Multimedia Systems (2018) 24:95–109
https://doi.org/10.1007/s00530-016-0529-2

REGULAR PAPER

Three-dimensional histogram shifting for reversible data hiding

Juan Zhao[1] · **Zhitang Li**[2,3]

Received: 14 November 2014 / Accepted: 31 August 2016 / Published online: 15 September 2016

Abstract Histogram shifting is an important method of reversible data hiding. However, every pixel, difference, or prediction-error is respectively changed to hide a data bit in the traditional histogram shifting, which constrains the capacity-distortion embedding performance. An efficient three-dimensional histogram shifting is proposed for reversible data hiding in this paper. Take H.264 videos as covers to show this method. In a 4×4 quantized discrete cosine transform luminance block, which is not inferred by others, three alternating current coefficients are selected randomly as an embeddable group. According to the different values of the selected coefficient groups, they could be divided into different sets. Data could be hidden according to these sets. In the traditional histogram shifting, only one information bit could be hidden with at most one modification of one coefficient, whereas two data bits could be hidden at the same cost by using the proposed scheme. The superiority of the presented technique is verified through experiments.

Keywords Reversible data hiding · Three-dimensional histogram shifting · Reversible watermarking · Reversible steganography · H.264

Communicated by Q. Tian.

✉ Juan Zhao
lotusj@hust.edu.cn

[1] School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan 430023, China

[2] Network Center, Huazhong University of Science and Technology, Wuhan 430074, China

[3] National Engineering Laboratory for Next Generation Internet Access System, Wuhan 430074, China

1 Introduction

With the rapid development of video sharing technology, it is essential to research the algorithms of hiding data into videos for copyright protection, covert communication, integrity authentication, and so on [1, 2]. H.264 is a state-of-the-art video compression standard and has become the most widely deployed video codec. At the H.264 encoder, a residual pixel block is obtained by subtracting a prediction block from its original pixel block in YUV video. After lossy compression [discrete cosine transformation (DCT) and quantization], the residual block becomes a quantized DCT (QDCT) block, which has one direct current coefficient and some alternating current (AC) coefficients. After entropy encoding (lossless compression) of each QDCT macro block (MB), YUV video is encoded into an H.264 video. Therefore, the information hidden by changing QDCT coefficients can be fully extracted after entropy decoding. It is most common to embed data into QDCT coefficients, but the distortion caused by hiding data will spread and accumulate [3, 4]. Using general data hiding methods to embed information will cause the permanent distortion of host multimedia. However, hiding data into QDCT coefficients with a reversible data hiding (RDH) algorithm, we could completely restore the value of QDCT coefficient after extracting information, so we can save and enjoy important videos without information and distortion caused by hiding data. Consequently, there will not be too many network videos with secret information so that it is difficult for others to find stego cover. In addition, RDH methods can also be applied in video error concealment [5–8] and some sensitive application fields [9–13] such as multimedia archive management, medical multimedia sharing, military affair, remote sensing, and law enforcement.

In recent years, many RDH methods such as lossless compression [14–16], difference expansion [5, 6, 8, 11, 13, 17–22], histogram shifting (HS) [7, 23–34] and integer transform [10, 12] have been presented. In a RDH framework based on lossless compression [14], the compressible parts, which are extracted nondestructively from the original cover, are compressed by a lossless compression algorithm. Then the to-be-hidden information is attached to the back of the compressed parts. Correspondingly, the receiver extracts the information from the end of the sequence and recovers the original cover by decompressing the compressed parts. When this method is used to hide information, little embedding capacity and high computational complexity are two issues which should be resolved. Furthermore, in the lossless compression scheme [16], a recursive code construction and a lossless compression algorithm are used to hide data. However, it is not easy to use the recursive construction to hide information into H.264 video since this video is encoded by treating each MB sequentially [35]. In the difference expansion algorithm [17], the difference between two adjacent pixels in a pixel pair was expanded to hide one data bit. Moreover, prediction-error expansion is used to improve the hiding performance of difference expansion by expanding the difference between the pixel and its prediction [19].

The peak of image histogram is used to hide information in the HS method proposed by Ni et al. [23]. In order to hide one data bit, each pixel value is changed at most by adding or subtracting 1. Li et al. [31] proposed a general framework of HS-based RDH, which could be utilized to construct a RDH algorithm by simply designing shifting and embedding functions. In order to achieve a better capacity-distortion trade-off, all kinds of prediction methods are used to get sharp difference histograms [36].

However, in general HS-based RDH methods, each pixel, difference or prediction-error is singly changed for hiding a data bit, which constrains the capacity-distortion performance. In order to solve this limiting problem of the capacity-distortion performance and the embedding efficiency, an efficient RDH algorithm based on three-dimensional (3D) HS is proposed in this work. Stereo H.264 videos, encoded or decoded through multi-view coding (MVC), are just taken as covers in this paper. An arbitrary block, which does not predict others, is treated as an embeddable block, from which three QDCT AC coefficients are randomly chosen as an embeddable unit. Coefficient units are divided into disjoint regions. On the basis of the regions of coefficient units, the 3D histogram is expanded or shifted for hiding data reversibly. In order to embed two data bits, two coefficients may be modified in the conventional HS, whereas only one coefficient may be changed by using the proposed scheme. Compared with some state-of-the-art methods, the presented algorithm has superior payload-distortion performance, which is verified by the experimental results.

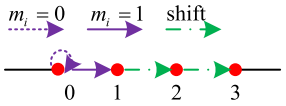

Fig. 1 Conventional HS

The rest of the paper is organized as follows: Section 2 presents the leading idea of 3D HS. In Sect. 3, the proposed RDH algorithm for MVC video and its implementation details are described. The hiding performance of the presented algorithm is evaluated via experimental results in Sect. 4. Finally, the conclusions of the paper are made in Sect. 5.

2 RDH method using HS

2.1 Conventional HS

Ni et al.'s HS method [16] could be used for hiding data into QDCT coefficients of MVC video. Denote a QDCT coefficient as F_{x1} and the marked QDCT coefficient as F'_{x1}. Information could be hidden by expanding and shifting one-dimensional (1D) histogram as shown in Fig. 1 and (1), where $m_i \in \{0, 1\}$ is a to-be-embedded data bit.

$$F'_{x1} = \begin{cases} 0, & \text{if } (F_{x1} = 0) \wedge (m_i = 0) \\ 1, & \text{if } (F_{x1} = 0) \wedge (m_i = 1) \\ F_{x1} + 1, & \text{if } F_{x1} > 0 \end{cases} \tag{1}$$

The positive coefficients are shifted for making vacant space. When the value of F_{x1} is 0, one data bit can be hidden, where the value of the coefficient F_{x1} will become 1 if m_i is 1, and will not be changed if m_i is 0. Accordingly, the hidden information m_i could be extracted from the marked QDCT coefficient F'_{x1}, and the value of QDCT coefficient can be completely restored as follows:

1. If $F'_{x1} = 0$, the extracted information bit $m_i = 0$ and the original coefficient $F_{x1} = 0$.
2. If $F'_{x1} = 1$, the extracted information bit $m_i = 1$ and the original coefficient $F_{x1} = 0$.
3. If $F'_{x1} > 1$, there is no hidden data in the coefficient and the original coefficient $F_{x1} = F'_{x1} - 1$.

In this method, the 1D coefficient histogram is defined by (2), where # is the cardinal number of a set, s_1 is a non-negative integer.

