COMPRESSED-DOMAIN VIDEO WATERMARKING FOR H.264

Maneli Noorkami, Russell M. Mersereau

Center for Signal and Image Processing
Georgia Institute of Technology
Atlanta, GA 30332-0250, USA
Email: {maneli,rmm}@ece.gatech.edu

ABSTRACT

This paper presents a novel, low complexity watermarking algorithm for H.264 that exploits the specific features of this new standard. The security of the algorithm is based on the randomness of the watermark location. The coefficient with the embedded watermark within the macroblock is determined by a public key extracted from the macroblock and a secret key possessed by the copyright owner. It is proposed to use the relative change of the DC coefficients of the 4×4 blocks in a macroblock to generate the public key. The watermark is embedded in the quantized AC coefficients of I frames, and the compressed video bit rate is only increased 0.5% on average. Watermark detection can be done without the original video signal, and the algorithm is appropriate for real-time applications. Simulation results show that watermark embedding preserves the perceptual quality of the video. However, the degradation induced by an adversary's attempt to remove the watermark creates visible artifacts.

1. INTRODUCTION

The advent of digital television, the appearance of digital versatile disks (DVD), and the transfer of video files over the internet demonstrate that digital video is becoming an important part of the broadcasting, entertainment, and communication industries. Although network bit rates are increasing and the storage capacity of hard disks and flash memory is greater than ever, it is still not possible to transmit or store raw digital video. Thus, video compression has been introduced as a means for efficient transmission and storage of video signals. Different video coding standards, such as H.261, H.262(MPEG-2) and H.263 have been developed over the last ten years. MPEG-2 has had great success, but anticipation for a new standard that exploits recent advances in efficient compression has led to the development of the MPEG-4 Visual and H.264 standards. A detailed description of the H.264 standard is given in [1],[2].

This research is partially supported by Texas Instruments.

Because of the high compression efficiency of H.264, it is expected to replace MPEG-2 and H.263 in their current applications. The high performance of H.264 is likely to increase the use and distribution of compressed video signals substantially. Thus, issues of copyright protection and authentication that are appropriate for this standard become very important. One method of copyright protection is the addition of a watermark to the video signal. The embedded watermark should be imperceptible to a human observer, but reliably detectable by a watermark detector.

In most video applications, watermark detection should be performed in real time. Detecting the watermark in an image can take as long as a few seconds; this delay is unacceptable for video displayed at a rate of 25 frames per second. Because of the large size of raw video files, video signals are usually stored and distributed in a compressed format. To embed the watermark in video, it is impractical to first decode the video sequence, embed the watermark in it, and then re-encode it. Thus, designing low complexity video watermarking algorithms that embed the watermark in the compressed domain is attractive. Previous work on video watermarking in the compressed domain focused on embedding the watermark into the MPEG-2 bit-stream sequence [3]-[6]. The residual blocks in the MPEG-2 standard are compressed using the DCT transform, quantization, reordering, run-level-coding and variable length coding. The watermark is embedded in the unquantized DCT coefficients in [3],[4], and in the quantized DCT coefficients in [5]. The algorithm presented in [6] embeds the watermark in the variable length codeword (VLC) domain.

This paper presents a low complexity watermarking algorithm for H.264 that exploits the specific features of this new standard. The watermark is embedded in the quantized AC coefficients of I frames. The coefficient with the embedded watermark within a macroblock is determined by a key that is specific to that macroblock. This requires a long key stream sequence. To avoid this problem, the key is generated using some features of the macroblock itself that are visible to everyone, and the copyright owner's secret key.

The paper is organized as follows. Section 2 presents

0-7803-9134-9/05/$20.00

the proposed approach. Section 2.1 discusses the procedure of public key generation from macroblocks. Watermark embedding and detection are explained in Section 2.2 and Section 2.3, respectively. Simulation results are given in Section 3. Finally, Section 4 concludes the paper.

2. PROPOSED APPROACH

The proposed H.264 watermarking algorithm works at the macroblock level of I frames. A macroblock contains a 16×16 sample region of a video frame. I frames are chosen for watermark embedding because their existence is crucial for the video signal. Also, P and B frames are highly compressed by motion compensation and there is less room to embed a watermark in them. Macroblocks in an I frame are intra-coded, which means that each 16×16 or 4×4 luma region and each 8×8 chroma region is predicted from samples coded previously in the same slice. Usually the 16×16 luma prediction mode, I16, is chosen for the smooth areas of the frame, and the 4×4 luma prediction mode, I4, is chosen for the detailed areas of the frame. Each 4×4 block of residual data is transformed by an integer transform after intra prediction. If the macroblock is coded in the 16×16 intra prediction mode, the DC coefficients of all 4×4 blocks are transformed by a 4×4 Hadamard transform after the 4×4 integer transform to decorrelate these coefficients further.

