Robust Video Watermarking of H.264/AVC

Jing Zhang, *Student Member, IEEE*, Anthony T. S. Ho, *Senior Member, IEEE*, Gang Qiu, and
Pina Marziliano, *Member, IEEE*

Abstract—A robust video watermarking scheme of the state-of-the-art video coding standard H.264/AVC is proposed in this brief. 2-D 8-bit watermarks such as detailed company trademarks or logos can be used as inconvertible watermark for copyright protection. A grayscale watermark pattern is first modified to accommodate the H.264/AVC computational constraints, and then embedded into video data in the compressed domain. With the proposed method, the video watermarking scheme can achieve high robustness and good visual quality without increasing the overall bit-rate. Experimental results show that our algorithm can robustly survive transcoding process and strong common signal processing attacks, such as bit-rate reduction, Gaussian filtering and contrast enhancement.

Index Terms—2-D grayscale watermark pre-processing, copyright protection, H.264/AVC, video watermarking.

I. INTRODUCTION

USAGE of digital multimedia has witnessed a tremendous growth during the last decade as a result of their notable benefits in efficient storage, ease of manipulation and transmission. Unfortunately, the very nature of the digital media makes the work of pirates and hackers easier, since it enables perfect copies with no loss of value. Digital video watermarking can be a promising solution in a digital-rights-management system.

Since most digital video commercial products are sold and circulated in compressed standard-compliant format, numerous compressed-domain methods have been proposed. For example, Langelaar *et al.*[1] proposed the *differential energy watermark* (DEW) algorithm. Hartung *et al.* [2] proposed to embed spread-spectrum watermark into compressed video. Most of previous works, for example [1]–[4], were focusing on MPEG-2 and are evaluated in the context of high bit-rates. As for low bit-rate (≤ 1 Mbit/s) video, Alattar *et al.* [5] extended Hartung's method for MPEG-4 video watermarking. However, few of them deal with the newest H.264/AVC video coding standard, specially designed for low bit-rate applications such as videoconferencing and video surveillance [6].

J. Zhang was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. She is now with iWOW Communication Pte Ltd., Singapore 319637.

A. T. S. Ho was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. He is now with the School of Electronics and Physical Sciences, University of Surrey, Guildford GU2 7XH, U.K.

G. Qiu was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. He is now with Vixs, Toronto, ON M2J 1R3, Canada.

P. Marziliano is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: epina@ntu.edu.sg).
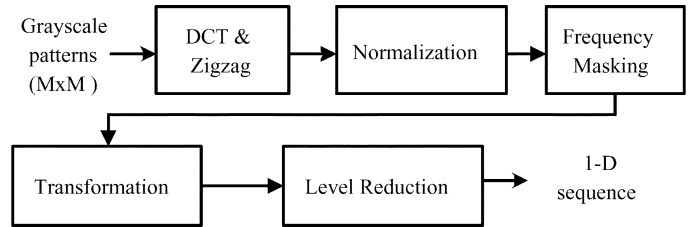
Fig. 1. Flowchart of watermark pre-processing steps.

The H.264/AVC standard impressively improves the coding efficiency with many enhanced functional features [6]. Recent research into H.264/AVC included the fast implementation of integer discrete cosine transform (DCT) [7] and variable block size motion compensation [8]. However, higher compression ratio leads to the difficulty in balancing among tradeoff requirements for watermarking H.264/AVC video data [10], [11]. In this brief, a grayscale watermark pre-processing is adapted for H.264/AVC. This pattern insertion process greatly increase the robustness and capacity providing more textured details than messages that can be treated as watermarks for copyright protection of company trademarks or logos. It can also be extended to include applications for noninvertible watermarking and annotation purposes.

The next section introduces the grayscale watermark pre-processing method. Section III describes the watermarking method. Section IV gives the simulation results followed by conclusions in Section V.

