

Robust Digital Image-in-video Watermarking for the Emerging H.264/AVC Standard

Jing Zhang and Anthony T. S. Ho

Center for Information Security
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798

Email: Jingzhang@pmail.ntu.edu.sg; etsho@ntu.edu.sg

Abstract—A novel grayscale watermark pre-processing and a robust video watermarking algorithm for the emerging video coding standard H.264/AVC are proposed in this paper for the copyright protection application. This algorithm can insert grayscale watermark patterns such as detailed trademarks or logos into the low bit-rate H.264/AVC videos in the compressed domain with good robustness and high capacity. The marked video sequences maintain good visual quality and the same overall consuming bit-rate. The proposed algorithm can robustly survive transcoding process and common signal processing, such as bit-rate reduction, Gaussian filtering and contrast enhancement.

I. INTRODUCTION

The progress in multimedia storage and transmission technology has allowed an ever increasing amount of information in digital format for archiving and transmission. However, the ease of copying and reproducing digital data is more likely to encourage the work of pirates and hackers, since it enables perfect copies with no loss in data quality. Digital watermarking represents a valid solution to the above problem, since it is possible to identify the source, author, creator, owner, distributor or authorized consumer of digitized images, video recordings or audio recordings. A digital watermark is an identification code, permanently embedded into digital data, carrying information pertaining to copyright protection.

Digital video watermarking algorithms can be sorted according to the embedding domain, such as the spatial domain, compressed domain, and bitstream domain. Compressed-domain watermarking methods arouse great interests among researchers. Hartung et al. [1] embedded spread-spectrum watermark into the MPEG-2 bitstream with drift compensation, but the capacity is reduced. Arena et al. [2] watermarked the MPEG-2 video using interleaved coding. The differential energy watermark method has been extended to video from still images by watermarking the I-frames of an MPEG stream as in [3]. However, most of the previous works were focusing on MPEG standards for high bit-rate applications rather than low bit-rate videos (≤ 1 Mbit/s).

The H.264/AVC standard is specially designed for low bit-rate applications. It significantly improves the coding efficiency with the help of many enhanced functional features, such as tree-structured motion compensation, Lagrangian Optimization and 4×4 Integer DCT [4]. However, higher

compression ratio leads to the difficulty in balancing among tradeoff requirements for watermarking video data [5], [6].

This paper describes a digital watermarking scheme of H.264/AVC video format, working directly in the compressed domain. Although the idea of compressed-domain watermarking of videos is not new, our work addresses the highly compressed format of the H.264/AVC standard. The scheme makes an accurate usage of the video encoder for watermark embedding, and experimental results exhibit robust results for this low bit-rate format under transcoding and common signal processing. In the digital rights management system, company trademarks and logos are always considered as excellent watermark for copyright protection [5]. Hence, this paper also proposes a novel pre-processing for grayscale watermark patterns which can efficiently compress the 2-D grayscale patterns for embedding. This can also help to solve the low capacity problem for watermarking the H.264/AVC video due to the high compression performance of this state-of-the-art video standard.

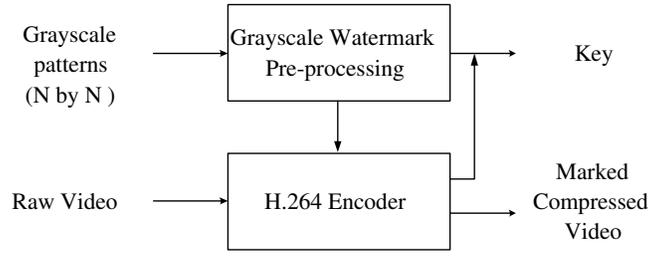
The grayscale watermark pre-processing method is introduced in the next section. The proposed watermarking method based on the emerging H.264/AVC standard will also be described. The simulation results and conclusions will be given in Section III and Section IV, respectively.

II. PROPOSED WATERMARKING ALGORITHM

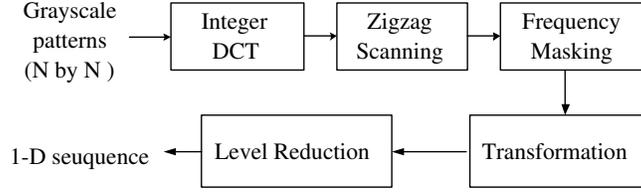
In this paper, 2-D texture grayscale trademarks and logos are first pre-processed and then embedded as the robust watermarks for copyright protection for H.264/AVC video data. Figure 1 shows the watermarking embedding scheme consisting of grayscale watermark pre-processing and H.264/AVC encoder with watermark embedding.

