Considering the reconstruction loop for data hiding of intra- and inter-frames of H.264/AVC

Zafar Shahid, Marc Chaumont, William Puech

SIViP (2013) 7:75–93
DOI 10.1007/s11760-011-0225-9

ORIGINAL PAPER

Considering the reconstruction loop for data hiding of intra- and inter-frames of H.264/AVC

Zafar Shahid · Marc Chaumont · William Puech

Received: 7 April 2010 / Revised: 26 February 2011 / Accepted: 8 March 2011 / Published online: 5 April 2011

Abstract This paper presents and analyzes a new approach to data hiding that embeds in both the *intra*- and *inter*-frames from the H.264/AVC video codec. Most of the current video data hiding algorithms take into account only the *intra*-frames for message embedding. This may be attributed to the perception that *inter*-frames are highly compressed due to the motion compensation, and any embedding message inside these may adversely affect the compression efficiency significantly. Payload of the *inter*-frames is also thought to be less, compared with the *intra*-frames, because of the lesser residual data. We analyze data hiding in both *intra*- and *inter*-frames over a wide range of QP values and observe that the payload of the *inter* is comparable with that of the *intra*-frames. Message embedding, in only those non-zero quantized transform coefficients (QTCs) which are above a specific threshold, enables us to detect and extract the message on the decoding side. There is no significant effect on the overall bitrate and PSNR of the video bitstream because instead of embedding message in the compressed bitstream, we have embedded it during the encoding process by taking into account the reconstruction loop. For the non-zero QTCs, in the case of *intra*-frames, we benefit from the spatial masking, while in the case of *inter*-frames, we exploit the motion and texture masking. We can notice that the data hiding is done during the compression process and the proposed scheme takes into account the reconstruction loop. The proposed scheme does not target robustness and the obtained payload is higher, with a better trade-off in terms of quality and bitrate, as compared with previous works.

Keywords Video data hiding · Non-zero quantized transform coefficients · H.264/AVC · Reconstruction loop · Intra- and inter-frames

1 Introduction

Many multimedia applications have emerged in the last decade thanks to the rapid growth in processing powers and network bandwidths. The relative ease, with which digital data can be copied or modified, necessitates its proper protection and authentication. Digital video watermarking has emerged as an important research field to protect the copyrighted multimedia data. Watermarking, steganography, and more generally data hiding are used in many applications for owner identification, copyright protection, integrity, and metadata embedding. For a video codec, data hiding can be carried out in either spatial or frequency domain. Data embedded in spatial domain can be lost because of the lossy stage of quantization. In the frequency domain, data hiding is done normally in the QTCs. Normally, for large videos with real-time constraints, the data hiding process is made part of the video encoder. In this context, few specific methods have been developed for the MPEG video standards [1,13]. The purpose of this paper is to investigate the payload capacity of *intra*- and *inter*-frames, since a typical video consists of an *intra* followed by a trail of *inters*. Challenge lies in the fact that the bitrate may rise significantly because of the message embedding. To overcome this limitation, the message has been embedded in only those QTCs which have a magnitude beyond a certain threshold.

Z. Shahid · M. Chaumont · W. Puech (✉)
LIRMM Laboratory, UMR 5506 CNRS, University of Montpellier II, 161, rue Ada, 34392 Montpellier Cedex 05, France
e-mail: william.puech@lirmm.fr

Z. Shahid
e-mail: zafar.shahid@lirmm.fr

M. Chaumont
e-mail: marc.chaumont@lirmm.fr

In H.264/AVC video codec, both the intra- and inter-predictions should be taken into account by embedding the hidden message inside the reconstruction loop. It is crucial, especially while streaming over heterogeneous networks, to keep the bitrate intact. Hence, bitrate escalation, due to the message embedding, must be taken into account by the rate distortion module.

The rest of the paper is organized as follows. In Sect. 2, first, we present the H.264/AVC video codec, with its integer transform (IT) and the quantization process, followed by an overview of the previous watermarking and data hiding techniques related to this video standard. We present, in Sect. 3, the proposed method by elaborating its embedding and extraction steps while taking into account the reconstruction loop. Sect. 4 contains the experimental results and a performance analysis of both *intra*- and *inter*-frames after embedding in more than one least significant bits (LSBs). In the said section, we also present a comparison of message embedding in a given video bitstream inside and outside the reconstruction loop. Finally, in Sect. 5, we present some concluding remarks about the proposed method.

2 H.264/AVC data hiding, challenges and prospects

Since significant changes have been incorporated in the H.264/AVC standard as compared to the previous video coding standards. An overview of H.264/AVC, with an emphasis on transform and quantization, is presented in Sect. 2.1. It is followed, in Sect. 2.2, by an overview of the previous watermarking and data hiding techniques already proposed in the literature for H.264/AVC. We have used capital letters to represent matrices e.g., A, Y, W and small letters along with index to represent the elements of matrices e.g., $x(i, j)$ represents jth element in ith row of matrix X.

2.1 Overview of H.264/AVC

The H.264/AVC standard [12] has some additional features as compared to previous video standards. The *baseline* standard has a 4 × 4 transform in contrast to 8 × 8 transform of the previous standards. DCT transform has been replaced by the integer transform (IT), which can be implemented by just additions and shifts in 16-bit arithmetic without any multiplication and hence requires lesser number of computations. The 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 at outperforming the previous video coding standards [36]. The block diagram of H.264/AVC is shown in Fig. 1.

The 4 × 4 IT has two main advantages. Firstly, it can be exhaustively implemented with simple 16-bit additions and shifts. Secondly, in contrast to floating point arithmetic, which gives different results on different platforms, there is no problem of mismatch on the encoder and decoder side for the integer arithmetic. A macro-block (MB) is divided into 16 blocks of 4 × 4 pixels which are processed one by one.

In the *intra*-mode, H.264/AVC has three alternatives, namely, *Intra*_4 × 4, *Intra*_16 × 16, and *I_PCM*. In *Intra*_16 × 16 mode, Hadamard transform is additionally employed to encode the DC coefficients. In the *Intra*_16 × 16 mode, the entire MB is predicted from top and left neighboring pixels and has 4 modes namely *horizontal*, *vertical*, *DC*, and *plane* modes. In the *Intra*_4 × 4 mode, each 4 × 4 luma block is predicted from top and left pixels of the reconstructed 4 × 4 neighbors. This alternative has nine different prediction modes. The *I_PCM* mode is used to limit the maximum size of the encoded block and is directly entropy encoded by skipping the transform and quantization stages. The scanning of these 4 × 4 blocks, inside MB, is not in a raster scan fashion, as illustrated with the help of numbers in Fig. 2. In the case of *Intra*_16 × 16 mode, Hadamard transform coefficients are sent first.

Transform and quantization process are embedded with each other to save the processing power and to avoid multiplications. Let a 4 × 4 block is defined as $X = \{x(i, j)|i, j\epsilon\{0, 3\}\}$ as shown in Fig. 1. $x(i, j)$ is predicted from its neighboring blocks, and we get the residual block:

$$e(i, j) = P(x(i, j), b_1(i, j), b_2(i, j), b_3(i, j), b_4(i, j)), \tag{1}$$

where $b_k(i, j)$ are the pixels from the reconstructed top and left blocks from intra-prediction, and $P(.)$ is the prediction function. For example, for a vertical prediction mode, the prediction will be performed from top block as $P(x, a, b, c, d) = x - a$, where a is the reconstructed block at top. From [25], the forward and inverse IT 4 × 4 matrices (A, A_{inv}) are as follows:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} A_{inv} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 1 & -1/2 \end{bmatrix}. \tag{2}$$

The residual block E is then transformed using the following equation:

$$Y = AEA^T, \tag{3}$$

where $E = \{e(i, j)|i, j\epsilon\{0, 3\}\}$ is in the spatial domain, and $Y = \{y(i, j)|i, j\epsilon\{0, 3\}\}$ is in the frequency domain. Scalar multiplication and quantization are defined as:

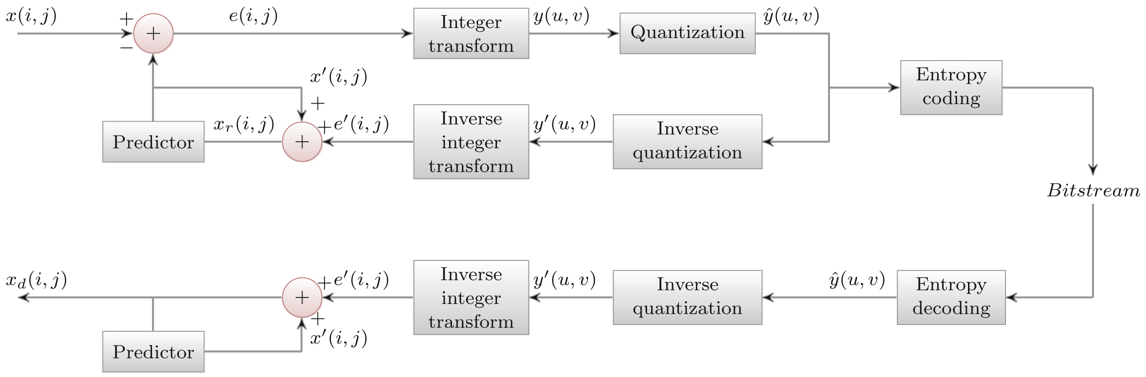


Fig. 1 Detailed block diagram explaining the prediction, the transform and the quantization steps in the H.264/AVC

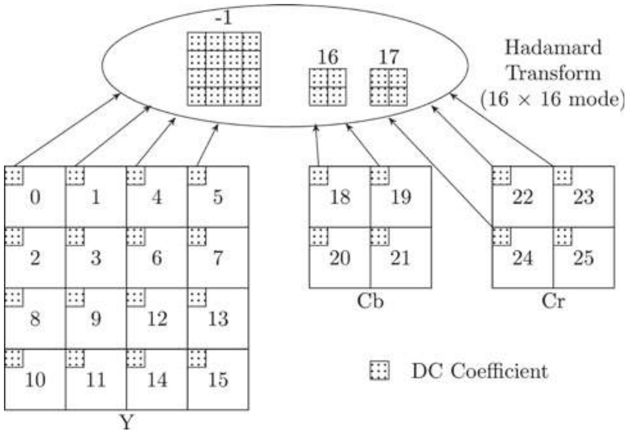


Fig. 2 Order of transmission of the luma and the chroma $Intra\_4 \times 4$ blocks inside MB

$$\hat{y}(u,v) = sign\{y(u,v)\}[(|\ y(u,v)\ | \times Aq(u,v) + Fq(u,v) \times 2^{15+Eq(u,v)})/2^{(15+Eq(u,v))}], \quad (4)$$

where $\hat{y}(u,v)$ is a QTC, $Aq(u,v)$ is the value from the 4×4 quantization matrix, and $Eq(u,v)$ is the shifting value from the shifting matrix. Both $Aq(u,v)$ and $Eq(u,v)$ are indexed by QP. $Fq(u,v)$ is the rounding off factor from the quantization rounding of factor matrix. This $\hat{y}(u,v)$ is entropy coded and sent to the decoder side.

On the decoder side, inverse quantization is carried out according to the expression:

$$y'(u,v) = \{[(\hat{y}(u,v) \times (Bq(u,v) \times 2^4)) \times 2^{Eq(u,v)}] + 2^3\}/2^4,$$

where $Bq(u,v)$ and $Eq(u,v)$ are the values from the inverse 4×4 quantization matrix and the shifting factor, respectively. $y'(u,v)$ is then inverse transformed to get $E' = (A_{inv} Y' A_{inv}^T + 2^5)/2^6$. The decoded residual signal $e'(i,j)$ is then added to the predicted signal to reconstruct the original signal back.

