Considering The Reconstruction Loop For Watermarking Of Intra And Inter Frames Of H.264/Avc
Zafar Shahid, Peter Meuel, Marc Chaumont, William Puech

To cite this version:

HAL Id: lirmm-00416223
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00416223
Submitted on 13 Sep 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 for watermark embedding. In this paper, we analyze watermark embedding in intra as well as in inter and we note that watermark embedding capability of inter is comparable to that of intra. 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. 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. The proposed scheme does not target robustness. Rather the main contribution of our scheme is higher payload as compared to payloads obtained in previous works.

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. 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 MPEG video standards [2, 4]. The purpose of this paper is to investigate the payload capability of intra and inter, since video data consists of intra followed by trail of inters. Challenge lies in the fact that bitrate may rise significantly because of watermark embedding. To overcome this limitation, watermark has been embedded in only those QTCs which have magnitude over a certain threshold.

In Section 2, we present the H.264/AVC video codec and previous related watermarking techniques. We present the proposed algorithm by elaborating the embedding and extraction steps in Section 3. Section 4 contains experimental results for both intra and inter 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
2.1 Overview of H.264/AVC
In this section, an overview of H.264/AVC with an emphasis on integer transform (IT) and quantization is presented.

H.264/AVC [1] has some additional features as compared to previous video standards. In baseline profile of H.264/AVC, it has $4 \times 4$ transform in contrast to $8 \times 8$ 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 [11]. The block diagram of H.264/AVC is shown in Fig. 1.

A macroblock (MB) is divided into 16 blocks of $4 \times 4$ pixels and they are processed one by one. In intra mode, H.264/AVC has three modes, Intra$_4 \times 4$, Intra$_{16} \times 16$ and I_PCM . In Intra$_{16} \times 16$ 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$_{4} \times 4$ mode, each $4 \times 4$ luma block is predicted from top and left pixels of reconstructed $4 \times 4$ 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 $4 \times 4$ block as shown in Fig 1. First of all, it is predicted from its neighboring blocks and we get the residual block:

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

Intra prediction is performed from the reconstructed neighboring pixels where $B_i$ are the reconstructed neighboring blocks. Forward and inverse integer transform $4 \times 4$ matrices $(A,A_{inv})$ are given in [7]. This residual block $E$ is
then transformed using the the forward transform matrix $A$ as $Y = A E A^T$. Scalar multiplication and quantization are defined as:

$$
\hat{Y} = \text{sign} \{Y\} \{ | Y | \otimes A_q + F_q \times 2^{15+E_q} \} > (15+E_q),
$$

where $\hat{Y}$ is quantized transformed coefficient. $A_q$ is the $4 \times 4$ quantization matrix and $E_q$ is the shifting matrix. Both $A_q$ and $E_q$ are indexed by QP. $F_q$ is the quantization rounding factor matrix. Right shift operator is applied to every elements of $4 \times 4$ matrix. This $\hat{Y}$ is entropy coded and sent to the decoder side. On the decoder side, inverse quantization is given by the expression $Y' = \{[\hat{Y} \otimes (B_q << 4)] << E_q + 8 \} > > 4$, where $B_q$ and $E_q$ are the inverse $4 \times 4$ quantization matrix and the shifting factor respectively. $Y'$ is then inverse transformed to get $E' = (A_{wv}Y' A'_{wv} + 32) > > 6$. 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 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 techniques are not viable in the transform domain since they embed watermark drawn from a Gaussian distribution. Golikeri et al. have proposed robust watermarking algorithm for H.264/AVC [3], in which they have used visual models developed by Watson [10] 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, $16 \times 16$ mode, they have also embedded the watermark in Hadamard transform coefficients. Their algorithm is robust but have very small payload. To avoid processing intensive decoding followed by re-encoding along with watermarking, some methods have suggested embedding watermark in entropy coding stage [5, 6, 8, 10]. Such algorithms face two major limitations. First, payload of such algorithms is very less which is up to few bytes per second as explained in [4]. Second, there is a continuous drift which degrades the visual quality significantly. To avoid drift, drift compensation signal should be added to decoder. Noorkami and Merserau have presented a technique to embed watermark in both intra and inter frames [9]. They propose to embed watermark in all 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 embedding of watermark only in non-zero QTCs, their method does not affect the compression efficiency of run-length coding. But performance of context-based adaptive variable length coding (CA VLC) gets affected and as a result, some increase in bitrate is observed, since there are lot of QTCs whose magnitude is 1 and CA VLC encodes trailing ones (T1’s) separately.

### 3. PROPOSED ALGORITHM

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. This threshold 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. We have not targeted robustness here. Rather we have demonstrated the high payload capability of the proposed scheme which is very high as compared to other schemes. Hence, the proposed scheme can be used in application where robustness is not required, e.g., broadcasting and hiding of meta data.

