LOSSLESS STEGANOGRAPHY ON ORTHOGONAL VECTOR FOR 3D H.264 WITH LIMITED DISTORTION DIFFUSION

Juan Zhao[1] and Zhitang Li[2]

[1]School of Mathematics &Computer Science,
Wuhan Polytechnic University, Wuhan 430048, China
[2]Institute of Computer Science and Technology, Huazhong University
of Science and Technology, Wuhan 430074, China

ABSTRACT

In order to improve the undetectability, a lossless algorithm based on orthogonal vectors with limited distortion diffusion for 3D H.264 video is proposed in this paper. Inter-view distortion drift is avoided by embedding data into frames, which do not predict other views. Three conditions and pairs of coefficients are proposed to prevent intra-frame distortion diffusion. Several quantized discrete cosine transform coefficients are chosen from an embeddable luminance 4×4 block to construct a carrier vector, which is modified by an offset vector. When the carrier vector and the offset vector are orthogonal or near to be orthogonal, a data bit can be hidden. Experimental results indicate that the method is effective by enhancing peak signal-to-noise ratio with 7.5dB and reducing the Kullback-Leibler divergence with 0.07 at least. More than 1.7×10^{15} ways could be utilized for constructing the vectors, so it is more difficult for others to steal data.

KEYWORDS

Lossless Steganography, Reversible Data Hiding, Orthogonal Vector, 3D H.264, Distortion Drift

1. INTRODUCTION

The most significant requisite of steganography, which can be used to hide secret message into innocuous-looking media, is imperceptibility in covert communication. However, the host media will be distorted permanently if general steganography algorithms are used to hide information. Permanent damage is unbearable to medical images, military, law enforcement [1] and other sensitive fields. Therefore, lossless steganography (a kind of reversible data hiding)[2], videlicet reversible steganography or distortion-free steganography, which can restore the impaired video after extracting the secret information, has been a hot topic. Besides the sensitive fields, lossless steganography methods can also be employed in error concealment and other fields.

Histogram shifting (HS) [3-10] and difference expansion (DE) [11-15] are two principal lossless steganography approaches. In addition, integer pair swap (IPS) [16], pair-wise logical computation [17], and some other methods [18][19] have been employed to hide information reversibly. In typical HS algorithm [3], information is hidden into the peak points of a medium histogram. The number of embeddable data bits depends on the pixel number of the peak point in the histogram. The hierarchical relationships of original images are used, and the difference values between pixels are altered to hide data in [5]. All pixels are classified into wall pixels and non-wall pixels in [4],

David C. Wyld et al. (Eds): SEAS, CMCA - 2021
pp. 35-52, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110203

where interpolation and direction order are used for hiding data. In [6], the closest adjacent pixels are used to predict the visited pixel value and evaluate its just noticeable difference. Prediction-error shift is used to improve the embedding performance in [8, 9]. In [9], multiple pairs of expansion bins are utilized for each histogram , and the multiple-expansion-bin-selection for optimal embedding is formulated as an optimization problem.

In difference expansion method [11], the difference between two neighboring pixels is doubled to hide message. The secret information and a compressed location map are hidden into the difference values. To increase the payload, 16 bits are embedded into a 4×4 pixel block in a two-dimensional DE scheme [12].The host image is divided into non-overlapped equal-sized blocks in the high-fidelity technique [13] based on prediction-error expansion and pixel-value-ordering. Bidirectional difference expansion is used in [14] with three steps.

However, in traditional steganography algorithms, the selection space of embeddable location is small. How to improve the undetectability and security of lossless steganography algorithm is a key issue of covert communication. Video has so many frames that it can ensure adequate storage space. [20] Hence, through embedding a little information into a frame, we can guarantee video quality and improve the invisibility and undetectability. H.264 is the standard for video compression with high compression efficiency. 3D H.264 video is encoded or decoded through multi-view coding, which is an extension of H.264. At the encoder, intra-frame, inter-frame and inter-view predictions are used to compress the original YUV videos into a 3D H.264 video. In order to watch the video on the screen, the 3D H.264 video needs to be decompressed into YUV videos by intra-frame, inter-frame and inter-view predictions at the decoder. So if one block of a video is changed to embed data, the other blocks of the same frame, other frames or views in the corresponding YUV videos may also be modified, it is called intra-frame, inter-frame, or inter-view distortion drift [21], which is not considered in the literatures on 3D video data hiding [22-24].

In order to improve the undetectability and security of steganography, we present a novel lossless steganography algorithm based on orthogonal vector (If the inner product of two vectors is zero, it is called that the two vectors are orthogonal) for 3D H.264 video with limited distortion drift in this paper. Inter-view distortion propagation is avoided by embedding information into frames that do not predict other views. Embeddable blocks based on restrictive conditions are selected to prohibit intra-frame distortion shift. A carrier vector is composed of some quantized discrete cosine transform (QDCT) coefficients or coupling coefficients in one embeddable 4×4 block. An offset vector is set for recording the modification of the carrier vector. By dividing the inner product of the carrier vector and offset vector into several disjoint intervals, the information is hidden according to the interval of inner product. The carrier vector is not altered for hiding information 0; otherwise, the offset vector is added to or subtracted from the carrier vector according to the inner product of the two vectors. The receiver extracts information and restores the carrier according to the interval of inner product.

Compared with the current methods, the contributions of this paper are presented as follows. A. Three conditions and two sets of coupling coefficients are proposed for limiting intra-frame distortion drift. B. When the three conditions are used to avoid intra-frame distortion drift, over 1.7×10^{15} ways of constructing the carrier vector and its offset vector increase the difficulty for others stealing data.

The rest of the paper is organized as follows. The way of avoiding distortion diffusion for 3D H.264 video is brought in Section 2. Section 3 describes the lossless steganography algorithm and Section 4 gives the experimental results. At last, the paper is concluded in Section 5.

2. DISTORTION DIFFUSION PREVENTION

The original block denoted by B^O in the original YUV video is processed by equation (1) at the 3D H.264 encoder.

$$B^O - B^P = B^{RO} \quad (1)$$

Where B^P is the prediction block and B^{RO} is the residual block. Undergoing discrete cosine transformation and quantization, the residual block B^{RO} becomes a QDCT block (denoted by Y). Finally, YUV videos will be changed into 3D H.264 video by entropy encoding of QDCT blocks. Because this entropy encoding is a lossless compression process, at the decoder, the data embedded in some QDCT coefficients could be extracted completely after entropy decoding (lossless decompression). Undergoing the inverse quantization and inverse discrete cosine transform, the QDCT block Y becomes a residual block denoted by B^R (Cause discrete cosine transformation and quantization are loss compressions, B^R is different from the original residual block B^{RO}), which will be added to the prediction block B^P for reconstructing the video.

The prediction block B^P of a block in a frame could be computed through inter-view prediction, inter-frame prediction, or intra-frame prediction. Horizontal inter-frame prediction and vertical inter-view prediction of a 3D H.264 video with hierarchical B coding and two views are illustrated in Figure 1[7]. There are 16 frames in one group of picture, where each view has eight frames. Only intra-frame prediction is used for I_0 frame, so the distortion of other frames will not affect I_0 frame. However, hiding data into I_0 frame will sway all the frames in the two adjacent groups of pictures predicted by I_0 frame. By contrast, embedding data into P_0 frame will not lead to inter-view distortion drift because P_0 frame in the right view does not predict frames in the left view. Furthermore, inter-view distortion drift and inter-frame distortion drift could be avoided by embedding data into b_4 frames. Therefore, compared with embedding data into I_0 frames, better video quality can be achieved through embedding data into P_0 or b_4 frames. However, b_4 frame is located at the lowest level and is easy to be discarded during the process of transmission in the network. Compared with b_4 frame, better video quality cannot be acquired through embedding data into P_0 frame. However, P_0 frame is a key picture at the highest level, resulting that it cannot be lost easily during the process of the network transmission. Consequently, stronger robustness can be obtained by hiding data into P_0 frame compared with b_4 frame. Therefore, the best combination of video quality and robustness can be obtained by hiding data into P_0 frame, so it is selected to embed information in this paper.

