



**HAL**  
open science

## A histogram shifting based RDH scheme for H. 264/AVC with controllable drift

Zafar Shahid, William Puech

► **To cite this version:**

Zafar Shahid, William Puech. A histogram shifting based RDH scheme for H. 264/AVC with controllable drift. *Electronic Imaging*, Feb 2013, San Francisco, CA, United States. pp.86650S, 10.1117/12.2012741 . lirmm-00830966

**HAL Id: lirmm-00830966**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00830966>**

Submitted on 6 Jun 2013

**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.

# A Histogram Shifting based RDH scheme for H.264/AVC with Controllable Drift

Zafar Shahid and William Puech

Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II  
161, rue Ada, 34392 MONTPELLIER Cedex 05, FRANCE

zafar.shahid@lirmm.fr, william.puech@lirmm.fr

## ABSTRACT

This paper presents an efficient method for high payload reversible data-hiding in H.264/AVC intra bitstream with a minimal drift, which is controllable and proportional to the payload. In contrast to previously presented open-loop reversible data hiding techniques for H.264/AVC bitstream, which mainly perform drift-compensation, we propose a new reversible data hiding technique for H.264/AVC intra bitstream which avoids drift. The major design goals of this novel H.264/AVC reversible data hiding algorithm have been runtime-efficiency, high perceptual quality with a minimal effect on bit-rate. The data is efficiently embedded in the compressed domain in an open-loop fashion, *i.e.*, all prediction results are reused. Nevertheless intra-drift is avoided, as only specific solution patterns are added, which are solutions of a system of linear equations that guarantee the preservation of the block's edge pixel values.

## 1. INTRODUCTION

Data hiding is a technique that embeds data into cover media contents such as audios, images and videos. In normal data hiding, the cover media is permanently distorted because of irreversible operations of embedding. This is acceptable for most of the applications but, there are some sensitive fields in which permanent distortion of the cover media is not acceptable *e. g.* military, law enforcement, satellite imaging and medical systems. In such fields, it is desirable to reverse the marked media back to the original cover media in a lossless manner after the hidden data are retrieved. Reversible data hiding (RDH) embeds the message into a cover media in a reversible manner and the original content is restored without any degradation at the time of extraction of the embedded data.

In H.264/AVC, the data could be embedded at various stages in the compression pipeline, *i.e.*, pre-compression or spatial domain, transform coding stage, QTCs (quantized transform coefficients) stage and entropy coding stage. From a runtime perspectives, all these stages can be divided into two main categories, namely closed-loop (within the reconstruction loop or in spatial domain) and open loop (after the reconstruction loop) as shown in Fig. 1. Embedding within the reconstruction loop (closed-loop) avoids drift, shows a small impact on rate distortion (RD) performance and offers a high payload.<sup>1</sup> The main disadvantage is that it requires re-computation of all prediction decision for the embedding of each single stego-message. Thus it requires the computationally most complex parts of video compression and therefore cannot be used for applications with runtime constraints or in applications which require numerous stego-messages to be embedded *e.g.*, active fingerprinting usually requires separate stego-message (each buyer's fingerprinting code) to be embedded in each copy.

On the other hand, if the embedding process occurs after the reconstruction loop (open-loop), the most complex parts of the compression pipeline are omitted. Open-loop data hiding embeds the data in compressed domain *i.e.* syntax elements (before entropy encoding), *binstrings* (in case of CABAC entropy coding engine) or bitstream domain (bit substitutions of the final bitstream). Thus open-loop data-hiding either requires only the entropy coding stage or no stages from the compression. Therefore open-loop reversible data-hiding is extremely runtime-efficient compared to closed-loop data hiding. In open-loop data hiding, prediction is performed from stego/watermarked content on the decoder side and from the original content on the encoder side. Thus there is a mismatch between encoder and decoder side predictors which accumulates over time. This accumulating mismatch is referred as drift and makes open-loop data hiding very challenging, as visual quality constraints are hard to meet. As video data are usually stored and distributed in a compressed format (very often H.264/AVC), it is often impractical to first decode the video sequence, embed the data and then re-compress it. A low-complexity video data hiding solution requires that the data hiding is performed in an open-loop fashion.

