IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 23, 2018, accepted August 23, 2018, date of publication September 13, 2018, date of current version October 17, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869148

A Robust and Improved Visual Quality Data Hiding Method for HEVC

YUNXIA LIU[1], HONGGUO ZHAO[1], SHUYANG LIU[2], CONG FENG[1], AND SI LIU[1]
[1]College of Information Science and Technology, Zhengzhou Normal University, Zhengzhou 450053, China
[2]School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China

Corresponding author: Yunxia Liu (liuyunxia0110@hust.edu.cn)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61572447.

ABSTRACT This paper presents a novel and robust data hiding method based on *H.265*/High Efficiency Video Coding (*HEVC*) standard. To improve the robustness of data hiding, the embedded data are first encoded into the encoded data by using the *BCH* syndrome code (*BCH code*) technique. To improve the visual quality of data hiding, three groups of the prediction directions are provided to limit the intra-frame distortion drift. Then, the encoded data are embedded into the multi-coefficients of the selected 4×4 luminance discrete sine transform blocks, which meet the groups. It is experimentally proven that the proposed method can achieve greater robustness, better visual quality, and higher embedding capacity than previously studied methods.

INDEX TERMS Data hiding, H.265/HEVC, intra-frame distortion drift, robustness.

I. INTRODUCTION

High efficiency video coding (*H.265/HEVC*) is the latest video coding standard developed by Joint Collaborative Team for Video Coding (*JCTVC*) [1], [2]. The main achievement of *H.265* is its significant improvement in compression performance, with approximately 50% reduction in bit rate for producing videos of similar visual quality when compared to *H.264/AVC* [3]. However, high risk is associated with covert communications as well as tampering and illegal copying of digital content with the ever-growing nature of the Internet and multimedia technologies such as the multimedia cloud [4] and 8K ultra-high definition (*UHD*) video applications [5]. Thus, data hiding is an active research area.

Data hiding is a technique that embeds data into media content [6]–[13], and the technique has been used in many applications such as protection on ownership declaration, privacy protection and covert communication [10]. Data hiding technique mainly focuses on two main concerns: the distortion which is introduced by data hiding [31], [36], [37] and the robustness when data hiding are faced with various attacks [25], [38]–[40]. Based on these attentions, the existing literature examining data hiding mainly utilizes the encoder encrypted bitstreams [14]–[17], syntax elements [18], [19], [31], and discrete cosine transform (*DCT*) coefficients [20]–[23], [26], [27], [41] to embed data.

The distortion introduced by data hiding plays a key role in the data hiding process, and many scholars have attempted to solve the problem of distortion derived from data hiding. The method proposed in [16] embedded data into the encrypted bitstreams by modifying the digital stream. The challenge of [16] is that the data hiding action could be detected by analyzing the statistical characteristics of the bitstreams. The method in [17] proposed a data hiding scheme with Slepian-Wolf coding to ensure the security of data hiding in the encrypted domain. The syntax element is another field for data hiding [18], [19]. Zhang *et al.* [18] proposed a scheme by employing the coded block pattern (*CBP*) as the data carrier. Jing *et al.* [19] proposed a scheme by modulating the search points of motion vectors during the quarter-pixel motion estimation process. *DCT* domain is also an active field for data hiding. The method in [20] embedded data into the quantized *DCT* coefficients in high frequency with diagonal directions for copyright protection. Zhao *et al.* [21] presented a two-dimensional histogram shift strategy for embedding data into *DCT* coefficients. To further reduce the distortion introduced by data hiding, Ma *et al.* [22] proposed the paired coefficients of *DCT*, one coefficient for embedding and another for compensating embedded distortion. Compared to [22], Ma *et al.* [23] proposed an improved method, which combined the paired-coefficients of *DCT* and the prediction directions to avert distortion drift. However, these existing methods cannot be applied for *H.265* because of new coding block structure and prediction technologies.