2.2 Previous work on video watermarking and data hiding

Many digital watermarking and data hiding schemes have been proposed in the literature for both image and video data. Data hiding methods can be classified into three broad categories namely robust [6,20,33,38,39], semi-fragile [14] and fragile [22]. Different watermarking and data hiding techniques offer various combinations of rate, distortion, and robustness. For each application, a particular watermark algorithm can be selected depending on its requirements. For example, applications for copyright protection would require using a robust watermark, while applications for proving integrity would employ a fragile or semi-fragile watermark.

As far as standard video codecs are concerned, five encoding stages can be identified where embedding can take place, namely the pre-compression stage, the video codec structure, the transform stage, the quantization step, and the bitstream as illustrated in Fig. 3. The embedding is primarily motivated by the goal of integrating data hiding and compression to reduce overall real-time video processing complexity.

In the pre-compression stage, marked as stage 1 in Fig. 3, message embedding is performed before the compression process [5,8,9,15,23,26]. Embedding can be performed either in the pixel domain or in some transform domain e.g., DCT, DFT, DWT. Temporal aspect of the video can also be exploited for watermark embedding by taking into account multiple frames at the same time [3,30]. In spatial domain, LSB modification is a very simple method to embed hidden message into the cover object [17]. This method may survive against attacks such as cropping but any addition of noise or lossy compression is likely to defeat the message extraction. In [17], a method has been proposed to use PRNG to decide the pixels for LSB substitution. This may improve the security but still vulnerable to the substitution of the LSB(s) with a constant value. LSB modification may be imperceptible but statistically still discernible. In [2], an upper bound for the LSB payload has been defined so that it remains statistically invisible.

In [4], the transform domain has been employed for LSB embedding. Here, the signature image is first quantized using vector quantization in order to hide larger message. The bit-by-bit message is then directly embedded into LSB of DCT coefficients of the image. Using the embedding replicate of

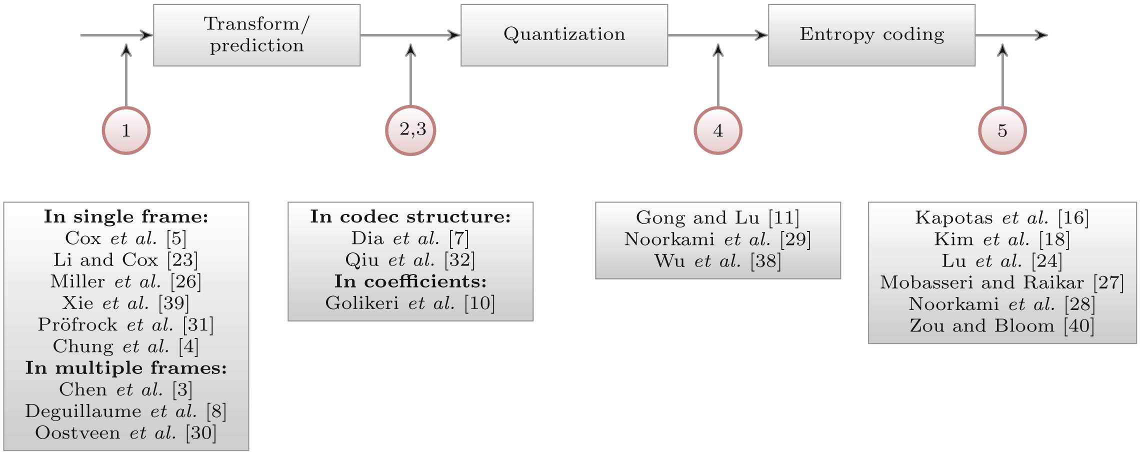


Fig. 3 Classification of the watermarking schemes on the basis of the working domain: *1* pre-compression, *2* video codec structure, *3* transform domain, *4* Quantized transform domain, *5* bitstream

the message provides further robustness against the signal processing attacks.

In [31], Pröfrock et al. have presented a watermarking technique in the spatial domain, which is robust to H.264/AVC video compression. Hidden message is embedded in the perceptually significant parts of the video in an imperceptible manner, by changing the spatial position of the object borders. Borders are defined by new normed center of gravity (NCG). Influence of lossy compression on NCGs is predicted, and watermark is embedded with enough robustness to compensate the lossy compression. A geometric warping process is proposed to quantize the NCG and embeds the watermark payload with a defined robustness. Xie et al. [39] have also proposed robust watermarking based on the on-off keying scheme which uses the DWT transform to embed the watermark.

The video codec structure is the second candidate domain for data hiding. Some researchers have proposed to embed the message in motion vectors [7,32]. In [21], Li et al. propose to perform robust watermarking for H.264/AVC by embedding the hidden message in a new syntax element named reference index. They also modify the current block to improve the robustness of the scheme by a geometric method with the least degradation in the video quality. These video watermarking techniques are vulnerable to re-encoding and conversion to other video codecs.

Hidden message can also be embedded in the transformed coefficients before quantization, as proposed by Golikeri et al. [10]. This kind of approach is illustrated by stage 3 in Fig. 3. They have used the visual models, developed by Watson [35], to choose the coefficients to be watermarked based on their frequency sensitivity, luminance masking, and contrast masking.

Some researchers have proposed algorithms to embed hidden message in the QTCs of H.264/AVC, as shown by stage 4 in Fig. 3. For example, Noorkami and Merserau [29] have presented a technique to embed message in both *intra*- and *inter*-frames in all the non-zero QTCs. They claim that visual quality of *inter*-frames is not compromised even if we embed message in all the non-zero QTCs. Owing to the embedding of the message, only in the non-zero QTCs, their method does not affect the compression efficiency of the run-length coding. The performance of context-based adaptive variable length coding (CAVLC), however, gets affected, and as a result, a controlled increase in the bitrate is eventually observed, since there are a lot of QTCs whose magnitude is 1 and CAVLC encodes *trailing ones* (T1's) separately. In [11], Gong and Lu embedded watermarks in the H.264/AVC video by modifying the quantized DC coefficients in the *luma* residual blocks. To increase the robustness while maintaining the perceptual quality of the video, a texture-masking-based perceptual model is used to adaptively choose the watermark strength for each block. To eliminate the effects of drift, a drift compensation algorithm is proposed which adds the drift compensation signal before embedding the watermark bit.

To avoid processing intensive decoding followed by re-encoding along with watermarking, some methods have suggested embedding the message into the compressed bitstream [18,24,27,28,40]. Kim et al. [18] suggest to hide the message in the sign bit of the trailing ones in CAVLC of H.264/AVC. Bitrate of the watermarked video remains exactly the same with the resultant PSNR greater than 43 dB. In [27], authentication of the H.264/AVC is performed by the direct watermarking of CAVLC codes. Zou and Bloom [40] have proposed to perform direct replacement of CAVLC. Kapotas et al. [16] have presented a data hiding method in H.264 streams for fragile watermarking. It takes advantage of the different block sizes used by the H.264 encoder during the inter-prediction stage, in order to hide the desired data.

The message can be extracted directly from the encoded stream without any need of the original host video. This approach can be mainly used for content-based authentication. Such algorithms face two major limitations. First, payload of such algorithms is very low—of the order of a few bytes per second [13]. Second, there is a continuous drift that degrades the visual quality significantly.

3 The proposed algorithm

In this paper, we have used LSB modification approach in the DCT domain and the hidden message is not embedded in all the non-zero QTCs. Rather, we have embedded the message in only those QTCs which are above a certain threshold. The threshold value depends on the number of message bits being embedded. This offers two advantages. First, it makes it possible to extract the message on the decoder side. Second, it does not affect the compression efficiency of the entropy coding engine significantly. We have not targeted robustness here. Rather, we have demonstrated the high payload capability of the proposed scheme which is very high as compared with other schemes. Hence, the proposed scheme can be used in application where robustness is not required, e.g., broadcasting and hiding of metadata.

In H.264/AVC, intra-prediction is performed in the spatial domain. Hence, even for the *intra*-mode, the transform is performed on prediction residuals. In contrast to previous methods, which embed hidden message in the DC coefficients, we have embedded the message in all those non-zeros QTCs having magnitude above a certain threshold, with the exception of the DC coefficients. Our algorithm is not robust but has a very high payload. For $Intra\_4 \times 4$ mode, we have not embedded message in the DC QTCs, while for $Intra\_16 \times 16$ mode, we have not modified the Hadamard transform coefficients either, since DC QTCs contain most of the energy and embedding message in these may affect the video quality and the bitrate significantly. We have embedded the message in the LSBs of QTCs keeping in view the following points:

- QTC, which we want to use for data hiding, 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 used for data hiding 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, the message is embedded in such a fashion that it can be completely extracted on the decoder side.

3.1 The watermark embedding

The embedding process is performed on QTCs of Eq. 4 as:

$$\hat{y}_w(u, v) = f(\hat{y}(u, v), M, [K]), \tag{5}$$

where $f()$ is the data hiding process, M is the hidden message and K is an optional key.

3.1.1 Analysis of embedding after the encoding loop

Message embedding can be done in QTC before the entropy coding, as shown in Fig. 4. It is analogous to embedding the message in a compressed bitstream. This includes two data hiding approaches. The first approach embeds the message in the VLC domain, and the bitstream needs only be entropy decoded to use this approach e.g., as proposed by Lu et al. [24]. Another approach embeds the message in DCT domain, and for this approach, bitstream has to be entropy decoded and inverse quantized. An example of this approach is differential energy watermarking scheme proposed by Langelaar et al. [19].

Embedding the message after the reconstruction loop creates two problems. Firstly, we start reconstruction on the encoder side with QTC $\hat{y}(u, v)$, while on the decoder side we start decoding with watermarked QTC $\hat{y}_w(u, v)$. This may result in a mismatch on the decoder side, which may keep on increasing because of the prediction process. Because of this mismatch, the difference in PSNR may be considerable, even for *intra*-frames, let alone the *inter*-frames. Secondly, the rate distortion (RD) bit allocation algorithm works in the quantization module, and any change in bitrate/quality tradeoff, because of the watermarking of QTCs, is not taken into account.

3.1.2 Embedding within the encoding loop

To solve both the aforementioned problems, message embedding should be performed inside the reconstruction loop as shown in Fig. 5. In this case, we have the same watermarked QTC $\hat{y}_w(u, v)$ on both encoder and decoder side for prediction, and the RD bit allocation algorithm is working on $\hat{y}_w(u, v)$ for both *intra*- and *inter*-frames. In the next section, we present the data embedding while taking into account the reconstruction loop.