$$h(s_1) = \#\{F_{x1} | F_{x1} = s_1\} \tag{2}$$

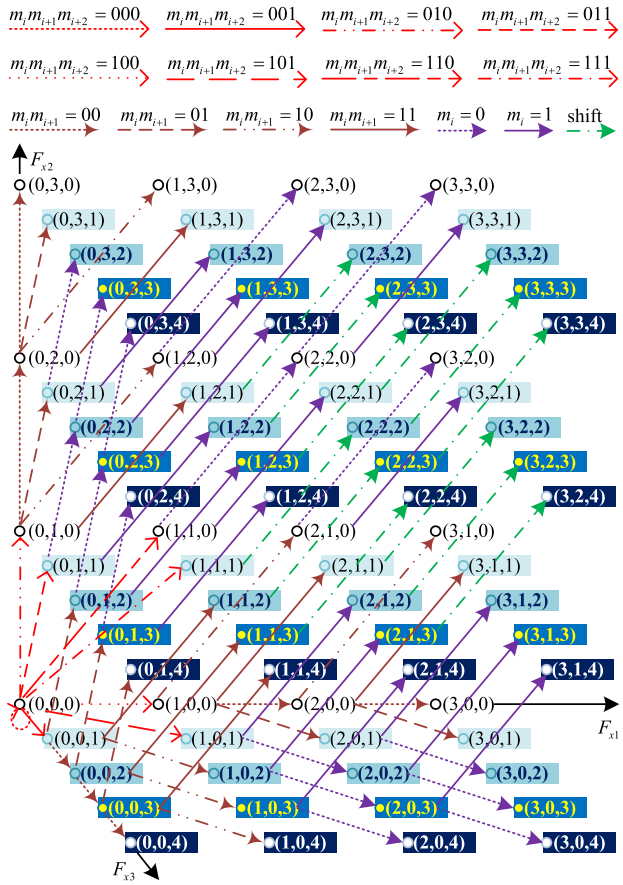

Fig. 2 Conventional 3D HS

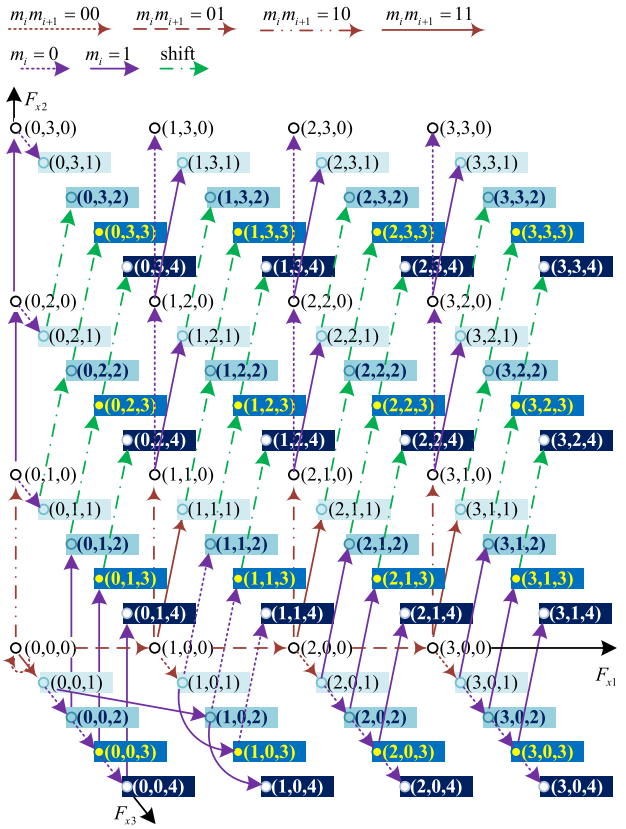

Fig. 3 Proposed 3D HS

If the scheme shown in Fig. 1 is used to embed data into each of the three QDCT coefficients denoted by F_{x1}, F_{x2}, and F_{x3}, the mapping will be a traditional 3D HS as shown in Fig. 2, where 3D histogram is defined as

$$w(s_1, s_2, s_3) = \#\{(F_{x1}, F_{x2}, F_{x3}) | F_{x1} = s_1, F_{x2} = s_2, F_{x3} = s_3\} \tag{3}$$

where s_2 and s_3 are nonnegative integers.

2.2 Proposed 3D HS

In histogram shifting schemes, different positions are used to represent different information. As shown in Fig. 2, eight adjacent places are needed to store eight kinds of three information bits (000, 001, 010, 011, 100, 101, 110, and 111), four adjacent positions are needed to represent four sorts of two information bits (00, 01, 10, and 11), and two neighboring positions are used to record two kinds of one data bit (0 and 1). When only one position could be used, no message can be hidden, and the original position should be shifted to its neighboring place. It can be observed that the maximum cost of each QDCT coefficient group in Fig. 2 is 3, which may bring obvious distortion.

In order to reduce the expense, we first search different positions to store different information with at most one change. If only nonnegative coefficient groups are used to store information, four positions [(0,0,0), (1,0,0), (0,1,0), and (0,0,1)] could be used for four sorts of two data bits. When the value of coefficient group (F_{x1}, F_{x2}, F_{x3}) is (0,0,0), which can be used to represent two data bits 00 with no modification, it could be expanded to its neighboring positions (1,0,0), (0,1,0), and (0,0,1) for signifying two data bits 01, 10, and 11 with one modification, respectively. When the value of coefficient group (F_{x1}, F_{x2}, F_{x3}) is (F_{x1},0,0) ($F_{x1} > 0$), it could be expanded to its neighboring positions (F_{x1}, 0, 1), (F_{x1} + 1,0,0), and (F_{x1}, 1,0) for signifying two data bits 00, 01, and 10 with one modification, respectively. And it can be expanded to (F_{x1}, 1, 1) for signifying two data bits 11, where the cost is 2. When F_{x1}, F_{x2}, F_{x3} are all positive integers, the group is shifted to (F_{x1}, F_{x2} + 1, F_{x3} + 1), where the cost is 2. In Fig. 2, by contrast, the group is shifted to (F_{x1} + 1, F_{x2} + 1, F_{x3} + 1), where the cost is 3. In this way, compared with the convention HS, we can get a more efficient 3D HS scheme, as shown in Fig. 3, where the set (denoted as J) of all the points could be divided into ten disjoint sets defined as follows:

$$J_1 = \{(0,\ 0, 0)\}$$
$$J_2 = \{(F_{x1}, 0, 0)|F_{x1} > 0\}$$
$$J_3 = \{(0, F_{x2}, 0)|F_{x2} > 0\}$$
$$J_4 = \{(0, 0, 1)\}$$
$$J_5 = \{(0, 0, F_{x3})|F_{x3} > 1\}$$
$$J_6 = \{(F_{x1}, F_{x2},\ 0)|F_{x1} > 0, F_{x2} > 0\}$$
$$J_7 = \{(1,\ 0, F_{x3})|F_{x3} > 0\}$$
$$J_8 = \{(F_{x1}, 0, F_{x3})|F_{x1} > 1, F_{x3} > 0\}$$
$$J_9 = \{(0, F_{x2}, F_{x3})|F_{x2} >\ 0, F_{x3} > 0\}$$
$$J_{10} = \{(F_{x1}, F_{x2}, F_{x3})|F_{x1} > 0, F_{x2} > 0, F_{x3} > 0\}$$

We divide the chosen coefficient groups into different sets and hide information based on the set the value of the coefficient group resides in. Accordingly, the embedding process can be described below.

1. if the coefficient group $(F_{x1},\ F_{x2},\ F_{x3}) \in J_1$, the marked coefficient group denoted by $(F'_{x1}, F'_{x2}, F'_{x3})$ will be

$$(F'_{x1}, F'_{x2}, F'_{x3}) = \begin{cases} (F_{x1}, F_{x2}, F_{x3}), & \text{if } m_i m_{i+1} = 00 \\ (F_{x1} + 1, F_{x2}, F_{x3}), & \text{if } m_i m_{i+1} = 01 \\ (F_{x1}, F_{x2} + 1, F_{x3}), & \text{if } m_i m_{i+1} = 10 \\ (F_{x1}, F_{x2}, F_{x3} + 1), & \text{if } m_i m_{i+1} = 11 \end{cases} \tag{4}$$

2. if the coefficient group $(F_{x1},\ F_{x2},\ F_{x3}) \in J_2$, the marked coefficient group $(F'_{x1}, F'_{x2}, F'_{x3})$ will be

$$(F'_{x1}, F'_{x2}, F'_{x3}) = \begin{cases} (F_{x1}, F_{x2}, F_{x3} + 1), & \text{if } m_i m_{i+1} = 00 \\ (F_{x1} + 1, F_{x2}, F_{x3}), & \text{if } m_i m_{i+1} = 01 \\ (F_{x1}, F_{x2} + 1, F_{x3}), & \text{if } m_i m_{i+1} = 10 \\ (F_{x1}, F_{x2} + 1, F_{x3} + 1), & \text{if } m_i m_{i+1} = 11 \end{cases} \tag{5}$$

3. if the coefficient group $(F_{x1},\ F_{x2},\ F_{x3}) \in J_3 \cup J_5$, the marked coefficient group $(F'_{x1}, F'_{x2}, F'_{x3})$ will be

$$(F'_{x1}, F'_{x2}, F'_{x3}) = \begin{cases} (F_{x1}, F_{x2}, F_{x3} + 1), & \text{if } m_i = 0 \\ (F_{x1}, F_{x2} + 1, F_{x3}), & \text{if } m_i = 1 \end{cases} \tag{6}$$

4. if the coefficient group $(F_{x1},\ F_{x2},\ F_{x3}) \in J_4$, the marked coefficient group $(F'_{x1}, F'_{x2}, F'_{x3})$ will be

$$(F'_{x1}, F'_{x2}, F'_{x3}) = \begin{cases} (F_{x1}, F_{x2}, F_{x3} + 1), & \text{if } m_i = 0 \\ (F_{x1} + 1, F_{x2}, F_{x3} + 1), & \text{if } m_i = 1 \end{cases} \tag{7}$$

