CONSIDERING THE RECONSTRUCTION LOOP FOR WATERMARKING OF INTRA AND INTER FRAMES OF H.264/AVC

Z. Shahid, P. Meuel, M. Chaumont and W. Puech

LIRMM Laboratory, UMR 5506 CNRS, University of Montpellier II
161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE
zafar.shahid@lirmm.fr, peter.meuel@lirmm.fr, marc.chaumont@lirmm.fr, william.puech@lirmm.fr

ABSTRACT

This paper presents design and analysis of watermarking of *intra* and *inter* frames in H.264/AVC video codec. Most of video watermarking algorithms take into account only *intra* frames for watermark embedding. This is due to the perception that *inter* frames are highly compressed by motion compensation and embedding watermark in them can affect the compression efficiency significantly. In this paper, we analyze watermark embedding in *intra* as well as in *inter* frames over the whole RD curve and we note that watermark embedding capability of *inter* is comparable to that of *intra* frames. Watermark embedding, in only those non-zero quantized transform coefficients (QTCs) which are above a specific threshold, enables us to detect and extract the watermark on the decoding side. For the non-zero QTCs, in *intra* frames we exploit the spatial masking and in *inter* frames we exploit motion and texture masking for watermark embedding. There is not significant compromise on quality and bitrate of the video bitstream because we have taken into account the reconstruction loop during the watermarking step.

1. INTRODUCTION

Many multimedia applications have emerged in the last decade because of rapid growth of processing power and network bandwidth. As digital data can be easily copied or modified, it must be protected and authenticated. Digital video watermarking has emerged as an important research field to protect the copyrighted multimedia data. Watermarking is used in many applications for owner identification, copyright protection and integrity. In video data, watermarking can be carried out either in spatial or frequency domain. Watermarking in spatial domain can be lost because of the lossy stage of quantization. In the frequency domain, watermarking is done normally in QTCs. For this purpose, few specific methods have been developed for standard such as MPEG-4 [4, 2].

In section 2, first we present the H.264/AVC video codec with its integer transform (IT) and the quantization process followed by an overview of previous watermarking techniques related to this video standard. We present the proposed algorithm by elaborating the embedding and extraction steps in Section 3. Section 4 contains experimental results and performance analysis for both *intra* and *inter* frames including payload capability, bitrate and quality trade off for embedding in more than one LSBs. Finally, in Section 5, we present the concluding remarks about the proposed algorithm.

2. H.264/AVC WATERMARKING, CHALLENGES AND PROSPECTS

Since significant changes have been incorporated in H.264/AVC as compared to the previous video coding standards, an overview of H.264/AVC with an emphasis on transform and quantization is presented Section 2.1. It is followed in Section 2.2 by previous watermarking techniques for H.264/AVC which have been proposed in literature.

2.1 Overview of H.264/AVC

H.264/AVC [1] has some additional features as compared to previous video standards. In *baseline* profile of H.264/AVC, it has 4×4 transform in contrast to 8×8 transform of previous standards. DCT transform has been replaced by IT which can be implemented by only additions and shifts in 16 bit arithmetic without any multiplication and hence requires lesser number of computations. H.264/AVC codec uses a uniform scalar quantization. For *Inter* frame, H.264/AVC supports variable block size motion estimation, quarter pixel accuracy, multiple reference frames, improved skipped and direct motion inference. For *Intra* frame, it offers additional spatial prediction modes. All these additional features of H.264/AVC are aimed to outperform previous video coding standards [9]. The block diagram of H.264/AVC is shown in Fig. 1.

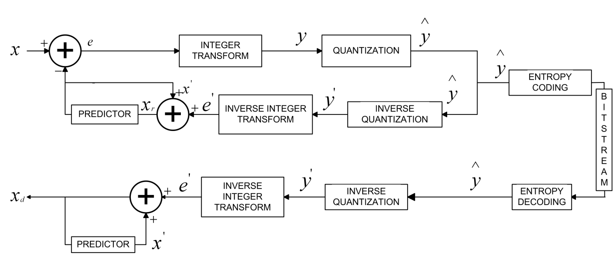


Figure 1: Detailed block diagram explaining prediction, transform and quantization steps in H.264/AVC.