3.2 Hidden message—aware rate distortion

Many encoding parameters, like the prediction modes, the quantization parameter (QP) value and the motion vectors are adjusted in the encoding process based on the video content and the required quality. Since a video data are very diverse in nature, both spatially and temporally, these parameters vary from scene to scene. The bit allocation algorithms are used to

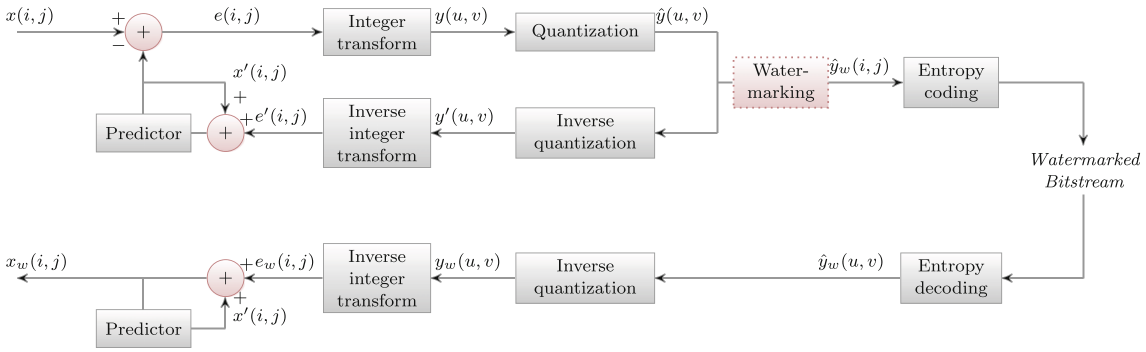


Fig. 4 Block diagram of the H.264/AVC along with the data hiding module outside the reconstruction loop

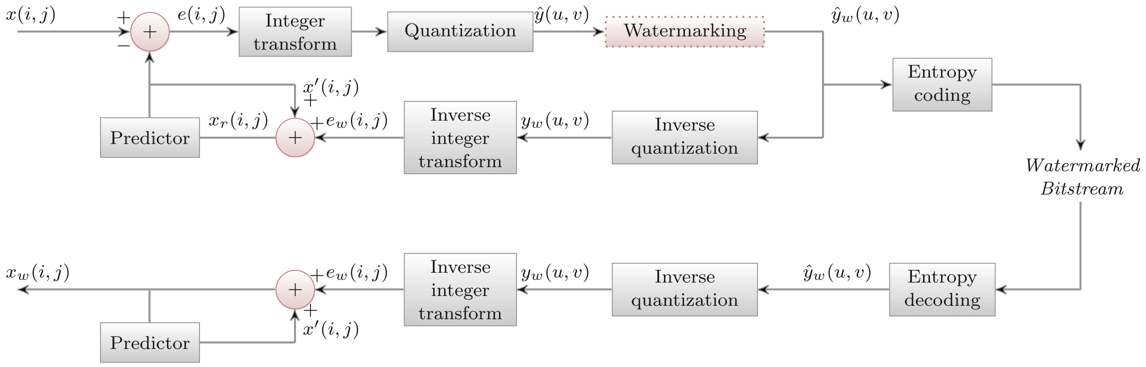


Fig. 5 The proposed data hiding method inside the reconstruction loop

find the best-suited values of these parameters to achieve the trade-off between bitrate and quality. For RD, the 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, which depends on the QP value, and R is the number of bits to encode a MB. In H.264/AVC, several modes are supported to encode a MB as *intra* or *inter*, as shown in Fig. 6. To obtain the cost J, for a specific prediction mode P, we first predict the MB for that mode to get the residual E. We then apply IT on the residual E, followed by quantization with some QP value, to get the QTCs which are then entropy coded. The number of bits, R, consists of MB header bits and data bits. The residual is, then, reconstructed by performing inverse quantization and inverse IT to give the reconstructed residual E'. Generally, the distortion, D, may be the sum of absolute differences (SAD), the sum of squared differences (SSD), or the sum of absolute transformed differences (SATD) between E and E'. Thus, we end up with the cost J for encoding this MB in the prediction mode, P. In a similar fashion, we find the 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 message in a video bitstream affects quality of the picture. It also affects the bitrate because this frame is used for the prediction after the reconstruction. Hence, RD optimization should take into account the embedding of the hidden message 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 to get QTC_w, which are then entropy coded to find the number of bits R_w to encode MB and reconstructed to measure the distortion D_w. By moving the message-embedding process to the inside of the reconstruction loop constitutes the best suitable mode for the watermarked blocks.

3.3 The embedding strategy

For message embedding in the video bitstream, we developed a strategy to embed message in the 1 LSB, 2 LSBs, or ‘1 & 2’ LSBs together. For the n LSB mode, the hidden message is embedded in a QTC in n LSBs if its magnitude is greater than $2^n - 1$. Owing to this threshold, detection and extraction of the message are performed on the decoder side. Algorithm 1 describes the embedding of 1 watermark bit (WMBit) in the LSB of $|QTC|$. Here, the threshold value is 1. If $|QTC|$ is less than or equal to 1, it will remain unchanged. For $|QTC| \geq 2$, output will either be the same or will get modified by ±1, depending on whether the WMBit is ‘0’ or ‘1’.

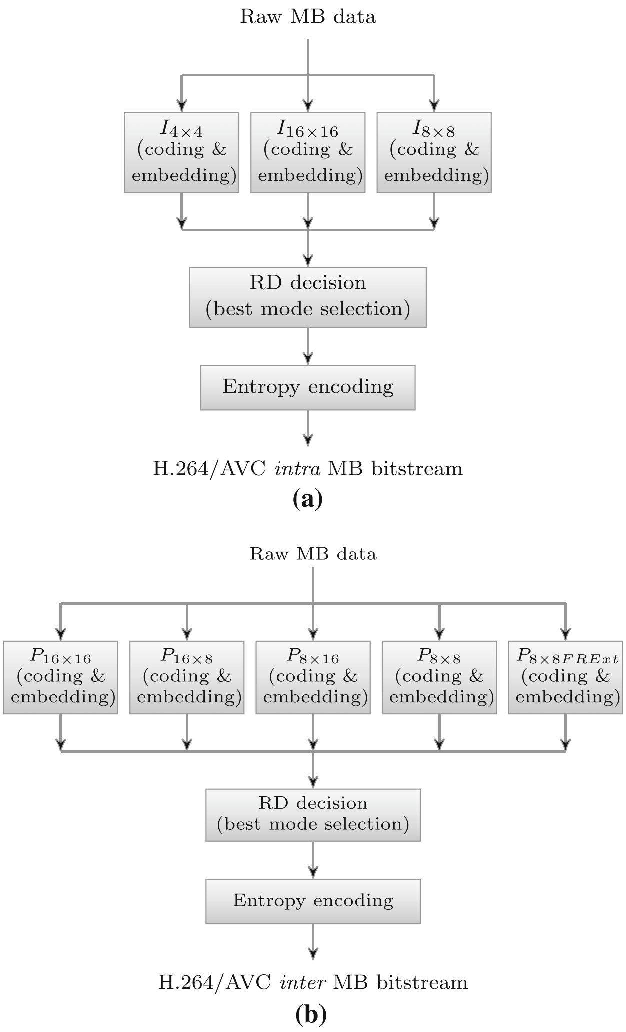

Raw MB data
$I_{4\times4}$ (coding & embedding)
$I_{16\times16}$ (coding & embedding)
$I_{8\times8}$ (coding & embedding)
RD decision (best mode selection)
Entropy encoding
H.264/AVC *intra* MB bitstream
(a)
Raw MB data
$P_{16\times16}$ (coding & embedding)
$P_{16\times8}$ (coding & embedding)
$P_{8\times16}$ (coding & embedding)
$P_{8\times8}$ (coding & embedding)
$P_{8\times8FRExt}$ (coding & embedding)
RD decision (best mode selection)
Entropy encoding
H.264/AVC *inter* MB bitstream
(b)

Fig. 6 Different prediction modes in the H.264/AVC for: **a** the *intra*-MBs **b** the *inter*-MBs

In this case, we have a 50% probability that the coefficient will remain unchanged even after being watermarked.

Algorithm 1 The embedding strategy in the 1 LSB.

```
if $|QTC| > 1$ then
    $|QTC_w| \leftarrow |QTC| - |QTC|\ mod\ 2 + WMBit$
end if
end
```

Similarly, Algorithm 2 outlines the embedding of 2 bits in 2 LSBs of a QTC. By keeping the threshold more than ‘3’, we can extract the hidden message on the decoder side successfully. QTC will remain unchanged if $|QTC| < 4$, otherwise 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.

Algorithm 2 The embedding strategy in the 2 LSBs.

```
if $|QTC| > 3$ then
    $|QTC_w| \leftarrow |QTC| - |QTC|\ mod\ 4 + WMBits$
end if
end
```

We can also perform a ‘1 & 2’ LSBs embedding together. In this case, we embed message in 0, ‘1 & 2’ LSBs depending on value of $|QTC|$, as shown in Algorithm 3. So we embed 2 WMBits if $|QTC|$ is high enough or 1 WMBit if $|QTC| > 1$.

Algorithm 3 The embedding strategy in the ‘1 & 2’ LSBs.

```
if $|QTC| > 3$ then
    $|QTCw| \leftarrow |QTC| - |QTC|\ mod\ 4 + WMBits$
else
    if $|QTC| > 1$ then
        $|QTCw| \leftarrow |QTC| - |QTC|\ mod\ 2 + WMBit$
    end if
end if
end
```

3.4 The hidden message extraction

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

$$M = g(\hat{y}_w(u, v), [K]), \tag{6}$$

where $g()$ is the data hiding detection/extraction process, $\hat{y}_w(u, v)$ is the watermarked QTC, and K an optional key required for extraction. When using ‘1 & 2’ LSBs watermark extraction, $g()$ can be given as shown in Algorithm 4.

Algorithm 4 The extraction strategy using the ‘1 & 2’ LSBs.

```
if $|QTC| > 3$ then
    $WMBits \leftarrow |QTC_w|\ mod\ 4$
else
    if $|QTC| > 1$ then
        $WMBit \leftarrow |QTC_w|\ mod\ 2$
    end if
end if
end
```

4 Experimental results

For experimental simulations, we have used the reference implementation of H.264[1] and applied our method on 150 frames of each of the nine selected standard video sequences

[1] We have used reference software JSVM 10.2 in AVC mode.

in the CIF format. Each of them represents different combinations of motion (fast/slow and pan/zoom/rotation), color (bright/dull), contrast (high/low), and objects (vehicle, buildings, people). The video sequences ‘bus’, ‘city’, and ‘foreman’ contain camera motion, while ‘football’ and ‘soccer’ contain camera panning and zooming along with object motion and texture in the background. The video sequences ‘harbor’ and ‘ice’ contain high luminance images with smooth motion. ‘Mobile’ sequence contains a still complex background and motion in the foreground. In case of the *intra*- and *inter*-sequences, *intra-period* has been set to 15 for all the simulations. In these simulations, the hidden message, being embedded, is of noise type and has been generated using a pseudorandom number generator.

Peak signal to noise ratio (PSNR) is a widely used objective video quality metric. However, it does not perfectly correlate with a perceived visual quality due to the non-linear behavior of the human visual system (HVS). Structural similarity index (SSIM) [34] takes into account the structural distortion measurement, since the HVS is highly specialized in extracting the structural information from the viewing field. SSIM has a better correlation to the subjective impression. SSIM ranges from −1 to 1. SSIM is 1 when both the images are the same. To present the visual comparison of original and watermarked video sequences, both PSNR and SSIM values are presented.

The detailed results for both *intra*- and *inter*-frames are explained in Sects. 4.1 and 4.2, respectively. They also contain the comparison of our scheme with: (1) the outside loop embedding and (2) the message embedding in LSBs of all the QTCs. A comparative analysis of our proposed method with other recent techniques has been presented in Sect. 4.3.

4.1 Analysis of *intra*-frames

In *intra*-frames, non-zero QTCs are present in those parts which contain texture and edges. These are the spatial masking areas, and the hidden message is naturally embedded in these areas of *intra*-frames. To analyze the effect of message embedding on payload, bitrate and PSNR, the video sequences have been encoded at QP values of 18 and 36.