5. if the coefficient group $(F_{x1},\ F_{x2},\ F_{x3}) \in J_6$, the marked coefficient group $(F'_{x1}, F'_{x2}, F'_{x3})$ will be

$$(F'_{x1}, F'_{x2}, F'_{x3}) = \begin{cases} (F_{x1}, F_{x2} + 1, F_{x3}), & \text{if } m_i = 0 \\ (F_{x1}, F_{x2} + 1, F_{x3} + 1), & \text{if } m_i = 1 \end{cases} \tag{8}$$

6. if the coefficient group $(F_{x1},\ F_{x2},\ F_{x3}) \in J_7$, the marked coefficient group $(F'_{x1}, F'_{x2}, F'_{x3})$ will be

$$(F'_{x1}, F'_{x2}, F'_{x3}) = \begin{cases} (F_{x1}, F_{x2} + 1, F_{x3} + 1), & \text{if } m_i = 0 \\ (F_{x1}, F_{x2}, F_{x3} + 2), & \text{if } m_i = 1 \end{cases} \tag{9}$$

7. if the coefficient group $(F_{x1},\ F_{x2},\ F_{x3}) \in J_8$, the marked coefficient group $(F'_{x1}, F'_{x2}, F'_{x3})$ will be

$$(F'_{x1}, F'_{x2}, F'_{x3}) = \begin{cases} (F_{x1}, F_{x2}, F_{x3} + 1), & \text{if } m_i = 0 \\ (F_{x1}, F_{x2} + 1, F_{x3} + 1), & \text{if } m_i = 1 \end{cases} \tag{10}$$

8. if the coefficient group $(F_{x1},\ F_{x2},\ F_{x3}) \in J_9 \cup J_{10}$, no information is hidden, and the marked coefficient group $(F'_{x1}, F'_{x2}, F'_{x3})$ will be taken as $(F_{x1},\ F_{x2} + 1,\ F_{x3} + 1)$.

After data is hidden by using these mapping rules, we can extract the information based on the set which the value of the marked coefficient group may reside in, and recover the value of the marked coefficient group according to the reverse process of embedding.

2.3 Embedding capacity and distortion

When the 1D HS is used for hiding data, the embedding capacity denoted as EC is $h(0)$. For QDCT coefficients, the embedding distortion denoted as ED in terms of l^2-error can be formulated as

$$\text{ED} = \frac{1}{2}h(0) + \sum_{s=1}^{+\infty} h(s_1) \tag{11}$$

The embedding capacities of the conventional 3D HS and the proposed 3D HS, denoted as EC_{con} and EC_{pro}, can be calculated by (12) and (13).

$$\text{EC}_{\text{con}} = 3 \sum_{(F_{x1},F_{x2},F_{x3}) \in J_1} w(F_{x1}, F_{x2}, F_{x3}) + 2 \sum_{(F_{x1},F_{x2},F_{x3}) \in J_2 \cup J_3 \cup J_4 \cup J_5} w(F_{x1}, F_{x2}, F_{x3}) + \sum_{(F_{x1},F_{x2},F_{x3}) \in J_6 \cup J_7 \cup J_8 \cup J_9} w(F_{x1}, F_{x2}, F_{x3}) \tag{12}$$

$$\text{EC}_{\text{pro}} = 2 \sum_{(F_{x1},F_{x2},F_{x3}) \in J_1 \cup J_2} w(F_{x1}, F_{x2}, F_{x3}) + \sum_{(F_{x1},F_{x2},F_{x3}) \in J_3 \cup J_4 \cup J_5 \cup J_6 \cup J_7 \cup J_8} w(F_{x1}, F_{x2}, F_{x3}) \tag{13}$$

For QDCT coefficients, the embedding distortion in terms of l^2-error of the conventional 3D HS and the proposed 3D HS, denoted as ED_{con} and ED_{pro}, can be formulated as

$$\begin{aligned} ED_{con} = & \frac{3}{2} \sum_{(F_{x1},F_{x2},F_{x3})\in J_1} w(F_{x1},F_{x2},F_{x3}) \\ & + 2 \sum_{(F_{x1},F_{x2},F_{x3})\in J_2\cup J_3\cup J_4\cup J_5} w(F_{x1},F_{x2},F_{x3}) \\ & + \frac{5}{2} \sum_{(F_{x1},F_{x2},F_{x3})\in J_6\cup J_7\cup J_8\cup J_9} w(F_{x1},F_{x2},F_{x3}) \\ & + 3 \sum_{(F_{x1},F_{x2},F_{x3})\in J_{10}} w(F_{x1},F_{x2},F_{x3}) \end{aligned} \tag{14}$$

and

$$\begin{aligned} ED_{pro} = & \frac{3}{4} \sum_{(F_{x1},F_{x2},F_{x3})\in J_1} w(F_{x1},F_{x2},F_{x3}) \\ & + \frac{5}{4} \sum_{(F_{x1},F_{x2},F_{x3})\in J_2} w(F_{x1},F_{x2},F_{x3}) \\ & + \sum_{(F_{x1},F_{x2},F_{x3})\in J_3\cup J_5} w(F_{x1},F_{x2},F_{x3}) \\ & + \frac{3}{2} \sum_{(F_{x1},F_{x2},F_{x3})\in J_4\cup J_6\cup J_8} w(F_{x1},F_{x2},F_{x3}) \\ & + 2 \sum_{(F_{x1},F_{x2},F_{x3})\in J_7\cup J_9\cup J_{10}} w(F_{x1},F_{x2},F_{x3}) \end{aligned} \tag{15}$$

According to (12) and (13), it can be inferred that the difference of embedding capacity between the presented 3D HS and the conventional 3D HS is

$$EC_{con} - EC_{pro} = \sum_{(F_{x1},F_{x2},F_{x3})\in J_1\cup J_3\cup J_4\cup J_5\cup J_9} w(F_{x1},F_{x2},F_{x3}) \tag{16}$$

According to (14) and (15), it can be inferred that the difference of embedding distortion between the presented 3D HS and the conventional 3D HS is

$$\begin{aligned} ED_{con} - ED_{pro} = & \frac{3}{4} \sum_{(F_{x1},F_{x2},F_{x3})\in J_1\cup J_2} w(F_{x1},F_{x2},F_{x3}) \\ & + \sum_{(F_{x1},F_{x2},F_{x3})\in J_3\cup J_5\cup J_6\cup J_8\cup J_{10}} w(F_{x1},F_{x2},F_{x3}) \\ & + \frac{1}{2} \sum_{(F_{x1},F_{x2},F_{x3})\in J_4\cup J_7\cup J_9} w(F_{x1},F_{x2},F_{x3}) \end{aligned} \tag{17}$$

Therefore, compared with the conventional 3D HS, although the capacity obtained by our method is lower, the distortion is decreased greatly. Two examples are given to show the advantage of the proposed method.

1. For the group $(F_{x1}, F_{x2}, F_{x3}) = (0, 0, 0) \in J_1$, in the proposed 3D HS, the distortion is 0, 1, 1 and 1 when (m_i, m_{i+1}) is (0, 0), (0, 1), (1, 0), and (1, 1), respectively. However, in the conventional 3D HS, the distortion is 2 if (m_i, m_{i+1}) is (1, 1). Therefore, it can be inferred that when the quantity of data hidden by the proposed method is the same as that hidden by the conventional 3D HS, the proposed method could achieve preferable quality compared with the traditional 3D HS.
2. For the group $(F_{x1}, F_{x2}, F_{x3}) = (2, 0, 0) \in J_2$, in the proposed 3D HS, the cost is 1, 1, 1 and 2 when (m_i, m_{i+1}) is (0, 0), (0, 1), (1, 0), and (1, 1), respectively. In the conventional 3D HS, the cost is 1, 2, 2 and 3, respectively. Accordingly, for the coefficient groups in the set J_2, in order to hide the same number of information, the presented 3D HS can be used to obtain better cover quality compared with the conventional 3D HS.

In general, when the coefficient group $(F_{x1}, F_{x2}, F_{x3}) \in J_2$, the same embedding capacity, i.e., $2\sum_{(F_{x1},F_{x2},F_{x3})\in J_2} w(F_{x1},F_{x2},F_{x3})$, can be acquired by the two methods, but lower distortion will be caused by using the proposed scheme, i.e. $\frac{5}{4}\sum_{(F_{x1},F_{x2},F_{x3})\in J_2} w(F_{x1},F_{x2},F_{x3}) < 2\sum_{(F_{x1},F_{x2},F_{x3})\in J_2} w(F_{x1},F_{x2},F_{x3})$. When the coefficient group $(F_{x1}, F_{x2}, F_{x3}) \in J_1$, the proposed method's efficiency (which is capacity/distortion) is $\left[2\sum_{(F_{x1},F_{x2},F_{x3})\in J_1} w(F_{x1},F_{x2},F_{x3})\right] / \left[\frac{3}{4}\sum_{(F_{x1},F_{x2},F_{x3})\in J_1} w(F_{x1},F_{x2},F_{x3})\right] = \frac{8}{3}$, and the conventional method's efficiency is $\left[3\sum_{(F_{x1},F_{x2},F_{x3})\in J_1} w(F_{x1},F_{x2},F_{x3})\right] / \left[\frac{3}{2}\sum_{(F_{x1},F_{x2},F_{x3})\in J_1} w(F_{x1},F_{x2},F_{x3})\right] = 2 < \frac{8}{3}$. Similarly, if the coefficient group (F_{x1}, F_{x2}, F_{x3}) belongs to other sets, when the same quantity of data is hidden, the better video quality and hiding efficiency can be achieved by the proposed 3D HS compared with the conventional 3D HS.