In H.264/AVC, intra prediction is performed in the spatial domain. So even for intra mode, IT is performed on prediction residuals. Since DC QTCs contain most of the energy and embedding watermark in them affects the video quality and the bitrate significantly, we have not embedded watermark in DC QTCs for Intra,$4 \times 4$ mode. While for Intra,$16 \times 16$ mode, we have not modified the Hadamard transform coefficients. We have embedded watermark in LSBs of AC 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 QTCs with magnitude ‘1’ and in CA VLC, they are encoded as T1’s, a separate syntax element. Thus changing of number of T1’s will affect the compression efficiency of CA VLC.

- Finally watermark is embedded in such a fashion that it can be completely extracted on the decoder side.

### 3.1 Embedding process within encoder loop

From equation (2) we apply the embedding with:

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

where $f()$ is the watermarking process, $W$ the watermark signal and $K$ is the optional watermarking key. Watermark embedding can be done in QTC before entropy coding. This embedding creates two problems. Firstly, we start 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 also working on $\hat{Y}_w$.

### 3.2 Watermark aware Rate Distortion

Many encoding parameters like prediction modes, quantization parameter (QP) 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, \( \lambda \) is the Lagrangian parameter 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 QTCs which are then entropy coded. Then, residual is reconstructed by performing inverse quantization and inverse IT to give the reconstructed residual \( E' \). 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 PSNR and bitrate of the picture. Hence, RD optimization should take into account the embedding of watermark. In this case, simplified Lagrangian cost function is \( J_w = D_w + \lambda R_w \), for finding the cost for a specific prediction mode. Instead of QTCs, watermarked QTCs are entropy coded to find the number of bits \( R \) to MB and reconstructed to measure the distortion \( D \). By moving the watermarking process inside the reconstruction loop, it incorporates the best suitable mode for the watermarked blocks.

### 3.3 Embedding strategy

For watermark embedding in QTCs, we developed a strategy to embed it in 1 LSB, 2 LSBs or 1 & 2 LSBs together. For the embedding of 1 WMBit in LSB of \( |QTC| \):

If \( |QTC| \) is less than 2, it will remain unchanged. For \( |QTC| \geq 2 \), output will be either same or will get modified by \( \pm 1 \), depending on whether \( \text{WMBit} \) 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 \( QTC \):

\[
\begin{align*}
\text{if } (|QTC| > 3) \text{ then} & \\
|QTC| &= |QTC| & \text{if } (|QTC| > 3) \text{ then} \\
\text{WMBit} &= \text{WMBit} & \text{if } (|QTC| > 3) \text{ then} \\
|QTC| &= |QTC| & \text{if } (|QTC| > 3) \text{ then}
\end{align*}
\]

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 relatively higher. 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 \( |QTC| \):

\[
\begin{align*}
\text{if } (|QTC| > 3) \text{ then} & \\
|QTC| &= |QTC| & 0x\text{fffffff}; \\
\text{WMBit} &= \text{WMBit} & 0x00000003; \\
|QTC| &= |QTC| & \text{WMBit}.
\end{align*}
\]

3.4 Watermark extraction

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

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

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 example, for 1 LSB watermark extraction, \( g() \) can be given as:

\[
\begin{align*}
\text{if } (|QTC| > 1) \text{ then} & \\
\text{WMBit} &= \text{WMBit} & 0x00000001;
\end{align*}
\]

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 nine 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 \emph{intra} frames, and then with \emph{intra} and \emph{inter} frames for 1 LSB, 2 LSBs and 1 & 2 LSBs embedding modes.
For intra frames we have encoded 150 frames of each sequence. Non-zero QTCs are present in those parts of frames which contain texture and edges. These are spatial masking areas and watermark is embedded in those areas of the frames. To analyze payload simulation, we have used foreman video sequence for the QP values of 18 and 36 as shown in Table 1. At QP value of 18, we have large number of QTCs which can be watermarked and hence payload is high for all the modes. At QP value of 36, we have adequate number of QTCs for 1 LSB mode so we have enough number of WMBits embedded. But payload for 2 LSBs mode is lower as fewer QTCs have magnitude above threshold for this mode. With the embedding of WMBits, QTCs 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 have lesser QTCs to be watermarked, hence less degradation in quality is observed.