Figure 7a, b illustrate the payload for each *intra*-frame in the *foreman* for the QP values of 18 and 36 for all of the three data hiding modes, namely the 1 LSB, the 2 LSBs and the 1 & 2 LSBs. At a QP value of 18, we have a large number of QTCs which can be watermarked and hence payload is high for all the modes. At a QP value of 36, we have an adequate number of QTCs for the 1 LSB mode and hence enough number of WMBits to embed. But the payload for the 2 LSBs mode is, however, lower since fewer QTCs have magnitudes above the threshold for this mode. Figure 8a, b show the effect of message embedding on the bitrate. Owing to the fact that we have neither modified the QTCs with magnitude ‘0’ and T1’s, the bitrate has increased only slightly. This increase in bitrate is due to two reasons. One, watermarked reconstructed QTCs are used for the prediction of the future MBs, which results in more residuals and hence increase in the bitrate. Two, the absolute value of QTCs increases gradually in the inverse scan order, and the entropy coding is designed for this distribution. After the WMBit embedding, this order may get disturbed and depends on the WMBits being embedded. With the embedding of the WMBits, the QTCs are modified and hence there is a decrease in the PSNR as shown in Fig. 9a, b. At the QP value of 18, higher number of coefficients are watermarked and hence a greater reduction in the PSNR is observed. While at the QP value of ‘36’, we have lesser QTCs to be watermarked, hence less degradation in the quality is observed. At the QP value of 36, few QTCs have magnitude above threshold, for the 2 LSBs embedding, and very few WMBits can be embedded in this mode. But we have adequate number of QTCs with magnitude greater than 1,

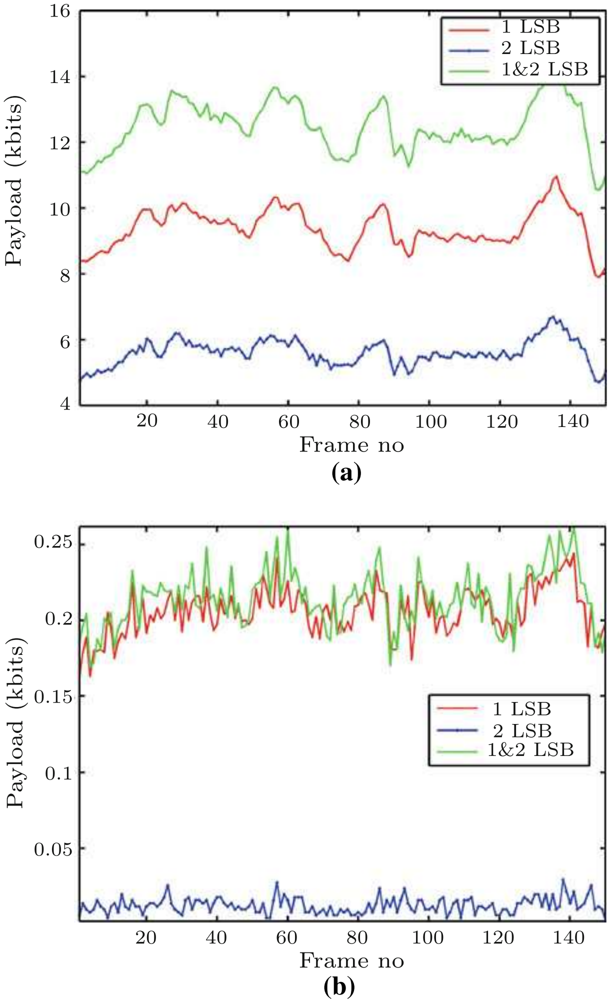


Fig. 7 Analysis of the payload capability for the hidden message embedding of the *intra*-frames in *foreman* for the QP value of: **a** 18 **b** 36

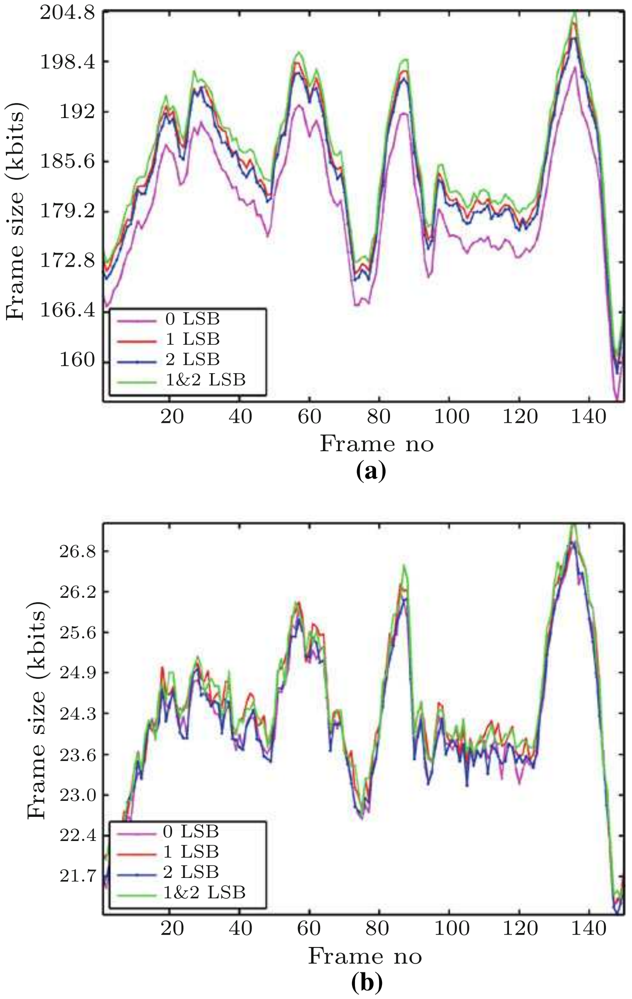


Fig. 8 Analysis of the bitrate variation for the hidden message embedding of the *intra*-frames in *foreman* for the QP values of: **a** 18 **b** 36

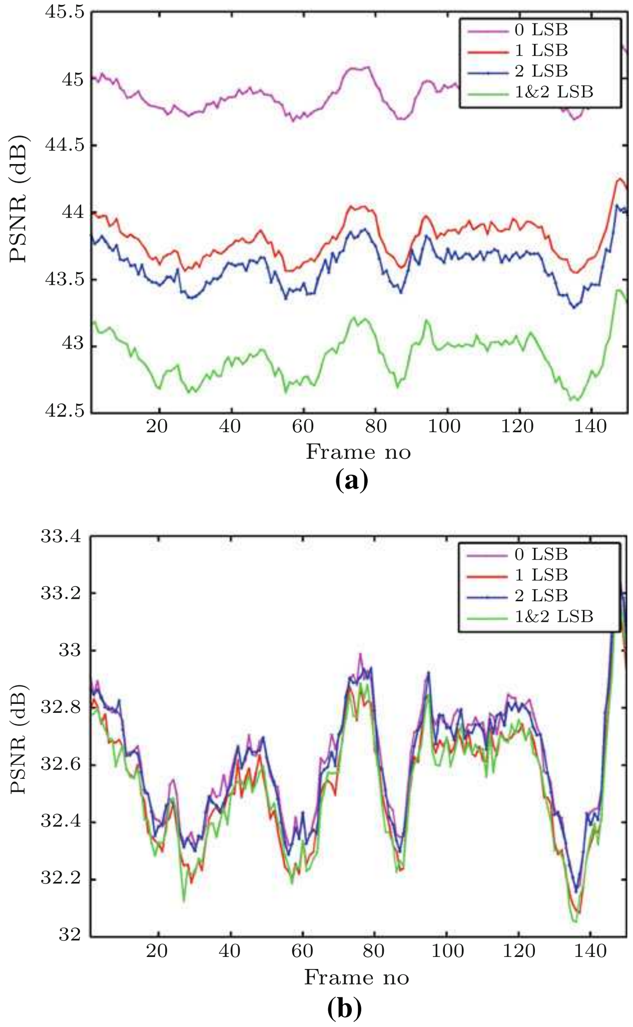


Fig. 9 Analysis of the change in PSNR for the hidden message embedding of the *intra*-frames *foreman* for the QP values of: **a** 18 **b** 36

resulting in enough number of WMBits embedded for this mode. Table 1 contains the payload, bitrate, and PSNR analysis for the *foreman* and the *football* sequences at the QP values of 18 and 36. For the 1 & 2 LSBs mode, with the QP value of 18, the increase in the bitrate is 3.34 and 5.68%, the payload equals to 312.10 and 396.80 kbps, and the decrease in the PSNR is 1.95 and 2.03 dB for the *foreman* and the *football*, respectively.

Table 1 Results for the *intra* for the *foreman* and the *football* sequences

QP	Hiding mode (LSBs)	*Foreman*			*Football*		
		Payload (kbps)	Bitrate (kbps)	PSNR (dB)	Payload (kbps)	Bitrate (kbps)	PSNR (dB)
	0	0	4504	44.88	0	4730	45.32
18	1	234.38	4622	43.80	293.88	4928	44.13
	2	140.13	4600	43.61	188.23	4920	43.99
	1&2	312.10	4654	42.93	396.80	4999	43.29
	0	0.00	603.2	32.63	0.00	750.1	32.52
36	1	5.15	609.6	32.54	8.26	759.9	32.40
	2	0.30	603.2	32.61	0.84	752.5	32.50
	1&2	5.35	609.6	32.53	8.73	760.4	32.39

Table 2 Comparison of the bitrate, the payload and the PSNR at the QP 18 for the data hiding embedding inside and outside the reconstruction loop for the *intra* sequence with the 1&2 LSBs mode

Seq.	Orig		LSB-inside Loop (Proposed method)			LSB-outside Loop		
	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)
Bus	44.52 (0.9932)	6.90	40.43 (0.9897)	7.32	856.00	24.35 (0.9232)	6.93	891.93
City	44.52 (0.9921)	6.07	41.66 (0.9881)	6.33	611.36	24.91 (0.9306)	6.10	640.82
Crew	45.22 (0.9883)	4.34	43.80 (0.9864)	4.48	290.64	27.72 (0.9412)	4.37	305.74
Football	45.32 (0.9901)	4.62	43.29 (0.9877)	4.88	396.80	27.54 (0.9464)	4.64	435.15
Foreman	44.88 (0.9872)	4.40	42.93 (0.9851)	4.55	312.10	24.54 (0.9432)	4.43	315.49
Harbour	44.30 (0.9967)	7.20	40.38 (0.9935)	7.70	895.74	22.22 (0.9386)	7.22	937.46
Ice	47.29 (0.9898)	2.25	45.08 (0.9886)	2.34	197.28	29.34 (0.9588)	2.26	194.80
Mobile	44.59 (0.9958)	10.38	41.07 (0.9930)	10.63	895.74	22.85 (0.9369)	10.38	1640.64
Soccer	45.19 (0.9889)	4.42	43.29 (0.9860)	4.68	373.39	26.81 (0.9434)	4.45	410.98
Avg.	45.09 (0.9913)	5.62	42.44 (0.9887)	5.88	536.56	25.59 (0.9403)	5.64	641.45

Table 2 shows the SSIM/PSNR, bitrate, and payload trade-off of our scheme for the *intra*-sequences of all the benchmark video sequences at the QP value of 18. It also contains a comparison with the embedding after the reconstruction loop. For our algorithm, decrease in the SSIM/PSNR is 0.0026/2.65 dB, whereas it is 0.0510/19.5 dB for embedding after the encoding loop, on the average (Fig 10).

For a subjective quality comparison, Fig. 11 contains frame # 0 of the *foreman* and the *football* sequences. Artifacts can be observed in the *intra*-frame because of the message embedding after the encoding loop. Ghost artifacts are encircled red in these video frames. For flat regions, a change in luminance can also be observed in the video frames. These artifacts are because of the spatial prediction from the top and the left blocks. If the encoding loop is not taken into account, during the embedding process, a drift or 'increasing error' will be created between the encoder and the decoder and different values will be used for the prediction on the encoder and the decoder side. Hence, distortion will increase gradually from the top-left corner to the bottom-right corner of the image.