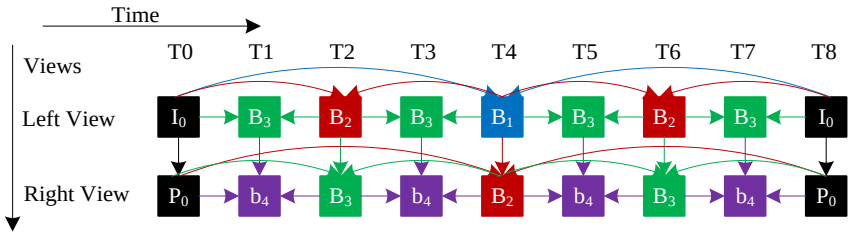


Figure 1. Prediction structure of3D H.264 with two views

At the encoder, through using inter-view prediction, inter-frame prediction, or intra-frame prediction, the prediction block B^P of a block is achieved to compute its residual block B^{RO}, which will be changed into a QDCT block Y by the 4×4 discrete cosine transform and quantization formulated as

$$Y = round\left[(C_f \cdot B^{RO} \cdot C_f^T) \otimes (E_f / Q)\right] = \begin{bmatrix} Y_{00} & Y_{01} & Y_{02} & Y_{03} \\ Y_{10} & Y_{11} & Y_{12} & Y_{13} \\ Y_{20} & Y_{21} & Y_{22} & Y_{23} \\ Y_{30} & Y_{31} & Y_{32} & Y_{33} \end{bmatrix}, \quad (2)$$

where Q is the quantization step size, $\otimes$is a math operator, by which each element in the former matrix is multiplied by the value at the corresponding position in the latter matrix,

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

When a data bit is hidden into one QDCT block Y by modifying some QDCT coefficients, the QDCT block after embedding data is denoted by Y^{Emb}. Let ΔY denote the modification caused by hiding information, it can be computed as

$$\Delta Y = Y^{Emb} - Y. \quad (3)$$

At the decoder, the residual block acquired by inverse quantization and 4×4 inverse discrete cosine transform is denoted by B^R , which can be calculated by

$$B^R = round\left[C_d^T \cdot (Y \otimes E_d) \cdot C_d\right], \quad (4)$$

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

When a data bit is embedded through modifying some QDCT coefficients of one block, the residual block after embedding data is denoted by B^{REmb}. Let ΔB^R depress the variation of the residual block between before and after hiding information. It can be computed as

$$\Delta B^R = B^{REmb} - B^R = round\left[C_d^T \cdot (\Delta Y \cdot Q \otimes E_d) \cdot C_d\right]. \quad (5)$$

Take the QDCT coefficient Y_{32} as an example to explain the distortion caused by hiding data. Suppose r is added to Y_{32}, the modification of the QDCT block for hiding data is $\Delta Y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & r & 0 \end{bmatrix}$,

then the alteration of the corresponding block in YUV video is

$$\Delta B^R = \frac{1}{2} Qabr \begin{bmatrix} 1 & -1 & -1 & 1 \\ -2 & 2 & 2 & -2 \\ 2 & -2 & -2 & 2 \\ -1 & 1 & 1 & -1 \end{bmatrix}. \quad (6)$$

Similarly, changing any one other QDCT coefficient in a 4×4 block will cause the variation of the whole block in the corresponding YUV video. In the same way, for an 8×8 block, modifying one QDCT coefficient will alter the whole 8×8 block, whose affected region is bigger than that of the 4×4 block. In addition, only two kinds of transformations, 4×4 transformation and 8×8 transformation, are used in 3D H.264 standard. Hence, the 4×4 transform block is selected to embed information in this paper.

It can be inferred that the edge pixels denoted by $c_0 \ldots c_{12}$ (shown in Figure 2) may be changed by hiding data into some QDCT coefficients of the blocks $B_{i,j-1}$ (the position of a block is expressed by I and j), $B_{i-1,j-1}$, $B_{i-1,j}$, and $B_{i-1,j+1}$. On one hand, when inter-view prediction or inter-frame prediction is employed to compute the prediction block denoted by $B^P_{i,j}$, the block $B_{i,j}$ will not be affected by the change of $c_0 \ldots c_{12}$, because $B^P_{i,j}$ is calculated by referring other frames. On the other hand, when intra-frame prediction is used by the current block $B_{i,j}$, its prediction block $B^P_{i,j}$ will be reckoned by the pixels $c_0 \ldots c_{12}$. Therefore, the hiding induced deviation of the blocks $B_{i,j-1}$, $B_{i-1,j-1}$, $B_{i-1,j}$, and $B_{i-1,j+1}$ will propagate to the block $B_{i,j}$. This is called as intra-frame distortion drift. However, according to the intra-frame prediction modes shown in Figure 2, it can be seen that the block $B_{i,j}$ is not affected by the thirteen pixels $c_0 \ldots c_{12}$ at the same time. For instance, when intra-frame prediction mode 0 is used by the current block $B_{i,j}$, only pixels c_1, c_2, c_3, andc_4 are used to predict the block $B_{i,j}$, so the distortion of the blocks $B_{i,j-1}$, $B_{i-1,j-1}$, and $B_{i-1,j+1}$ will not drift to the block $B_{i,j}$. Similarly, we can conceive the influence of $B_{i,j}$ over $B_{i,j+1}$, $B_{i+1,j+1}$, $B_{i+1,j}$, and $B_{i+1,j-1}$. Therefore, some conditions could be used to prevent intra-frame distortion drift.

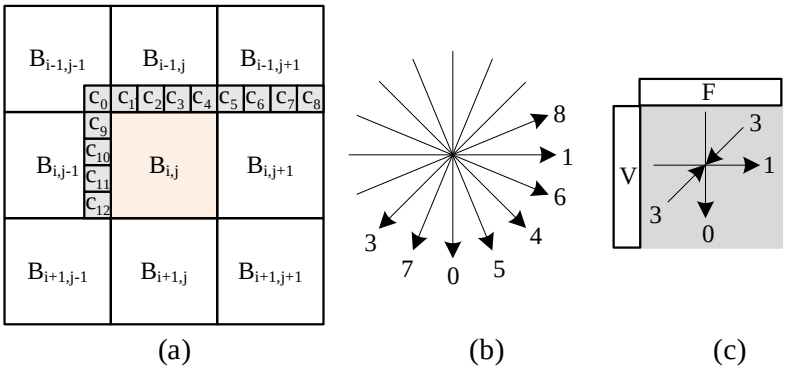


Figure 2. Intra-frame prediction mode (a) block position (b) the predictive direction of 4×4 and 8×8 luma block.(c) the predictive direction of 16×16 luma block.(In mode 2, all elements are predicted with the average of upper pixels denoted by F and left pixels denoted by V, i.e. Mean (F+V))

The prediction mode of intra-frame is denoted by *prediction Mode*. The mode type of macro block (MB) is denoted by *mb_type*. Let p be the *mb_type* of inter-view or inter-frame prediction. If the *mb_type* of the block $B_{i,j+1}$ is p, the prediction block $B^P_{i,j+1}$ is calculated by referring another frame, so the current block $B_{i,j}$ will not predict the block $B_{i,j+1}$. Otherwise, when the intra-frame prediction modes 0, 3, 7 indicated in Figure 2(b), or 0 in Figure 2 (c) are used by the block $B_{i,j+1}$, it can be seen from the predictive directions that the block $B_{i,j+1}$ will not be predicted by the current block $B_{i,j}$. Therefore, if information is embedded into the QDCT coefficients of the current block $B_{i,j}$, whose right adjacent block $B_{i,j+1}$ meets **Condition 1**, the evoked distortion will not drift to its adjacent block $B_{i,j+1}$.