The proposed algorithm embeds the watermark in one quantized coefficient of a macroblock. The security of the algorithm is based on the randomness of the selected coefficient for watermark embedding. Embedding the watermark in only one coefficient in a macroblock does not induce visible artifacts. However, the attacker cannot identify which coefficient has been chosen. Therefore, he has to change at least half of the coefficients to make watermark detection impossible. However, changing half of the coefficients results in visible artifacts in the video signal, and renders the video useless.

The selection of the coefficient in the i^{th} macroblock for watermark embedding is under the control of a key. If the same key is used for every frame, the watermarking algorithm is vulnerable to a self-collusion attack. Thus, a very long key stream sequence is required. Transmitting a long key, however, would make the algorithm impractical. This problem is solved by generating the key from a combination of a public key, Kp_i, extracted from some feature of the macroblock and a secret key, Ks, possessed by the copyright owner. The public key is extracted from each macroblock and passed as the plaintext to a cryptographic system with the secret key, Ks. The ciphertext generated by the cryptographic system is the key for that macroblock. Since the security demands of watermarking systems are less than those of cryptographic systems, a fast and simple cryptographic scheme can be used for this purpose. Currently, a shift cipher with modulus 2, key Ks, and plaintext Kp_i is used. Two bits of this key determine the selected 8×8 block in the macroblock, $b8_i$, another two bits determine the selected 4×4 block in the 8×8 block, $b4_i$, and 4 bits determine the selected AC coefficient in that 4×4 block, cw_i, for watermark embedding.

Kp should be extracted from some feature of the macroblock that cannot be changed by the attacker without degrading the perceptual quality of the video. It is proposed to use the relative change of the DC coefficients of the 4×4 blocks in a macroblock to extract the public key. Only 4×4 intra-predicted macroblocks are used for watermarking. The 16×16 intra-predicted macroblocks are not used for two reasons. First, the 16×16 intra prediction mode is used for smooth regions of the frame, and watermark embedding causes visible artifacts there. Second, the extra Hadamard transform for this macroblock decorrelates the DC coefficients even more, and many of these coefficients are zero.

2.1. Extracting the public key

To make the extracted public key robust, a feature of the macroblock should be used to which the human eye is sensitive. One such feature is the DC coefficients of the 4×4 blocks. If the DC coefficient itself is used for public key extraction, the attacker can change the DC coefficient of every block by the same amount, which would make watermark detection impossible. This would result in a darker or brighter frame, but the perceptual quality of each frame would be preserved. However, if the relative difference of the DC coefficients of 4×4 blocks is used to determine the public key, the attacker has to increase or decrease the DC coefficient of one block or more to make the public key extraction impossible for the copyright owner. This results in visible artifacts.

To determine the public key, Kp_i, the quantized DC coefficients of 4×4 blocks of the i^{th} I4 macroblock are extracted, and put in a vector of DC coefficients, DC_i, in a key-scrambled fashion. DC_i can have up to 24 elements, 16 elements from the luma component, Y, and 4 components from each of the chroma components, Cb and Cr. The structure of the public key generation is shown in Figure 1. The j^{th} bit of Kp_i is derived from DC_i as follows:

$$Kp_i(j) = \begin{cases} 0 & DC_i(j) > DC_i(j+1) \\ 1 & DC_i(j) \leq DC_i(j+1) \end{cases} \quad (1)$$

To show the robustness of the extracted public key, one of the DC coefficients in a macroblock is changed to erase one bit of Kp_i. The resulting I frame is shown in Figure 2. It can be observed that changing only one DC coefficient in a macroblock lowers the perceptual quality of the video frame.