II. WATERMARK PRE-PROCESSING

A. Classification

The preliminary analysis for watermark pre-processing has been focused on grayscale patterns with characters. Therefore, we need to classify the 26 capital letters of the English alphabet and ten numerical numbers for the adaptive process, such as *Transformation*, which will be explained later. Let us define four categories as follows.

- *Category $\mathcal{C}$*: C, J, O, Q, S, U, 0, 3, 6, 8, and 9.
- *Category $\mathcal{L}$*: E, F, H, I, L, T, and 1.
- *Category $\mathcal{L} + \mathcal{C}$*: B, D, G, P, R, 2, 4, and 5.
- *Category $\mathcal{L} + \mathcal{D}$*: A, K, M, N, V, W, X, Y, Z, and 7.

In the above, $\mathcal{C}, \mathcal{L}$, and $\mathcal{D}$ denote curves, lines, and diagonals, respectively.

B. Proposed Method

Fig. 1 shows the flowchart of the watermark pre-processing steps. Watermark patterns can then be reconstructed from the 1-D output sequence obtained from the pre-processing. Fig. 2 shows the watermark embedding scheme.
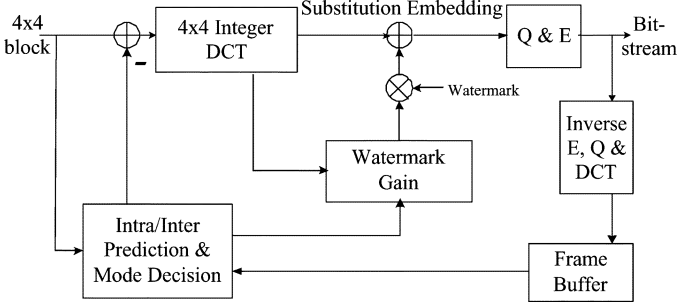
Fig. 2. Watermark embedding scheme.

TABLE I
PARAMETERS a AND m FOR WATERMARK PRE-PROCESSING ($\mathcal{C}$, $\mathcal{L}$ AND $\mathcal{D}$
DENOTE CURVES, LINES AND DIAGONALS, RESPECTIVELY)

k	$\mathcal{C}$		$\mathcal{L}$		$\mathcal{L}+\mathcal{C}$		$\mathcal{L}+\mathcal{D}$	
	a	m	a	m	a	m	a	m
0	0	166	0	140	0	195	0	157
1	500	150	430	140	440	145	420	150
2	100	40	250	100	400	100	300	100
3	50	40	150	60	100	60	160	60
4	200	60	250	80	300	100	200	50
5	300	100	400	120	350	120	450	200
6	300	70	400	70	120	50	260	80

TABLE II
RECONSTRUCTED WATERMARKS AFTER PRE-PROCESSING
AND THE CORRESPONDING CORRELATIONS

Pattern	$\mathcal{C}$	$\mathcal{L}$	$\mathcal{L}+\mathcal{C}$	$\mathcal{L}+\mathcal{D}$	Others	
Original	689	ILT	BRG	WXY	NTU	
Group(A)	689	ILT	BRG	WXY	NTU	
$\rho(A)$	0.97	0.99	0.98	0.99	0.99	0.96
Group(B)	689	ILT	BRG	WXY	NTU	
$\rho(B)$	0.92	0.99	0.95	0.97	0.98	0.91

1) DCT and Zigzag-Scanning: The grayscale pattern is first decomposed into nonoverlapping 4×4 blocks, denoted by $p(i,j)$ $(0 \leq i,j \leq 3)$. Then, each block $p(i,j)$ is transformed by

$$P(u,v) = \text{IntDCT}\{p(i,j)\}, \qquad 0 \leq u,v \leq 3 \qquad (1)$$

where $\text{IntDCT}\{\cdot\}$ represents the 4×4 integer DCT with low-complexity and higher accuracy and is uniquely used in H.264/AVC [7]. This transform is approximated to the traditional DCT used in all prior video standards. However, it has many fundamental differences which contribute to a better performance for video coding [9] and digital watermarking [10].