A macroblock (MB) is divided into 16 blocks of 4×4 pixels and they are processed one by one. In *intra* mode, H.264/AVC has three modes, $Intra\_4\times4$, $Intra\_16\times16$ and I_PCM . In $Intra\_16\times16$ mode, Hadamard transform is further used to encode DC coefficients. Entire MB is predicted from top and left neighboring pixels and has 4 modes namely *horizontal*, *vertical*, *DC* and *plane* modes. In $Intra\_4\times4$ mode, each 4×4 luma block is predicted from top and left pixels of reconstructed 4×4 neighbors and has 9 prediction modes. I_PCM mode is used to limit the maximum size of encoded block and bypass transform and quantization stages.

Transform and the quantization process are embedded with each other to save the processing power and to avoid multiplications. Let X be a 4×4 block as shown in Fig 1. First of all, it is predicted from its neighboring blocks and we get the residual block:

$$E = P(X, B_1, B_2, B_3, B_4). \qquad (1)$$

In H.264, intra prediction is performed from the reconstructed neighboring pixels where B_i are the reconstructed neighboring blocks. Forward and inverse integer transform 4×4 matrices (A, A_{inv}) are given in [5]. This residual block E is then transformed using the the forward transform matrix A:

$$Y = AEA^T. \qquad (2)$$

Scalar multiplication and quantization are defined as:

$$\hat{Y} = sign\{Y\}[(|Y| \otimes Aq + Fq \times 2^{15+Eq}) >> (15+Eq)], \quad (3)$$

where $\hat{Y}$ is quantized transformed coefficient. Aq is the 4×4 quantization matrix and Eq is the shifting matrix. Both Aq and Eq are indexed by QP. Fq is the quantization rounding factor matrix. Right shift operator is applied to every elements of 4×4 matrix. This $\hat{Y}$ is entropy coded and sent to the decoder side. On the decoder side, inverse quantization is given by the expression $Y' = \{[(\hat{Y} \otimes (Bq << 4)) << Eq] + 8\} >> 4$, where Bq and Eq are the inverse 4×4 quantization matrix and the shifting factor respectivly. Y' is then inverse transformed to get $E' = (A_{inv} Y' A_{inv}^T + 32) >> 6$. The decoded residual signal E' is then added to the predicted signal to reconstruct the original signal back.

2.2 Previous H.264/AVC watermarking techniques

In literature, very few watermarking techniques have been proposed for H.264/AVC. Most of them propose to embed the message only in the *intra* frames. This is due to the inherent complexity and compression efficiency of this video standard. Because of IT, traditional spread spectrum technique are not viable since they embed watermark drawn from a Gaussian distribution. Golikeri *et al.* have proposed watermarking algorithm for H.264/AVC [3], in which they have used visual models developed by Watson [8] to choose the coefficients to be watermarked based on frequency sensitivity, luminance masking and contrast masking. They embed watermark in transformed coefficients before quantization. In the case of $Intra\_16\times16$ mode, they have also embedded the watermark in Hadamard transform coefficients. Some methods have suggested embedding watermark in entropy coding stage [10, 6]. In that case, there is a drift which degrades the visual quality significantly. To avoid drift, either we have to do reversible watermarking in which signal is extracted before decoding or drift compensation signal should be added to decoder [4]. Noorkami and Merserau have presented a technique to embed watermark in both *intra* and *inter* frames [7]. They propose to embed watermark in all the non-zero QTCs. They claim that visual quality of *inter* frames is not compromised even if we embed watermark in all non-zero QTCs. Owing to insertion of watermark only in non-zer QTCs, their method does not affect the compression efficiency of run-length coding. But performance of context-based adaptive variable length coding (CAVLC) gets affected and as a result, some increase in bitrate is observed. Since there are lot of QTCs with magnitude '1' and CAVLC encodes *trailing ones* (T1's) separately, it affects the coding efficiency of entropy coding engine. In this paper, we have not embedded watermark in all non-zero QTCs, rather we have embedded watermark in only those QTCs which are above a certain threshold which depends upon the number of watermark bits (WMBits) being embedded. It has two advantages. First, it makes it possible to extract the watermark on the decoder side. Second, it does not affect much the compression efficiency of entropy coding engine.