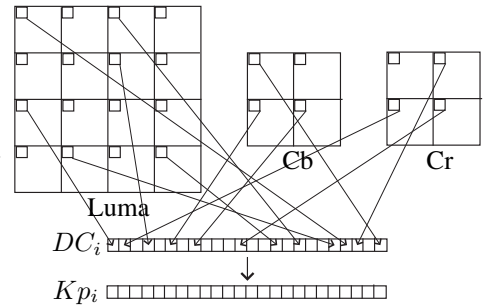
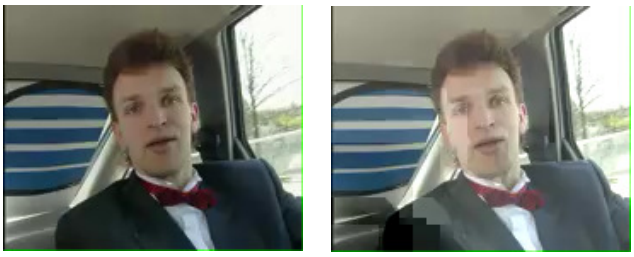


Fig. 1. Structure of public key generation.



(a) Original I frame (b) I frame after attack

Fig. 2. Examples of degradation induced by changing one DC coefficient in a macroblock.

2.2. Watermark embedding

If a compressed video bitstream is to be watermarked, it has to be decoded to some extent. The closer the watermark embedding operation is to the entropy coding level, the less computationally complex and more suitable for real-time application the algorithm becomes. However, the closer the embedding is to the DCT transform operation, the less the degradation induced by watermark embedding becomes. In this paper we propose to embed the watermark in the reordered quantized AC coefficients. Because quantization is a lossy operation, it is desirable to embed the watermark after quantization to avoid possible erasure of the watermark. Furthermore, entropy coding and decoding is a fast procedure, and watermark embedding and detection can be done in real-time.

As discussed in Section 2, several bits of Kp_i are used to select the coefficient, cw_i, in the macroblock for watermark embedding. Kp_i has more bits than required. In future work, the extra key bits will be used to make the algorithm more secure and robust. To embed the watermark W_i in macroblock i, cw_i is modified as follows:

if $W_i = 0$,

$$cw_i = \begin{cases} cw_i - 1 & \text{if } cw_i \bmod 2 = 1 \\ cw_i & \text{if } cw_i \bmod 2 = 0 \end{cases} \quad (2)$$

if $W_i = 1$,

$$cw_i = \begin{cases} cw_i & \text{if } cw_i \bmod 2 = 1 \\ cw_i - 1 & \text{if } cw_i \bmod 2 = 0 \end{cases} \quad (3)$$

The maximum change made to the quantized coefficient selected for watermark embedding is one level. Thus, the modification of the unquantized DCT coefficient is as large as the quantization step size for that block. Consequently, the degradation induced by watermark embedding is proportional to the quantization error.

Entropy coding produces short codewords for frequently occurring values and longer codewords for less frequently occurring values. Generally the values closer to zero occur more frequently. The watermark embedding method moves some values closer to zero and some values further. Thus, the average bit rate remains more or less the same. There is a coded block pattern parameter in H.264 that indicates which blocks within a macroblock contain coded coefficients. This parameter is computed before watermark embedding. Thus, if the watermarking algorithm selects a block of all zero coefficients to embed the watermark, this parameter does not let the change take place, and helps to control the bit rate. When the detector finds that the selected embedded coefficient is in an all zero block, it knows that the watermark has not been embedded in that macroblock.

2.3. Watermark detection

Watermark detection is performed after entropy decoding. The detector uses the DC coefficients of the macroblock with its secret key to find the location of the watermark. The watermark bit in the macroblock is determined as follows:

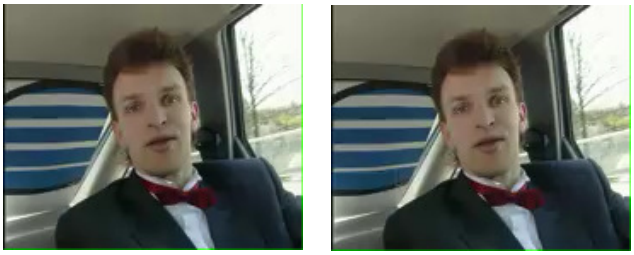
$$\hat{W}_i = \begin{cases} 0 & \text{if } cw_i \bmod 2 = 0 \\ 1 & \text{if } cw_i \bmod 2 = 1 \end{cases} \quad (4)$$

3. SIMULATION RESULTS

The proposed watermarking algorithm was implemented in the H.264 reference software version JM86 downloaded from http://iphome.hhi.de/suehring/tml/. The standard video sequence carphone (QCIF, 176×144) was used for simulation. Figure 3 shows one frame of this sequence coded with the H.264 codec with and without the watermark.