The robustness of data hiding has become one interesting field [24]–[27], as the embedded data sometimes cannot recover from unreliable situations such as signal processing, re-quantization and geometric attacks. The method in [24] combined quantization index modulation and singular value decomposition for watermarking to effectively resist scaling attacks. However, this technique is designed for stereo audio. *BCH* code is explored for robust data hiding and resistance to re-quantization and unsafe transmission attacks. The methods in [25]–[27], [33], and [42]–[44] used the *BCH* code technique to improve the robustness, and the methods are compatible with *H.264*-standard-based videos. However, none of these algorithms is suitable for the robust data hiding in *H.265/HEVC*.

Recently, scholars have examined data hiding methods on *H.265* [13], [28]–[30], [45]. Tew and Wong [28] proposed a data hiding method based on the flexible coding block size feature in *HEVC* and adaptively changed the prediction block size corresponding to the embedded data. And Swati *et al.* [29] embedded watermark into each quantized transform coefficients (*QTC*) by means of modifying the *LSB* of the selected *QTC*. However, these data hiding methods have not focused on the reduction of distortion drift and the robustness of data hiding. Additionally, the relevance of *DST* coefficients was not considered. To further enhance the visual quality and robustness of data hiding, the robustness and relevance of *DST* coefficients are analyzed and applied in this paper.

The goal of this paper is to present a data hiding method that can improve visual quality, robustness performance and can be used in video steganography, covert communication and error concealment.

The main contributions and novelty in this paper are highlighted as follows:

1) *Three groups of prediction modes*: To reduce the distortion drift derived from data hiding, three groups of intra-frame prediction modes have been classified and utilized for improving the visual quality during data hiding process. With the analysis of distortion drift generation, three groups of prediction modes could effectively reduce the distortion drift from the prediction process.

2) *Two sets of multi-coefficients*: To furtherly improve the visual quality of data hiding, two sets of 4×4 luminance *DST* blocks multi-coefficients have been provided and utilized during data hiding process. Based on the analysis of transformation process in *HEVC*, the relevance of *DST* coefficients is used to make the embedding error to be zero. Furthermore, the combination between multi-coefficients and groups of prediction modes could fully eliminate the distortion drift in *HEVC*.

3) *Robust data hiding scheme with BCH code for HEVC*: To improve the robustness performance of data hiding, *BCH* code is also included in the entire data hiding scheme. To the best of our knowledge, the robust data hiding based on *BCH* code has not been investigated to be compatible to *H.265/HEVC* standard. It could be promising and effective for data hiding resistance to the error bits, especially for high-definition *(HD)* video with unsafe network transmission scenes.

The remainder of this paper is organized as follows. Section 2 gives some relatively key features of *HEVC* including intra-frame prediction and 4×4 transform process. Section 3 provides the proposed techniques about intra-frame distortion drift avoidance and the architecture of the robust data hiding scheme. The experimental results are presented in Section 4, the conclusions and future works are presented in Section 5.

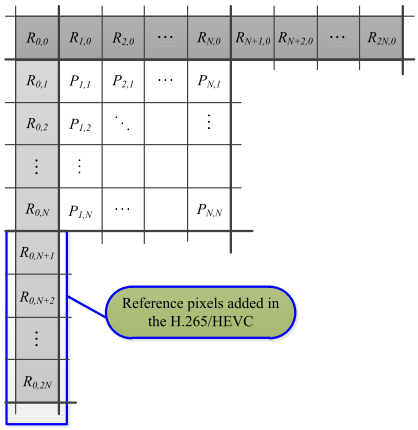


FIGURE 1. Labeling of prediction block pixels.