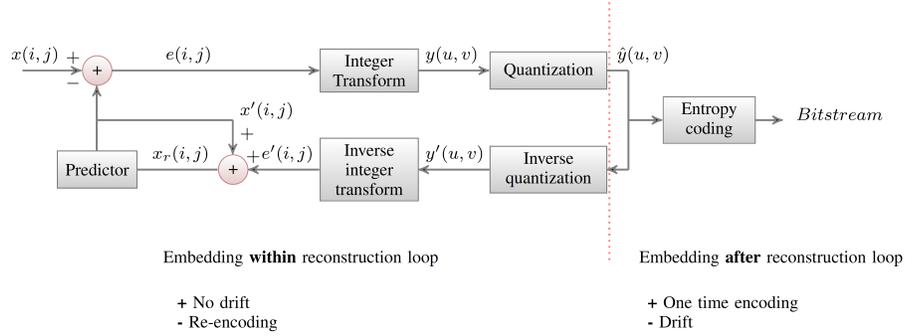


Figure 1: Two main categories of data hiding in a video codec: 1) closed-loop (during the encoding process or spatial domain), 2) open-loop.

In this paper, we present an algorithm for open-loop reversible data hiding in H.264/AVC bitstream, which has a high payload. At the same time, the embedding is imperceptible and has a minimal controllable intra-drift, by keeping the predictor pixels unchanged.

The remainder of the paper is organized as follows. Recent work on H.264/AVC reversible data hiding is presented in Section 2. The proposed technique including the derivation of the linear system, data embedding and extraction are presented in Section 3. Experimental results are presented in Section 4 to validate the efficiency of the proposed scheme. Final conclusions are drawn in Section 5.

## 2. RECENT WORK

For video content, data hiding is often strongly tied to the compression. The previous work on video data hiding can be divided into two main approaches: 1) closed-loop data hiding, *i.e.*, embedding is conducted before or during compression 2) open-loop data-hiding, *i.e.*, data hiding is conducted in entropy coding stage or directly applied in the bitstream domain. There are many reversible data hiding techniques for still images both in pixel domain and compressed domain.<sup>2-4</sup> Hayat *et al.* have presented a technique to use RDH for scalable 3-D terrain visualization for JPEG2000 images.<sup>3</sup> While in Ref. 4, the authors proposed a technique to perform RDH in encrypted domain for image. For video content, there are very few algorithms for RDH till now. It is because of the complexity of the prediction process and the presence of the reconstruction loop.

Some researchers have applied traditional reversible data hiding methods to H.264/AVC directly in a closed-loop fashion.<sup>5-7</sup> In Ref. 5, Lie *et al.* utilize the difference expansion<sup>8</sup> to perform RDH in QTCs for error concealment. The proposed scheme is capable of achieving PSNR improvement of up to 1.48 *dB*, at a considerable bitrate, when the packet loss rate is 20%. Chung *et al.*<sup>6</sup> presented an algorithm which utilizes the histogram shifting (HS) on QTCs to embed motion vector (MV) for error concealment. According to the HS rules, the algorithm should find the peak point and zero point to shift and embed data from embedding target. In Ref. 7, Lin and Li proposed a reversible data hiding method based on HS. They suggested to utilize only the last level of each  $4 \times 4$  transform block instead of all QTCs in order to avoid bitrate increase and has high payload. The main limitation of all these methods is that they are closed-loop methods and needs complete re-compression to embed a different message.

In open-loop paradigm, the drift problem is first discussed in data hiding/watermarking for compressed videos and a drift compensation method to counteract this problem is provided by Hartung and Girod.<sup>9</sup> Recently, a RDH scheme is proposed to restrain the distortion for video data hiding using drift compensation for Mpeg-4 video codec in Ref. 10. In Ref. 11, the authors propose to compute the prediction error blocks and then slightly modify the error values through shifting the prediction errors based on H.264/AVC intra prediction to contain the distortion drift. Liu *et al.* recently proposed to perform RDH outside the reconstruction loop,<sup>12</sup> while avoiding the intra prediction drift. They proposed to perform embedding only in those  $4 \times 4$  blocks, which are not used for prediction of future blocks. The selected blocks are then used for data-hiding using a modified version of histogram shifting technique. The main limitation of this technique is that it has very less payload, since there are not many blocks which are not used for prediction of future blocks in normal scenarios.