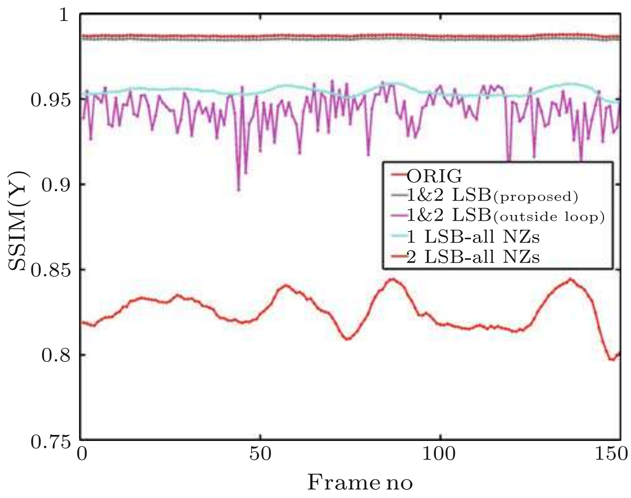


Fig. 10 SSIM of proposed scheme with: *1* the embedding outside the reconstruction loop, *2* the naive embedding in all the QTCs for the *intra*-frames for the *foreman* sequence at the QP value 18

To show the efficiency of the QTC selection criteria of our scheme, we have compared it with a naive data hiding in LSBs of all the QTCs, while taking into account the reconstruction loop. Figure 12 shows the frame # 0 of the *foreman* and the *football*. In contrast to our algorithm, one can note the noise artifacts in the frames in which message embedding is performed in all the QTCs using the naive 1 LSB and 2 LSBs

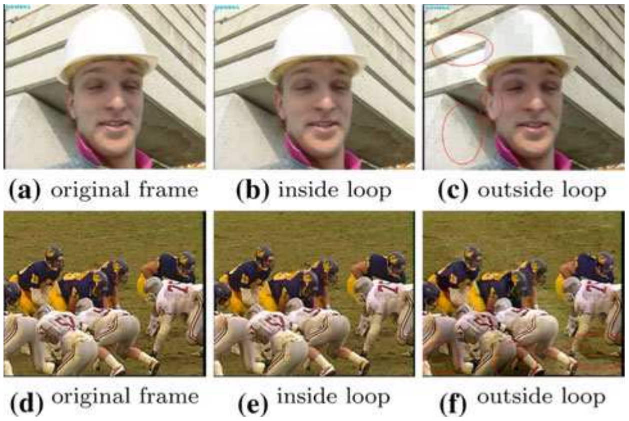


Fig. 11 Artifacts created in *intra* due to outside loop data hiding embedding with the 1&2 LSBs mode with the QP 18 for the frame # 0 of *foreman* and *football*

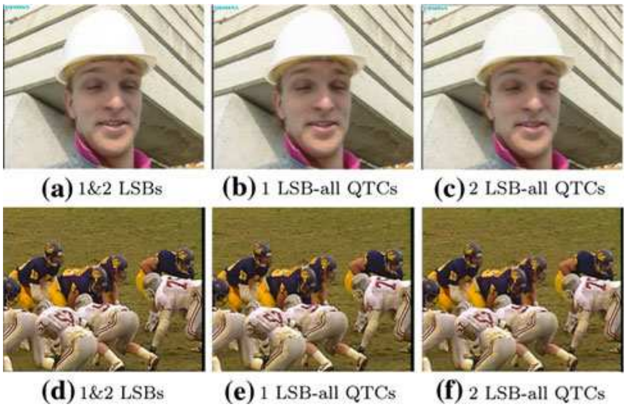


Fig. 12 Visual comparison of the 1&2 LSBs mode with the 1 LSB and the 2 LSBs embedding in all the QTCs for the *intra*-frame # 0 for the QP value of 18

embeddings. This noise is due to the introduction of new frequencies as a result of the conversion of the zero QTCs to non-zeros. Table 3 shows the SSIM/PSNR, bitrate, and payload analysis for the naive 1 LSB and 2 LSBs embeddings. We have compared it with the 1&2 LSBs mode of our algorithm, which has the highest trade-offs in terms of the SSIM/PSNR and the bitrate. When the *skip-mode* is off, the payload for the CIF resolution at 25 fps will be 3712.5 and 7425 kbps for naive 1 LSB and 2 LSBs embeddings, respectively. One can note that the decrease in the SSIM/PSNR for the 1&2

Table 3 Comparison of the bitrate and the PSNR of our scheme with the data hiding embedding in all the QTCs for the *intra* sequence at the QP 18

Seq.	Orig		1&2 LSBs mode (Our method)			1 LSB-all NZs		2 LSBs-all NZs	
	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB)	Bitrate (mbps)
Bus	44.52 (0.9932)	6.90	40.43 (0.9897)	7.32	856.00	38.87 (0.9760)	9.55	32.71 (0.9087)	12.67
City	44.52 (0.9921)	6.07	41.66 (0.9881)	6.33	611.36	38.99 (0.9723)	8.74	32.74 (0.8935)	11.92
Crew	45.22 (0.9883)	4.34	43.8 (0.9864)	4.48	290.64	39.53 (0.9564)	7.61	32.73 (0.8257)	11.34
Football	45.32 (0.9901)	4.62	43.29 (0.9877)	4.88	396.80	39.44 (0.9615)	8.01	32.69 (0.8432)	11.61
Foreman	44.88 (0.9872)	4.40	42.93 (0.9852)	4.55	312.10	39.39 (0.9545)	7.59	32.76 (0.8253)	11.41
Harbour	44.3 (0.9967)	7.20	40.38 (0.9935)	7.70	895.74	38.68 (0.9880)	9.81	32.72 (0.9531)	12.94
Ice	47.29 (0.9898)	2.25	45.08 (0.9886)	2.34	197.28	39.78 (0.9312)	6.71	32.53 (0.7258)	11.00
Mobile	44.59 (0.9958)	10.38	41.07 (0.9930)	10.63	895.74	38.63 (0.9820)	12.20	32.47 (0.9302)	14.31
Soccer	45.19 (0.9890)	4.42	43.29 (0.9860)	4.68	373.39	39.3 (0.9564)	7.85	32.70 (0.8305)	11.41
Avg.	45.09 (0.9913)	5.62	42.44 (0.9887)	5.88	536.56	39.18 (0.9642)	8.68	32.67 (0.8596)	12.07

The reconstruction loop has been taken into account for the data hiding embedding in all the QTCs

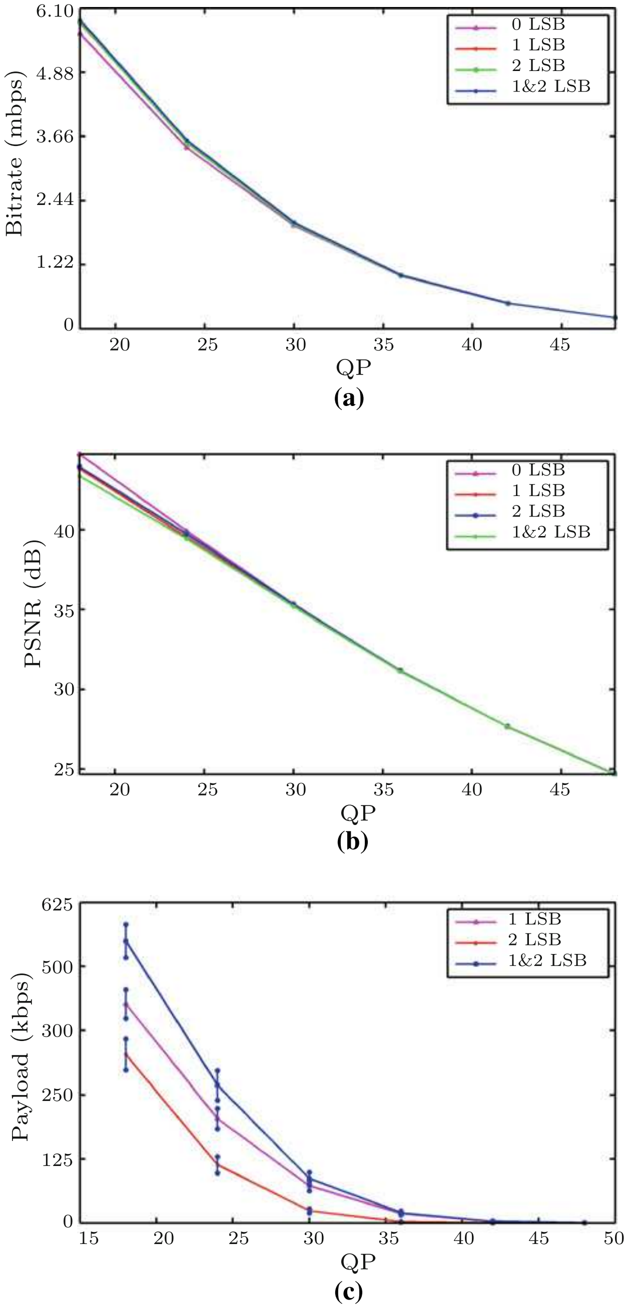


Fig. 13 Analysis of the data hiding for the *intra*-frames for all the nine benchmark video sequences over the whole range of the QP values for: **a** the bitrate, **b** the PSNR, **c** the payload. The standard deviation of the payload for all the QP values is also shown

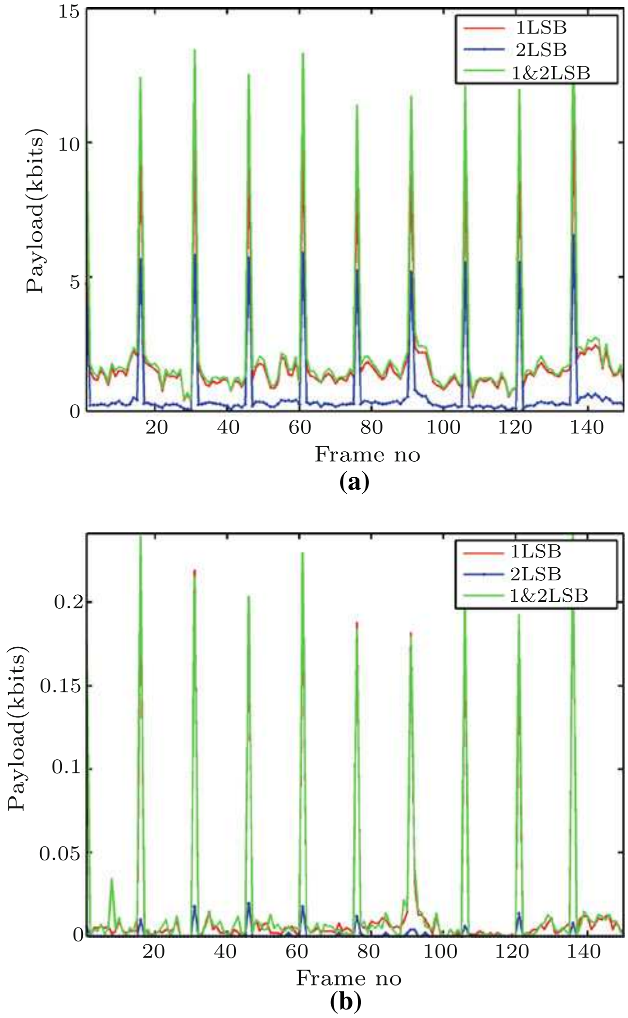
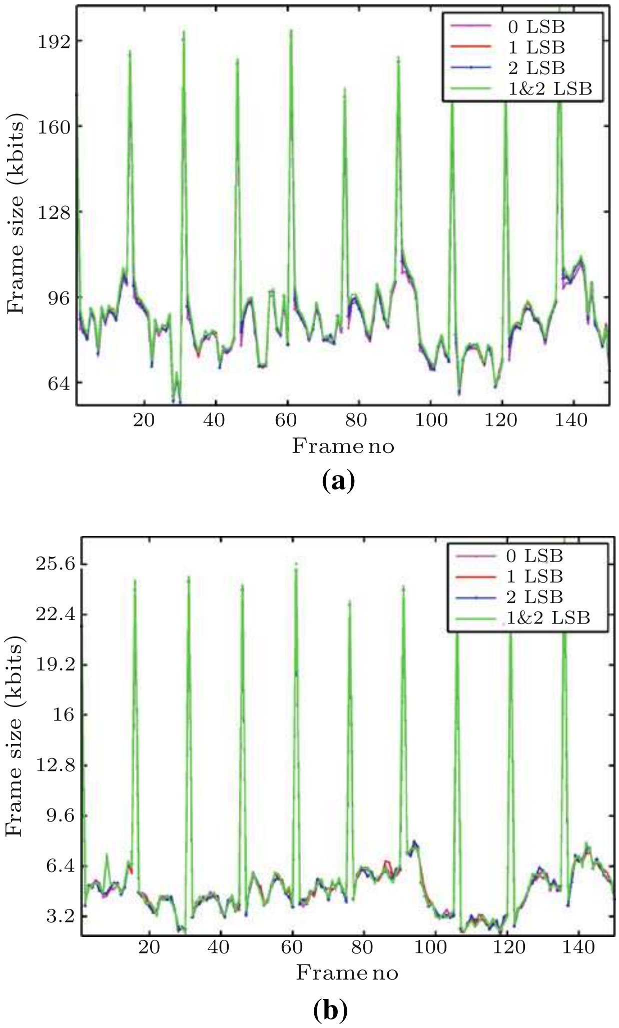