Additionally, the presented scheme can be applied in the image or the video RDH algorithms since the difference or the prediction-error histogram is similar to the coefficient histogram. When the presented scheme is used in some media, especially a gray-scale image with 8 storage bits [24], the overflow/underflow problem should be treated. However, this problem need not be considered when the information is embedded into QDCT coefficients of H.264 video [6].

3 The proposed RDH algorithm for MVC video

3.1 Embeddable blocks limiting distortion drift

The original YUV videos captured by cameras should be compressed to decrease the network transmission load. In order to diminish the spatial redundancy of video sequences,

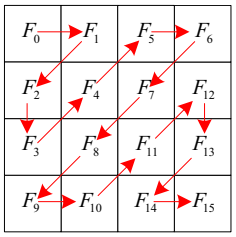

Fig. 4 A 4 × 4 QDCT block with zigzag scan

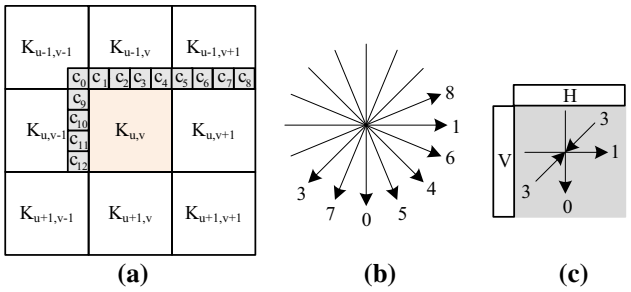

Fig. 5 Intra-frame prediction mode: **a** block position, **b** the predictive direction of 4 × 4 and 8 × 8 luma block and **c** the predictive direction of 16 × 16 luma block. In mode 2 not shown in the figure, all elements are predicted with the average of upper pixels denoted by H and left pixels denoted as V, i.e., mean $(H + V)$

parallax prediction, inter-frame prediction, and intra-frame prediction are employed in MVC standard to calculate prediction block. Then the prediction block is subtracted from the original block in the YUV video at MVC encoder, where the residuary block denoted as K^{R0} undergoes 4 × 4 (or 8 × 8) DCT and quantization as shown in

$$F = round\left[(C_f K^{R0} C_f^T) \otimes (E_f/Q)\right] \tag{18}$$

where F is a QDCT block with 16 QDCT coefficients numbered by zigzag scan as shown in Fig. 4,

$$C_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, \quad E_f = \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}, \quad a = 1/2,\ b = \sqrt{2/5},$$

the matrix C_f^T is the transpose of C_f, Q is the quantization step size, $\otimes$ is a mathematical operator, which indicates that each value in the former matrix is multiplied by the value at the corresponding position in the latter matrix.

If one data bit is hidden into one QDCT block F by changing some QDCT coefficients, the QDCT block F will be altered to a marked block denoted by F', and the deviation (denoted as ΔF) introduced by hiding data is

$$\Delta F = F' - F \tag{19}$$

In order to reconstruct YUV videos that the users watch on the screen, at the decoder, the prediction block is computed and added to the residual block denoted by K^R, which is achieved by lossless decompression (entropy decoding) and lossy decompression (inverse quantization and inverse 4 × 4 (or 8 × 8) DCT) as shown in

$$K^{\mathrm{R}} = round\left[C_d^T (F \otimes E_d) C_d\right] \tag{20}$$

where

$$C_d = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}, \quad E_d = \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}.$$

When a data bit is hidden through modifying some QDCT coefficients of a block, the residual pixel block K^R will be turned into a marked pixel block denoted as $K^{R'}$, and the mutation denoted by ΔK^R is

$$\Delta K^R = K^{R'} - K^R = round\left[C_d^T (\Delta F \cdot Q \otimes E_d) C_d\right]. \tag{21}$$

Take the QDCT coefficient F_{13} for example to show the distortion caused by embedding information. Suppose we add an integer denoted as r to the value of F_{13}, that is, the change of the QDCT block for embedding information is

$\Delta F = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 0 \end{bmatrix}$, then the change of corresponding pixel block in YUV is $\Delta K^R = \frac{1}{2} Qabr \begin{bmatrix} 1 & -2 & 2 & -1 \\ -1 & 2 & -2 & 1 \\ -1 & 2 & -2 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$.

It can be seen that the modification of one QDCT coefficient causes the distortion of the whole 4 × 4 transform block in the corresponding YUV video. Similarly, altering one QDCT coefficient in an 8 × 8 transform block will vary the whole 8 × 8 pixel block, whose range is bigger than the range of 4 × 4 block. Thus 4 × 4 transform blocks are chosen for hiding data in this paper.

Correspondingly, it can be inferred that the boundary pixels denoted as $c_0 \ldots c_{12}$, shown in Fig. 5, may be modified by embedding information into some QDCT coefficients of the blocks $K_{u,v-1}$ (integers u and v are used to denote the position of a block), $K_{u-1,v-1}$, $K_{u-1,v}$, and $K_{u-1,v+1}$. Additionally, if intra-frame prediction is utilized by the current block $K_{u,v}$, its prediction block will be calculated by the pixels $c_0 \ldots c_{12}$. Consequently, the embedding induced deviation of the blocks $K_{u,v-1}$, $K_{u-1,v-1}$, $K_{u-1,v}$, and $K_{u-1,v+1}$ will drift to the block $K_{u,v}$. Otherwise, when the prediction block of the block $K_{u,v}$ is counted by using inter-frame prediction or parallax prediction, i.e., referring another frame as shown in Fig. 6, any modification of

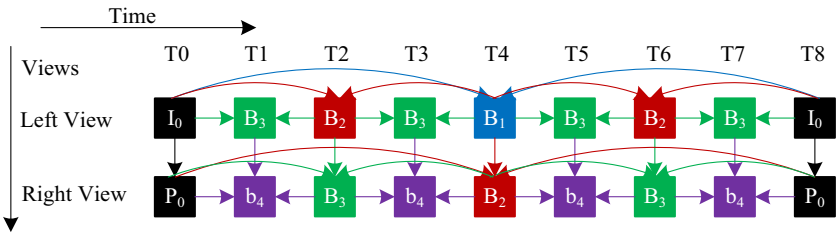

Fig. 6 Prediction structure of MVC video with two views

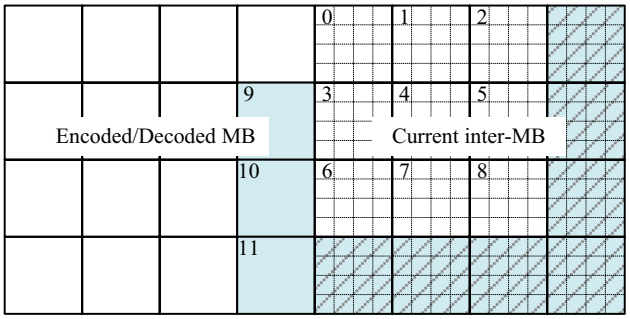

Fig. 7 4 × 4 blocks without intra-frame distortion drift

adjacent blocks in the same frame will not have an impact on the block $K_{u,v}$.

A 16 × 16 MB with inter-frame prediction or parallax prediction is denoted as inter-MB. If the current block $K_{u,v}$ is one of the nine 4 × 4 blocks numbered by 0…8, which are not located at the bottom or the rightmost column of the inter-MB as shown in Fig. 7, its adjacent blocks $K_{u,v+1}$, $K_{u+1,v+1}$ and $K_{u+1,v}$ will be in the current inter-MB. In addition, its neighboring block $K_{u+1,v-1}$ may be either in the current inter-MB or one of the blocks numbered by 9, 10, and 11 in the encoded MB at the encoder (or the decoded MB at the decoder). Therefore, these adjacent blocks $K_{u,v+1}$, $K_{u+1,v+1}$, $K_{u+1,v}$, and $K_{u+1,v-1}$ will not be influenced by the block $K_{u,v}$, and the blocks numbered 0…8 in an inter-MB could be chosen as embeddable blocks to embed data without causing intra-frame distortion drift.