Open-loop video data hiding algorithms face major challenges. First, the payload of such algorithms is significantly reduced, *i.e.*, up to a few bytes per second as explained in Ref. 9. Second, if drift compensation is not applied there is a continuous drift which significantly distorts the visual quality. If drift compensation is applied, the algorithms face considerable bit-rate increases.

To deal with this problem, we are presenting a reversible data hiding based on H.264/AVC with a minimal intra-frame distortion drift, which is controllable and proportional to the payload. The novelty of the proposed algorithm lies in the fact that the distortion is contained even if the stego-block is used for prediction of future blocks. To achieve these goals, drift is avoided instead of compensated which allows to almost keep the increment in original bit-rate proportional to the payload, as no compensation signals need to be coded.

### 3. PROPOSED ALGORITHM

The main challenge in the design of an H.264/AVC open-loop data hiding algorithm is drift which is a result of *intra* and *inter* prediction. In the proposed reversible data hiding scheme, a stego-message is embedded in the transform domain outside the reconstruction loop as shown in Fig. 2. It is performed in such a fashion that only those pixels are modified which are not used for the intra prediction of future blocks.

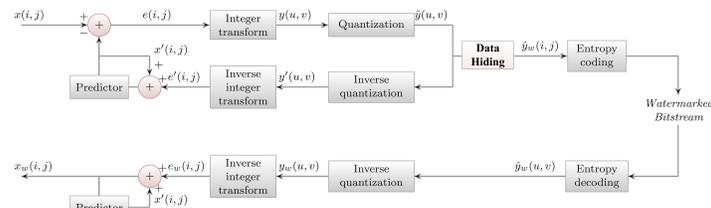


Figure 2: Open-loop reversible data hiding for H.264/AVC.

In Section 3.1, we present the solutions of a system of linear equations, which allow to embed a stego-message in the transform coefficients without modifying boundary pixels which are used for prediction of future blocks. Data embedding step is explained in Section 3.2, while data extraction step is presented in Section 3.3.

#### 3.1 Intra-drift-free data hiding: solution of a system of linear equations

In H.264/AVC, *intra* prediction is performed in spatial mode, using pixels from top and left blocks. Two types of spatial predictions are available:  $I_{4 \times 4}$  and  $I_{16 \times 16}$ . The  $I_{4 \times 4}$  mode is based on predicting each  $4 \times 4$  luma block separately. In this paper, we are presenting the algorithm for a watermark embedding in the  $4 \times 4$  blocks of *intra* frames.

Fig. 3 shows the border pixels which are used for prediction of future blocks. For the current block, pixels labeled  $A$  to  $M$  from top and left block are used. Similarly,  $d, h, l, p, m, n, o$  pixels will be used for future prediction. Since it is only the residual  $r$  which is transformed and entropy coded and is used for data-hiding in open-loop techniques, if the residual  $r$  for the pixels at positions  $d, h, l, p, m, n, o$  in Fig. 3 are unchanged, then the original pixels will remain unchanged.

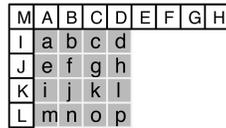


Figure 3: Labeling of the pixels used for *intra* prediction of  $4 \times 4$  blocks.

Since we aim to embed the data in the residual in the compressed domain, we need to explore the relationship between the spatial domain pixel values and the compressed domain coefficients. In the following, capital letters refer to matrices, *e.g.*,  $C, Y, W$  and small letters with indices refer to the elements of a matrix, *e.g.*,  $x_{ij}$  represents  $j^{th}$  element in  $i^{th}$  row of matrix  $X$ .