Condition 1. $$\left(mb\_type_{B_{i,j+1}} = p\right) \cup \left[predictionMode_{B_{i,j+1}} \in \{0,3,7\}_{(4\times4 \text{ or } 8\times8)} \cup \{0\}_{(16\times16)} \right]$$

If the *mb_type* of the block $B_{i+1,j}$ is p, the prediction block $B^P_{i+1,j}$ is not calculated by referring

the current block $B_{i,j}$. Otherwise, when the intra-frame prediction modes 1 and 8 indicated in Figure 2(b), or 1 in Figure 2 (c) are used by the block $B_{i+1,j}$, it can be seen from the predictive directions that the block $B_{i+1,j}$ will not be predicted by the current block $B_{i,j}$. Similarly, when the *mb_type* of the block $B_{i+1,j-1}$ is *p*, or the intra-frame prediction modes 0, 1, 2, 4, 5, 6, and 8 indicated in Figure 2(b), or 0, 1, 2, and 3 in Figure 2 (c) are used by the block $B_{i+1,j-1}$, the block $B_{i+1,j-1}$ will not be predicted by the current block $B_{i,j}$. Therefore, if information is embedded into the QDCT coefficients of the current block $B_{i,j}$, whose adjacent blocks $B_{i+1,j}$ and $B_{i+1,j-1}$ meet **Condition 2**, the evoked distortion will not drift to its adjacent blocks$B_{i+1,j}$and $B_{i+1,j-1}$.

Condition 2.
$$\left[\begin{array}{l}\left(mb\_type_{B_{i+1,j}}=p\right)\cup \\ \left(predictionMode_{B_{i+1,j}}\in\{1,8\}_{(4\times4\text{ or }8\times8)}\cup\{1\}_{(16\times16)}\right)\end{array}\right]\cap$$
$$\left[\left(mb\_type_{B_{i+1,j-1}}=p\right)\cup\left(\begin{array}{l}predictionMode_{B_{i+1,j-1}}\in \\ \{0,1,2,4,5,6,8\}_{(4\times4\text{ or }8\times8)}\cup\{0,1,2,3\}_{(16\times16)}\end{array}\right)\right]$$

If the *mb_type* of the block $B_{i+1,j+1}$ is *p*, the prediction block $B^P{}_{i+1,j+1}$ is not calculated by referring the current block $B_{i,j}$. Otherwise, when the intra-frame prediction modes 0, 1, 2, 3, 7, and 8 indicated in Figure 2(b), or 0, 1, 2, and 3 in Figure 2(c) are used by the block $B_{i+1,j+1}$, it can be seen from the predictive directions that the block $B_{i+1,j+1}$ will not be predicted by the current block $B_{i,j}$. Therefore, if data is embedded into the QDCT coefficients of the block $B_{i,j}$, whose adjacent block $B_{i+1,j+1}$ meets **Condition 3**, the evoked distortion will not drift to its adjacent block$B_{i+1,j+1}$.

Condition 3. $\left(mb\_type_{B_{i+1,j+1}}=p\right)\cup\left[predictionMode_{B_{i+1,j+1}}\in\{0,1,2,3,7,8\}_{(4\times4\text{ or }8\times8)}\cup\{0,1,2,3\}_{(16\times16)}\right]$

It is obvious that if information is hidden into the QDCT coefficients of the current block $B_{i,j}$, whose adjacent blocks meet **Condition 1**, **Condition 2**, and **Condition 3**, the evoked distortion will not drift to any neighboring blocks. Therefore, these three conditions could be used for hiding data without intra-frame distortion drift. When **Condition 1**, **Condition 2**, and **Condition 3** could not be satisfied at the same time, intra-frame distortion drift could be prevented by compensation method. When data is hidden into some QDCT coefficients of the current block $B_{i,j}$, suppose the alternation of the QDCT block is $\Delta Y=\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\-r&0&r&0\end{bmatrix}$, then the alteration of the corresponding block in YUV video is

$$\Delta B^R = Qabr\begin{bmatrix}0&-1&-1&0\\0&2&2&0\\0&-2&-2&0\\0&1&1&0\end{bmatrix}. \qquad (7)$$

When the values of QDCT coefficients Y_{30} and Y_{32} are changed at the same time (if *r* is added to Y_{32}, *r* should be subtracted from Y_{30} correspondingly), values on the last column of matrix ΔB^R are zero. It shows that pixels on the last column of the current block $B_{i,j}$ are not altered by hiding data. Because pixels on the last column may predict the neighboring blocks $B_{i,j+1}$ or $B_{i+1,j+1}$, so the distortion of the current block $B_{i,j}$ will not infect the two blocks. Accordingly, the pair of coefficients (Y_{30}, Y_{32}) could be coupled to confine intra-frame distortion drift partly. When data is hidden into some QDCT

coefficients of the current block $B_{i,j}$, assume the alternation of the QDCT block is $\Delta Y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2r \end{bmatrix}$,

then the alteration of the corresponding block in YUV video is

$$\Delta B^R = \frac{5}{4} Qb^2 r \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -2 & 2 & -1 \\ -1 & 2 & -2 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (8)$$

When the values of QDCT coefficients Y_{13} and Y_{33} are changed at the same time (if r is added to Y_{13}, $2r$ should be subtracted from Y_{33} correspondingly), the values on the bottom row of matrix ΔB^R are zero. It shows that pixels on the bottom row of the current block $B_{i,j}$ are not changed by hiding data. Because pixels on the last column may predict the neighboring blocks $B_{i+1,j}$,$B_{i+1,j-1}$or$B_{i+1,j+1}$, so the distortion of the current block $B_{i,j}$ will not infect the three blocks. Accordingly, the pair of coefficients (Y_{13}, $2Y_{33}$) could be coupled to prevent intra-frame distortion drift partly. Similarly, we can get two sets of coupling coefficients, which are denoted by C_{set} and R_{set}.

$$C_{set} = \{(Y_{00}, Y_{02}), (Y_{10}, Y_{12}), (Y_{20}, Y_{22}), (Y_{30}, Y_{32}), (Y_{01}, 2Y_{03}), (Y_{11}, 2Y_{13}), (Y_{21}, 2Y_{23}), (Y_{31}, 2Y_{33})\}$$
$$R_{set} = \{(Y_{00}, Y_{20}), (Y_{01}, Y_{21}), (Y_{02}, Y_{22}), (Y_{03}, Y_{23}), (Y_{10}, 2Y_{30}), (Y_{11}, 2Y_{31}), (Y_{12}, 2Y_{32}), (Y_{13}, 2Y_{33})\}$$

When data is hidden into any coupling coefficients in C_{set} (if r is added to the former coefficient, r or $2r$ should be subtracted from the latter coefficient correspondingly), the values on the rightmost column of matrix ΔB^R are 0, so the distortion of the block $B_{i,j}$ will not propagate its neighboring blocks $B_{i,j+1}$ and $B_{i+1,j+1}$. Hiding data into any coupling coefficients in R_{set}, we can make the values at the bottom row of matrix ΔB^R be zero, so the distortion of block $B_{i,j}$ will not affect its adjacent blocks $B_{i+1,j}$,$B_{i+1,j-1}$, and $B_{i+1,j+1}$. It can be seen that the coupling coefficients in R_{set} could be combined with **Condition 1** to avoid intra-frame distortion drift, and the coupling coefficients in C_{set} can be combined with **Condition 2** to eliminate intra-frame distortion drift. For instance, when the adjacent blocks of the block $B_{i,j}$ do not satisfy **Condition 1 to 3** at the same time, if **Condition 1** is satisfied by the adjacent block $B_{i,j+1}$, it will not be affected by the block $B_{i,j}$. In addition, if coupling coefficients such as (Y_{02}, Y_{22}) of R_{set} are selected from the block $B_{i,j}$ for hiding data, the distortion of the block $B_{i,j}$ will not drift to its neighboring blocks $B_{i+1,j}$, $B_{i+1,j-1}$, and $B_{i+1,j+1}$.

3. LOSSLESS ALGORITHM BASED ON ORTHOGONAL VECTOR

The presented lossless steganography algorithm based on orthogonal vector for 3D H.264 video is depicted in Figure 3. At first, the information to be hidden is encrypted, and the 3D H.264 video is entropy decoded to gain the QDCT coefficients and intra-frame prediction modes.