II. RELATED TECHNICAL BACKGROUNDS

A. INTRA-FRAME PREDICTION OF H.265/HEVC

Intra-prediction has been used in the I-frame of *H.265/HEVC* to reduce the spatial redundancy of the video sequences. The internal pixels of a 4×4 prediction block are predicted by using the boundary pixels of the upper and left reference pixels of adjacent blocks. As illustrated in Fig. 1, the pixels $P_{1,1}$ to $P_{N,N}$ of the prediction block are predicted by the reference pixels $R_{0,0}$, $R_{1,0} \cdots R_{2N,0}$, $R_{0,1} \cdots R_{0,2N}$. And the utilization of pixels $R_{0,N+1}$ to $R_{0,2N}$ in the standard H.265 provides more potential prediction modes, which will effectively improve the prediction accuracy (e.g., prediction modes with 2 – 9 in Fig. 2).

Intra-frame prediction process used in *HEVC* provides 33 angular prediction modes for each 4×4 luminance block; Fig.2 describes orders and directional orientations for these prediction modes [1]. One prediction block can be made the most probable choice from these 33 angular, planar and *DC* prediction modes. After the selection of prediction modes, the prediction pixels can be generated from the reference pixels and the chosen prediction modes. With more boundary pixels are used as reference pixels, these 33 angular prediction modes are provided for intra-frame *HEVC* standard instead of 9 angular prediction modes in *H.264/AVC* [32].

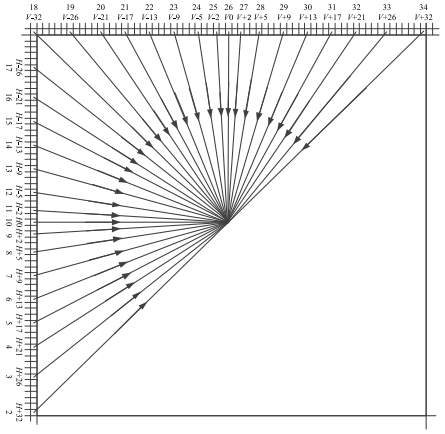


FIGURE 2. 4×4 luminance block prediction modes.

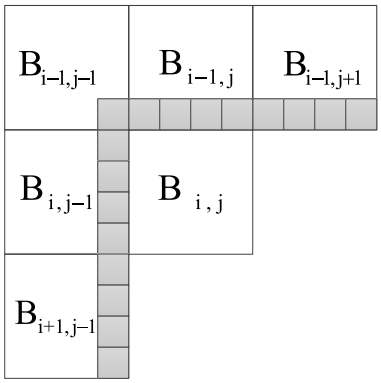


FIGURE 3. Prediction block $B_{i,j}$ and the adjacent encoded blocks.

B. INTRA-FRAME DISTORTION DRIFT IN HEVC

The intra-frame distortion drift emerges, which is introduced by embedding data into the 4×4 prediction blocks. As illustrated in Fig. 3, we assume that the current 4×4 prediction block is $B_{i,j}$ and that each sample of $B_{i,j}$ is the sum of the predicted and residual values. The predicted values of $B_{i,j}$ are calculated from the reference pixels which are gray in Fig. 3. If we embed data into the adjacent blocks $B_{i-1,j-1}, B_{i,j-1}, B_{i-1,j}, B_{i+1,j-1}$, *and* $B_{i-1,j+1}$, the embedding errors would propagate to $B_{i,j}$. This data hiding distortion that accumulates from the errors of the adjacent blocks to the current block is defined as the intra-frame distortion drift.

C. TRANSFORM AND INVERSE TRANSFORM PROCEDURE IN HEVC

In order to limit the dynamic range and improve the precision for transform computation, the integer discrete sine transform and inverse transform is used in *HEVC* standard. First, we provide a table of notations to this section in order to present the transformation process in *HEVC* briefly and effectively in Table 1.