3. PROPOSED ALGORITHM

In H.264/AVC, intra prediction is performed in the spatial domain. So even for *intra* mode, the transform is performed on prediction residuals. We have used the reference implementation of H.264 JSVM 10.2 in AVC mode for video sequences in CIF resolution. To keep the watermark imperceptible, we have not modified the DC coefficients. For $Intra\_16\times16$ mode, we have not modified the Hadamard transform coefficients either. We have embedded watermark in LSBs of QTCs keeping in view the following points:

- QTC which we want to watermark should be non-zero. If a QTC with zero magnitude becomes non-zero in the course of embedding, it will highly affect the compression efficiency of run-length encoding.
- QTC to be watermarked should be preferably greater than '1' because there are many QTC with magnitude '1' and in CAVLC, they are also encoded as T1's. Thus changing of number of T1's will affect the compression efficiency of CAVLC.
- Finally watermark is embedded in such a fashion that it can be completely extracted on the decoder side.

Our watermark algorithm fulfils all the above conditions.

3.1 Embedding process within encoder loop

From equation (3) we apply the embedding with:

$$\hat{Y}_W = f(\hat{Y}, W, [K]). \qquad (4)$$

where $f()$ is the watermarking process, W the watermark signal and K is the watermarking key ([] shows the optional argument). Watermark embedding can be done in QTC before entropy coding. This embedding creates two problems. Firstly, we started reconstruction on the encoder side with QTC $\hat{Y}$ while on the decoder side we start decoding with watermarked QTC $\hat{Y}_w$. This results in a mismatch on the decoder side, which keeps on increasing because of the prediction process. Because of this mismatch, the difference in PSNR is very significant even for *intra* frames, let alone the *inter* frames. Secondly, Rate Distortion (RD) bit allocation algorithm works in quantization module and any change in bitrate/quality trade off because of the watermarking of QTC is not being taken into account.

To solve both the problems, watermark embedding should be performed inside the reconstruction loop as shown in Fig. 2. In this case, we have the same watermarked QTC $\hat{Y}_w$ on both encoder and decoder side for prediction and RD bit allocation algorithm is working on $\hat{Y}_w$.

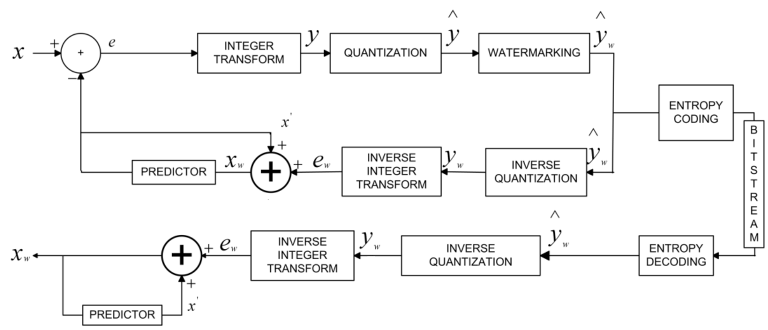


Figure 2: Watermarking by modifying the 1, 2 or 1 & 2 least significant bits inside the reconstruction loop.