3.1. Information Embedding

Denote the threshold as H. We select a 4×4 luminance QDCT block of P_0 frame in the right view according to $|Y_{00}| \geq H$ (threshold H=0,1,2,…. The bigger the threshold H is, the fewer embeddable blocks will be found, and the less the distortion will be. Compared with a block with small $|Y_{00}|$, the distortion caused by hiding data into a block with big $|Y_{00}|$ is less.). If **Conditions 1 to 3** are satisfied at the same time, the current block is chosen as an embeddable block. In the 4×4 block, a QDCT coefficient could be changed for hiding data, whereas only 16 selections can be used. If the third party identifies the marked block and the steganography algorithm, the probability for calculating the hidden data bit directly is $1/16 = 0.0625$. In order to reduce the probability of being cracked, we

embed data by choosing n QDCT coefficients from the block to make up a carrier vector denoted by $\psi = (x_1, x_2, \ldots, x_i, \ldots, x_n)(n \in [2,16])$ such as $\psi = (Y_{22}, Y_{32}) = (2, 0)$. Denote the carrier vector after hiding data as $\psi' = (x_1', x_2', ..., x_i', ..., x_n')$. In order to express the size of the modified value on carrier vector for embedding data reversibly, we construct a non zero offset vector denoted as $\partial = (z_1, z_2, \ldots, z_i, \ldots, z_n)$ such as $\partial = (0,1)$. Let φ be the included angle from the carrier vector ψ to the offset vector ∂, and φ' be the included angle from ψ' to ∂. Denote the length of the carrier vector as $|\psi| = (x_1^2 + x_2^2 + ... + x_n^2)^{1/2}$. If the length $|\psi|$ or the included angle φ is changed for hiding data, more computations need to be done since the modification of QDCT coefficient must be computed. So we hide data by changing the direction of the carrier vector, as shown in Figure 4 and (9), where the value of QDCT coefficient is changed directly and simply.

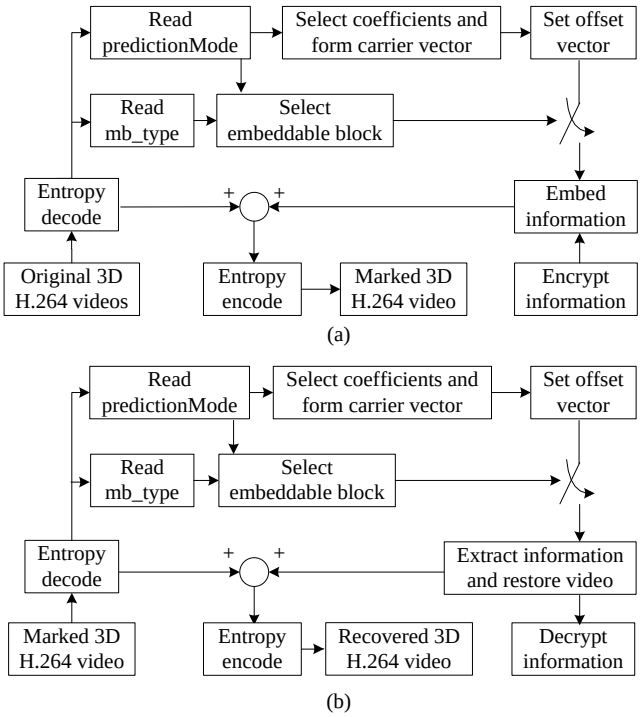


Figure 3. The flowchart of presented algorithm. (a) Embedding. (b) Extraction

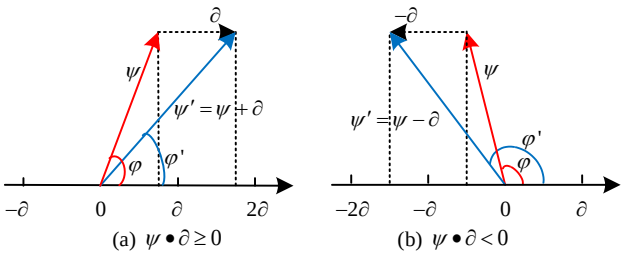


Figure 4. The modification of carrier vector ψ

Denote a bit of information as $u \in \{0,1\}$[10]. If the information u is 0, the carrier vector is not changed. Otherwise, in order to embed data reversibly, the carrier vector is modified.

$$\psi' = \begin{cases} \psi + \partial, \text{ if } (\psi \bullet \partial \geq 0) \\ \psi - \partial, \text{ if } (\psi \bullet \partial < 0) \end{cases} \Leftrightarrow x_i' = \begin{cases} x_i + z_i, \text{ if } (\psi \bullet \partial \geq 0) \\ x_i - z_i, \text{ if } (\psi \bullet \partial < 0) \end{cases} \tag{9}$$

Where $\psi \bullet \partial$ ($=x_1z_1+x_2z_2+\ldots+x_nz_n=|\psi||\partial|\cos\varphi$) is the inner product of ψ and ∂.

In order to get minimum offset $|\partial|$, we could set only one z_i in ∂ is 1 or -1 and the others are 0. Nonzero z_i represents the embedding position and the modified value of the embedding coefficient x_i.

Proposition. If $|\psi \bullet \partial| < \partial \bullet \partial$, then a bit of information u could be hidden reversibly.

Proof. If $\psi \bullet \partial = 0$, then ψ and ∂ are orthogonal, denoted as $\psi \perp \partial$. At this time, $\cos\varphi = 0$. So we can infer that $\lim_{\cos\varphi \to 0} (\psi \bullet \partial) = 0$.

Suppose $|\psi \bullet \partial| = \partial \bullet \partial$, we can reason that

$$|\psi \bullet \partial| = \partial \bullet \partial \Leftrightarrow \begin{cases} \psi \bullet \partial = \partial \bullet \partial \\ \psi \bullet \partial = -\partial \bullet \partial \end{cases} \Leftrightarrow \begin{cases} (\psi - \partial) \bullet \partial = 0 \\ (\psi + \partial) \bullet \partial = 0 \end{cases} \tag{10}$$

All points in the space from the flat $(\psi - \partial) \bullet \partial = 0$ to $(\psi + \partial) \bullet \partial = 0$ could be used for hiding message. Hence our embedding condition could be

$$|\psi \bullet \partial| < \partial \bullet \partial \Leftrightarrow |\psi||\partial||\cos\varphi| < |\partial||\partial| \Leftrightarrow |\psi||\cos\varphi| < |\partial|. \tag{11}$$

(11) shows that information could be hidden when the projection $|\psi||\cos\varphi|$ of the carrier vector ψ on the offset vector ∂ is less than the length of ∂. When the information u is 0, the carrier is not changed, so the projection $|\psi'|\cos\varphi' \in (-|\partial|, |\partial|)$. In order to hide information 1, the carrier vector ψ is altered as shown in (9) and Figure 4, the projection $|\psi|\cos\varphi \in [0, |\partial|)$ is turned into $|\psi'|\cos\varphi' \in [|\partial|, 2|\partial|)$ by adding the offset vector ∂ to the carrier vector ψ. The projection $|\psi|\cos\varphi \in (-|\partial|, 0)$ is changed into $|\psi'|\cos\varphi' \in (-2|\partial|, -|\partial|)$ by subtracting the offset vector ∂ from the carrier vector ψ. When the value of $|\psi||\cos\varphi|$ belongs to the interval $[|\partial|, \infty)$ or $(-\infty, -|\partial|]$, message could not be hidden. In order to distinguish the interval of information 1, the interval $[|\partial|, \infty)$ is changed to be $[2|\partial|, \infty)$ by adding the offset vector ∂ to the carrier vector ψ, and the interval $(-\infty, -|\partial|]$ is changed to be $(-\infty, -2|\partial|]$. Then we could extract information and recover the carrier vector according to different intervals of $|\psi'|\cos\varphi'$. Therefore, this proposition is proved.