The  $4 \times 4$  residual matrix  $X$  is derived by the  $4 \times 4$  transform coefficient matrix  $Y$  using inverse integer transform as:

$$X = C^T Y C$$

The transform matrix  $C$  is defined in the H.264/AVC standard:

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0.5 & -0.5 & -1 \\ 1 & -1 & -1 & 1 \\ 0.5 & -1 & 1 & -0.5 \end{bmatrix}$$

For data hiding purpose, transform coefficient matrix  $Y$  may be modified to  $Y'$ , such that the border pixels of  $X' = C^T Y' C$  are the same as those in  $X$ , *i.e.*:

$$\begin{aligned} C^T Y' C - C^T Y C &= X' - X \\ &= D \\ &= \begin{bmatrix} d_{00} & d_{01} & d_{02} & 0 \\ d_{10} & d_{11} & d_{12} & 0 \\ d_{20} & d_{21} & d_{22} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

where:

$$\begin{aligned} Y' &= Y + N, \\ N &= \begin{bmatrix} n_{00} & n_{01} & n_{02} & n_{03} \\ n_{10} & n_{11} & n_{12} & n_{13} \\ n_{20} & n_{21} & n_{22} & n_{23} \\ n_{30} & n_{31} & n_{32} & n_{33} \end{bmatrix} \end{aligned}$$

We get 7 linear equations in  $\vec{n} = [n_{00}, \dots, n_{33}]^T$  for the zero entries in  $D$ , which can be written short hand as:

$$P\vec{n} = \vec{0}$$

The solution of this system of linear equations is spanned by a 9 element basis. Any linear combination of these basis vectors can be added to the transform block without changing the border pixels.

### 3.2 Reversible data hiding

In a traditional histogram based RDH scheme, the main source of drift is the histogram shifting (HS). HS is the shifting of all the QTCs after the peak by 1, In case of QTCs, the histogram peak is at 0 as shown in distribution of QTCs in Fig. 4.a. Fig. 4.b shows the HS process in which all the QTCs after the peak are shifted. As illustrated in Section 3.1, a solution pattern of system of linear equation (or their linear combination) can be added to the QTCs of a  $4 \times 4$  block without changing the border pixels and hence, avoiding drift.

Fig. 5 shows the analysis of drift accumulation because of traditional RDH and the proposed scheme. QTCs responsible for drift are colored red, while green QTCs cause no drift. QTCs responsible for drift (red) in a traditional RDH are represented in Fig. 5.a. While Fig. 5.b shows QTCs responsible for drift in our proposed scheme. Since we are using a solution of system of linear equations to embed data in QTCs, there is **no drift** because of histogram shifting. It is evident that 0 message bit is the only source of drift in our case. If the message is random having equal probability of 1 and 0 bits. The drift is caused by half of the bits of payload and is controllable. On the contrary, drift is because of all the QTCs except 0 in traditional histogram based RDH scheme.

Fig. 6 shows the block diagram and Algorithm 1 details the proposed scheme. First of all, the system of linear equations is used to detect the stego-blocks. A transform block is **stego-block**, if it does not contain any negative QTC in the solution

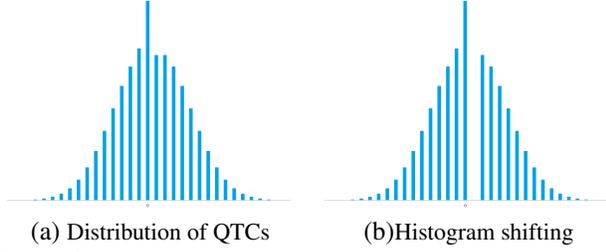


Figure 4: Distribution of transform coefficients and histogram shifting for reversible data hiding.

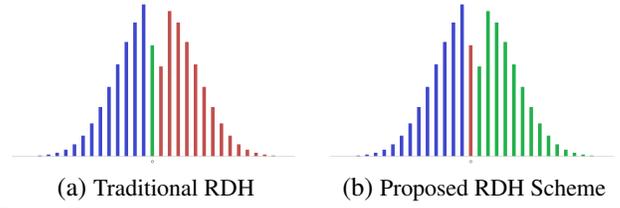


Figure 5: Drift comparison wherein red QTCs are responsible for drift and green QTCs cause no drift.