The coefficients $P(u,v)$ of the 2-D 4×4 integer DCT transformed matrix are zigzag-scanned into a 1-D sequence of 16-integer DCT coefficients $P(k)$, where $k \in [0,15]$ indicates the zigzag position.

2) Normalization and Frequency Mask: After integer DCT, several lowest frequencies possess sufficient information for the pattern reconstruction. To effectively discard these unnecessary high frequency coefficients, all 16 DCT coefficients $P(k)$ in each 4×4 block are normalized as follows:

$$P^n(k) = \frac{|P(k)|}{\sum_{k=0}^{15} |P(k)|}, \qquad 0 \leq k \leq 15. \qquad (2)$$

From the experimental results, the four normalized coefficients $P^n(k)$ with the largest values are at different zigzag scanning positions k for patterns with characters from various categories.

- Category $\mathcal{C}$: $k \in \mathbf{K}_{\mathcal{C}} = \{0,1,2,5\}$.
- Category $\mathcal{L}$: $k \in \mathbf{K}_{\mathcal{L}} = \{0,1,3,5\}$.
- Category $\mathcal{L}+\mathcal{C}$: $k \in \mathbf{K}_{\mathcal{L}\mathcal{C}} = \{0,1,2,5\}$.
- Category $\mathcal{L}+\mathcal{D}$: $k \in \mathbf{K}_{\mathcal{L}\mathcal{D}} = \{0,1,5,6\}$.

Therefore, four different sets $\mathbf{K}_{\mathcal{C}}, \mathbf{K}_{\mathcal{L}}, \mathbf{K}_{\mathcal{L}\mathcal{C}}$ and $\mathbf{K}_{\mathcal{L}\mathcal{D}}$ are generated from the normalized coefficients $P^n(k)$. Next, only the DCT coefficients $P(k)$ with the index k belonging to these four sets can be kept, and the coefficients with k not belonging to the these sets are discarded. This technique is known as *adaptive frequency masking*, and these sets act as the frequency masks.

3) Transformation: After masking, there is a significant difference between the dynamic range of every two adjacent coefficients. In order to narrow down these dynamic ranges for better distortion control, the remaining coefficients $P(k)$ are transformed as follows:

$$P^t(k) = \frac{P(k) + a(k)}{m(k)}, \qquad 0 \leq k < 15 \qquad (3)$$

where $a(k)$ is added to $P(k)$ to keep all the result values positive, and $m(k)$ acts as compression quantization step to narrow down the dynamic range.

In order to effectively represent the original pattern, $a(k)$ and $m(k)$ both change adaptively to zigzag position k of the corresponding coefficients, and also to different categories of characters as mentioned. Based on our experiments, the $a(k)$ and $m(k)$ pairs corresponding to the seven lowest frequencies are shown in Table I.

4) Level Reduction: The coefficients $P^t(k)$ after transformation have multiple values, $P^t(k) \in [1,7]$. They can be further simplified to be binary, denoted by $w(k)$ as

$$w(k) = P^t(k)\text{mod}2 \qquad (4)$$
$$Key(k) = P^t(k)/2 \qquad (5)$$

where k is in the sets $\mathbf{K}_{\mathcal{C}}, \mathbf{K}_{\mathcal{L}}, \mathbf{K}_{\mathcal{L}\mathcal{C}}$, or $\mathbf{K}_{\mathcal{L}\mathcal{D}}$ and the sequence Key can be stored in a key parameter file for future reconstruction.

Table II shows two groups of watermark samples reconstructed directly after pre-processing without watermarking, and the corresponding normalized correlation. All patterns have the same size $M \times M$, where M is equal to 32. A grayscale

pattern (*NTU* logo) with a hybrid combination from different character categories and an *EEE* logo are also illustrated. In the case of *NTU*, the pattern can be separated into three parts and each part undergoes the pre-processing separately.