The carrier vector ψ is changed for embedding data as shown in Figure 4 and (12). Information 0 and 1 are hidden into different intervals based on the value of $|\psi|\cos\varphi$. According to (11), we can infer the equivalent relations and the real embedding process (13), a corresponding version of (12).

$$
\begin{aligned}
&\text{if } \{(|\psi||\cos\varphi| < |\partial|) \wedge (u = 0)\} \\
&\quad \psi' = \psi \Rightarrow |\psi'||\cos\varphi'| < |\partial| \\
&\text{else if } \{(0 \le |\psi|\cos\varphi < |\partial|) \wedge (u = 1)\} \\
&\quad \psi' = \psi + \partial \Rightarrow |\partial| \le |\psi'|\cos\varphi' < 2|\partial| \\
&\text{else if } \{(-|\partial| < |\psi|\cos\varphi < 0) \wedge (u = 1)\} \\
&\quad \psi' = \psi\text{-}\partial \Rightarrow -2|\partial| < |\psi'|\cos\varphi' < -|\partial| \\
&\text{else if } (|\psi|\cos\varphi \ge |\partial|) \\
&\quad \psi' = \psi + \partial \Rightarrow |\psi'|\cos\varphi' \ge 2|\partial| \\
&\text{else if } (|\psi|\cos\varphi \le -|\partial|) \\
&\quad \psi' = \psi\text{-}\partial \Rightarrow |\psi'|\cos\varphi' \le -2|\partial|
\end{aligned}
\tag{12}
$$

When a 4×4 luminance QDCT block $B_{i,j}$ in P_0 frame meets $|Y_{00}| \ge H$, **Condition 1** is satisfied by its right adjacent block $B_{i,j+1}$, but **Condition 2** or **3** is not satisfied, several coupling coefficients could be chosen from R_{set} to combine with **Condition 1** to prevent intra-frame distortion diffusion. If a 4×4 block in P_0 frame satisfies $|Y_{00}| \ge H$, **Condition 2** is satisfied by its adjacent blocks $B_{i+1,j}$ and $B_{i+1,j-1}$, but **Condition 1** is not satisfied, some coupling coefficients in C_{set} could be managed to unite with **Condition 2** for removing distortion propagation.

$$
\begin{aligned}
&\text{if } (|\psi \bullet \partial| < \partial \bullet \partial) \wedge (u = 0) \\
&\quad \psi' = \psi \Rightarrow |\psi' \bullet \partial| < \partial \bullet \partial. \\
&\text{else if } (0 \le \psi \bullet \partial < \partial \bullet \partial) \wedge (u = 1) \\
&\quad \psi' = \psi + \partial \Rightarrow \partial \bullet \partial \le \psi' \bullet \partial < 2\partial \bullet \partial. \\
&\text{else if } (-\partial \bullet \partial < \psi \bullet \partial < 0) \wedge (u = 1) \\
&\quad \psi' = \psi\text{-}\partial \Rightarrow -2\partial \bullet \partial < \psi' \bullet \partial < -\partial \bullet \partial. \\
&\text{else if } (\psi \bullet \partial \ge \partial \bullet \partial) \\
&\quad \psi' = \psi + \partial \Rightarrow \psi' \bullet \partial \ge 2\partial \bullet \partial. \\
&\text{else if } (\psi \bullet \partial \le -\partial \bullet \partial) \\
&\quad \psi' = \psi\text{-}\partial \Rightarrow \psi' \bullet \partial \le -2\partial \bullet \partial.
\end{aligned}
\tag{13}
$$

Coupling coefficients denoted as (q_1,y_1), (q_2,y_2),…, $(q_S, y_S)(S \in [2, 8])$ are divided into embedding coefficients and compensation coefficients. The former can compose an embedding carrier vector denoted as $\psi_c = (q_1, q_2, \ldots, q_i, \ldots, q_S)$. The latter could form a compensation vector denoted as $\omega = (y_1, y_2, \ldots, y_i, \ldots, y_S)$. We construct a non zero offset vector denoted as $\partial_c = (e_1, e_2, \ldots, e_i, \ldots, e_S)$. Denote the carrier vector after hiding data as $\psi'_c = (q'_1, q'_2, ..., q'_i, ..., q'_S)$, and the compensation vector after hiding data as $\omega' = (y'_1, y'_2, ..., y'_i, ..., y'_S)$. When the carrier vector ψ_c is changed, the compensation vector ω is modified accordingly.

3.2. Information Extraction and Video Recovery

After the 3D H.264 video is entropy decoded, if $|Y_{00}| \ge H$ and **Conditions 1 to 3** are all satisfied, the current 4×4 QDCT block is chosen as an embeddable block, the process of information extraction and video recovery is exhibited in (14).

$$
\begin{aligned}
&\text{if } \left(\left|\psi' \bullet \partial\right| < \partial \bullet \partial\right) \\
&\quad u = 0,\ \ \psi = \psi' \Rightarrow \left|\psi \bullet \partial\right| < \partial \bullet \partial \\
&\text{else if } \left(\partial \bullet \partial \le \psi' \bullet \partial < 2\partial \bullet \partial\right) \\
&\quad u = 1,\ \ \psi = \psi' - \partial \Rightarrow 0 \le \psi \bullet \partial < \partial \bullet \partial \\
&\text{else if } \left(-2\partial \bullet \partial < \psi' \bullet \partial < -\partial \bullet \partial\right) \\
&\quad u = 1,\ \ \psi = \psi' + \partial \Rightarrow -\partial \bullet \partial < \psi \bullet \partial < 0 \\
&\text{else if } \left(\psi' \bullet \partial \ge 2\partial \bullet \partial\right) \\
&\quad \psi = \psi' - \partial \Rightarrow \psi \bullet \partial \ge \partial \bullet \partial \\
&\text{else if } \left(\psi' \bullet \partial \le -2\partial \bullet \partial\right) \\
&\quad \psi = \psi' + \partial \Rightarrow \psi \bullet \partial \le -\partial \bullet \partial
\end{aligned}
\tag{14}
$$

The process in (14) is the reverse process of (13). When $|\psi' \bullet \partial| < \partial \bullet \partial$, the hidden data bit u is 0. The original carrier vector ψ is not changed for hiding information 0, i.e., $\psi = \psi'$, so the values of QDCT coefficients do not need to be recovered. Identifying the intervals $[\partial \bullet \partial, 2\partial \bullet \partial)$ and $(-2\partial \bullet \partial, -\partial \bullet \partial)$ of $\psi' \bullet \partial$, we extract one bit of information 1. The interval $(\partial \bullet \partial, 2\partial \bullet \partial)$ is altered to be its original interval $[0, \partial \bullet \partial)$ by subtracting the offset vector ∂ from the carrier vector ψ'. The interval $(-2\partial \bullet \partial, -\partial \bullet \partial)$ is altered to be its original interval $(-\partial \bullet \partial, 0)$ by adding the offset vector ∂ to the carrier vector ψ'. The values of QDCT coefficients are recovered correspondingly. When the value of $\psi' \bullet \partial$ belongs to the interval $[2\partial \bullet \partial, \infty)$ or $(-\infty, -2\partial \bullet \partial]$, there is no hidden message. The interval $[2\partial \bullet \partial, \infty)$ is altered to be its original interval $[\partial \bullet \partial, \infty)$ by subtracting the offset vector ∂ from the carrier vector ψ'. The interval $(-\infty, -2\partial \bullet \partial]$ is altered to be its original interval $(-\infty, -\partial \bullet \partial]$ by adding the offset vector ∂ to the carrier vector ψ'. The video is exactly restored after extracting information in this way.

In P_0 frame of the right view, once $|Y_{00}| \ge H$ is satisfied by a 4×4 luminance QDCT block, but **Conditions 1** to 3 are not satisfied concurrently, only **Condition 1** or **2** is satisfied, pairs of coefficients from R_{set} or C_{set} are used to extract the information and restore the video. At last, the video is entropy encoded and the information is decrypted with keys.

Denote the frame number as N, and the information length as L. The computational efficiency of the proposed algorithm is related to the information length L and the frame number N. Therefore, the computational complexity of the proposed algorithm can be denoted by $O(N \times L)$.

Furthermore, the proposed way of limiting distortion drift could be used for 2D or 3D H.264 video with other structure. The presented lossless steganography algorithm could be used for hiding information in other media that can be grouped (some elements in a group can be selected to build a carrier vector). In addition, when the proposed method is applied in some media, especially a gray-scale image with 8 storage bits, the overflow/underflow problem should be treated[14]. However, this problem need not be considered when the data is hidden into QDCT coefficients of H.264 video.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

Nine test videos (the size of each frame is 640×480) Akko & Kayo, Ballroom, Crowd, Exit, Flamenco, Objects, Race, Rena, and Vassar [25] are utilized to do experiments with JM18.4 [26]. The parameter intra-period is 8 and two YUV files are encoded to a 3D H.264 video with 233

frames. 30 P_0 frames in the right view are used to hide data. The capacity of a video sequence is the average number of bits embedded into one P_0 frame of all the P_0 frames in that sequence. The peak signal-to-noise ratio (PSNR) value, the structural similarity (SSIM) value, and the Kullback-Leibler divergence (KLD) value obtained through comparing the marked YUV video with the original YUV video are the averages of all the frames. The difference of PSNR (DPSNR) and the difference of SSIM (DSSIM) are discrepancies before and after hiding data. The embedding efficiency denoted as e is defined by $e=L_{emb}/L_{cha}$, where L_{emb} is the number of embedded bits, and L_{cha} is the quantity of changed bits.