<table>
<thead>
<tr>
<th>QP 18</th>
<th>Payload</th>
<th>Frame Size</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>I frames</td>
<td>Kbits/frame</td>
<td>Kbytes</td>
<td>dB</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2.815</td>
<td>44.883</td>
</tr>
<tr>
<td>LSB1</td>
<td>9.375</td>
<td>2.889</td>
<td>43.801</td>
</tr>
<tr>
<td>LSB2</td>
<td>5.605</td>
<td>2.875</td>
<td>43.605</td>
</tr>
<tr>
<td>LSB1&amp;2</td>
<td>12.484</td>
<td>2.909</td>
<td>42.928</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QP 36</th>
<th>Payload</th>
<th>Frame Size</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>I frames</td>
<td>Kbits/frame</td>
<td>Kbytes</td>
<td>dB</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.377</td>
<td>32.628</td>
</tr>
<tr>
<td>LSB1</td>
<td>0.206</td>
<td>0.381</td>
<td>32.536</td>
</tr>
<tr>
<td>LSB2</td>
<td>0.012</td>
<td>0.377</td>
<td>32.612</td>
</tr>
<tr>
<td>LSB1&amp;2</td>
<td>0.214</td>
<td>0.381</td>
<td>32.526</td>
</tr>
</tbody>
</table>

Table 1: Overall analysis of watermark embedding in intra frames for foreman sequence.

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 WM-Bit embedding, this order may get disturbed and depends upon the WMBits being embedded. Fig. 3.a and 3.b illustrate the payload for each intra frame in foreman for QP=18 and QP=36.

For experimental simulation of intra & inter frames, intra period has been set 15. Non-zero QTCs are found in the parts of frames containing motion and texture. Watermark is embedded in those areas as these are temporal masking areas in inter frames. After intra frame, first few inter frames are better predicted and contain lesser residual errors. Hence watermark embedding affects quality and compression ratio, but after a few inter frames, residual errors increase and watermark embedding does not affect so 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. Table 2 and 3 show the average change in frame size for foreman sequence at QP value of 18 and 36 for intra & inter frames. Fig. 4.a and 4.b illustrate the payload for I and P frames in foreman for QP values of 18 and 36. On average, the change in file size is 3.2%, 2.7% and 2.8% for intra, inter and intra & inter respectively for a QP value of 18.

<table>
<thead>
<tr>
<th>QP 18</th>
<th>Payload</th>
<th>Frame Size</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>I frames</td>
<td>Kbits/frame</td>
<td>Kbytes</td>
<td>dB</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2.818</td>
<td>44.876</td>
</tr>
<tr>
<td>LSB1</td>
<td>9.352</td>
<td>2.892</td>
<td>43.800</td>
</tr>
<tr>
<td>LSB2</td>
<td>5.586</td>
<td>2.878</td>
<td>43.605</td>
</tr>
<tr>
<td>LSB1&amp;2</td>
<td>12.452</td>
<td>2.913</td>
<td>42.917</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QP 18</th>
<th>Payload</th>
<th>Frame Size</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>P frames</td>
<td>Kbits/frame</td>
<td>Kbytes</td>
<td>dB</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1.317</td>
<td>44.541</td>
</tr>
<tr>
<td>LSB1</td>
<td>1.378</td>
<td>1.343</td>
<td>44.302</td>
</tr>
<tr>
<td>LSB2</td>
<td>0.280</td>
<td>1.333</td>
<td>44.449</td>
</tr>
<tr>
<td>LSB1&amp;2</td>
<td>1.530</td>
<td>1.354</td>
<td>44.230</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QP 18</th>
<th>Payload</th>
<th>Frame Size</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>I + P</td>
<td>Kbits/frame</td>
<td>Kbytes</td>
<td>dB</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1.417</td>
<td>44.563</td>
</tr>
<tr>
<td>LSB1</td>
<td>1.909</td>
<td>1.446</td>
<td>44.269</td>
</tr>
<tr>
<td>LSB2</td>
<td>0.633</td>
<td>1.436</td>
<td>44.392</td>
</tr>
<tr>
<td>LSB1&amp;2</td>
<td>2.258</td>
<td>1.458</td>
<td>44.144</td>
</tr>
</tbody>
</table>

Table 2: Overall analysis of watermark embedding in intra and inter frames for foreman sequence at QP=18.

Between 1 LSB and 2 LSBs embedding modes, 1 LSB performs better having higher payloads and minimum increase in bitrate. In 2 LSBs mode, 2 bits are embedded at the same time and thus magnitude of compromise is higher. 2 LSBs mode should be used in combination with 1 LSB mode as we have done in 1 & 2 LSBs embedding mode to get higher payload. One can note that just like bitrate, pay-
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 bit rate and quality of the video bitstream. Intra and inter frames can be used for LSB embedding by taking into account the reconstruction loop. Experimental results have demonstrated that inter frames can be equally good for watermark embedding owing to its motion and texture masking.

Acknowledgment

This work is supported by VOODDO (2008-2011) project of ANR and the region of Languedoc Roussillon, France.

REFERENCES