In Group A of Table II, the 7 lowest frequencies ($k \in [1, 7]$) in each 4×4 block are remained without adaptive frequency masking. In Group B, normalization and adaptive frequency masking are conducted, and only four coefficients are remained after masking. The number of coefficients for watermarking has been reduced to 43.8% in Group A and 25.0% in group B.

The reconstructed grayscale watermarks are highly correlated to the original as given in Table II. The watermark patterns with more straight lines, such as in Category $\mathcal{L}$, can obtain better reconstruction than patterns in Category $\mathcal{C}$ and $\mathcal{L} + \mathcal{D}$. This is due to the fact that watermarks with more curves always have more energy in higher DCT frequencies, while most of the high frequencies are discarded during the pre-processing.

III. PROPOSED WATERMARKING METHOD

The proposed grayscale watermark pre-processing can change a 2-D 8-bit watermark pattern ($M \times M$) into a binary sequence $\mathbf{w} = \{w(n), n = 0, \ldots, M^2/4 - 1\}$, where $w(n) \in \{0, 1\}$ and $M^2/4$ is the reduced watermark length. Our watermarking scheme is applied after the pre-processing and the binary sequence $\mathbf{w}$ will be used for embedding.

A. Embedding

Watermark information $\mathbf{w} = \{w(n), n = 0, \ldots, M^2/4 - 1\}$ is firstly mapped to a bi-polar vector as follows:

$$w^b(n) = (-1)^{w(n)}, \qquad n = 0, 1, \ldots, M^2/4 - 1. \quad (6)$$

The spread-spectrum watermark method is defined as follows:

$$b(m) = w^b(n), \qquad nS \leq m < (n+1)S \quad (7)$$
$$w^s(m) = p(m) \cdot b(m), \qquad m = 0, \ldots, NS - 1 \quad (8)$$

where $p(m) = \pm 1$ is a pseudonoise bit. S is the spreading factor, determined experimentally according to the video frame size and watermark size.

The significant compression efficiency of the H.264/AVC standard means that the remaining DCT coefficients are perceptually important. Thus, during embedding, only one certain middle frequency $X_{u_o, v_o}(m)$ in the diagonal positions in a 4×4 DCT block is substituted by

$$X_{u_o, v_o}(m) \leftarrow \tilde{X}_{u_o, v_o}(m) = \alpha \cdot \beta(m) \cdot w^s(m) \quad (9)$$

where u_o and v_o indicate the position of the coefficient, $w^s(m)$ is the spread-spectrum watermark and both α and $\beta(m)$ are positive gain factors.

The gain factor α is empirically decided by the user. While the local gain factor $\beta(m)$ is obtained from the DC coefficient $X_{0,0}(m)$ and the AC coefficients $X_{u,v}(m)$:

$$\beta = \max \left[0, -X_{0,0}(m) + \mu \sum_{1 \leq u, v \leq 3} |X_{u,v}(m)| \right] \quad (10)$$

where μ is the the weighting factor. For relatively dark and "busy" (or textured) regions, the local gain factor increases the embedding power for higher robustness. For relatively bright and smooth regions, where human eyes are more readily to spot any distortions, the local gain factor should be decreased to minimize the watermark visibility.

To increase compression efficiency, the H.264/AVC standard supports a tree-structured motion compensation [9] with seven inter prediction modes for each macroblock B as well as two Intra prediction modes. A larger partition size is suitable for still areas and a smaller partition size is suitable for areas with detailed motion [8].

After embedding $w^s(m)$ into all the available Inter/Intra modes (denoted by $\mathcal{O}$), the best mode $\tilde{o}$ for the marked macroblock $\tilde{B}$ is selected by minimizing the expression in (11) within the constrained R and minimized D, using Lagrangian Optimization technique [9] of H.264/AVC

$$\tilde{o} = \arg \min_{o \in \mathcal{O}} \left(D(\tilde{\boldsymbol{B}}, o) + \lambda R(\tilde{\boldsymbol{B}}, o) \right) \quad (11)$$

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

B. Retrieval

At the H.264/AVC Decoder, the retrieval approach is carried out as follows.