3.2 Watermark aware Rate Distortion

Many encoding parameters like prediction modes, quantization parameter (QP) value and motion vectors are adjusted in the encoding process based on video content and required quality. Since video data is very diverse in nature both spatially and temporally, these parameters vary from scene to scene. Bit allocation algorithms are used to find the most suitable values of these parameters to achieve the trade off between bitrate and quality. For RD, Lagrangian bit-allocation is widely used owing to its simplicity and effectiveness. The simplified Lagrangian cost function is $J = D + \lambda R$, where J represents the cost of encoding for a given MB, D is the distortion, λ is the Lagrangian parameter depending on QP value and R is the number of bits to encode a MB. To obtain the cost J for a specific prediction mode P, we first predict the MB for that mode to get residual E. We then apply IT followed by quantization with some QP value to get QTC which are then entropy coded. The number of bits R consists of MB header bits and data bits. Then, residual is reconstructed by performing inverse quantization and inverse IT to give the reconstructed residual E'. Generally, the distortion D is either sum of absolute differences (SAD), sum of squared differences (SSD) or sum of absolute transformed differences (SATD) between E and E'. Thus, we end up with the cost J for encoding this MB in the P mode. In a similar fashion, we find cost J for all other prediction modes. The mode which yields the minimum cost is selected as the RD optimized mode for this MB.

Embedding a watermark in a video bitstream affects quality of the picture. It also affects the bitrate because this frame is used for prediction after reconstruction. Hence, RD optimization should take into account the embedding of watermark in QTCs in order to select the best prediction mode. In this case, simplified Lagrangian cost function is $J_w = D_w + \lambda R_w$, for finding the cost for a specific prediction mode. QTCs are first watermarked and then are entropy coded to find the number of bits R to encode MB and reconstructed to measure the distortion D instead of QTC. By moving the watermark embedding process inside the reconstruction loop, it incorporates the best suitable mode for the watermarked blocks.

3.3 Embedding strategy

For watermark embedding in a video bitstream, we developed a strategy to embed watermark in 1 LSB, 2 LSBs and 1 &2 LSBs together. For the embedding of 1 WMBit in LSB of $|QTC|$:

$$
\begin{array}{l}
if\ (|QTC| > 1)\ then \\
\left\{ \begin{array}{lcccl} |QTC| & = & |QTC| & \& & 0xfffffffe; \\ WMBit & = & WMBit & \& & 0x00000001; \\ |QTC| & = & |QTC| & | & WMBit. \end{array} \right.
\end{array} \tag{5}
$$

If $|QTC|$ is less than 2, it will remain unchanged. For $|QTC| \geq 2$, output will be either same or will get modified by ± 1, depending on whether $WMBit$ is '0' or '1'. In this case, we have 0.5 probability that the coefficient will remain unchanged even after being watermarked.

For embedding of 2 bits in 2 LSBs of QTC:

$$
\begin{array}{l}
if\ (|QTC| > 3)\ then \\
\left\{ \begin{array}{lcccl} |QTC| & = & |QTC| & \& & 0xfffffffc; \\ WMBit & = & WMBit & \& & 0x00000003; \\ |QTC| & = & |QTC| & | & WMBit. \end{array} \right.
\end{array} \tag{6}
$$

By keeping the threshold more than '3', we can extract the watermark message on decoder side successfully.

QTC will remain unchanged if $|QTC| < 4$. If $QTC \geq 4$, it will get modified depending on whether WMBits are '00', '01', '10' or '11'. In this case, we have only 0.25 probability that the coefficient will remain unchanged even after being watermarked. So the compromise in PSNR is of relatively higher value. We can also adapt a 1 & 2 LSB embedding together. In this case, we embed watermark in 0, 1 or 2 LSBs depending on value of $|QTC|$:

$$
\begin{array}{l}
if\ (|QTC| > 3)\ then \\
\left\{ \begin{array}{lcccl} |QTC| & = & |QTC| & \& & 0xfffffffc; \\ WMBit & = & WMBit & \& & 0x00000003; \\ |QTC| & = & |QTC| & | & WMBit. \end{array} \right. \\
else\ (if\ |QTC| > 1)\ then \\
\left\{ \begin{array}{lcccl} |QTC| & = & |QTC| & \& & 0xfffffffe; \\ WMBit & = & WMBit & \& & 0x00000001; \\ |QTC| & = & |QTC| & | & WMBit. \end{array} \right.
\end{array} \tag{7}
$$

So we insert 2 WMBits if $|QTC|$ is higher enough or 1 WMBit if $|QTC| > 1$.