Besides intra-frame distortion drift, the inter-frame and the parallax distortion drift will also decrease the video quality. As illustrated in Fig. 6, hierarchical B coding is used in the prediction scheme of MVC video with two views. For one group of picture (GOP) with 16 frames, there are eight frames in each view. The horizontal prediction is inter-frame, and the vertical prediction is parallax. I_0 frame and P_0 frame are pivotal pictures at the highest level. Only intra-frame prediction is used for I_0 frame so that it will not be affected by hiding data into other frames, but hiding data into an I_0 frame will infect all the P_0, B_1, B_2, B_3, and b_4 frames in the two GOPs predicted by I_0 frame. In contrast, hiding information into P_0 or b_4 frames in the right view will not cause parallax distortion drift as P_0 or b_4 frames are not referred by the frames in the left view, where hiding information into b_4 frames also will not cause inter-frame distortion drift. In addition, only b_4 frames may be affected by hiding data into B_3 frames, and six B_3 frames in one GOP could afford enough redundancy space. Accordingly, compared with hiding data into I_0 frame, better video quality could be obtained by hiding information into P_0, B_3 or b_4 frames, which could be selected by users on demand.

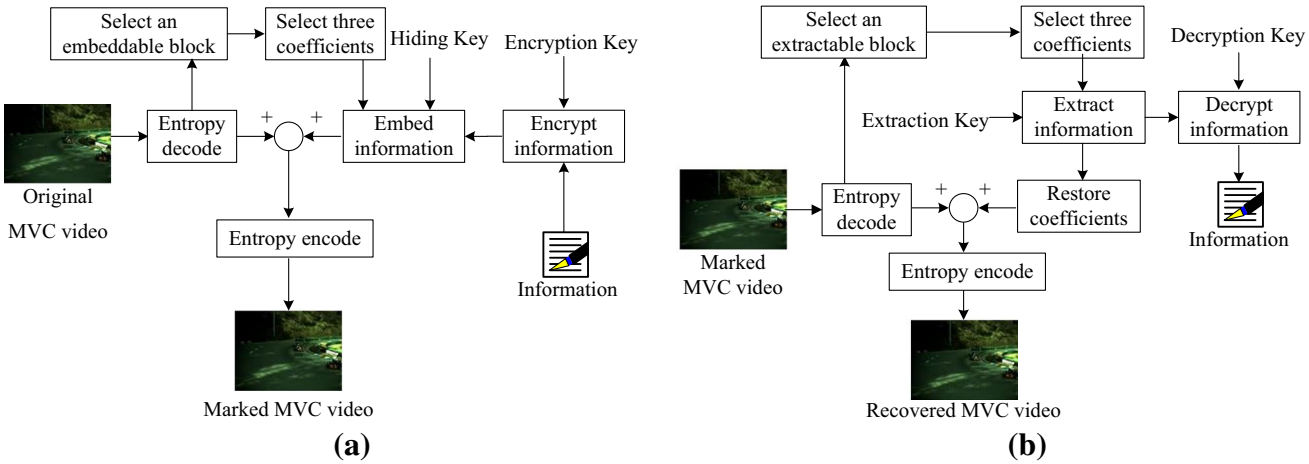

Fig. 8 The flowchart of the presented RDH algorithm for MVC video: **a** information embedding and **b** information extraction and video recovery

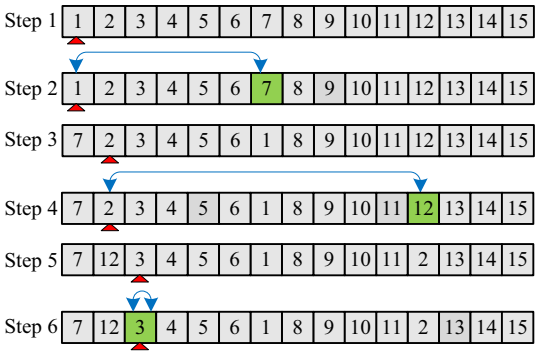

Fig. 9 The random selection of three embeddable coefficients

3.2 Embedding procedure

The presented RDH algorithm for MVC video is shown in Fig. 8. The sender entropy decodes the MVC video to choose embeddable blocks from some QDCT inter-MB, where the MBs not selected for hiding data will be entropy encoded directly. The information is hidden into three QDCT coefficients (F_{x1}, F_{x2}, F_{x3}) chosen from each embeddable 4 × 4 block. The marked MVC video will be gotten by entropy encoding each MB after hiding data. Correspondingly, the receiver could entropy decode the marked MVC video to extract the embedded data from the marked QDCT coefficients that could be recovered completely later.

The way of selecting three QDCT coefficients (F_{x1}, F_{x2}, F_{x3}) from a 4 × 4 luminance block is shown in Algorithm 1. Random function is utilized to choose three coefficients from 15 AC coefficients in a block with zigzag scan described in Fig. 4. Figure 9 shows an example of selecting three embeddable coefficients.

Algorithm 1 Random selection of three QDCT coefficients.

Input: Random seed S
Output: Three QDCT coefficients (F_{x1}, F_{x2}, F_{x3})
Initialize the random seed with srand(S);
For i=1 to 15 do
  Assign a value to the ith element of the integer array X[] with $X[i]=i$;
End for
For i=1 to 3 do
  Generate the ith random number with xi=rand()%(16-i)+i;
  If xi!=i
    Swap the positions of $X[xi]$ and $X[i]$;
  End if
End for
Three QDCT coefficients (F_{x1}, F_{x2}, F_{x3}) are chosen as an embeddable coefficient group.

Step 1 The cursor starts from the first position marked by 1
Step 2 When 7 is chosen randomly, the embeddable coefficient F_{x1} is F_7, and the positions of 7 and 1 pointed by the cursor are swapped
Step 3 Move the cursor forward to point at 2
Step 4 If 12 is selected at random, the embeddable coefficient F_{x2} is F_{12}, and the places of 12 and 2 are swapped
Step 5 Move the cursor forward to point at 3
Step 6 When 3 is selected randomly, the embeddable coefficient F_{x3} is F_3. Therefore, the selected coefficient group (F_{x1}, F_{x2}, F_{x3}) is (F_7, F_{12}, F_3)

It can be seen that there are 15 ways to choose F_{x1} from 15 QDCT AC coefficients, 14 ways to select F_{x2} from the rest 14 QDCT AC coefficients, and 13 ways to select F_{x3} from the rest 13 QDCT AC coefficients. Accordingly, the optional quantity of selecting (F_{x1}, F_{x2}, F_{x3}) is $15 \times 14 \times 13 = 2730$. When a marked block is found by the third party, the probability for directly guessing the hidden data bit is $1/2730 \approx 3.66 \times 10^{-4}$.

It is more difficult to find small area of distortion compared with large area of distortion. Therefore, it is necessary to limit the distortion region in a MB. In the hiding procedure shown in Algorithm 2, an embeddable 4 × 4 block, which could be used to hide data without causing intra-frame distortion drift, is selected randomly from 9 blocks shown in Fig. 7. In this way, only one 4 × 4 block may be modified for hiding data in one MB. In addition, a positive integer denoted as Z is set to generate a random threshold denoted by U so that we can randomly select embeddable blocks according to $|F_0| \geq U$. High threshold U will constrain the quantity of embeddable blocks, so the distortion region will be limited. Consequently, the application of arbitrary embeddable positions including blocks and coefficients could be employed to reduce the distortion of statistical histogram and enhance the undetectability of RDH scheme.

Algorithm 2 Embedding procedure of the proposed RDH algorithm for MVC video.

Input: Original MVC video, Message M, RSA public key, Seed S.
Output: The marked MVC video.
Encrypt the seed S with a RSA public key;
Hide the seed S into I_0 frame with a public RDH algorithm;
Encrypt the message M with a RSA public key;
Compress the message with a lossless compression algorithm into $M'=\{m_i\}$ where $m_i \in \{0,1\}$;
Do
  For each P_0, B_3 or b_4 frame do
    For each inter-MB with 4×4 luma transformation do
      Generate a random block number g=rand()%9;
      Generate a random threshold by U=rand()%Z;
      If the current coefficient F_0 of the gth 4×4 QDCT block (shown in Fig.7) satisfies $|F_0| \geq U$
        Select three QDCT coefficients (F_{x1}, F_{x2}, F_{x3}) with Algorithm 1;
        Embed information by following Fig.3 and (7);
      End if
    End for
  End for
While M' is not embedded completely.

3.3 Extraction and recovery procedures

Algorithm 3 demonstrates the procedure of information extraction and video restoration. The same random seed can be used to generate some same random sequences. Therefore, when the sender and receiver use the same random seed S, the embeddable QDCT blocks and coefficients employed by the sender can be in one-to-one correspondence with the extractable QDCT blocks and AC coefficients utilized by the receiver. Finally, the receiver could extract the embedded data and restore the video fully according to the reverse process of Fig. 3.

