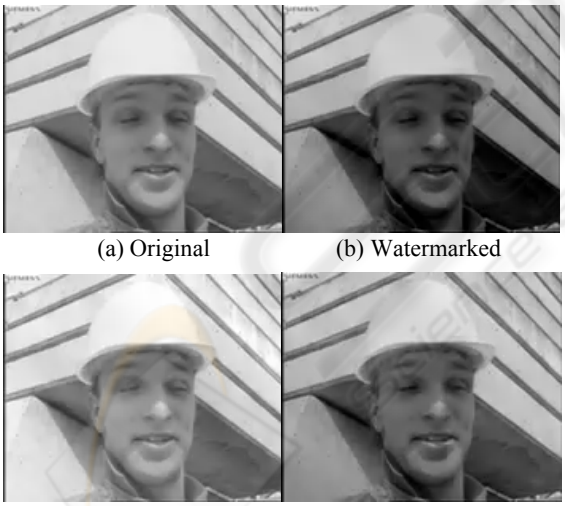
Table 1: Comparison of processing time.

	Proposed	JM decoder	JM encoder
Time(s)	0.562	0.515	7.234

It is observed that the proposed method only consumes a little more CPU time than JM decoder, thus can satisfy the real-time requirement.

4.2.4 Drift Compensation

Our embedding strategy brings a drift compensation in order to weaken the error propagation. The results show that there is almost no degradation. On the other hand, in the case of no compensation, the drift accumulation incurs a visible degradation especially for a large capacity. First let wm_i always be -1 in Equation 2 (Noorkami and Mersereau, 2005), the image turns darker and darker in the right bottom corner, as shown in Fig. 6b. In another comparison, add 1 to DC coefficient when Ws_i=1 and subtract 1 from it when Ws_i=0. The image gets either brighter or darker according to different random sequences, as shown in Fig. 6c and 6d.



(a) Original (b) Watermarked

(c) Watermarked (d) Watermarked

Figure 6: Examples of no compensation.

5 CONCLUSIONS

In this paper, we improved the block classification algorithm by restricting intra prediction mode in H.264 video coding. Our experiments verify that the proposed method can attain accurate block classification result based on DCT coefficients residue without sacrificing compression ratio too much. It is further used in an adaptive real-time watermarking scheme. The algorithm is fast and appropriate for real-time applications, also accomplishes the compensation for error propagation. Simulation results show that the proposed method has a large capacity and preserves the perceptual quality of the video.

However, the proposed watermark embedding and detection procedures are very simple. It is robust against modifications in the H.264 bitstream domain, but cannot resist against signal processing attacks. Because any operation including re-encoding may change the prediction modes and subsequently the residuals in the blocks, the watermark will be destroyed. In future work, a new algorithm will be designed to improve robustness, especially against re-encoding attacks.

REFERENCES

Richardson, I.E.G., 2003. *H.264 and MPEG-4 video compression*. John Wiley & Sons Ltd., West Sussex, England.

Noorkami, M. and Mersereau, R.M., 2005. Compressed-domain video watermarking for H.264. In *Proc. of International Conference on Image Processing (ICIP'05)*, pp. 890~893.

Sakazawa, S., Takishima, Y., and Nakajima, Y., 2006. H.264 native video watermarking method. In *IEEE International Symposium on Circuits and Systems (ISCAS 2006)*, pp. 1439-1442.

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

Tong, H.H.Y. and Venetsanopoulos, A.N., 1998. A perceptual model for JPEG applications based on block classification, texture masking, and luminance masking. In *Proc. of International Conference on Image Processing (ICIP'98)*, pp. 428-432.

Zhang, Y., Lu, Z.X., and Lu, H.M., 2008. A novel block classification method for H.264 compressed video. To appear in *Proc. of 16th International Conference on Computer Communications and Networks (ICCCN'08)*.

Wang, Z.N., Yang, J., Peng, Q., Ma, Z., and Zhu, C.Q., 2007. A fast transform domain based algorithm for H.264/AVC intra prediction. In *IEEE International Conference on Multimedia and Expo 2007 (ICME'07)*, pp. 1563-1566.