1) One watermark coefficient $\hat{w}^s(m)$ is extracted according to the polarity of the marked quantized coefficient $\hat{X}_{u,v}(m)$ in each 4×4 block: the watermark $\hat{w}^s(m)$ is equal to the sign of $\hat{X}_{u,v}(m)$.
2) The correlation sum H_n between the extracted watermark $\hat{w}^s(m)$ and the original spread-spectrum pseudonoise sequence $\mathbf{p}$ is computed as follows:

$$H_n = \sum_{m=nS}^{(n+1)S-1} p(m) \cdot \hat{X}_{u_o, v_o}(m). \quad (12)$$

3) The extracted bit $\hat{w}(n)$ can be derived from the correlation sum with an appropriate threshold τ ($\tau = 0$ in our experiments): $\hat{w}(n) = 1$ if $H_n > \tau$; $\hat{w}(n) = 0$ if $H_n < -\tau$; otherwise $\hat{w}(n)$ is deemed to be lost and set to a very large negative value.

After extracting the sequence $\hat{w}(n)$ from the compressed H.264 video, the inverse approach of the watermark pre-processing reconstructs the extracted grayscale watermark.

IV. EXPERIMENTS AND RESULTS

A. Test Conditions

The proposed watermarking technique has been integrated into the H.264 JM-8.6 reference software [12]. The first 100 frames of the video sequences: *Foreman*, *Stefan*, *Coastguard*, *Flower Garden*, *Container Ship*, and *Silent* are used in the experiments. All video clips are coded in CIF format (352×288
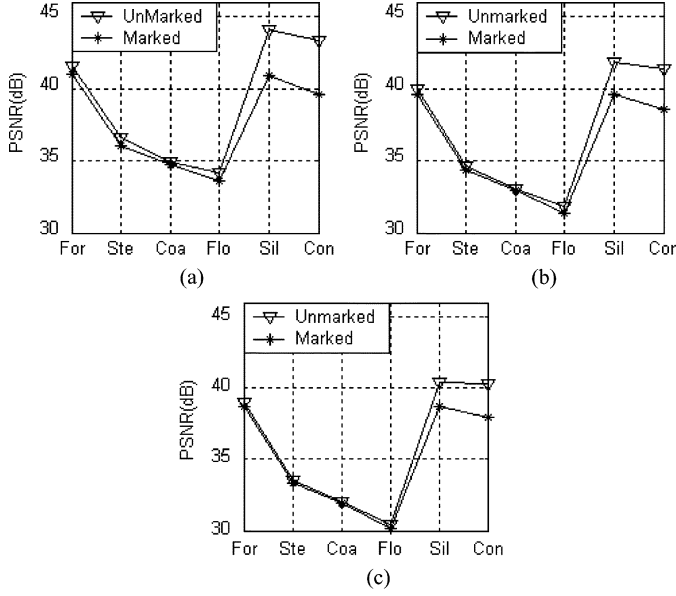
Fig. 3. PSNR(dB) of unmarked and marked *Foreman*, *Stefan*, *Coastguard*, *Flower*, *silent*, and *Container* (denoted by *For*, *Ste*, *Coa*, *Flo*, *Sil*, and *Con* in the horizontal axis) (768, 512, 396 kbit/s, CIF-size).
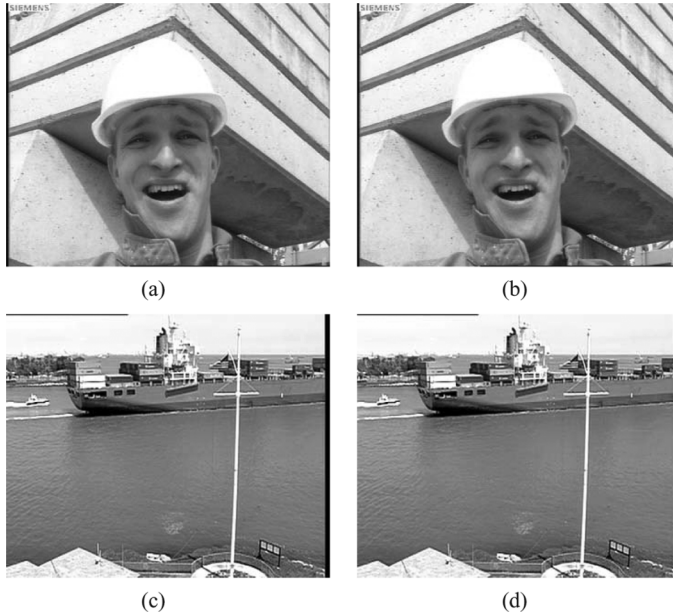