Fig. 14 Analysis of the payload capability for the message embedding of the *intra- & inter*-frames for *foreman* for the QP values: **a** 18 **b** 36

LSBs mode of our algorithm is 0.0026/2.65 dB, in contrast to the naive versions where it is 0.0271/5.91 dB for the naive 1 LSB embedding and 0.1307/12.42 dB for the naive 2 LSBs embedding for all the benchmark sequences. Increase in the bitrate is 4.6% for our algorithm, in comparison with 54.44 and 114.76% for the naive 1 LSB and the 2 LSBs modifications, respectively. Hence, the trade-offs for the SSIM/PSNR and the bitrate are so high that the naive LSB embeddings cannot be used in practical applications. A framewise SSIM comparison of our scheme has been presented in Fig. 10 with the outside loop embedding and embedding in all the QTCs. One can note that visual quality of the proposed scheme is consistently preserved for the *intra*-sequence.

4.1.1 Analysis over a whole range of QP values for the intra-frames

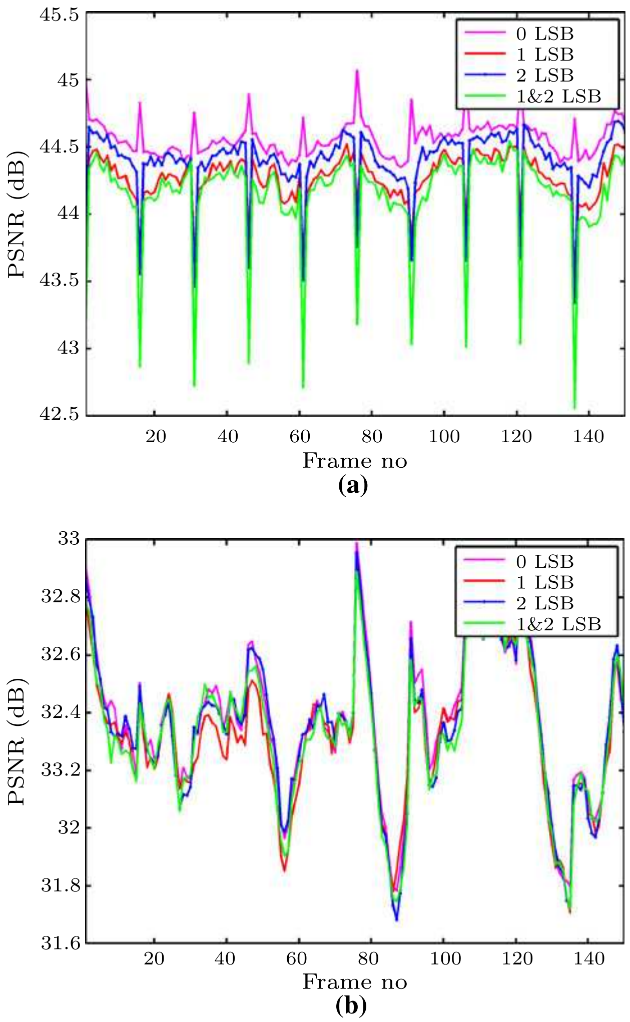
The results of our data hiding algorithm over a whole range of QP values—18, 24, 30, 36, 42, and 48—have been demonstrated in Fig. 13 for all the nine video sequences. Figure 13a illustrates the change in the bitrate over the whole range for 1 LSB, 2 LSBs, and 1 & 2 LSBs. Figure 13b shows the PSNR while Fig. 13c illustrates the payload capacity at various QP values. The PSNR graph is linear, and from the QP value of 36 onward, there is no considerable degradation in the quality. It is important to note that unlike the PSNR, bitrate and



(a)

(b)

Fig. 15 Analysis of the change in bitrate for the message embedding of the *intra- & inter*-frames for *foreman* for the QP values: **a** 18 **b** 36



(a)

(b)

Fig. 16 Analysis of the change in PSNR for the message embedding of the *intra- & inter*-frames for *foreman* for the QP values: **a** 18, **b** 36

payload graphs are non-linear. The reason is that the quantization value is an exponential function of the QP value and PSNR is a logarithmic measure.

4.2 Analysis of the *inter*-frames

The *inter*-frames contain both *intra*- and *inter*-MBs. The prediction is performed from the preceding video frames in the case of *inter*-MBs, and from the top and the left blocks in the case of *intra*-MBs. The non-zero QTCs are found in the parts of frames containing motion and texture. The message is naturally embedded in these temporal masking areas of the *inter*-frames. In a video sequence, after every *intra*-frame, first few *inter*-frames are better predicted and contain lesser residual errors. Hence, any message embedding affects more the quality and the compression ratio. But as we go away from *intra*-frames, accumulated errors appear and message embedding does not affect much the quality of the *inter*-frames. On the average, after 5 *inter*-frames followed by *intra*-frame, the ratio of the payload to the size of the *inter*-frames is comparable with that of the *intra*-frames, especially at the lower QP values.

Figures 14, 15, and 16 have been used for the payload, the bitrate, and the PSNR analysis, respectively at the QP values of 18 and 36 for the *foreman* sequence. One can note that the payload is adequate at the QP value of 18, and it decreases sharply. In fact, at the QP value of 36, we have very few QTCs with magnitudes above the threshold for the 1 LSB mode, let alone the 2 LSBs mode. Tables 4, 5 show the payload, the bitrate, and the PSNR analysis for the *foreman* and the *football* sequences at the QP value of 18 and 36 for the *intra*-& *inter*-frames. For the *foreman* sequence with the 1 & 2 LSBs mode at the QP value of 18, the increase in the bitrate is 2.81 and 2.89%, the payload is 38.25 and 56.45 kbps, and the PSNR decrease is 0.22 and 0.20 dB for the *inter* and the *intra*& *inter*, respectively. In contrast, for the *football*

Table 4 Data hiding results with the *intra* and the *inter* frames of the *foreman* and the *football* video sequences for the QP value of 18

QP	Hiding mode (LSBs)	*Foreman*			*Football*		
		Payload (kbps)	Bitrate (kbps)	PSNR (dB)	Payload (kbps)	Bitrate (kbps)	PSNR (dB)
	0	0	4.40	44.88	0	4.67	45.29
I	1	233.80	4.51	43.80	295.62	4.86	44.10
	2	139.65	4.49	43.61	196.25	4.85	43.90
	1&2	311.30	4.55	42.92	402.42	4.93	43.20
	0	0.00	2.05	44.54	0	3.47	44.76
p	1	34.45	2.09	44.30	172.91	3.58	43.65
	2	7.00	2.08	44.45	82.01	3.64	43.68
	1&2	38.25	2.11	44.23	207.43	3.70	43.05
I	0	0	2.21	44.56	0	3.55	44.79
+	1	47.73	2.25	44.27	181.09	3.67	43.69
P	2	15.83	2.24	44.39	89.62	3.72	43.70
	1&2	56.45	2.27	44.14	220.43	3.70	43.06

Table 5 Data hiding results with the *intra* and the *inter* frames of the *foreman* and the *football* video sequences for the QP value of 36

QP	Hiding mode (LSBs)	*foreman*			*football*		
		Payload (kbps)	Bitrate (kbps)	PSNR (dB)	Payload (kbps)	Bitrate (kbps)	PSNR (dB)
	0	0	601.6	32.61	0	781.3	32.42
I	1	4.96	608.5	32.52	9.08	791.2	32.30
	2	0.28	601.4	32.59	0.97	782.4	32.38
	1&2	5.19	609.4	32.52	9.70	792.6	32.28
	0	0	117.8	32.35	0	441.9	31.62
P	1	0.11	117.9	32.31	2.91	448.0	31.55
	2	0.003	117.1	32.34	0.12	443.0	31.62
	1&2	0.13	118.7	32.32	2.93	448.2	31.55
I	0	0	149.9	32.37	0	464.5	31.67
+	1	0.44	150.6	32.33	3.32	470.9	31.60
P	2	0.02	149.4	32.35	0.18	465.6	31.67
	1&2	0.47	151.3	32.33	3.38	471.0	31.60

sequence, the increase in the bitrate is 6.73 and 6.62%, the payload is 207.43 and 220.43 kbps, and the PSNR decrease is 1.71 and 1.73 dB for the *inter* and the *intra*& *inter*, respectively. The *football* sequence has greater payload capacity than the *foreman*, especially in the P frames. It is because of the texture and the high amount of motion in the *football* sequence.

The overall analysis of all the benchmark video sequences is given in Table 6. It also contains a comparison with the message embedding after the encoding loop. Decrease in the SSIM/PSNR for our scheme is 0.0106/1.38 dB, while it is 0.0687/19.2 dB for the embedding after the encoding loop. For a subjective quality comparison with the outside loop watermark embedding, Fig. 18 contains frame # 89 of the *foreman* and the *football* sequences and shows the artifacts in the *inter*-frame because of the watermark embedding after the encoding loop. One can note that the *inter*-frame is heavily distorted in this case. The results show that any data hiding after the encoding loop distorts the video fame in the case of *inter*. Hence, message embedding after the encoding loop is not a workable solution for state of the art video codecs because of the spatial and the temporal prediction.

For comparison, with the naive message embedding in the LSBs of all the QTCs, Fig. 19 shows frame # 28 of the *foreman* and the *football*. Just like the *intra*-frames, one can only note the noise artifacts in those frames in which message embedding is performed in all the QTCs using the 1 LSB and the 2 LSBs embeddings, owing to the introduc-

Table 6 Comparison of the bitrate, the payload and the PSNR for the message embedding inside and outside the reconstruction loop for the *intra* & *inter* sequence with the 1&2 LSBs mode

Seq.	Orig		LSB-inside Loop (Proposed method)			LSB-outside Loop		
	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)
Bus	44.27 (0.9923)	4.00	42.21 (0.9916)	4.22	278.38	24.12 (0.9101)	4.34	356.86
City	44.36 (0.9916)	2.57	43.80 (0.9912)	2.66	87.14	25.16 (0.9304)	2.76	115.12
Crew	44.80 (0.9859)	3.12	44.00 (0.9854)	3.24	111.90	28.48 (0.9020)	3.50	164.94
Football	44.79 (0.9876)	3.56	43.06 (0.9867)	3.79	220.43	26.94 (0.8898)	3.99	320.76
Foreman	44.56 (0.9856)	2.22	44.14 (0.9856)	2.28	56.46	25.04 (0.9323)	2.53	92.18
Harbour	44.18 (0.9963)	4.45	42.05 (0.9950)	4.72	330.15	22.46 (0.9331)	4.70	380.64
Ice	46.93 (0.9884)	1.06	46.27 (0.9886)	1.11	46.03	29.14 (0.9475)	1.29	89.51
Mobile	44.18 (0.9951)	5.70	41.00 (0.9937)	5.96	526.03	22.97 (0.9285)	5.98	581.07
Soccer	44.82 (0.9873)	2.35	43.99 (0.9870)	2.46	100.12	25.86 (0.9178)	2.64	153.12
Avg.	44.77 (0.9900)	3.23	43.39 (0.9894)	3.38	195.18	25.57 (0.9213)	3.52	250.47

The QP value is 18 and the *Intra period* is 15.