The number of watermark bits embedded in an I frame of the six standard video sequences, the percentage of watermark bits recovered after an H.264 re-encoding attack, and the percentage increase in the video bit rate after watermarking are given in Table 3. On average, the watermarking process increased the size of the compressed video by only 0.485%.

Two examples of a hypothetical adversary's attempt to remove the watermark are shown in Figure 4. One coefficient and half of the coefficients in each 4×4 block were



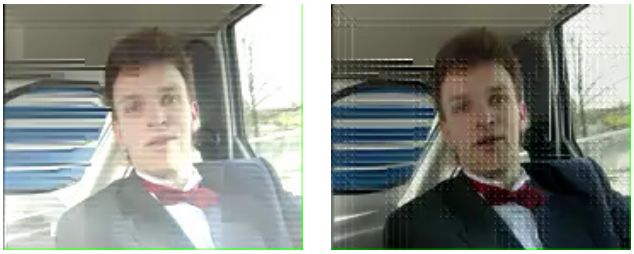
(a) Original H.264 frame (b) Watermarked H.264 frame

Fig. 3. Comparison of the watermarked H.264 frame with the original H.264 frame.

Table 1. Simulation results for six standard videos.

Video Sequence	Watermark Bits	Re-encoding Recovery Rate	Bit rate Increase
carphone	44	58%	0.80%
claire	22	83%	0.44%
mobile	85	85%	0.23%
mother	42	68%	0.69%
table	38	62%	0.31%
tempete	81	83%	0.44%

modified in the frame on the left and right, respectively. While the first attack only erases a few bits of the watermark, the second one erases half of the watermark bits. Both attacks result in a significant decrease in video quality. The proposed algorithm is not robust against signal processing attacks, because any simple signal processing operation, such as filtering, changes the prediction modes and subsequently the residuals in the I macroblocks of H.264. But, the algorithm is robust against modifications in the H.264 bitstream domain.



(a) One coefficient (b) Half of the coefficients

Fig. 4. Examples of adversary's attempt to remove the watermark by modifying the quantized AC coefficients.

4. CONCLUSION

A novel, low complexity watermarking algorithm was presented. The coefficient in a macroblock to be embedded with the watermark is selected by a key. The key is generated from a robust feature extracted from the macroblock and a secret key possessed by the copyright owner. The relative difference of the DC coefficients of 4×4 blocks is used as a robust feature. Using macroblock features for watermark embedding makes the algorithm more robust against self-collusion attacks by embedding the watermark in the same location in similar frames and different locations in dissimilar frames. The algorithm is fast and appropriate for real-time applications. Simulation results show that watermark embedding preserves the perceptual quality of the video.

Currently, the watermark embedding and detection procedures are very simple. In future work, the proposed algorithm will be combined with the algorithm proposed in [7] to also improve robustness against signal processing attacks. The possibility of using the prediction direction of the macroblock and exploiting the chroma components for watermark embedding is under investigation.

5. REFERENCES

[1] I. E. Richardson, *H.264 and MPEG-4 Video Compression*. Wiley, 2004.

[2] ISO/IEC 14496-10 and ITU-T Rec. H.264, *Advanced video coding*, 2003.

[3] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," in *Signal Processing*, vol. 66, pp. 283–301, 1998.

[4] T.-Y. Chung, M.-S Hong, Y.-N Oh, D.-H. Shin and S.-H. Park, "Digital watermarking for copyright protection of MPEG-2 compressed video," *IEEE Transaction on Consumer Electronics*, vol. 44, pp. 895–901, 1998.

[5] D. Simitopoulos, S. A. Tsaftaris, N. V. Boulgouris, and M. G. Strintzis, "Compressed-domain video watermarking of MPEG streams," in *Proceedings 2002 IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 569–572, 2002.

[6] C.-S. Lu, J.-R. Chen, L. H.Y.M, and K.-C. Fan, "Real-time MPEG-2 video watermarking in the VLC domain," in *Proceedings 16th International Conference on Pattern Recognition*, vol. 2, pp. 552–555, 2002.

[7] M. Noorkami, R. M. Mersereau,"A H.264 video watermarking algorithm exploiting a human visual model," *Submitted to International Workshop on Digital Watermarking (IWDW)*, Siena, September 15-17, 2005.