Fig. 4. The 30th frames of *Foreman* and *Container* (768 kbit/s, CIF-size) without and with watermark. (a) Unmarked (41.55 dB). (b) Marked (41.00 dB). (c) Unmarked (43.30 dB). (d) Marked (39.54 dB).

pixels) at the frame rate 15 frames/s at the bit-rate 768 kbit/s, 512 kbit/s, and 396 kbit/s, respectively. The GOP structure comprises IPP..., compliant to the *Baseline Profile* of H.264/AVC. A small 16×16 grayscale watermark is used as the watermark pattern for our experiments.

B. Fidelity After Watermarking

Fig. 3 shows the average peak signal-to-noise ratio (PSNR) comparison results (decibels). On average, the watermarking leads to a decrease of approximately 1.1 dB. This is mainly
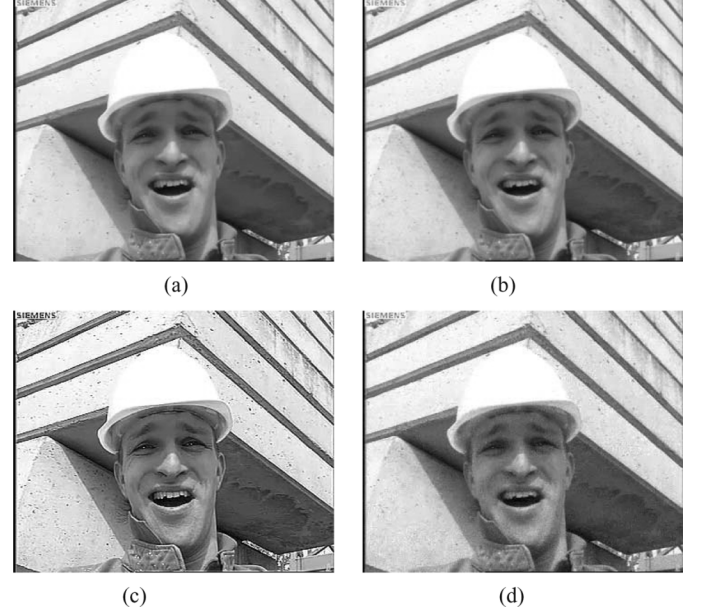

Fig. 5. The 30th frame of Foreman under different attacks (768 kbit/s, CIF size). (a) To $1/3$ bit-rate (35.11 dB). (b) 5×5 Gaussian filter (35.73 dB). (c) Contrast Enhancement (22.91 dB). (d) Gaussian Noise (31.04 dB).