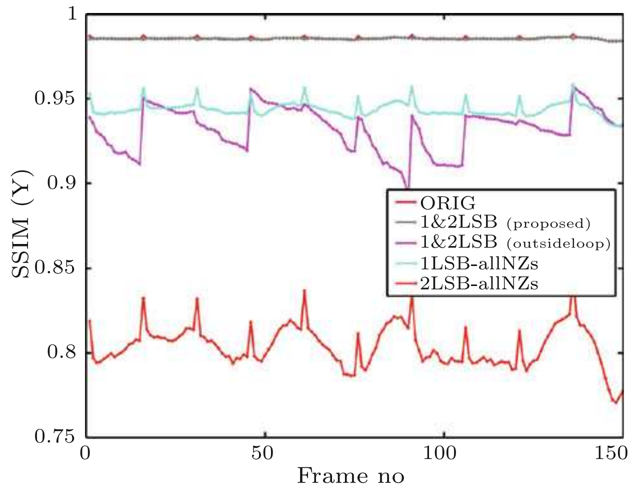


Fig. 17 SSIM of the proposed scheme with: *1* The embedding outside reconstruction loop, *2* The naive embedding in all the QTCs for the *intra- & inter*-frames for the *foreman* sequence at the QP value 18

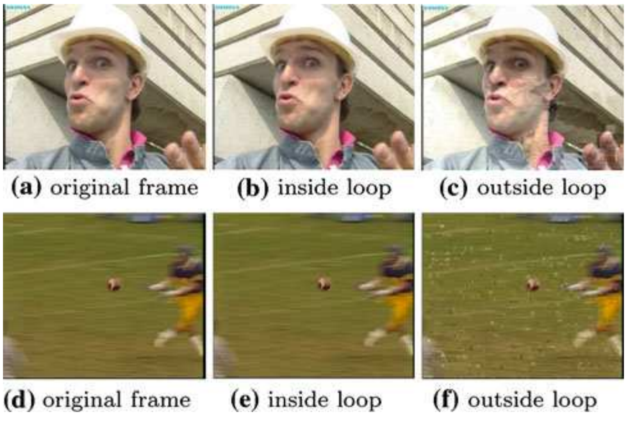


Fig. 18 Artifacts created in the *inter* due to the outside loop message embedding with the 1&2 LSBs mode with the QP 18 for the frame # 89 of *foreman* and *football*

tion of new frequencies. Table 7 shows the SSIM/PSNR, the bitrate, and the payload analysis at the QP value of 18. The payload, for the naive 1 LSB and 2 LSBs embedding, will be similar to the *intra*-frames, *i.e.* 3712.5 and 7425 kbps, respectively. Average decrease in the SSIM/PSNR for our algorithm is 0.0106/1.38 dB, while it is 0.0345/6.62 dB for the naive 1 LSB embedding and 0.1498/12.93 dB for the naive 2 LSBs

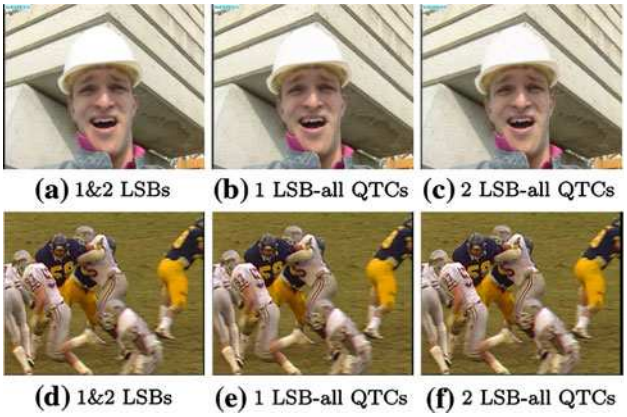


Fig. 19 Visual comparison of the 1&2 LSBs mode with the naive 1 LSB and 2 LSBs embeddings in all the QTCs. for the *inter*-frame # 28 for the QP value of 18

embedding. Increase in the bitrate is 4.6% for our algorithm, while it is 101.23 and 216.09% for the naive 1 LSB and 2 LSB modifications. Such high trade-offs for SSIM/PSNR and bitrate make it inappropriate for practical applications. Framewise SSIM comparison of our scheme in Fig. 17 for the *foreman* sequence at the QP value 18 verifies that the visual quality of the proposed scheme is consistently preserved for the *intra* & *intra*-sequence.

4.2.1 Analysis over whole range of QP values for inter-frames

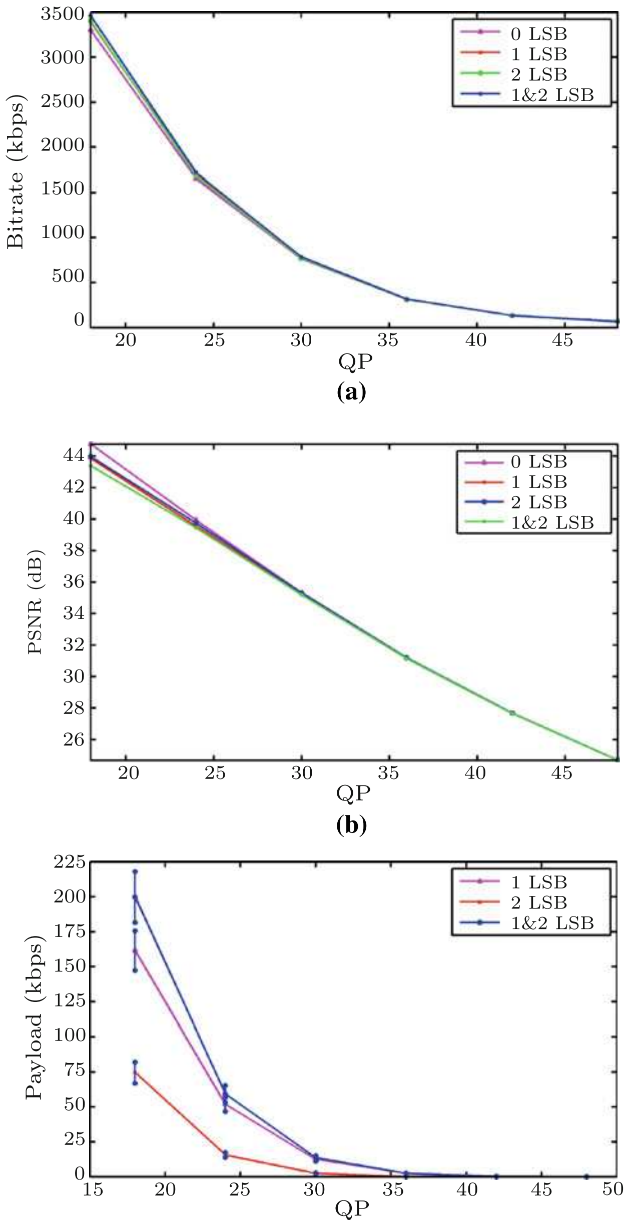
The overall performance analysis of all the nine video sequences with an *intra-period* of '15' is shown in the form of graphs in Fig. 20. Figure 20a illustrates the effect of data hiding on the bitrate, while Fig. 20b shows the change in the PSNR for different message embedding modes. Figure 20c illustrates the payload capability of our algorithm along with the standard deviation of the payload at different QP values. Among the 1 LSB and 2 LSBs embedding modes, 1 LSB performs better, having higher payloads and minimum increase in the bitrate. In the 2 LSBs mode, 2 bits are embedded in the same coefficient and thus the magnitude of compromise is higher. The 2 LSBs should be used in combination with the 1 LSB mode (1 & 2 LSBs embedding mode) when higher payload is required. One can note that just like the bitrate, the payload varies with the QP both in the case of *intra* and *inter*.

4.3 Comparative evaluation

For the sake of comparative evaluation of our scheme, we have compared it with seven other recent techniques

Table 7 Comparison of the bitrate and the PSNR of our scheme with the message embedding in all the QTCs for the *intra* & *inter* sequence with the QP 18

Seq.	Orig		1&2 LSBs mode (Proposed method)			1 LSB-all NZs		2 LSBs-all NZs	
	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)
Bus	44.27 (0.9923)	4.00	42.21 (0.9916)	4.22	278.38	37.86 (0.9703)	6.87	31.79 (0.8948)	10.36
City	44.36 (0.9916)	2.57	43.80 (0.9912)	2.66	87.14	38.28 (0.9677)	5.82	32.07 (0.8807)	9.82
Crew	44.80 (0.9859)	3.12	44.00 (0.9854)	3.24	111.90	38.33 (0.9431)	6.36	31.97 (0.8009)	10.16
Football	44.79 (0.9876)	3.56	43.06 (0.9867)	3.79	220.43	38.16 (0.9499)	6.87	31.77 (0.8187)	10.54
Foreman	44.56 (0.9856)	2.22	44.14 (0.9856)	2.28	56.46	38.45 (0.9436)	5.76	32.11 (0.8040)	9.85
Harbour	44.18 (0.9963)	4.45	42.05 (0.9951)	4.72	330.15	37.65 (0.9845)	7.02	31.79 (0.9442)	10.26
Ice	46.93 (0.9884)	1.06	46.27 (0.9886)	1.11	46.03	38.70 (0.9145)	6.08	31.51 (0.6863)	10.31
Mobile	44.18 (0.9951)	5.70	41.00 (0.9937)	5.96	526.03	37.50 (0.9777)	7.74	31.60 (0.9218)	10.57
Soccer	44.82 (0.9873)	2.35	43.99 (0.9870)	2.46	100.12	38.41 (0.9481)	5.98	31.95 (0.8100)	10.04
Avg.	44.77 (0.9900)	3.23	43.39 (0.9894)	3.38	195.18	38.15 (0.9555)	6.50	31.84 (0.8402)	10.21



(a)

(b)

Fig. 20 Analysis of the data hiding for the *intra- & inter*-frames for all the nine benchmark video sequences over the whole range of the QP values for: **a** the bitrate, **b** the PSNR, **c** the payload. The standard deviation of the payload for all the the QP values is also shown

which are as follows: the particle swarm optimization-based dither modulation scheme [37], the bitstream replacement of CAVLC bitstream [40], the hybrid watermarking scheme [32], the data hiding in the H.264/AVC compressed video [18], the reference index-based watermarking of the H.264/AVC [21], the spread transform scalar Costa scheme [10], and the robust watermarking for P frames [29]. These techniques are different from each other in several aspects e.g., the working domain (pixel, transform, or bitstream) and the embedding algorithm (dither modulation, spread transform, or quantization based). The comparison has been made based on several important characteristics of the data hiding systems and summarized in Table 8.