A. Grayscale Watermark Pre-processing

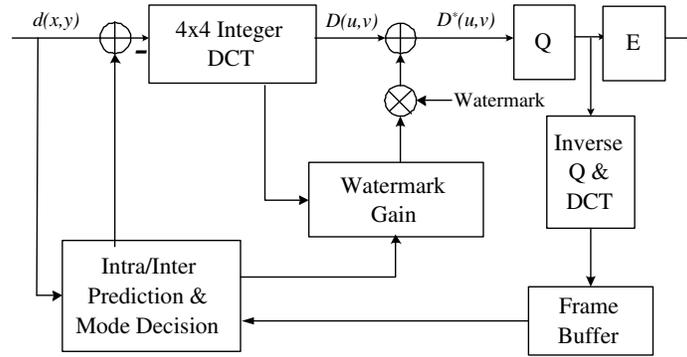
The advancement in video coding technology has led to a significant decrease in redundancy for watermark embedding. For example, in bi-predicted frames of a video clip with low motion at a low coding bit-rate, most coefficients after prediction are zero [6]. Accordingly, the watermark should be pre-processed to decrease the data size for embedding, if complex watermarks in such as 2-D grayscale company trademarks are used. A grayscale watermark pre-processing



(a) Video Watermarking scheme



(b) Grayscale Watermark Pre-processing



(c) H.264 Encoder with Watermark embedding

Fig. 1. Watermark embedding scheme.

is proposed to deal with this problem in 4 primary steps as follows.

1) *4 × 4 Integer DCT*: The grayscale pattern is first decomposed into non-overlapping 4×4 blocks, denoted by $m(x, y)$ ($0 \leq x, y \leq 3$). Each block $m(x, y)$ is transformed by

$$M(u, v) = \text{DCT}\{m(x, y)\} \quad 0 \leq u, v \leq 3 \quad (1)$$

where $\text{DCT}\{\cdot\}$ represents the less complex but more accurate integer DCT with the fixed block-size 4×4 which is uniquely used in H.264/AVC [4]. This transform has many fundamental differences from the traditional DCT, which contribute to a better performance for video coding [4] and digital watermarking [5].

2) *Zigzag Scanning and Frequency Masking*: After zigzag scanning, from the 4×4 transformed block we obtain 16 coefficients $M(j)$, where $j \in [0, 15]$ indicates the position in zigzag scanning. There are several lowest frequencies which

possess the significant energy with sufficient information for the pattern reconstruction. To achieve higher efficiency, a mask is applied to select the top 6 coefficients and discard the other components.

3) *Transformation*: After masking, there is a significant difference between the dynamic range of every two adjacent coefficients. We need to narrow the dynamic ranges of these coefficients. Otherwise this can introduce undesirable distortion during watermarking. Thus, the remaining coefficients $M(j)$ are transformed to reduce the dynamic range as follows:

$$M'(j) = \frac{M(j) + A(j)}{B(j)} \quad 0 \leq j < 15 \quad (2)$$

where $A(j)$ and $B(j)$ represent the corresponding transformation parameters to j , where $j \in [0, 15]$ indicates the position in zigzag scanning.

4) *Level Reduction*: The coefficients $M'(j)$ after transformation have multiple values, $M'(j) \in [1, 7]$. They can be

further simplified to be binary, denoted by $w(j)$ as

$$w(j) \equiv [M'(j)]_{mod2} \quad (3)$$

$$K(j) = M'(j)/2 \quad (4)$$

Watermark patterns can then be reconstructed from the 1-D output sequence w obtained from the pre-processing and the key file K .

After the grayscale watermark pre-processing, a 2-D grayscale watermark pattern with the size $N \times N$ (with $N \times N$ spacial pixels) can be compressed into a binary sequence w with the shortest length $3N^2/8$. This significantly decreases the watermarking payload for every frame by approximately 63%.

B. H.264/AVC Watermarking Method

1) *Watermark Embedding*: In the next stage, the binary sequence w is first mapped into a bipolar sequence as follows:

$$w'(i) = (-1)^{w(i)} \quad i \in N \quad (5)$$

where i and N indicate the position index and the length of the binary sequence w .

The bipolar sequence w' is secondly modulated by the Spread-Spectrum method in [7] to obtain the watermark sequence for embedding, denoted by w^s .