In addition, the computational efficiency of the presented RDH algorithm depends on the video frame number denoted by N_F and the information length denoted by L_I. Therefore, the computational complexity of the presented algorithm can be denoted by O ($N_F \times L_I$).

Algorithm 3 Extraction procedure of the presented RDH algorithm for MVC video.

Input: A marked MVC video, RSA private key.
Output: The recovered MVC video and message M.
Extract the seed S from I_0 frame with the public RDH algorithm;
Decrypt the seed S with a RSA private key;
Do
  For each P_0, B_3 or b_4 frame do
    For each inter-MB with 4×4 luma transformation do
      Generate a random block number g=rand()%9;
      Generate an integer threshold using U=rand()%Z;
      If the current coefficient F_0 of the gth 4×4 block (shown in Fig.7) satisfies $|F_0| \geq U$
        Select three QDCT coefficients (F_{x1}, F_{x2}, F_{x3}) by using Algorithm 1;
        Extract information and recover the video by following the reverse process of Fig.3;
      End if
    End for
  End for
While M' is not extracted completely
Decompress the message M';
Decrypt the message into M with the RSA private key.

4 Experimental results and discussions

The presented algorithm has been effectuated in the H.264 reference software version JM18.4 [37]. The nine video sequences (640 × 480) [38] shown in Fig. 10 act as test samples. Two YUV files are encoded to a MVC video with 233 frames, which include 30 I_0 frames, 30 P_0 frames and 116 b_4 frames. The parameter intra-period was set as 8. The capacity of a video sequence is the mean number of bits hidden in one I_0/P_0/b_4 frame of all the I_0/P_0/b_4 frames in that sequence. The peak signal-to-noise ratio (PSNR) value and the structural similarity (SSIM) value, which are achieved by comparing the marked YUV video with the original YUV video, are the averages of all the frames. The embedding efficiency e is defined as

$$e = L_{hide} / \sum_{i=1}^{i=L_{modi}} Lcha_i, \tag{22}$$

where L_{hide} is the number of hidden bits, and L_{modi} is the number of modified bits, and $Lcha_i$ is the changed size of a modified coefficient.

If the parameter code block pattern of a block is 0, there is no QDCT coefficient hoarded in the block which only contains zero coefficients actually. Therefore, this block could not be modified for hiding information so that large visual distortion can be eliminated. The distribution of changeable QDCT coefficients in blocks with nonzero code block pattern is shown in Table 1. The schemes hiding data into P_0 and b_4 frame with intra-frame distortion drift are denoted as P_0_drift and b_4_drift. The schemes embedding information into P_0 and b_4 frame without intra-frame distortion drift are denoted as P_0_interMB and b_4_interMB, in which only embeddable blocks 0…8 shown in Fig. 7 are considered. The probability of changeable zero coefficients is denoted as p_0. For P_0_drift, p_0 is about 0.925. For P_0_interMB, p_0 is about 0.927. For b_4_drift, p_0 is about 0.935. For b_4_interMB, p_0 is about 0.937. The overwhelming majority of changeable QDCT coefficients are zero, which indicates that the peak of the QDCT coefficient histogram is rather steep. Thus, fine payload-distortion performance can be obtained by using HS to embed data. Li et al.'s two-dimensional HS method [33] could be used for embedding information into QDCT coefficients of MVC video. When the value of coefficient pair (F_{x1}, F_{x2}) meets $F_{x1}\ F_{x2} = 0$, it is expanded to its neighboring positions ($F_{x1} + 1$, $F_{x2} - 1$) or ($F_{x1} - 1$, $F_{x2} + 1$) for signifying one data bit 1, where the cost is 2. In order to hide two data bits 11 into zero coefficients, the cost is 4, whereas the cost is 1 by using our method that takes full advantage of the coefficient distribution.

Table 2 shows the embedding performance of four schemes where the threshold U is 0. Different hiding capacities can be achieved by hiding data into different videos. Therefore, different amounts of data are hidden into different videos. The proposed 3D-HS-based RDH method is used for hiding data in these schemes except Ma et al.'s scheme [3], which is denoted by Ma. Compared with Ma, where data is hidden into I_0 frame without causing intra-frame distortion drift, P_0_drift increases PSNR, SSIM and embedding efficiency e with 0.163 dB, 0.00005, and 1.2 for hiding 500 bits of data in one frame of Crowd, respectively. Compared with P_0_drift, PSNR, SSIM and embedding efficiency e can be enhanced with 0.003 dB, 0.00013 and 0.03 by P_0 _interMB for Crowd, respectively. Compared with P_0_ interMB, PSNR, SSIM and embedding efficiency e can be enhanced with 0.083 dB, 0.00033 and 0.05 by b_4_ interMB for Crowd, respectively. On the whole, compared with Ma, b_4_interMB is superior by enhancing PSNR,

Fig. 10 Test videos (the size of each frame is 640 × 480). **a** Akko&Kayo, **b** Ballroom, **c** Crowd, **d** Exit, **e** Flamenco, **f** Objects, **g** Race, **h** Rena and **i** Vassar

Table 1 Average numbers of different QDCT coefficient values in one frame of Ballroom

Schemes	The value of changeable QDCT coefficient										
	−5	−4	−3	−2	−1	0	1	2	3	4	5
P_0_drift	3	6	16	64	634	22,513	907	128	29	11	5
P_0_interMB	1	3	8	32	340	12,570	505	69	16	6	3
b_4_drift	1	3	7	28	163	6586	205	35	10	4	2
b_4_interMB	0	1	3	14	89	3697	112	18	5	2	1

SSIM, and embedding efficiency e with 0.156 dB, 0.00008 and 0.71 at least, respectively.

In Table 3, three QDCT AC coefficients F_2, F_5, and F_3 are used for hiding information. Compared with the conventional 3D HS, the presented 3D HS is better by improving PSNR, SSIM, and embedding efficiency e with 0.02 dB, 0.00004, and 0.41 at least, respectively. Accordingly, the marked frame of Akko&Kayo, Exit and Race are shown in Fig. 11. Figures (a–c) are the marked frame obtained by employing the conventional 3D HS to embed data. The rest figures are the marked frame achieved by using the proposed 3D HS to hide information. Their original frames are shown in Fig. 10. It is easy to find apparent distortion in the frames (a–c). There is a large distortion on the top left corner of frame (a). Many squares can be seen on the back of the man in frame (b). On the top middle of frame (c), the distortion is on the trees. By contrast, little distortion could be seen in the frames (d–f). The experimental results verify that superior visual quality could be obtained by using the proposed 3D HS method to hide information.

Table 2 Hiding performance of four schemes for hiding data into one frame on average

Sequences	Amount of hidden data		Ma	P_0_drift	P_0 _interMB	b_4_interMB
Akko&Kayo	100	PSNR(dB)	39.438	39.573	39.574	39.725
		SSIM	0.96808	0.96810	0.96816	0.96837
		e	1.62	2.58	2.59	2.61
Ballroom	400	PSNR(dB)	36.728	36.895	36.899	36.986
		SSIM	0.94510	0.94516	0.94527	0.94556
		e	1.11	2.36	2.38	2.39
Crowd	500	PSNR(dB)	35.606	35.769	35.772	35.855
		SSIM	0.96897	0.96902	0.96915	0.96948
		e	1.13	2.33	2.36	2.41
Exit	60	PSNR(dB)	38.174	38.305	38.307	38.425
		SSIM	0.93608	0.93609	0.93610	0.93626
		e	1.35	2.48	2.49	2.54
Flamenco	300	PSNR(dB)	39.306	39.457	39.460	39.477
		SSIM	0.97160	0.97164	0.97171	0.97180
		e	1.55	2.44	2.46	2.49
Objects	50	PSNR(dB)	37.598	37.710	37.711	37.808
		SSIM	0.97504	0.97505	0.97506	0.97520
		e	1.71	2.30	2.29	2.42
Race	140	PSNR(dB)	37.470	37.614	37.616	37.626
		SSIM	0.96143	0.96146	0.96148	0.96150
		e	1.14	2.27	2.28	2.30
Rena	40	PSNR(dB)	41.409	41.517	41.518	41.590
		SSIM	0.97424	0.97425	0.97426	0.97437
		e	1.43	2.63	2.63	2.69
Vassar	70	PSNR(dB)	36.298	36.437	36.438	36.651
		SSIM	0.91558	0.91559	0.91561	0.91580
		e	1.22	2.35	2.36	2.41

Table 3 Embedding performance of the conventional 3D HS and the presented 3D HS for hiding 500 bits of information into the first P_0 frame

Sequences	Akko&Kayo	Ballroom	Crowd	Exit	Objects	Race	Rena	Vassar
Conventional 3D HS								
PSNR (dB)	40.51	37.58	37.50	39.06	37.62	39.94	42.68	37.24
SSIM	0.97144	0.94853	0.97749	0.94295	0.97395	0.96648	0.97616	0.92184
e	1.95	1.85	1.74	1.76	1.59	1.95	1.95	1.70
Presented 3D HS								
PSNR (dB)	40.56	37.60	37.52	39.09	37.65	39.98	42.77	37.25
SSIM	0.97171	0.94979	0.97753	0.94305	0.97409	0.96675	0.97637	0.92198
e	2.60	2.49	2.30	2.43	2.00	2.67	2.60	2.22