---

**Algorithm 1** Embedding of a message bit in a reversible manner.

---

```

1:  $k \leftarrow \text{SizeOfSolPatt}(QTCs)$ 
2:  $QTC[k] \leftarrow \text{SolOfLinEq}(QTCs)$ 
3:  $StegoBlk \leftarrow 1$ 
4:  $l \leftarrow 0$ 
5: while  $l < k$  do
6:   if  $QTC[l] < 0$  then
7:      $StegoBlk \leftarrow 0$  {Non-RDH block}
8:     Break
9:   end if
10:   $l \leftarrow (l + 1)$ 
11: end while
12: if  $StegoBlk$  then
13:   for all  $i = 1 \rightarrow k$  do
14:     if  $QTC[i] = 0$  then
15:        $StegoQTC[i] \leftarrow \text{MsgBit}$ 
16:     else {Perform HS}
17:        $StegoQTC[i] \leftarrow QTC[i]$ 
18:     end if
19:   end for
20: end if
21: end

```

---

pattern. In the solution pattern of a stego-block, if the QTC is 0, a message bit is embedded in it. If it is positive, HS is performed on it by incrementing it by 1.

Solution pattern affects the payload of the proposed scheme because of two factors. First is the number of QTCs in the solution pattern. The higher the number of QTCs, more bits can be embedded per  $4 \times 4$  block and hence, the higher will be the payload. Second is the frequency of QTCs. High frequency QTCs have higher probability of being 0 than low-frequency QTCs.

### 3.3 Data extraction and restoration of original QTCs

The extraction of data and restoration of original QTCs is performed in open-loop fashion as well, as shown in Fig. 7. The only information required for the extraction process is the solution pattern which is used for the embedding process. Let us denote the QTCs which are part of solution pattern as stego-QTCs. They are extracted from the bitstream by entropy decoding.

The whole extraction process is explained in Algorithm 2. First step is to check whether the  $4 \times 4$  block contains embedded data or not. If any stego-QTC of the solution pattern is negative, it means that this block does not contain any embedded data. Among the blocks which satisfies this condition, if stego-QTC is greater than 1, it does not contain any data and inverse HS is performed on this stego-QTC to extract original QTC by subtracting 1. Otherwise stego-QTC contains the embedded bit. In this case, the value of extracted bit is equal to stego-QTC, and extracted QTC will be 0. In this way, data are extracted from the stego H.264/AVC bitstream and the original bitstream is restored in a lossless fashion.

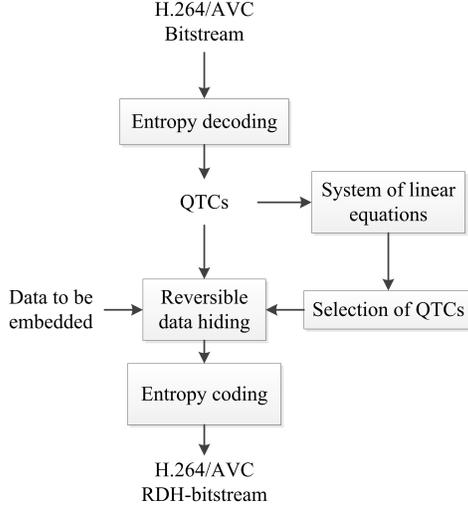


Figure 6: Proposed open-loop RDH scheme for H.264/AVC with controllable drift based on system of linear equations.

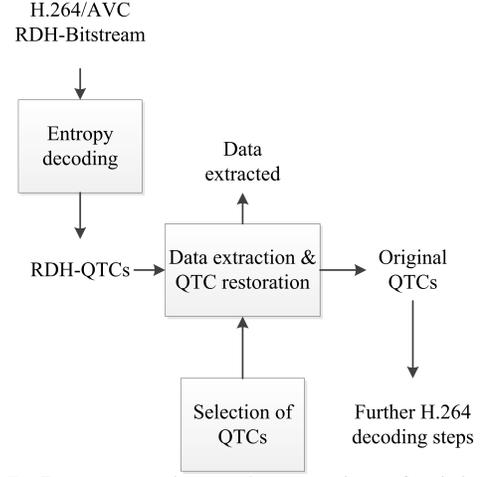


Figure 7: Data extraction and restoration of original QTCs for proposed scheme based on system of linear equations.