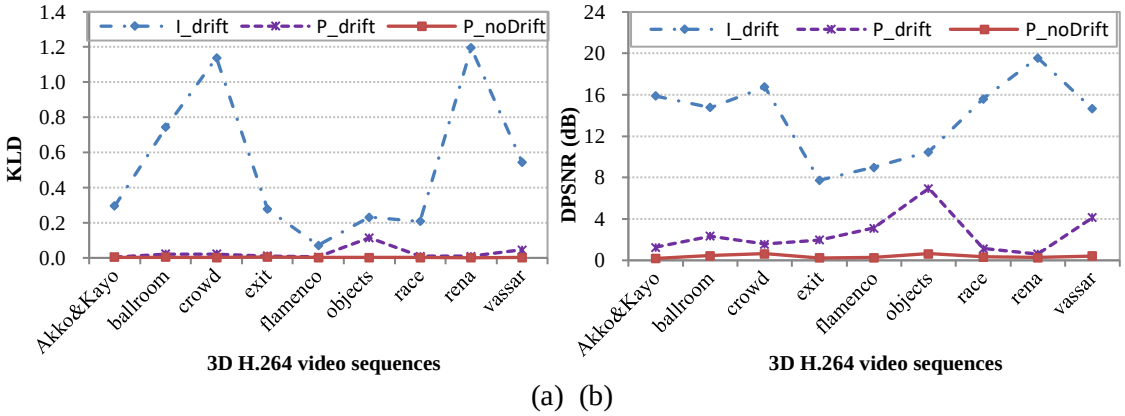
4.1. Effect of Distortion Drift Limitation

When the parameter code block pattern of a block is zero, there is no QDCT coefficient stored in the block which has all zero coefficients in fact. Therefore, not every block in a 640×480 frame, which has 307200 QDCT coefficients at most, could be changed, so grand visual distortion can be eliminated. The space meeting our conditions (**Condition**s **1** to **3**, **Condition 1** or **2**) is not too less than unconditional space. In addition, most QDCT coefficients are zero, which can guarantee enough capacity for embedding secret information. In order to prevent intra-frame and inter-view distortion drift, the embeddable coefficients of the proposed algorithm are chosen as shown in Table 1. Steganography with more rigorous condition will bring lesser capacity and preferable invisibility.

Denote the presented scheme without inter-view and intra-frame distortion drift as P_noDrift, the scheme without inter_view distortion drift as P_drift, and the scheme without limiting distortion drift as I_drift, where information is hidden into I_0 frame. Data is hidden into P_0 frame in P_noDrift and P_drift. The proposed lossless steganography algorithm based on orthogonal vector is used for hiding data in the three schemes, where $\partial = \partial_c = (0, 1)$ and the threshold H=0. The quantities of information embedded into the nine test videos are 750 bits, 1700 bits, 3800 bits, 630 bits, 880 bits, 650 bits, 2400 bits, 630 bits, and 1000 bits, respectively. As shown in Figure 5, the KLD and DPSNR values of I_drift are very large, which show that obvious distortion is caused by hiding data into I frame. So it is easy to be found by the third party, that is, its undetectability and security are weak. Through preventing inter-view distortion drift, the KLD and DPSNR value of P_drift are less than those of I_drift. Compared with I_drift, P_noDrift is superior by enhancing PSNR with 7.5dB and reducing KLD with 0.07 at least. By avoiding inter-view and intra-frame distortion drift, the KLD and DPSNR value of P_noDrift are about 0, which is hard to be detected by the third party. Therefore, the presented way to limit distortion drift is effective for improving the undetectability and security.

Table 1. Embeddable coefficients of different methods

Methods	**Conditions 1 to 3**	**Condition 1 (R_{set})**	**Condition 2 (C_{set})**
Ours	$\psi = (Y_{11}, Y_{22})$	$\psi_c = (Y_{01}, Y_{02})$ $\omega = (Y_{21}, Y_{22})$	$\psi_c = (Y_{10}, Y_{20})$ $\omega = (Y_{12}, Y_{22})$
HS	Y_{22}	(Y_{02}, Y_{22})	(Y_{20}, Y_{22})
DE	(Y_{11}, Y_{22})	(Y_{02}, Y_{22})	(Y_{20}, Y_{22})
IPS	(Y_{11}, Y_{22})	(Y_{02}, Y_{22})	(Y_{20}, Y_{22})



(a) (b)

Figure 5. KLD and PSNR of the schemes with or without distortion drift limitation

Correspondingly, the marked frames of Flamenco and Race are shown in Figure 6 (a) and (e) are the first original P_0 frames, (b) and (f) are the first marked I_0 frames obtained by using I_drift to hide data, (c) and (g) are the first marked P_0 frames obtained by using P_drift to hide data, (d) and (h) are the first marked P_0 frames obtained by using P_noDrift to hide data. It can be seen that there are large distortion in the frames (b) and (f). The distortion is around the people and floor in the frame (b). The road and trees are distorted in the frame (f). When data is hidden into the first P_0 frame, there is no distortion on the first I_0 frame. Compared with the frames (b) and (f), the distortion in the frames (c) and (g) is less. Furthermore, the distortion in the frames (d) and (h) is not obvious. It can be concluded from the results that superior visual quality and invisibility could be achieved by using the proposed way to prevent inter-view and intra-frame distortion drift.