At the encoder, for the transform block size of 4×4, the traditional two-dimensional *DST* transform is shown

TABLE 1. Notations of the transform and inverse transform procedure.

symbol	The meaning or definition of each symbol
$Y(k,l)$	The DST coefficient at the coordinate (k, l)
$x(m,n)$	The residual smaple at the coordinate (k,l)
Y	Matrix of the DST coefficients in 4×4 block
X	Matrix of the residual sample in 4×4 block
A	The transform matrix in 1-dimensional sine transform
H	Integer transform matrix in $HEVC$
$\tilde{Y}$	Quantized transform coefficients matrix in 4×4 block
2^{T_Shift}	Scaling factor in the scaling process
Q_{step}	Quantized step size
QP	Quantization parameter
Y'	Inverse quantization & scaling coefficients matrix
X'	Decoded residual sample matrix after inverse tranform
IT_Shift	Scaling parameter of inverse DST

as follows:

$$Y(k,l) = C(k)C(l)\sum_{m=1}^{4}\sum_{n=1}^{4} x(m,n)\sin\left[\frac{(2k-1)m\pi}{9}\right] \times \sin\left[\frac{(2l-1)n\pi}{9}\right] \quad k,l = 1,2,3,4 \qquad (1)$$

where k, l represent the horizontal or vertical coordinates in the transform block, respectively, and $C(k)$, $C(l)$ is fixed to a constant number of 2/3 for all values of k and l.

The two-dimensional sine transform in equation (1) can be computed by applying one-dimensional sine transform in the horizontal and vertical directions [1]. The one-dimensional sine transform can be represented with matrix as follows:

$$Y = AX$$

At here, X represents the original residual sample matrix, Y represents the transform coefficient matrix, and the transform matrix A can be defined as follow:

$$A_{mk} = C(k)\sin\left[\frac{(2k-1)m\pi}{9}\right], \quad m,k = 1,2,3,4 \qquad (2)$$

The $\{A_{m,k}\}$ can be represented as matrix A as follows:

$$A = \frac{2}{3}\begin{bmatrix} \sin\frac{\pi}{9} & \sin\frac{2\pi}{9} & \sin\frac{3\pi}{9} & \sin\frac{4\pi}{9} \\ \sin\frac{3\pi}{9} & \sin\frac{3\pi}{9} & 0 & -\sin\frac{3\pi}{9} \\ \sin\frac{4\pi}{9} & -\sin\frac{\pi}{9} & -\sin\frac{3\pi}{9} & \sin\frac{2\pi}{9} \\ \sin\frac{2\pi}{9} & -\sin\frac{4\pi}{9} & \sin\frac{3\pi}{9} & -\sin\frac{\pi}{9} \end{bmatrix} \qquad (3)$$

After rounding and scaling the transform matrix A in equation (3), an alternative integer transform H can be obtained

and applied to the 4×4 luma residual blocks for transformation process in *HEVC* [1], [32], with the integer transform matrix H is:

$$H = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}$$

According to the two-dimensional *DST* transform in equation (1), the core integer transform of 4×4 blocks in the *H.265/HEVC* [1] is shown as follows:

$$Y = HXH^T \tag{4}$$

Then, the post-scaling and quantization process is completed as in (2):

$$\tilde{Y} = (Y. \times MF)/2^{(qbits+T\_Shift)} \tag{5}$$

Where:

$$qbits = 14 + floor(QP/6)$$
$$MF = 2^{qbits}/Q_{step}$$

At the decoder, after re-scaling and inverse quantization step depicted equation (6) in *HEVC*, from the inverse DST described in (7), we can obtain the decoded residual sample matrix X':

$$Y' = \tilde{Y}. \times Qstep. \times 2^{6-shift} \tag{6}$$
$$X' = H^T Y' H \tag{7}$$

Where $shift = 6 + floor(QP/6) - IT\_Shift$.

D. BCH SYNDROME CODE SCHEME

BCH syndrome code is a technique that aims to retrieve data from the situation of bits error. Based on our previously studies [26], [27], [33], we have developed a binary *BCH* scheme to improve the robustness of the proposed data hiding method. Liu *et al.* [26] have confirmed its performance on the robustness of data hiding in *H.264/AVC*. As mentioned before, it's still significant for the design of data hiding method based on *H.265/HEVC*, especially for unsafe transmission scenes of *HD* video applications.