TABLE III
RECONSTRUCTED WATERMARKS AFTER PRE-PROCESSING AND
THE CORRESPONDING CORRELATIONS

Original Watermark	**K**
Reconstructed after WM Pre-processing	**K** (ρ =0.97)
Extracted from watermarked video	**K** (ρ =0.93)
Extracted after transcoding (to1/3 Bit-rate)	**K** (ρ =0.50)
Extracted after 5x5 Gaussian Filtering	**K** (ρ =0.72)
Extracted after Circular Filtering	**K** (ρ =0.87)
Extracted after Contrast Enhancement	**K** (ρ =0.83)
Extracted after Gaussian Noise	**K** (ρ =0.75)

due to the Lagrangian optimization in H.264/AVC and the proposed watermark gain factors in our algorithm. Lagrangian optimization has previously been applied to improving watermark detection for noisy interpolated images [4]. Fig. 4 shows both *Foreman* and *Container* as examples of video clips with significant motion and with minimal motion, respectively. In the experiments, no visible artifacts can be observed in all of the test video sequences.

C. Results of Robustness Tests

To test the robustness of the proposed algorithm, the watermarked videos are transcoded to $1/3$ of the original bit-rate. Common signal processing attacks have been applied, including 5×5 Gaussian low-pass filtering, circular averaging filtering, unsharp contrast enhancement, and additive Gaussian noise (variance $= 0.001$). Fig. 5 shows the examples of attacked *Foreman* (768 kbit/s, CIF-size). The visual quality decreases significantly, especially under contrast enhancement and Gaussian noise.
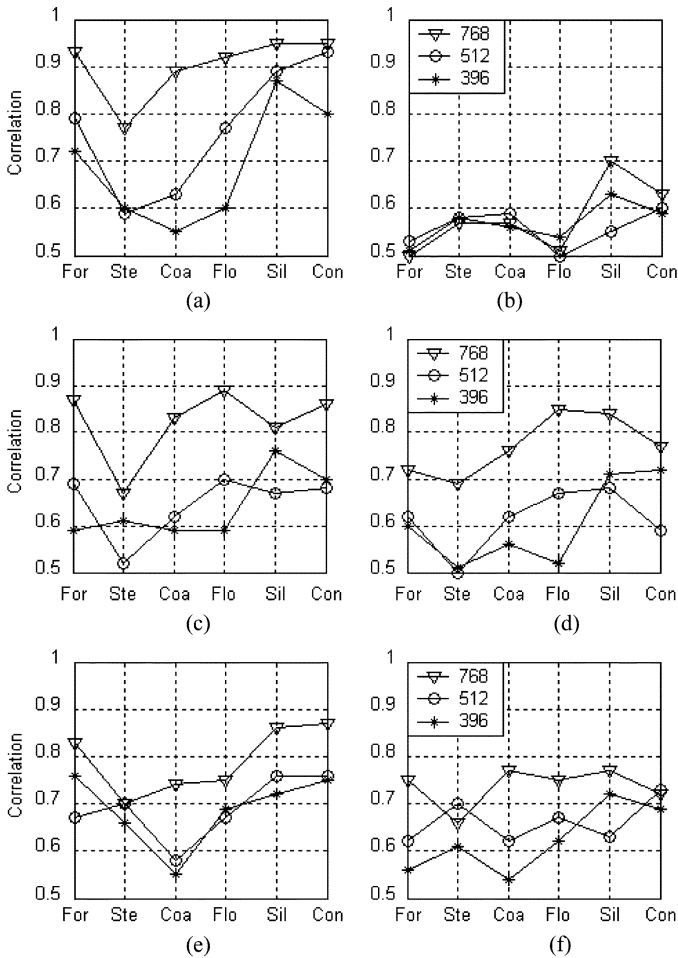
Fig. 6. Robustness under transcoding and common signal processing on *Foreman*, *Stefan*, *Coastguard*, *Flower*, *silent* and *Container* (denoted by *For*, *Ste*, *Coa*, *Flo*, *Sil*, and *Con* in the horizontal axis) (768 kbit/s, 512 kbit/s, and 396 kbit/s, CIF-size).