Due to the significant compression efficiency of the H.264/AVC standard, the remaining DCT coefficients of the residual data are perceptually essential. Thus, during the H.264/AVC encoding process, only one middle frequency $D(u, v)$ in the diagonal position in a 4×4 DCT block is substituted by

$$D^*(u, v) = \alpha \cdot \beta_i \cdot w_i^s \quad (6)$$

where u and v indicate the position of the coefficient, w_i^s is the spread-spectrum watermark component, both α and β_i are positive gain factors to maintain a good visual quality of the marked video, and i denotes the position of the block, $i \in R$.

The embedded watermark w_i^s is a bipolar component. Thus, from Equ. 6, the watermark information is embedded into the polarity the frequency component by replacing operation, instead of modulating the amplitude in traditional addition embedding methods:

$$\text{sign}(D^*(u, v)) = w_i^s \quad (7)$$

During video processing and transcoding process, the amplitude of the H.264/AVC compressed video data can be altered in a very wide dynamic range. The algorithm will be highly fragile if watermark is inserted by amplitude modification. Therefore, embedding watermark components into the data polarity can help increase the robustness of the watermarking scheme.

The gain factor α is derived empirically. However, it can be adaptively altered according to the best prediction mode of the current block. The local gain factor β_i is obtained from the DC coefficient $D(0, 0)$ and the AC coefficients $D(u, v)$:

$$\beta_i = \mu \sum_{\substack{1 \leq u \leq 3 \\ 1 \leq v \leq 3}} |D(u, v)| - D(0, 0) \quad (8)$$

where μ is the the weighting factor which balances the DC and AC coefficients.

From this formula, the local watermark gain is determined by the local motion activity from the integer DCT coefficients of the residual error. The DC coefficient represents the luminance of the block, and the sum of the AC coefficient $D(u, v)$ represents the *spatial activity* of the block. The performance tradeoff between the DC and AC coefficients can be adjusted by the weighting factor μ . If the current block is in an area with a significant amount of motion activity, the residual error should also contain motion information. Otherwise, the residual will be zeros or negligible if the block is located in the motionless background. The local gain factor can alter adaptively according to the local motion activity. The human visual system is more sensitive to the distortion in smooth areas than textured areas, motionless areas than areas with motion activity. The local watermark gain can be linked with perceptual masking adaptively with the visual content.

To increase compression efficiency, the H.264/AVC standard supports a tree-structured motion compensation [4] with 7 Inter-prediction modes for each macroblock \mathcal{S} as well as two Intra-prediction modes: 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 and 4×4 for Inter-prediction, $I16 \times 16$ and $I4 \times 4$ for Intra-prediction. A larger partition size is suitable for still areas and a smaller partition size is suitable for areas with detailed motion.

After embedding w_i^s into all the available Inter/Intra modes (denoted by \mathcal{I}), the best mode o^* for the marked macroblock MB^* is selected by minimizing the expression in Eq. (9) within the constrained R and minimized D , using Lagrangian Optimization technique [4] of H.264/AVC:

$$o^* = \arg \min_{o \in \mathcal{I}} \left(D(MB^*, o) + \lambda R(MB^*, o) \right) \quad (9)$$

where λ denotes the predetermined Lagrangian multiplier for mode choice, and D and R represent the distortion and consumed bits for encoding the current mode o , respectively.

2) *Watermark Retrieval*: For extraction, the original embedded watermark components are not needed. At the H.264/AVC Decoder, one watermark coefficient \hat{w}_i^s can be estimated by the sign of the corresponding marked quantized coefficient $\hat{D}_q(u, v)$ in each 4×4 block:

$$\hat{w}_i^s = \text{sign}(\hat{D}_q(u, v)) \quad 0 \leq u, v \leq 3, i \in R \quad (10)$$

The Hamming distance H between the extracted watermark \hat{w}^s and the original spread-spectrum chip sequence is computed. The extracted bit \hat{w}_i can be derived from the Hamming distance with an appropriate threshold τ . $\hat{w}_i = 1$ if $H > \tau$; $\hat{w}_i = 0$ if $H < -\tau$.

C. Grayscale Watermark Post-processing

After extracting the sequence \hat{w}_i from the compressed H.264 video, the grayscale watermark is reconstructed by performing an inverse operation of the watermark pre-processing.