3.4 Watermark extraction

During watermark extraction process, we can extract the watermark from watermarked QTCs as:

$$W = g(\hat{Y}_w, [K]), \tag{8}$$

where $g()$ is the watermark detection/extraction process, $\hat{Y}_w$ is the watermarked QTC, $[K]$ shows the optional key required for extraction of watermark. For 1 LSB watermark extraction, $g()$ can be given as:

$$\begin{array}{l} if\ (|QTC| > 1)\ then \\ \{\ WMBit \ = \ |QTC| \ \ \& \ \ 0x00000001; \end{array} \tag{9}$$

Watermark extraction process works in the same way for 2 LSBs and 1 & 2 LSBs modes.

4. EXPERIMENTAL RESULTS

For experimental simulation, we have used the reference implementation of H.264 JSVM 10.2 in AVC mode and applied our method on 9 benchmark video sequences in CIF resolution. Each of them represents different combinations of motion (fast/slow, pan/zoom/rotation), color(bright/dull), contrast (high/low) and objects (vehicle, buildings, people). We have first done the simulation only with *intra* frames, and then with *intra* and *inter* frames both for 1 LSB, 2 LSBs and 1 & 2 LSBs embedding.

For *intra* frames we have encoded 150 frames of each sequence. Here, we exploit the spatial masking to embed watermark in QTC. Watermark is embedded in those parts of frames which contain texture and edges. To analyze payload simulation, we have used *foreman* standard video sequence for the QP values of 18 and 36 as shown in Table 1. At QP value of 36, few QTCs have magnitude above threshold for 2 LSB insertions and very few WMBits can be inserted. But we have adequate number of QTCs with magnitude greater than '1' so we have enough number of WMBits inserted for this mode.

		Payload Kbits/frame	Frame Size Kbytes	PSNR dB
QP 18 I frames	0	0	2.815	44.883
	LSB1	9.375	2.889	43.801
	LSB2	5.605	2.875	43.605
	LSB1&2	12.484	2.909	42.928
QP 36 I frames	0	0	0.377	32.628
	LSB1	0.206	0.381	32.536
	LSB2	0.012	0.377	32.612
	LSB1&2	0.214	0.381	32.526

Table 1: Only with *intra* frames.

Owing to the fact that we have not changed zero QTCs and T1's, bitrate has increased only slightly. This increase is due to two reasons. One, watermarked reconstructed QTCs are used for prediction of future MBs which results in more residual and hence increase in bitrate. Secondly, absolute value of QTC increases gradually in inverse scan order and entropy coding is designed for this distribution. After WMBit insertion, this order may get disturbed and depends upon the WMBits being embedded. Fig. 3.a and b illustrate the payload for each *intra* frame in *foreman* for QP=18 and QP=36. With the insertion of WMBits, QTC are modified and hence there is a decrease in PSNR. At QP value of '18', higher number of coefficients are watermarked and hence a greater reduction in the PSNR. While at QP value of '36', we

(a)

(b)