---

**Algorithm 2** Extraction of message bits and restoration of original QTCs from a stego-block.

---

```

1:  $k \leftarrow \text{SizeOfSolPatt}(QTC)$ 
2:  $\text{StegoQTC}[k] \leftarrow \text{SolOfLinEq}(QTC)$ 
3:  $\text{StegoBlk} \leftarrow 1$ 
4:  $l \leftarrow 0$ 
5: while  $l < k$  do
6:   if  $\text{StegoQTC}[l] < 0$  then
7:      $\text{StegoBlk} \leftarrow 0$  {Non-RDH block}
8:     Break
9:   end if
10:   $l \leftarrow (l + 1)$ 
11: end while
12: if  $\text{StegoBlk}$  then
13:   for all  $i = 1 \rightarrow k$  do
14:     if  $\text{StegoQTC}[i] < 2$  then
15:        $QTC[i] \leftarrow 0$ 
16:        $\text{MsgBit} \leftarrow \text{StegoQTC}[i]$ 
17:     else {Perform HS}
18:        $QTC[i] \leftarrow \text{StegoQTC}[i] - 1$ 
19:     end if
20:   end for
21: end if
22: end

```

---

## 4. SIMULATION RESULTS

This section presents evaluation of the proposed reversible data hiding algorithm. We have used a reference implementation based on the H.264/AVC reference software JM 12.2\*. Several benchmark video sequences in QCIF resolution ( $176 \times 144$ ) containing different combinations of motion, texture and objects have been used for analysis of payload, increase in bitrate and change in PSNR. We have performed embedding in *luma* component of 200 frames of each video sequence.

The proposed reversible data hiding technique utilizes a solution pattern which avoids drift and is a solution of system of linear equations. In a stego-block, RDH is performed by embedding of 1 bit in all the QTCs selected by the solution

---

\*<http://iphome.hhi.de/suehring/tml/>

pattern with 0 magnitude. While the magnitude of all the positive QTCs, which lie in solution pattern is incremented by 1. The solution pattern should be chosen in accordance with the payload requirements. If moderate payload is required, solution pattern should contain the QTCs which are: 1) low frequency, 2) less in number. Otherwise, we can have a solution pattern having higher number of high frequency QTCs. Following are the samples of solution patterns derived from a system of linear equations:

$$Sol_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad Sol_2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

For these simulation, we have selected the solution pattern  $Sol_1$  having 4 low-frequency QTCs. Moreover, we are embedding only in those blocks, which have at least one non-zero QTC in the solution pattern. Hence we are embedding at most 3 bits per  $4 \times 4$  block. If we are conscious about visual quality and need a lesser payload, we may reduce the embedding to 1 or 2 bit per frame. Higher payload can be attained if data hiding of up to 4 bits is allowed in a  $4 \times 4$  block, but it will have a severe effect on visual quality of video. Moreover,  $Sol_2$  will be more appropriate for higher payload, since  $Sol_2$  contains more QTCs than  $Sol_1$  and more bits can be embedded in a single  $4 \times 4$  block.

Table 1: Analysis of the proposed algorithm in terms of payload, bit-rate and PSNR at QP Values 30 for benchmark sequences.

sequence	PSNR (dB)		framesize (kbytes)		payload
	Orig.	RDH	Orig.	RDH	(bits/frame)
bus	33.20	26.68	4.64	4.87	721
city	33.45	27.89	3.58	3.76	595
crew	36.31	32.23	1.66	1.75	231
football	32.89	25.95	5.15	5.38	818
foreman	35.41	27.76	2.30	2.40	341
harbour	32.80	26.63	4.98	5.20	793
ice	36.72	30.48	2.27	2.35	222
mobile	33.23	27.00	7.99	8.10	431
soccer	33.47	28.89	3.02	3.17	509
avg.	34.16	28.17	3.95	4.12	518

Table 2: Analysis of the proposed algorithm over whole range of QP values for benchmark video sequence *football*.