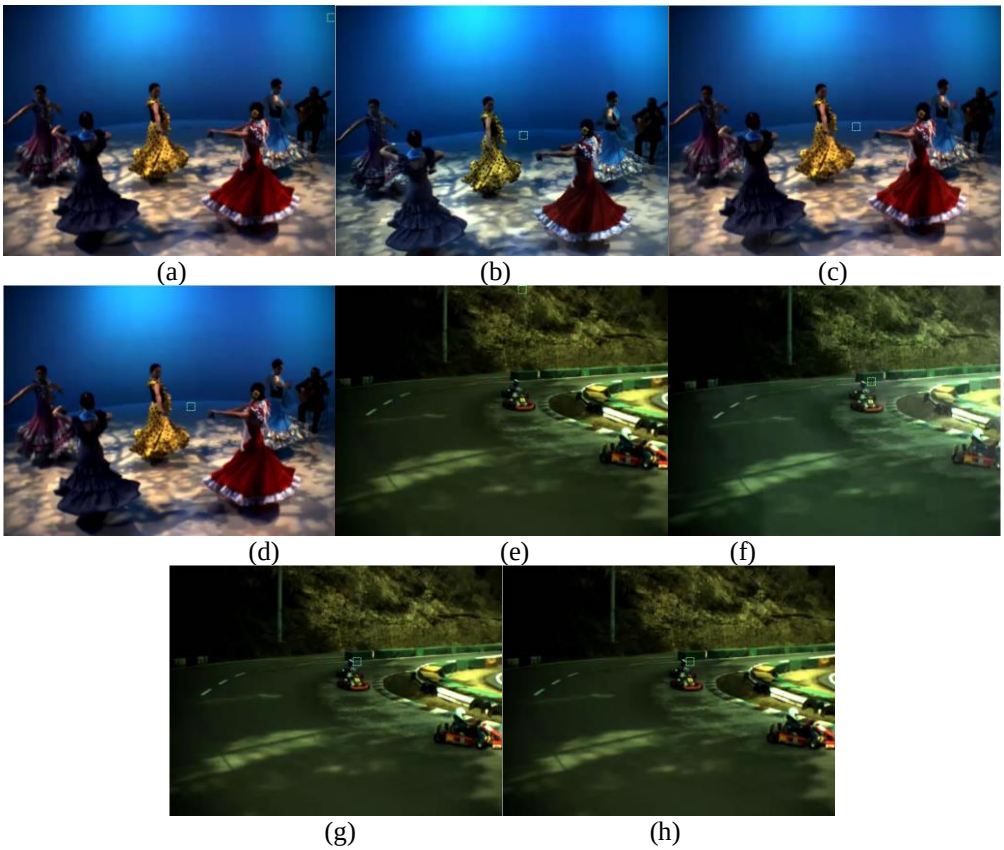
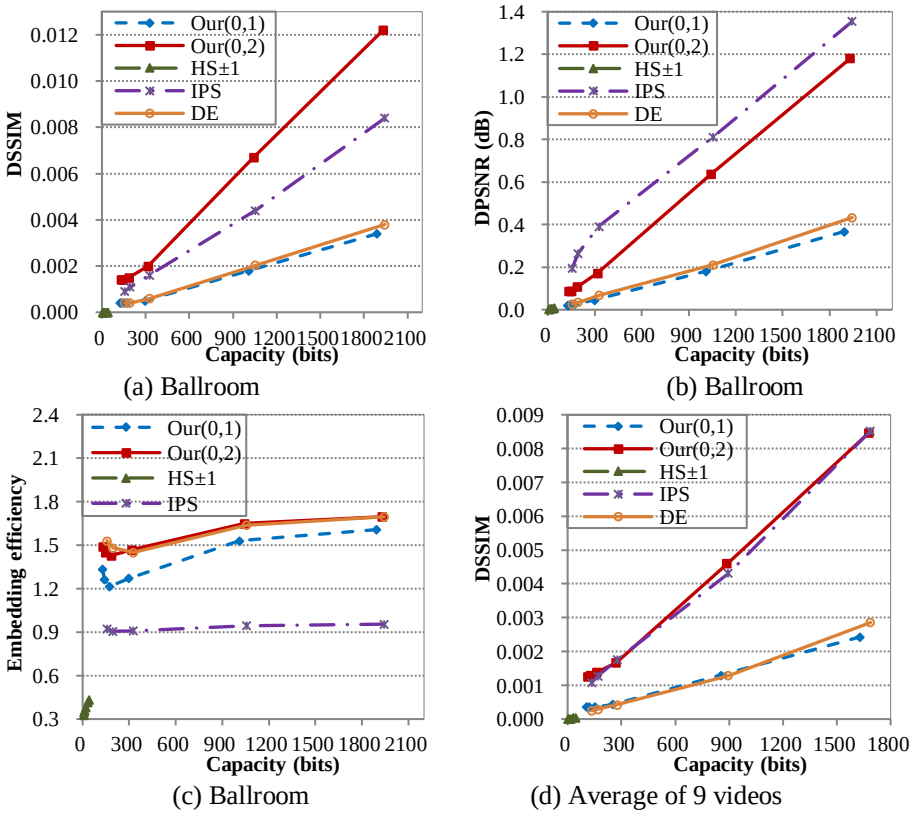


Figure 6. The original and marked frames of Flamenco and Race

4.2. Comparison of Different Lossless Steganography Methods

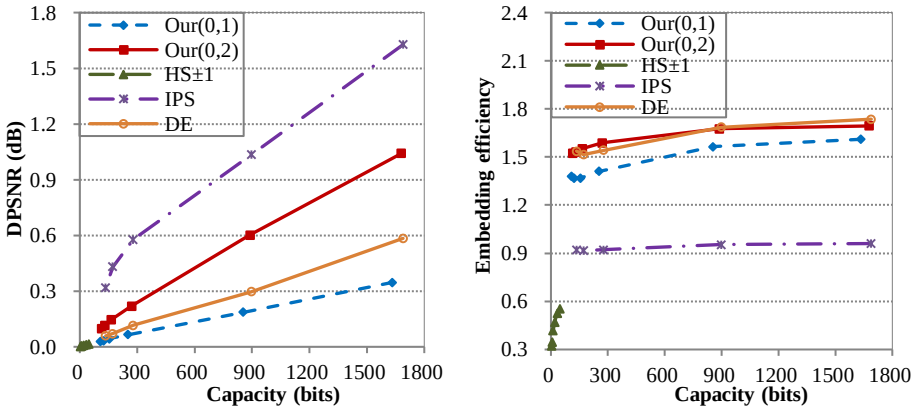
In order to compare the presented algorithm with other lossless steganography methods in the same environment, embeddable blocks, which could be used for hiding information without causing inter-view and intra-frame distortion drift, are selected from P_0 frames by using three conditions and coupling coefficients. HS[8], DE[15], and IPS [16] are three typical lossless steganography algorithms and could be used for video. Therefore, they are chosen for comparing with our algorithm. The embeddable coefficients of the proposed algorithm, HS[8], DE [15], and IPS [16] are shown in Table 1. Denote the proposed scheme with the offset vector $\partial = \partial_c = (0,1)$ as Our(0.1), and the proposed scheme with the offset vector $\partial = \partial_c = (0,2)$ as Our(0,2).The comparison of embedding performance for different schemes is portrayed in Figure 7, where the points of each line from left to right represent the embedding cases in which the threshold H is 4, 3, 2, 1, and 0, respectively.

Compared with other schemes, in order to embed the same quantity of information, the least DSSIM and DPSNR (i.e. the best SSIM and PSNR) can be obtained by using the proposed algorithm Our (0.1). The best SSIM and PSNR mean that the best video quality, invisibility and undetectability. Let HS0 be HS that is employed for embedding information into zero coefficients, and HS±1 be HS that is employed for embedding information into 1 or -1 coefficients. Given the same conditions and coupling coefficients to prevent intra-frame distortion propagation, HS0 is equivalent to the presented algorithm Our (0,1). So the lines of HS0 are thusly omitted in Figure 7. Little capacity, embedding efficiency, DSSIM and DPSNR could be got by using HS±1 to hide data.



(a) Ballroom (b) Ballroom

(c) Ballroom (d) Average of 9 videos



(e) Average of 9 videos (f) Average of 9 videos

Figure 7. Performance comparison of varied lossless algorithms

Compared with Our(0,1), higher capacity and better embedding efficiency can be achieved by using Our(0,2) to hide information. At the same time, the higher DSSIM and DPSNR are caused. That is, when schemes with bigger $|\partial|$ or $|\partial_c|$ are used to hide data, greater capacity can be got, and more distortion will be begot. Therefore, we need to control the modulus of∂ for limiting the distortion.

4.3. Discussions

When the threshold H is not considered, the optional number for picking n coefficients from 16 QDCT coefficients is the number of permutations, which is denoted by A_{16}^{n}. In an embeddable block satisfying **Conditions 1 to 3** at the same time, A_{16}^{n} ways could be used to create the carrier vector ψ. Assume the element z_i in the offset vector ∂ is 0, 1 or -1, there are 3^n-1 kinds of ∂ excluding a zero vector. Then the amount of selections for constructing ψ and $\partial$is calculated by (15). Even if an eavesdropper knows a marked block, the probability of guessing the embedded value directly is only$1/(1.3\times10^{21}) \approx 7.7\times10^{-22}$.

$$\sum_{n=2}^{16} A_{16}^{n}(3^n-1)=\sum_{n=2}^{16}\frac{16!(3^n-1)}{(16-n)!}\approx 1.3\times10^{21}. \tag{15}$$

When only one z_i in the offset vector ∂ is 1 or -1 and the others are 0, there are n ways to choose one z_i from n elements of ∂. This z_i could be 1 or -1, 2 ways could be used. So $2n$ alterations could be used to form ∂, and the optional count for constructing ψ and ∂ is obtained by (16). Even if the third party identifies a marked block and the data hiding algorithm, the probability for calculating the hidden data bit at once is only$1/(1.7\times10^{15})\approx5.9\times10^{-16}$.

$$2\sum_{n=2}^{16} nA_{16}^{n}=2\sum_{n=2}^{16}\frac{16!n}{(16-n)!}\approx 1.7\times10^{15}. \tag{16}$$

When only one z_i in the offset vector $\partial_c$is 1 or -1 and the others are 0, $2S$ alterations could be used to form ∂_c, and the optional count for constructing ψ_c and ∂_c is

$$2\sum_{S=2}^{8} S2^S A_8^S=2\sum_{S=2}^{8}\frac{8!S2^S}{(8-S)!}\approx 2.6\times10^{8}. \tag{17}$$

Given a pure payload, only 16 selections could be used by HS for picking one position from 16 QDCT coefficients to embed data and the elective sum for DE and IPS is $A_{16}^2 = 240$. It can be seen clearly that our lossless algorithm has an enormous choice, which can be united with large capacity to provide a broad operational space for secure and robust technologies someday. The proposed algorithm will be combined with random function, secret sharing, duplication code, error correcting code, and other methods to strengthen the security and robustness of covert communication in future.