In order to compare the proposed RDH algorithm based on 3D HS with other RDH schemes in the same environment, embeddable blocks, which can be utilized for hiding data without the parallax or the intra-frame distortion diffusion, are chosen from inter-MBs of P_0 frame, as shown in Fig. 7. Huang and Chang's scheme [30] is denoted as Huang. In Huang and our method, information is hidden into three QDCT AC coefficients F_2, F_5, and F_3, where the coefficient pair (F_2, F_5) is used by Shi et al.'s scheme [13] denoted as Shi, and Ou et al.'s scheme [36] denoted as Ou. Additionally, F_2 is used by Chung et al.'s scheme [7] denoted as Chung. At the leftmost point of every line

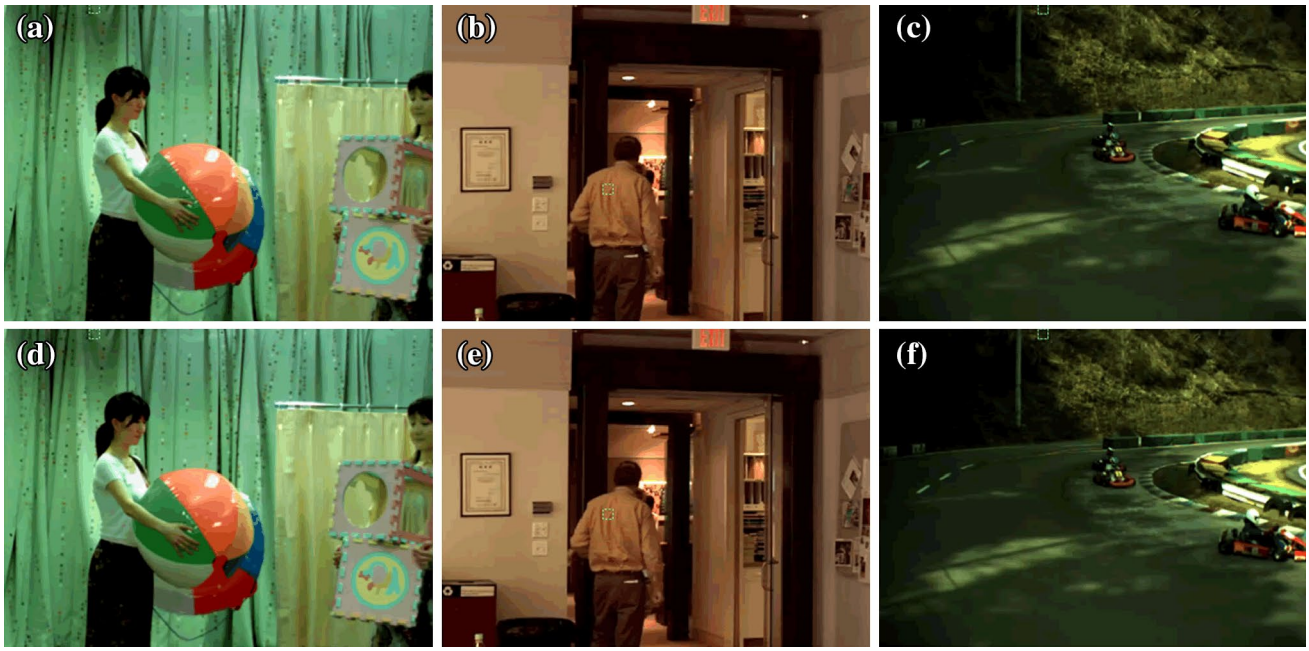

Fig. 11 The marked frames of Akko&Kayo, Exit and Race with conventional 3D HS and the proposed 3D HS

in Fig. 12, data is hidden into the embeddable blocks that meet $|F_0| \geq U$, where the threshold U is 0. The five points of each line from left to right express the hiding cases in which the threshold U is 4, 3, 2, 1 and 0, respectively.

It is obvious that the best PSNR, SSIM, embedding efficiency e, and the least bit-rate increase could be obtained by employing the presented method when the same number of data is embedded. We try to hide two data bits with at most one modification, so the embedding efficiency is improved, i.e., we can hide more information when the same distortion is caused. Consequently, if the same quantity of information is embedded, the less modification will be induced, and the video quality is better, which is verified by higher PSNR and SSIM. Additionally, little cost brings lower bit-rate increase that demonstrates hiding data with our method has little impact on the coding efficiency of H.264.

In Table 4, the threshold U is 0, and 700 bits of information are embedded into each P_0 frame on average. Compared with the other schemes, the presented method is superior by increasing PSNR and SSIM with 0.006 dB and 0.00005 at least, respectively. The best SSIM and PSNR mean that the best video quality can be gained by utilizing our algorithm. Furthermore, the embedding efficiency e of the presented scheme is much higher compared with that of other schemes, which demonstrates that when the same quantity of cover bits are changed, more bits of information could be embedded by our algorithm compared with other schemes.

5 Conclusions

An efficient 3D HS is presented for RDH algorithm in this paper. This new reversible method could be used for hiding data into image and video, where the image RDH such as 3D difference HS and 3D prediction-error HS will be applied and verified in our future work. We use the proposed 3D HS algorithm to embed data into QDCT coefficients of MVC video in this paper. Three coefficients chosen from each embeddable block are used for hiding two bits of information, where just one coefficient may be changed by adding 1 at most in most cases. Superior payload-distortion performance could be achieved by the proposed scheme compared with some state-of–the-art RDH methods. In order to improve the hiding capacity and decrease the distortion, the presented method will be generalized to hide over two data bits with at most one modification in future.

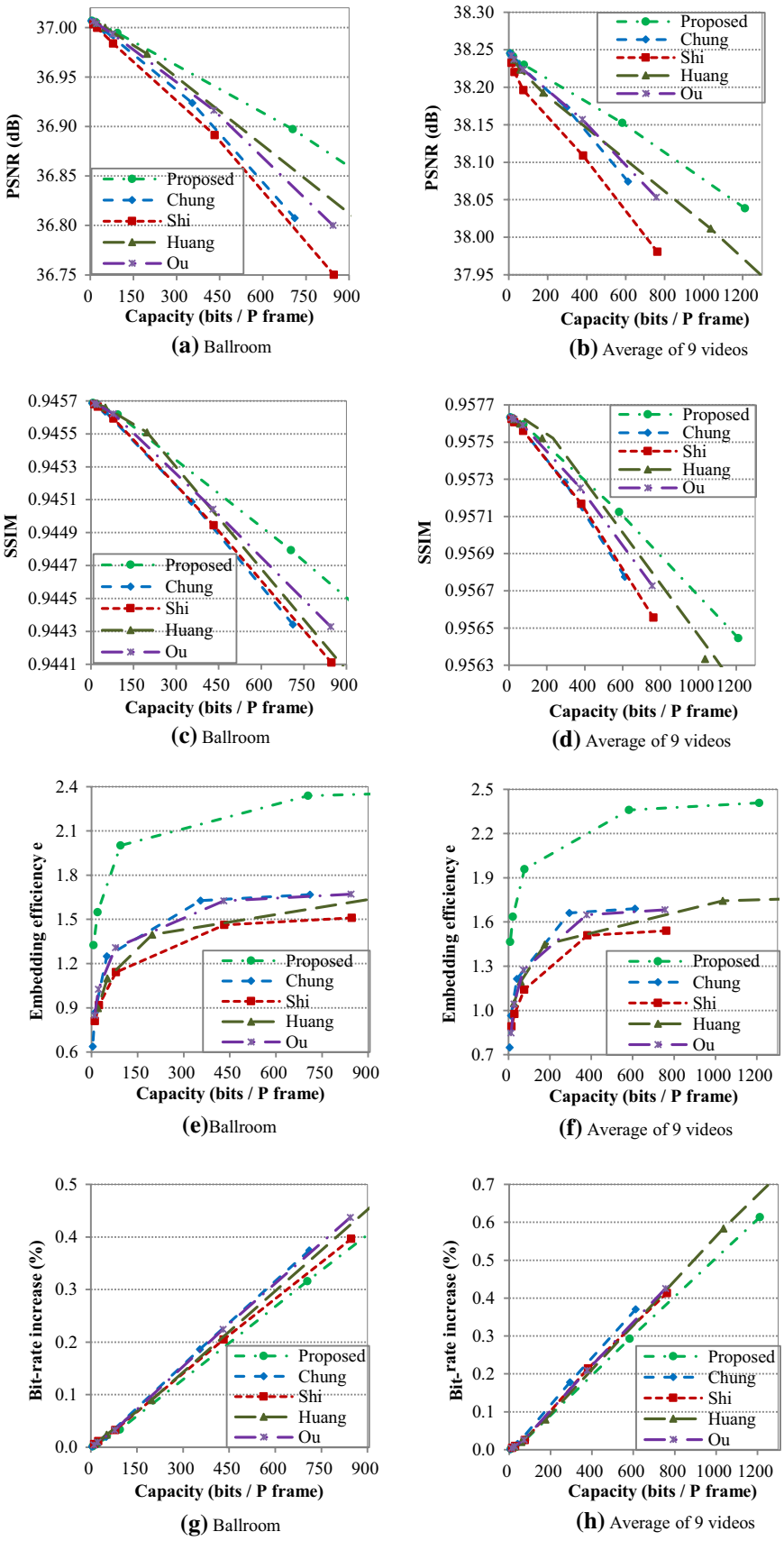

Fig. 12 Comparison of hiding performance with other RDH methods on MVC videos

Table 4 Embedding performance for hiding 700 bits of information into one P_0 frame on average