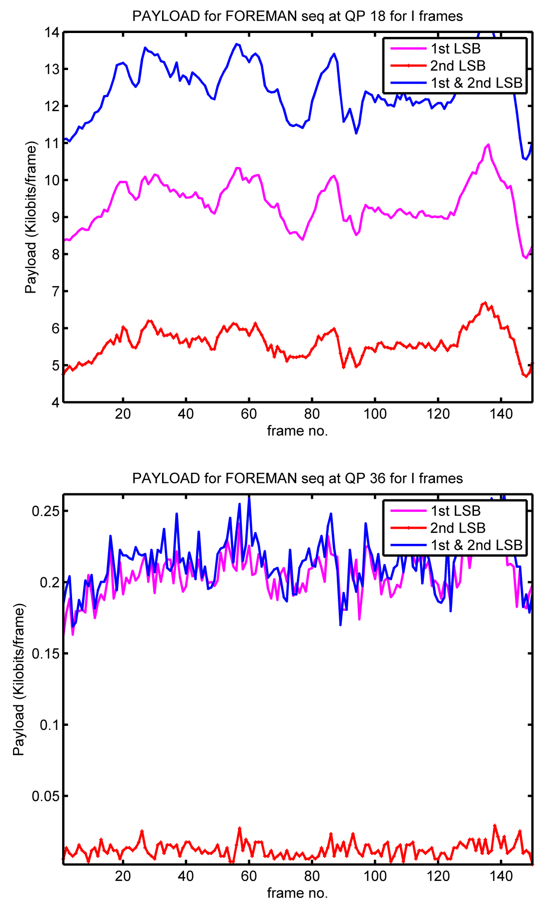


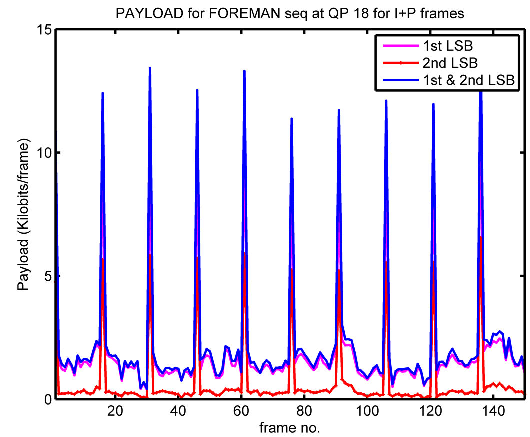
Figure 3: Analysis of payload capability for watermark embedding of *intra* frames in *foreman* for:a) QP=18, b) QP=36.

		Payload Kbits/frame	Frame Size Kbytes	PSNR dB
QP 18 I frames	0	0	2.818	44.876
	LSB1	9.352	2.892	43.800
	LSB2	5.586	2.878	43.605
	LSB1&2	12.452	2.913	42.917
QP 18 P frames	0	0	1.317	44.541
	LSB1	1.378	1.343	44.302
	LSB2	0.280	1.333	44.449
	LSB1&2	1.530	1.354	42.918
QP 18 I + P	0	0	1.417	44.563
	LSB1	1.909	1.446	44.269
	LSB2	0.633	1.436	44.392
	LSB1&2	2.258	1.458	44.144

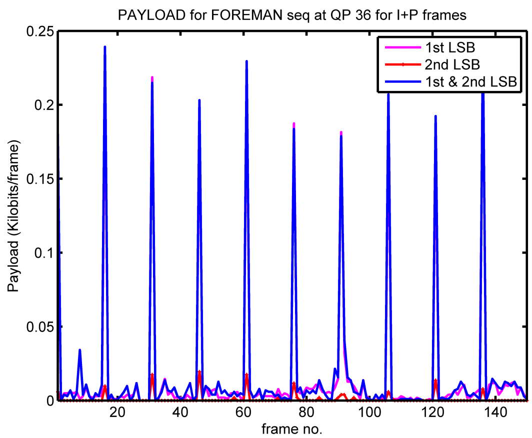
Table 2: With *intra* and *inter* frames for QP=18.

have lesser QTCs to be watermarked, hence less degradation in quality is observed.

For experimental simulation of *intra* & *inter* frames, *intra period* has been set 15. In *inter* frames, we exploit the temporal masking to embed watermark in QTC. The watermark is embedded in those parts of frames which contain motion and texture. After *intra* frame, first few *inter* frames are better



(a)



(b)

Figure 4: Analysis of payload capability for watermark embedding of *intra & inter* frames in *foreman* for: a) QP= 18, b) QP=36.