5. CONCLUSIONS

A lossless steganography algorithm based on orthogonal vector with limited distortion propagation for 3D H.264 video is proposed to acquire fine embedding performance. Inter-view and intra-frame distortion diffusion is prevented according to the prediction structure and modes of 3D video. Some conditions are used to select an embeddable block, where several coefficients are picked up to compose a carrier vector or compensation vector. One bit of message is embedded or extracted according to the value of the inner product of the carrier vector and offset vector. There is such a huge choice space to create and change a carrier vector that a large operational area is achieved for steganography. Compared with other algorithms, the proposed scheme has superior invisibility and undetectability because of the numerous ways of forming carrier vector and offset vector.

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of Hubei Prorince under Grant 2017CFB306 .

The authors are heartily grateful to the reviewers for their valuable comments improving the quality of the original manuscript.

REFERENCES

[1] X. P. Zhang, "Reversible data hiding with optimal value transfer," (in English), IEEE Transactions on Multimedia, Article vol. 15, no. 2, pp. 316-325, Feb 2013.

[2] C. Y. Yang, L. T. Cheng, and W. F. Wang, "An efficient reversible ECG steganography by adaptive LSB approach based on 1D FDCT domain," Multimedia Tools and Applications, vol. 79, no. 33-34, pp. 24449-24462, Sep 2020.

[3] Z. C. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," (in English), IEEE Transactions on Circuits and Systems for Video Technology, Article vol. 16, no. 3, pp. 354-362, Mar 2006.

[4] X. T. Wang, C. C. Chang, T. S. Nguyen, and M. C. Li, "Reversible data hiding for high quality images exploiting interpolation and direction order mechanism," (in English), Digital Signal Processing, Article vol. 23, no. 2, pp. 569-577, Mar 2013.

[5] Y. Y. Tsai, D. S. Tsai, and C. L. Liu, "Reversible data hiding scheme based on neighboring pixel differences," Digital Signal Processing, vol. 23, no. 3, pp. 919-927, May 2013.

[6] W. Hong, T. S. Chen, and M. C. Wu, "An improved human visual system based reversible data hiding method using adaptive histogram modification," Optics Communications, vol. 291, pp. 87-97, Mar 2013.

[7] J. Zhao and Z. Li, "Three-dimensional histogram shifting for reversible data hiding," Multimedia Systems, vol. 24, no. 1, pp. 95-109, February 19,2018 2018.

[8] X. Z. Xie, C. C. Chang, and Y. C. Hu, "An adaptive reversible data hiding scheme based on prediction error histogram shifting by exploiting signed-digit representation," Multimedia Tools and Applications, vol. 79, no. 33-34, pp. 24329-24346, Sep 2020.

[9] W. F. Qi, X. L. Li, T. Zhang, and Z. M. Guo, "Optimal Reversible Data Hiding Scheme Based on Multiple Histograms Modification," Ieee Transactions on Circuits and Systems for Video Technology, vol. 30, no. 8, pp. 2300-2312, Aug 2020.

[10] J. Zhao, Z. Li, and B. Feng, "A novel two-dimensional histogram modification for reversible data embedding into stereo H.264 video," Multimedia Tools and Applications, vol. 75, no. 10, pp. 5959-5980, May 1, 2016 2016.
[11] J. Tian, "Reversible data embedding using a difference expansion," (in English), IEEE Transactions on Circuits and Systems for Video Technology, Article vol. 13, no. 8, pp. 890-896, Aug 2003.
[12] O. M. Al-Qershi and B. E. Khoo, "Two-dimensional difference expansion (2D-DE) scheme with a characteristics-based threshold," Signal Processing, vol. 93, no. 1, pp. 154-162, Jan 2013.
[13] X. L. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," Signal Processing, vol. 93, no. 1, pp. 198-205, Jan 2013.
[14] W. Q. Wang, "A reversible data hiding algorithm based on bidirectional difference expansion," Multimedia Tools and Applications, vol. 79, no. 9-10, pp. 5965-5988, Mar 2020.
[15] C. Y. Weng, "DWT-based reversible information hiding scheme using prediction-error-expansion in multimedia images," Peer-to-Peer Networking and Applications, vol. 13, no. 2, pp. 514-523, Mar 2020.
[16] S. Maiti and M. P. Singh, "A novel reversible data embedding method for source authentication and tamper detection of H.264/AVC video," presented at the 5th International Conference on Information Processing, ICIP 2011, Bangalore, India, August 5, 2011 - August 7, 2011, 2011. Available: <Go to ISI>://WOS:000306579700044
[17] T. Zhang, X. L. Li, W. F. Qi, and Z. M. Guo, "Location-Based PVO and Adaptive Pairwise Modification for Efficient Reversible Data Hiding," Ieee Transactions on Information Forensics and Security, vol. 15, pp. 2306-2319, 2020.
[18] B. G. Mobasseri, R. J. Berger, M. P. Marcinak, and Y. J. NaikRaikar, "Data Embedding in JPEG Bitstream by Code Mapping," Ieee Transactions on Image Processing, vol. 19, no. 4, pp. 958-966, Apr 2010.
[19] X. T. Wu, C. N. Yang, and Y. W. Liu, "A general framework for partial reversible data hiding using hamming code," Signal Processing, vol. 175, Oct 2020, Art. no. 107657.
[20] Y. Liu, S. Liu, Y. Wang, H. Zhao, and S. Liu, "Video steganography: A review," Neurocomputing, vol. 335, pp. 238-250, 2019/03/28/ 2019.
[21] X. J. Ma, Z. T. Li, H. Tu, and B. C. Zhang, "A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift," (in English), IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 10, pp. 1320-1330, Oct 2010.
[22] J. Franco-Contreras, S. Baudry, and G. Doërr, "Virtual view invariant domain for 3D video blind watermarking," in Image Processing (ICIP), 2011 18th IEEE International Conference on, 2011, pp. 2761-2764: IEEE.
[23] A. Koz, C. Cigla, and A. A. Alatan, "Watermarking of free-view video," Image Processing, IEEE Transactions on, vol. 19, no. 7, pp. 1785-1797, 2010.
[24] A. Chammem, M. Mitrea, and F. Preteux, "Stereoscopic video watermarking: a comparative study," Annals of Telecommunications-Annales Des Telecommunications, vol. 68, no. 11-12, pp. 673-690, Dec 2013.
[25] (Feb). Video test sequences. Available: http://blog.csdn.net/do2jiang/article/details/5499464
[26] K. Sühring. (Aug). H.264/AVC software coordination (JM 18.4 ed.). Available: http://iphome.hhi.de/suehring/tml

AUTHORS

Juan Zhao received her B.S. degree from Henan Normal University, Xinxiang, China, in 2007, and PhD degree from Huazhong University of Science and Technology, Wuhan, China, in 2015. She is currently a lecturer in School of Mathematics &Computer Science, Wuhan Polytechnic University. Her research interests include data hiding, network security and multimedia security.

Zhitang Li received his M.E. degree in Computer Architecture from Huazhong University of Science and Technology, Wuhan, China, 1987, and PhD degree in Computer Architecture from Huazhong University of Science and Technology, Wuhan, China, 1992. His research interests include computer architecture, network security, and P2P networks. He was the director of China Education and Research Network in Central China. He was a vice president of Department of Computer Science and Technology, Huazhong University of Science and Technology, China. He has published more than one hundred papers in the areas of network security, computer architecture, and P2P networks.