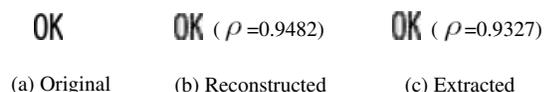


Fig. 2. Samples of The Original, Reconstructed and Extracted Watermarks from the 30th Frame of *Stefan* (768 kbit/s, CIF-size) and Normalized Correlation.

III. RESULTS AND ANALYSIS

In our experiments, the proposed H.264/AVC watermarking algorithm has been integrated into the H.264 JM-7.6 reference software [8]. The standard test video clips include *Foreman*, *Stefan*, *Coastguard*, *Flower Garden*, *Container Ship*, and *Silent*. All the first 100 frames are coded at 15 frames/s at 768 kbits/s, 512 kbits/s and 396 kbits/s, respectively. B-frames are not used in our experiments because there are very few nonzero DCT coefficients due to the efficient compression performance.

A small 16×16 grayscale watermark is used as the watermark pattern for our experiments. Figure 2 shows the reconstructed watermark without watermarking and the retrieved samples from the marked *Stefan* (768 kbit/s, CIF-size). These reconstructed watermark samples are highly correlated to the original pattern. This demonstrates and supports the feasibility of the proposed grayscale watermark pre-processing technique.

Figure 3 shows the frame samples from the unmarked and marked video clips with PSNR(dB). The unmarked and marked video clips are reconstructed from the compressed data without and with watermarks, respectively. The average PSNR values are computed by comparing the reconstructed video frames to the original raw video frames.

On average, the watermark insertion leads to a decrease of approximately 0.97 dB for all video clips coded at 768 kbits/s, 0.96 dB at 512 kbits/s and 0.64 dB at kbits/s, which are all less than 1 dB. In the experiments, no visible artifacts can be observed in all of the test video sequences.

Two categories of video attacks have been applied to the marked video to test the robustness of the watermarking algorithm: transcoding and common signal processing processes.

In the first group, we utilized the bit-rate reduction. During watermarking in the H.264/AVC encoding process, the Lagrangian Optimization technique targets the overall consuming bit-rate at 768 kbits/s, 512 kbits/s and 396 kbits/s. When transcoding is applied after watermarking, the bit-rates have been reduced to approximately 1/2 and even 1/3 of the original bit-rates. During re-encoding with the bit-rate reduction, most high frequency components which represent detailed texture will be discarded, especially when the bit-rates have been decreased to the 1/3. Our robust method can still robustly survive under such strong attacks.

After decoding the marked bitstream, common signal processing attacks are applied to the raw video frame by frame, including 5×5 Gaussian low-pass filtering, circular averaging filtering, unsharpened contrast enhancement, and additive Gaussian noise (mean=0, variance= 0.001).

Figure 4 shows the examples of attacked *Stefan* (768 kbit/s, CIF-size). The visual quality decreases significantly. For example, the PSNR values decrease approximately 6.7dB and 6.3dB when under contrast enhancement and Gaussian noise, respectively.

The robustness is measured by the normalized correlation between the original grayscale watermark and the extracted and results are presented in Table I and Table II.

Table I and Table II show that the proposed watermarking algorithm together with the watermark pre-processing can survive strong attacks, such as bit-rate reduction, contrast enhancement and Gaussian noise. The watermark patterns can tolerate the distortion introduced by some incorrect extracted watermark components, and perform better than binary watermark sequences. Moreover, the fact that the watermark components are embedding into the polarities of the DCT coefficients of the residual data, instead of modifying the amplitude also play an important role in increasing the robustness of the algorithm. Further more, by applying the spread-spectrum to the pre-processed watermark components before embedding repeatedly spreads the watermark components into every macroblock of each frame. This also increases the robustness of our algorithm.

IV. CONCLUSIONS

The proposed grayscale watermark pre-processing with the presented video watermarking algorithm for copyright protection technique performed well in terms of robustness and data capacity. The watermarked H.264/AVC video clips maintained the good visual quality and almost the same Bit-rate. Other more detailed copyright information, such as textured company trademarks or logos can also be used as watermark to further protect ownership and defend against illegal attacks.