QP	PSNR (dB)		framesize (kbytes)		payload
	Orig.	RDH	Orig.	RDH	(bits/frame)
12	50.57	46.28	19.42	19.51	124
18	44.62	40.29	13.09	13.18	308
24	38.54	27.73	8.81	8.98	665
30	32.89	25.95	5.15	5.38	818
36	28.47	23.29	2.64	2.78	499
42	24.49	22.14	1.25	1.35	240

Payload of the proposed scheme depends on type of video content and the QP value. The higher the texture in the video content, the higher is the payload. Table 1 shows payload, bit-rate increment and PSNR decrease for the benchmark sequences at QP value of 30. The original video content is fully restored if we extract the embedded data. The proposed RDH scheme has average payload of 518 bits per frame for video content of QCIF resolution. The average increase in bitrate is 4.28%. If the original video is not restored, the average decrease in PSNR is 5.99 dB, which is quite acceptable visual quality.

Video content can be encoded at different quality levels using different QP values. Hence, it is important to observe the performance of the proposed scheme on different QP values. Table 2 shows the analysis of our proposed scheme at

different QP values for benchmark video sequence *football*. It is evident that the proposed scheme conserves bit-rate and PSNR over whole range of QP values. Nevertheless, payload reduces with increase in QP value. It is because of the fact that we embed our watermark symbols only in blocks which contain at least one non-zero QTC in the solution pattern. The number of blocks having all four zero QTCs in the solution pattern increase with increase in the QP value.

Fig. 8 compares the visual quality of original and RDH video frames encoded at QP value of 30. It is evident that video quality is quite acceptable despite the fact that we are embedding the data in open-loop fashion (outside the reconstruction loop). It verifies that we have successfully minimized the effect of drift error. If we use a traditional reversible data hiding based on histogram shifting, the RDH video frame is quite distorted because of accumulation of drift errors.

In these experiments, we have embedded in solution pattern which has only 4 low-frequency QTCS in a  $4 \times 4$  block. We have selected a single solution to embed data in a reversible manner. Since the embedding position inside the  $4 \times 4$  block depends on the solutions of the linear system and payload is directly dependent on the solution pattern. The payload of the proposed scheme could be extended by using solution pattern having: 1) more QTCs for embedding (e.g.  $Sol_2$  in Eq. 1), 2) high frequency QTCs, since they have high probability of having '0' magnitude.

### 5. CONCLUSION

In this paper, we proposed an efficient high payload reversible data hiding algorithm for H.264/AVC in the compressed domain with a very controlled intra-drift. A system of linear equations is solved to determine additive data hiding patterns for the H.264  $4 \times 4$  transform coefficients that do not cause intra-drift in the pixel domain. The proposed scheme has a very controlled increase in bitrate, while the visual quality of the RDH video is quite acceptable. Another advantage of