Table III shows the reconstructed watermark without watermarking and the retrieved samples from the marked and attacked *Foreman* (768 kbit/s, CIF-size). These samples are highly correlated to the original pattern. The proposed watermarking algorithm together with the watermark pre-processing can still survive strong attacks, such as bit-rate reduction, contrast enhancement and Gaussian noise.

The robustness is measured by the normalized correlation between the original grayscale watermark and the extracted watermark, as illustrated in Fig. 6. The average normalized correlation per frame for all watermarked video streams at 768 kbits/s is approximately 0.90. After transcoding to the $1/3$ of the original bit-rate, the average correlation drops to approximately 0.58. After contrast enhancement, the average correlation only decreases approximately by 12%. Moreover, among the six test sequences, *Silent* and *Container Ship* give the best robustness after transcoding and signal processing attacks. This leads to the conclusion that more correct watermark bits are recovered

in the two sequences, because there is only minimal motion in the two sequences. Most of the watermark bits are automatically embedded into the textured background with high gain. These bits are relatively more robust, even though the introduced distortion is higher than the other four sequences with significant motion. These results validate the fundamental tradeoff between the fidelity of the video data and the watermark robustness.

V. CONCLUSION

The proposed grayscale watermark pre-processing technique greatly increases the robustness and capacity of the video watermarking algorithm for copyright protection and maintain the good visual quality and almost the same Bit-rate. Detailed copyright information, such as textured company trademarks or logos can be used as watermark to further protect ownership and defend against the illegal attacks. To protect copyright, how to combine the proposed embedding techniques with noninvertible watermarking [13] will be future research.

REFERENCES

[1] G. Langelaar and R. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Trans. Signal Process.*, vol. 10, no. 1, pp. 148–158, Jan. 2001.

[2] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Process.*, vol. 66, no. 3, pp. 283–301, May 1998.

[3] S. Arena, M. Caramma, and R. Lancini, "Digital watermarking applied to MPEG-2 coded video sequences exploiting space and frequency masking," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2000, pp. 1987–1990.

[4] A. Giannoula, N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Watermark detection for noisy interpolated images," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 53, no. 5, pp. 359–363, May 2006.

[5] A. M. Alattar, E. T. Lin, and M. U. Celik, "Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 787–800, Aug. 2003.

[6] T. Wiegang and G. J. Sullivan, Study of final committee draft of joint video specification (ITU-T rec. H.264/ISO/IEC 14496-10 AVC) Doc. JVT-G050d2, Joint video Team (JVT) of ISO/IEC MEPG & ITU-T VCEG(ISO/IECJTC1/SC29/WG11 and ITU-T SG16 Q.6), Dec. 2002.

[7] C.-P. Fan, "Fast 2-dimensional 4×4 forward integer transform implementation for H.264/AVC," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 3, pp. 174–177, Mar. 2006.

[8] C.-Y. Chen, S.-Y. Chien, Y.-W. Huang, T.-C. Chen, T.-C. Wang, and L.-G. Chen, "Analysis and architecture design of variable block-size motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 3, pp. 578–593, Mar. 2006.

[9] 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.

[10] J. Zhang and A. T. S. Ho, "An efficient digital image-in-image watermarking algorithm using the integer discrete cosine transform (IntDCT)," in *Proc. IEEE Joint Conf. 4th Int. Conf. Info., Commun. Signal Process. and 4th Pacific-Rim Conf. Multimedia*, Dec. 2003, vol. 2, pp. 1163–1167.

[11] 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. 17th Int. Conf. Pattern Recogn.*, Cambridge, U.K., Aug. 2004.

[12] K. Hühring, H.264/AVC Joint Model 8.6 (JM-8.6) Reference Software [Online]. Available: http://iphome.hhi.de/suehring/tml/

[13] Q. Li and E.-C. Change, , J. , Ed., "On the possibility of noninvertible watermarking schemes," in *IH 2004, Lecture notes in Computer science (LNCS)*. Berlin: Springer-Verlag, 2004, vol. 3200, pp. 13–24.