REFERENCES

- [1] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283-301, May 1998.
- [2] S. Arena, M. Caramma and R. Lancini, "Digital watermarking applied to MPEG-2 coded video sequences exploiting space and frequency masking," *The IEEE International Conf. on Image Processing*, pp. 1987-1990, Sep. 2000.
- [3] G. Langelaar and R. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Trans. Signal Processing: Image Commun.*, vol. 10, pp. 148-158, 2001.
- [4] T. Wiegang, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560 -576, Jul. 2003.
- [5] J. Zhang and A.T.S. Ho, "An efficient digital image-in-image watermarking algorithm using the integer discrete cosine transform (IntDCT)," in *IEEE Joint Conf. of 4th Int. Conf. Info., Commun. and Signal Processing and 4th Pacific-Rim Conf. Multimedia*, Dec. 2003.
- [6] G. Qiu, P. Marziliano, A.T.S. Ho, D.J. He, and Q.B. Sun, "A hybrid watermarking scheme for H.264/AVC video," in *Proc. the 17th International Conference on Pattern Recognition*, Cambridge, UK, Aug. 2004.
- [7] A.T.S. Ho, and F. Shu, "A Robust Spread-Spectrum Watermarking Method Using Two-Level Quantization," in *IEEE International Conference on Image Processing (ICIP 2004)*, Singapore, Oct. 2004.
- [8] H.264/AVC Joint Model 7.6 (JM-7.6) Reference Software. [Online]. Available: http://iphome.hhi.de/suehring/tml/download/old_jm/.



(a) Unmarked (34.14dB)



(b) Marked (33.58dB)



(c) Unmarked (36.58dB)



(d) Marked (36.00dB)

Fig. 3. The 20th frames of unmarked and marked *Flower* (a) and (b) and *Stefan* (c) and (d) (768 kbit/s, CIF-size).



(a) To 1/3 Bit-rate (31.04dB)



(b) Contrast Enhancement (29.32dB)



(c) Gaussian Noise (29.66dB)



(d) 5x5 Gaussian filter (30.13dB)

Fig. 4. The 20th frames of *Stefan* (768 kbit/s, CIF-size) after attacks

TABLE I

THE ROBUSTNESS AGAINST TRANSCODING ATTACKS, MEASURED BY NORMALIZED CORRELATION.(768, 512 AND 396 KBIT/S, CIF-SIZE)

Video	Original Bit-rate	No Attack	To 1/2 Bit-rate	To 1/3 Bit-rate
Fore-man	768	0.93	0.70	0.50
	512	0.79	0.67	0.53
	396	0.72	0.62	0.51
Stefan	768	0.77	0.69	0.57
	512	0.59	0.58	0.58
	396	0.60	0.59	0.58
coast-guard	768	0.89	0.70	0.57
	512	0.63	0.60	0.59
	396	0.55	0.56	0.54
Flower Garden	768	0.92	0.78	0.51
	512	0.77	0.67	0.50
	396	0.60	0.56	0.54
Silent	768	0.95	0.82	0.70
	512	0.89	0.73	0.55
	396	0.87	0.73	0.63
Con-tainer	768	0.95	0.79	0.63
	512	0.93	0.80	0.60
	396	0.80	0.69	0.59
Ave- -rage	528	0.84	0.68	0.57

TABLE II

THE ROBUSTNESS AGAINST COMMON SIGNAL PROCESSING (5×5 GAUSSIAN FILTER, CIRCULAR AVERAGING FILTER, UNSHARPENED CONTRAST ENHANCEMENT AND GAUSSIAN NOISE), MEASURED BY NORMALIZED CORRELATION.(768, 512 AND 396 KBIT/S, CIF-SIZE)

Video	Bit-rate	Gau- -ssian	Cir- -cular	Con- -trast	Noise
Fore-man	768	0.72	0.87	0.84	0.75
	512	0.62	0.69	0.67	0.62
	396	0.60	0.59	0.76	0.56
Stefan	768	0.69	0.67	0.70	0.66
	512	0.48	0.52	0.70	0.70
	396	0.51	0.61	0.66	0.61
coast-guard	768	0.76	0.83	0.74	0.77
	512	0.62	0.62	0.58	0.62
	396	0.56	0.59	0.55	0.54
Flower Garden	768	0.85	0.89	0.75	0.75
	512	0.67	0.70	0.67	0.67
	396	0.52	0.59	0.69	0.62
Silent	768	0.84	0.81	0.86	0.77
	512	0.68	0.67	0.76	0.63
	396	0.71	0.76	0.72	0.72
Con- -tainer	768	0.77	0.86	0.87	0.70
	512	0.59	0.68	0.76	0.73
	396	0.72	0.70	0.83	0.69
Ave- -rage	529	0.66	0.70	0.73	0.67