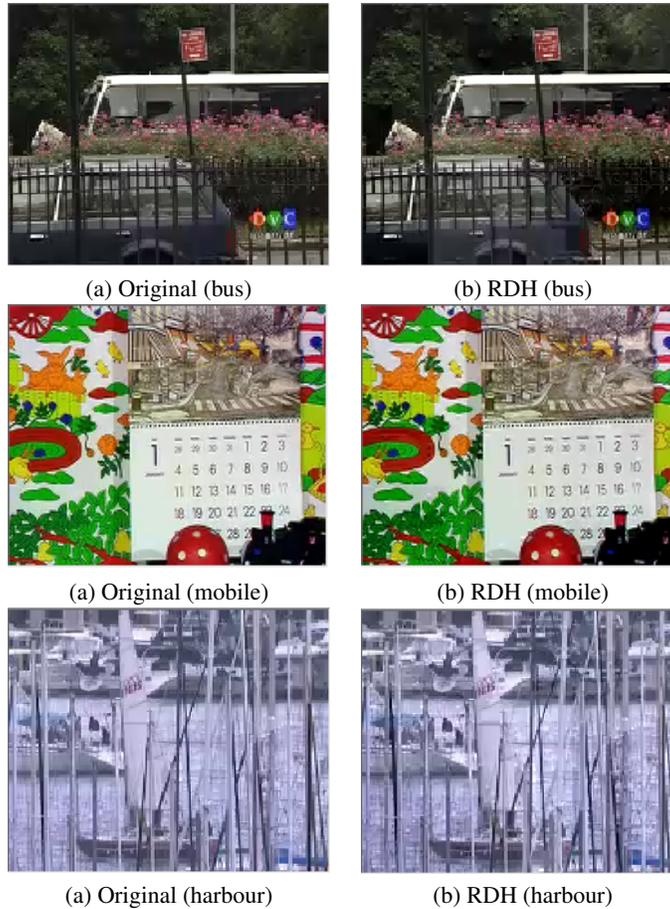


Figure 8: Visual quality of frame # 0 for benchmark video sequences at QP value of 30, which were: a) originally encoded, b) encoded with RDH.

our approach is its runtime-efficiency. Since the RDH is performed in an open-loop fashion, the proposed scheme is very suitable for the real-time applications, such as video streaming. In near future, we plan to enhance our algorithm both in terms of quality and capacity.

## REFERENCES

- [1] Shahid, Z., Chaumont, M., and Puech, W., "Considering the Reconstruction Loop for Data Hiding of Intra and Inter Frames of H.264/AVC," *Signal, Image and Video Processing, Springer* **5** (June 2011).
- [2] Chen, C.-C. and Kao, D.-S., "DCT-Based Reversible Image Watermarking Approach," in [*Proc. International Conference on Intelligent Information Hiding and Multimedia Signal Processing*], **2**, 489–492 (November 2007).
- [3] Hayat, K., Puech, W., and Gesquière, G., "Adaptively Synchronous Scalable Spread Spectrum (A4S) Data-Hiding Strategy for Three-Dimensional Visualization," *Journal of Electronic Imaging* **19**(2), 023011 (2010).
- [4] Puech, W., Chaumont, M., and Strauss, O., "A Reversible Data Hiding Method for Encrypted Images," in [*Proc. SPIE Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents*], (January 2008).
- [5] Lie, W., Lin, T., Tsai, D.-C., and Lin, G.-S., "Error Resilient Coding Based on Reversible Data Embedding Technique for H.264/AVC Video," in [*Proc. IEEE International Conference on Multimedia and Expo*], 1174–1177 (2005).
- [6] Chung, K. L., Huang, Y. H., Chang, P. C., and Liao, H. Y. M., "Reversible Data Hiding-Based Approach for Intra-Frame Error Concealment in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology* **20**, 1643–1647 (November 2010).
- [7] Lin, Y.-C. and Li, J.-H., "Reversible Watermarking for H.264/AVC Videos," *World Academy of Science, Engineering and Technology* **78**, 828–831 (2011).
- [8] Tian, J., "Reversible Data Embedding using a Difference Expansion," *IEEE Transactions on Circuits and Systems for Video Technology* **13**, 890–896 (August 2003).
- [9] Hartung, F. and Girod, B., "Watermarking of Uncompressed and Compressed Video," *Signal Processing* **66**, 283–301 (May 1998).
- [10] Zeng, X., Chen, Z., and Zhang, X., "Issues and Solution on Distortion Drift in Reversible Video Data Hiding," *Multimedia Tools and Applications* **52**, 465–484 (2011).
- [11] Fallahpour, M. and Megías, D., "Digital watermarking," ch. Reversible Data Hiding Based On H.264/AVC Intra Prediction, 52–60, Springer-Verlag, Berlin, Heidelberg (2009).
- [12] Liu, Y., Li, Z., and Ma, X., "Reversible Data Hiding Scheme Based On H.264/AVC without Distortion Drift," *Journal of Software* **7**, 1059–1065 (2012).