		Payload Kbits/frame	Bit Rate Kbytes	PSNR dB
QP 36 I frames	0	0	0.376	32.613
	LSB1	0.198	0.380	32.523
	LSB2	0.011	0.376	32.589
	LSB1&2	0.207	0.381	32.520
QP 36 P frames	0	0	0.074	32.353
	LSB1	0.005	0.074	32.315
	LSB2	1×10^{-4}	0.073	32.336
	LSB1&2	0.005	0.074	32.318
QP 36 I + P	0	0	0.094	32.370
	LSB1	0.017	0.094	32.329
	LSB2	8×10^{-4}	0.093	32.353
	LSB1&2	0.019	0.095	32.331

Table 3: With *intra* and *inter* frames for QP=36.

predicted and contains lesser drift errors. Hence watermark insertion affects quality and compression ratio. But after a few *inter* frames, drift errors appear and watermark insertion does not affect much the quality of *inter* frames. On average, after 5 frames, the payload/frame size ratio of *inter* frames is very close to that of *intra* frames. At QP value of 36, lesser QTCs have magnitude above 3 and hence fewer WMBit insertions in 2 LSB mode. Table 2 and 3 show the average change in size of frames for *foreman* sequence at QP value of 18 and 36 for *intra* & *inter* frames. Fig. 4.a and b illustrate the payload for I and P frames in *foreman* for QP=18 and QP=36. On an average basis, the change in file size of all the videos is 3.2% ,2.7% and 2.8% for *intra*, *inter* and *intra* & *inter* respectively with a QP value of 18.

5. CONCLUSION

We have designed and analyzed a new video watermarking scheme for H.264/AVC. Our scheme embeds RD optimized watermark in QTCs for both *intra* and *inter* frames. Our watermark offers consistent payload capability to H.264/AVC standard at different bitrates without adversely affecting the overall bite rate and quality of the video bitstream. Owing to spatial masking, *intra* frames can be used for LSB insertion and for *inter* frames, watermark insertion is done in temporal masking regions. Experimental results have demonstrated that *inter* frames can be equally good for watermark insertion owing to its motion and texture masking.

REFERENCES

[1] Draft itu-t recommendation and final draft international standard of joint video specification (itu-t rec. h.264 iso/iec 14496-10 avc). Technical report, Joint Video Team (JVT),Doc. JVT-G050, March 2003.

[2] A.M. Alattar, E.T. Lin, and M.U. Celik. Digital Watermarking of Low Bit-Rate Advanced Simple Profile MPEG-4 Compressed Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:787–800, 2003.

[3] A. Golikeri, P. Nasiopoulos, and Z.J. Wang. Robust Digital Video Watermarking Scheme for H.264 Advanced Video Coding Standard. *Journal of Electronic Imaging*, 16(4):043008, 2007.

[4] F. Hartung and B. Girod. Watermarking of Uncompressed and Compressed Video. *Signal Processing*, 66(3):283–301, 1998.

[5] H.S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky. Low-complexity transform and quantization in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):598–603, 2003.

[6] B.G. Mobasseri and Y.N. Raikar. Authentication of H.264 Streams by Direct Watermarking of CAVLC Blocks. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, page 65051W. SPIE, 2007.

[7] M. Noorkami and R.M. Mersereau. Digital Video Watermarking in P-Frames With Controlled Video Bit-Rate ncrease. *IEEE Transactions on Information Forensics and Security*, 3(3):441–455, September 2008.

[8] A.B. Watson. DCT Quantization Matrices Visually Optimized for Individual Images. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 1913, pages 202–216, September 1993.

[9] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[10] R.B. Wolfgang, C.I. Podilchuk, and E.J. Delp. Perceptual Watermarks for Digital Images and Video. *Proceedings of the IEEE*, 87(7):1108–1126, Jul 1999.