Videos	Proposed	Chung	Shi	Huang	Ou
Ballroom					
PSNR (dB)	36.875	36.808	36.784	36.843	36.869
SSIM	0.94478	0.94435	0.94426	0.94441	0.94451
Crowd					
PSNR (dB)	35.747	35.621	35.583	35.716	35.728
SSIM	0.96900	0.96849	0.96842	0.96871	0.96889
Race					
PSNR (dB)	37.628	37.599	37.597	37.604	37.620
SSIM	0.96149	0.96134	0.96136	0.96134	0.96144

Acknowledgments This work is supported by the National Natural Science Foundation of China (Grant Nos. 61272407 and 61370230). We are heartily grateful to the reviewers for their valuable comments improving the quality of the original manuscript.

References

1. Socek, D., Kalva, H., Magliveras, S.S., Marques, O., Culibrk, D., Furht, B.: New approaches to encryption and steganography for digital videos. Multimed. Syst. **13**(3), 191–204 (2007)
2. Song, X.G., Lian, S.G., Hu, W., Hu, Y.: Digital video watermarking based on intra prediction modes for audio video coding standard. Multimed. Syst. **20**(2), 195–202 (2014)
3. Ma, X.J., Li, Z.T., Tu, H., Zhang, B.C.: A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift. IEEE. T. Circ. Syst. Vid. **20**(10), 1320–1330 (2010)
4. Chen, W., Shahid, Z., Stutz, T., Autrusseau, F., Le Callet, P.: Robust drift-free bit-rate preserving H.264 watermarking. Multimed. Syst. **20**(2), 179–193 (2014)
5. Lie, W.N., Lin, T.C.I., Tsai, D.C., 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 (pp. 1175–1178) (2005)
6. Lin, S.F.D., Su, Y.L., Huang, J.Y.: Error resilience using a reversible data embedding technique in H.264/AVC. In: Proc. 7th WSEAS International Conference on Mulitmedia Systems and Signal Processing (MUSP '07) (pp. 112–117) (2007)
7. Chung, K.L., Huang, Y.H., Chang, P.C., Liao, H.Y.M.: Reversible data hiding-based approach for intra-frame error concealment in H.264/AVC. IEEE. T. Circ. Syst. Vid. **20**(11), 1643–1647 (2010)
8. Lin, S.D., Chuang, C.Y., Meng, H.C., Su, Y.L.: An error resilient technique using reversible data embedding in H.264/AVC. Int. J. Innovat. Comput. Inf. Control. **7**(5A), 2283–2290 (2011)
9. Zeng, X., Chen, Z., Xiong, Z.: Issues and solution on distortion drift in reversible video data hiding. Multimed. Tools Appl. **52**(2–3), 465–484 (2011)
10. Farrugia, R.A.: Reversible visible watermarking for H.264/AVC encoded video. Proc. International Conference on Computer as a Tool, EUROCON 2011—Joint with Conftele 2011 (pp. 1–4) (2011)
11. Ali, M.A., Edirisinghe, E.A.: Multi-layer watermarking of H.264/AVC video using differential expansion on IPCM blocks. In: Proc. IEEE International Conference on Consumer Electronics (pp. 53–54) (2011)
12. Maiti, S., Singh, M.P.: A novel reversible data embedding method for source authentication and tamper detection of H.264/AVC video. In: Proc. 5th International Conference on Information Processing, ICIP 2011. **157**, 349–355 (2011)
13. Shi, Y.J., Qi, M., Yi, Y.G., Zhang, M., Kong, J.: Object based dual watermarking for video authentication. Optik. **124**(19), 3827–3834 (2013)
14. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding for all image formats. In: Proc. Security and Watermarking of Multimedia Contents IV (pp. 572–583) (2002)
15. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless generalized-LSB data embedding. IEEE. T. Image Process. **14**(2), 253–266 (2005)
16. Zhang, W.M., Hu, X.C., Li, X.L., Yu, N.H.: Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression. IEEE. T. Image Process. **22**(7), 2775–2785 (2013)
17. Tian, J.: Reversible data embedding using a difference expansion. IEEE. T. Circ. Syst. Vid. **13**(8), 890–896 (2003)
18. Alattar, A.M.: Reversible watermark using the difference expansion of a generalized integer transform. IEEE. T. Image Process. **13**(8), 1147–1156 (2004)
19. Thodi, D.M., Rodriguez, J.J.: Expansion embedding techniques for reversible watermarking. IEEE. T. Image Process. **16**(3), 721–730 (2007)
20. Wu, H.C., Lee, C.C., Tsai, C.S., Chu, Y.P., Chen, H.R.: A high capacity reversible data hiding scheme with edge prediction and difference expansion. J. Syst. Softw. **82**(12), 1966–1973 (2009)
21. Lin, C.N., Buehrer, D.J., Chang, C.C., Lu, T.C.: Using quad smoothness to efficiently control capacity-distortion of reversible data hiding. J. Syst. Softw. **83**(10), 1805–1812 (2010)
22. Lee, C.F., Chen, H.L.: Adjustable prediction-based reversible data hiding. Digit. Signal Process. **22**(6), 941–953 (2012)
23. Ni, Z.C., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE. T. Circ. Syst. Vid. **16**(3), 354–362 (2006)
24. Lin, C.-C., Tai, W.-L., Chang, C.-C.: Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recogn. **41**(12), 3582–3591 (2008)
25. Tsai, P., Hu, Y.C., Yeh, H.L.: Reversible image hiding scheme using predictive coding and histogram shifting. Signal Process. **89**(6), 1129–1143 (2009)
26. Tsai, H.M., Chang, L.W.: Secure reversible visible image watermarking with authentication. Signal Process. image. **25**(1), 10–17 (2010)
27. Huang, H.C., Fang, W.C.: Techniques and applications of intelligent multimedia data hiding. Telecomm. Syst. **44**(3–4), 241–251 (2010)
28. Hong, W.E., Chen, T.S.: Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism. J. Vis. Comm. Image R. **22**(2), 131–140 (2011)
29. Lin, Y.C.: Reversible data-hiding for progressive image transmission. Signal Process. Image. **26**(10), 628–645 (2011)
30. Huang, H.C., Chang, F.C.: Hierarchy-based reversible data hiding. Expert Syst. Appl. **40**(1), 34–43 (2013)
31. Li, X.L., Li, B., Yang, B., Zeng, T.Y.: General framework to histogram-shifting-based reversible data hiding. IEEE. T. Image Process. **22**(6), 2181–2191 (2013)
32. Wang, Z.H., Lee, C.F., Chang, C.Y.: Histogram-shifting-imitated reversible data hiding. J. Syst. Softw. **86**(2), 315–323 (2013)
33. Li, X.L., Zhang, W.M., Gui, X.L., Yang, B.: A novel reversible data hiding scheme based on two-dimensional difference-histogram modification. IEEE Trans. Inf. Forensic. Secur. **8**(7), 1091–1100 (2013)
34. Zhao, J., Li, Z., Feng, B.: A novel two-dimensional histogram modification for reversible data embedding into stereo H.264 video. Multimed. Tools Appl, 1–22 (2015)

35. Xu, D.W., Wang, R.D., Wang, J.C.: A novel watermarking scheme for H.264/AVC video authentication. Signal Process. Image. **26**(6), 267–279 (2011)
36. Ou, B., Li, X.L., Zhao, Y., Ni, R.R., Shi, Y.Q.: Pairwise prediction-error expansion for efficient reversible data hiding. IEEE. T. Image Process. **22**(12), 5010–5021 (2013)
37. Sühring, K.: H.264/AVC software coordination (2012). http://iphome.hhi.de/suehring/tml
38. Video test sequences.: (2013) http://blog.csdn.net/do2jiang/article/details/5499464