The embedding domain is of vital importance for a watermarking system. The watermarking algorithms of Zou and Bloom [40], and Kim et al. work in the bitstream domain (outside the reconstruction loop). These algorithms can work on compressed video, without decoding it. These schemes, however, have lower payload and can lead to a considerable decrease in the PSNR because of the drift error. For [18], the PSNR drops to 43 dB even for the I frame, let alone the P frames. Moreover, a flickering effect is visible along the temporal dimension of the video. The video quality decreases with the increase in the encoded spatial resolution because of this drift error. The bitrate change is also another important aspect of these data hiding algorithms. For the algorithms that work in bitstream domain [18], there is no change in bitrate. For our algorithm, the increase in the bitrate is 4.6% with a decrease in the PSNR of 1.38 dB for the *intra-* & *inter*-sequence.

Different watermarking algorithms have different embedding space. Some of them work on the video data (pixels or coefficients), while other embed hidden message in the video header. Our scheme works inside the reconstruction loop, while utilizing the largest possible embedding space with minimal trade-off in terms of the bitrate and the SSIM/PSNR. It is in contrast to other schemes which either use the video header fields [21,40] or some part of the video data e.g., the luma [10,37], or the I frames only [18,32,37]; hence, not fully utilizing the embedding capacity. In [29], the watermark is embedded in all the non-zero QTCs, (even having the magnitude of 1). Let the *watermark cost* be the increase in the bitrate (in bits) per watermark bit. In the case of the *intra*, the *watermark cost* is 0.06 and 0.15 at the QP values of 18 and 36, respectively. These results are far better than 1.54, the result presented in [29]. In the case of *intra* and *inter*, we get the *watermark costs* of 0.15 and 0.42 at the QP values of 18 and 36, respectively, which is far better than 1.50, the result of the work presented in [29]. At the higher QP values, we do not have lot of non-zero QTCs which can be watermarked but the ratio between the payload and the bitrate is still conserved.

To summarize, our proposed scheme presents a good payload capability with a minimal trade-off for the PSNR and the bitrate. This trade-off is possible because of the selection of the AC QTCs with magnitudes above a certain threshold and embedding inside the reconstruction loop.

5 Conclusion

In this paper, we have designed and analyzed a new video data hiding scheme with a high payload for the H.264/AVC.

Table 8 Comparison of the proposed scheme with other recent watermarking techniques for the H.264/AVC video codec.

Wm Scheme	Working domain	Insertion technique	Payload	Bitrate increase	PSNR (dB)	Embedding space
Wu et al. [37]	Transform	Dither modulation	1 bit/4x4 block	Yes	Yes	DCT coeff. of *luma* (I frames only)
Zou and Bloom [40]	Bitstream	Replacement		Yes	Yes	Intra prediction (mode change)
Qiu et al. [32]	Transform/MB header	Replacement	1 bit/MB	Yes	Yes	AC coeffs (I frames) MVs (P frames)
Kim et al. [18]	Bitstream	Replacement	1 bit/MB	No	(up to 43 dB)	Signs-T1's (only I frames)
Li et al. [21]	Transform	Spread spectrum		3.22%	–0.75 dB	reference index (P frames) except 1st P frame)
Golikeri et al. [10]	Transform (before quantization)	Spread transform scalar Costa scheme	1 bit/MB	Yes	Yes	DC coeff. of *luma* (I and P frames, HVS based)
Noorkami et al. [29]	Transform	Spread spectrum	1 bit/MB	Yes	Yes	All non-zero QTCs (I and P frames)
Our scheme	Transform	Quantization based	195.18 kbps (20.19 bits/MB)	4.6%	–1.38 dB	AC coefficients (above threshold)

Our scheme embeds the RD optimized hidden message in the QTCs for both the *intra*- and *inter*-frames. Having a regard for the reconstruction loop, the proposed scheme offers consistent payload capability to the H.264/AVC standard at different bitrates without adversely affecting the overall bitrate and the SSIM/PSNR of the video bitstream. The watermark is embedded in those regions of the *intra*-frames, which contain edges and texture, and for the *inter*-frames, message embedding is naturally done in the temporal masking regions, which contains motion and texture. The experimental results have demonstrated that the *inter*-frames can be equally good for the message embedding owing to its motion and texture masking.

Acknowledgments This work is in part supported by the project VOODDO (2008–2011) of the french ANR and the region of Languedoc Roussillon, France.

References

1. Alattar, A.M., Lin, E.T., Celik, M.U.: Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video. IEEE Trans. Circuits Syst. Video Technol. **13**, 787–800 (2003)
2. Chandramouli, R., Memon, N.: Analysis of LSB based image steganography Techniques. In: Proceedings of the IEEE International Conference on Image Processing, vol. 3, pp. 1019–1022. (2001)
3. Chen, C., Ni, J., Huang, J.: Temporal statistic based video watermarking scheme robust against geometric attacks and frame dropping. In: IWDW '09: Proceedings of the 8th International Workshop on Digital Watermarking, pp. 81–95. Springer, Heidelberg (2009)
4. Chung, Y., Wang, P., Chen, X., Bae, C., Otoom, A., Tran, T.: A performance comparison of high capacity digital watermarking systems. In: KES, vol. 1, pp. 1193–1198. (2005)
5. Cox, I., Kilian, J., Leighton, F., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. **6**, 1673–1687 (1997)
6. Cox, R.D., Henri, C.J., Bret, P.M.: CD-based image archival and management on a hybrid radiology intranet. Can. J. Med. Radiat. Technol. **28**(3), (1997)
7. Dai, Y., Zhang, L., Yang, Y.: A new method of MPEG video watermarking technology. In: Proceedings of International Conference on Communication Technology, vol. 2, pp. 1845–1847. (2003)
8. Deguillaume, F., Csurka, G., O'Ruanaidh, J., Pun, T.: Robust 3D DFT video watermarking. SPIE **3657**, 113–124 (1999)
9. Ejima, M., Miyazaki, A.: A wavelet-based watermarking for digital images and video. In: Proceedings of IEEE International Conference on Image Processing, vol. 3, pp. 678–681. (2000)
10. Golikeri, A., Nasiopoulos, P., Wang, Z.: Robust digital video watermarking scheme for H.264 advanced video coding standard. J. Electron. Imaging **16**(4), 043008 (2007)
11. Gong, X., Lu, H.: Towards fast and robust watermarking scheme for H.264 video. In: Proceedings of IEEE International Symposium on Multimedia, pp. 649–653. (2008)
12. H264. 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)
13. Hartung, F., Girod, B.: Watermarking of uncompressed and compressed video. Signal Process **66**, 283–333 (1998)
14. He, D., Sun, Q., Tian, Q.: A semi-fragile object based video authentication system. In: Proceedings of International Symposium on Circuits and Systems, **3**, pp. 814–817 (2003)
15. Kang, X., Huang, J., Shi, Y., Lin, Y.: A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression. IEEE Trans. Circuits Syst. Video Technol. **13**(8), 776–786 (2003)
16. Kapotas, S., Varsaki, E., Skodras, A.: Data hiding in H.264 encoded video sequences. In: Proceedings of IEEE International Workshop on Multimedia Signal Processing, (2007)
17. Katzenbeisser, S., Petitcolas, F. (eds.): Information Hiding Techniques for Steganography and Digital Watermarking. Artech House Inc., Norwood (2000)

18. Kim, S., Kim, S., Hong, Y., Won, C.: Data hiding on H.264/AVC compressed video. In: Image Analysis and Recognition Lecture Notes in Computer Science, chapter 62, pp. 698–707 (2007)
19. Langelaar, G., Lagendijk, R.: Differential energy watermarking of DCT encoded images and video. IEEE Trans. Image Process. **10**, 148–158 (2001)
20. Lee, P., Chen, M.: Robust error concealment algorithm for video decoder. IEEE Trans. Consumer Electron. **45**(3), 851–859 (1999)
21. Li, J., Liu, H., Huang, J., Zhang, Y.: A robust watermarking scheme for H.264. In: Digital Watermarking volume 5450 of Lecture Notes in Computer Science, pp. 1–15. Springer, Heidelberg (2009)
22. Li, K., Zhang X.: Reliable adaptive watermarking scheme integrated with JPEG2000. In: Proceedings of 3rd International Symposium on Image and Signal Processing and Analysis (ISPA 2003). Rome, Italy (2003)
23. Li, Q., Cox, I.: Using perceptual models to improve fidelity and provide resistance to volumetric scaling for quantization index modulation watermarking. IEEE Trans. Inf. Forensics Security **2**(2), 127–139 (2007)
24. Lu, C., Chen, J., Fan, K.: Real-time frame-dependent video watermarking in VLC domain. Signal Process. Image Commun. **20**(7), 624–642 (2005)
25. Malvar, H., Hallapuro, A., Karczewicz, M., Kerofsky, L.: Low-complexity transform and quantization in H.264/AVC. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 598–603 (2003)
26. Miller, M., Doerr, G., Cox, I.: Applying informed coding and embedding to design a robust high capacity watermark. IEEE Trans. Image Process. **13**(2), 792–807 (2004)
27. Mobasseri, B., Raikar, Y.: Authentication of H.264 streams by direct watermarking of CAVLC blocks. In: Proceedings of Security, Steganography, and Watermarking of Multimedia Contents IX volume 6505 SPIE, San Jose, CA (2007)
28. Noorkami, M., Mersereau, R.: Compressed-domain video watermarking for h.264. In: Proceedings of IEEE International Conference on Image Processing, pp. 890–893. Genoa, Italy (2005)
29. Noorkami, M., Mersereau, R.: Digital video watermarking in P-frames with controlled video bit-rate increase. IEEE Trans. Inf. Forensics Security **3**(3), 441–455 (2008)
30. Oostveen, J., Kalker, T., Haitsma J.: Visual hashing of digital video: applications and techniques. In: Proceedings of SPIE, Applications of Digital Image Processing XXIV. vol. 4472, pp. 121–131. San Diego CA, USA (2009)
31. Pröfrock, D., Schlauweg, M., Müller, E.: A new uncompressed-domain video watermarking approach robust to H.264/AVC compression. In: Proceedings of IASTED International Conference on Signal Processing, Pattern Recognition, and Applications, pp. 99–104. Anaheim, CA, USA (2006)
32. Qiu, G., Marziliano, P., Ho, A.T.S., He, D., Sun, Q.: A hybrid watermarking scheme for h.264/avc video. Pattern Recognition, International Conference on **4**, pp. 865–869 (2004)
33. Ramkumar, M., Akansu, A.: Robust protocols for proving ownership of images. In: Proceedings of International Conference on Information Technology: Coding and Computing, pp. 22–27 (2000)
34. Wang, Q., Zhao, D., Ma, S., Lu, Y., Huang, Q., Ga, W.: Context-based 2D-VLC for video coding. In: Proceedings of IEEE International Conference on Multimedia and Expo, pp. 89–92 (2004)
35. Watson, A.: DCT quantization matrices visually optimized for individual images. In: Proceedings of SPIE Society of Photo-Optical Instrumentation Engineers, vol. 1913, pp. 202–216 (1993)
36. Wiegand, T., Sullivan, G., Bjntegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. **13**, 560–576 (2003)
37. Wu, C., Zheng, Y., Ip, W., Chan, C., Yung, K., Lu, Z.: A fexible H.264/AVC compressed video watermarking scheme using particle swarm optimization based Dither modulation. AEU Int. J. Electron. Commun. (2010)
38. Wu, G., Wang, Y., Hsu, W.: Robust watermark embedding/detection algorithm for H.264 video. J. Electron. Imaging **14**(1), (2005)
39. Xie, F., Furon, T., Fontaine, C.: On-Off keying modulation and Tardos fingerprinting. In: Proceedings of ACM Workshop on Multimedia and security, pp. 101–106. New York, NY, USA (2008)
40. Zou, D., Bloom, J.: H.264/AVC stream replacement technique for video watermarking. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1749 –1752 (2008)