The core generalized parity-check matrix B for *BCH(n, k, t)* can be represented as follows:

$$\mathrm{B} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & \cdots & (\alpha^3)^{n-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha^{2t-1} & (\alpha^{2t-1})^2 & \cdots & (\alpha^{2t-1})^{n-1} \end{bmatrix}$$

where α is the primitive element in Galois field $GF(2^m)$, m is the order of the Galois field $GF(2^m)$, n is the codeword length and k is the code dimension, and t is the number of error bits that *BCH* could correct up. Based on the methods in [26], [27], and [33], this paper has adopted the same binary *BCH* code algorithm to improve the robustness of data hiding in the scheme as described in Section 3.3.

III. PROPOSED METHOD FOR DATA HIDING

A. PROPOSED TECHNIQUE FOR AVOIDING INTRA-FRAME DISTORTION DRIFT

Developed from the prediction shown in Section 2.1, we propose three groups to avoid intra-frame distortion drift. For convenience, the following definitions are given: the block on the right of the current prediction block is defined as right-block, the block on the top of the right-block is defined as top-right-block, the block on the bottom of the current block is defined as down-block, the block on the right of the down-block is defined as right-down-block, and the block on the left of the down-block is defined as left-down-block.

Group 1: The prediction modes of left-down-block $\in \{2, 3, \ldots, 26\}_{4\times4}$ and down-block $\in \{2, 3, \ldots, 10\}_{4\times4}$ It means that left-down-block with prediction modes 2, 3 . . . or 26 and down-block with prediction modes 2, 3 . . . or 10 are defined as Group 1, as shown in Fig. 4(a).

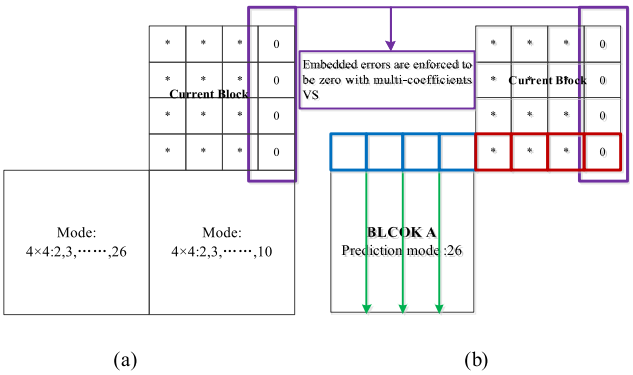


FIGURE 4. Current prediction block meets Group 1.

Suppose a prediction block meets Group 1, the residual coefficients on the lowest row will not be used for the intra-frame prediction of its left-down-block and down-block in the future. That is, if the embedded errors of the column furthest to the right of the current block are also zero (as depicted with purple lines in Fig. 4), the embedding-induced errors will not propagate to all the adjacent prediction blocks. As shown in Fig. 4(b), if the prediction mode of left-down-block A is 26, the directional orientation of 26 is vertical downward. During the prediction process of block A, the reference samples are the coefficients marked as blue instead of the lowest row coefficients of current block marked as red. That is, if we embed data into current block, the embedding errors on the lowest row will not be used as reference samples in the future prediction process of its left-down-block A.

Group 2: The prediction modes of top-right-block $\in \{10, 11, \ldots, 34, DC\}_{4\times4}$ and right-block $\in \{26, 27, \ldots, 34\}_{4\times4}$ It means that top-right-block with prediction modes 10, 11, . . . , 34 or DC and the right-block with prediction modes 26, 27, . . . , 34 are defined as Group 2, as shown in Fig. 5(a).

Suppose a prediction block meets Group 2, the residual coefficients of the column furthest to the right will not be used for the intra-frame prediction of its top-right-block and

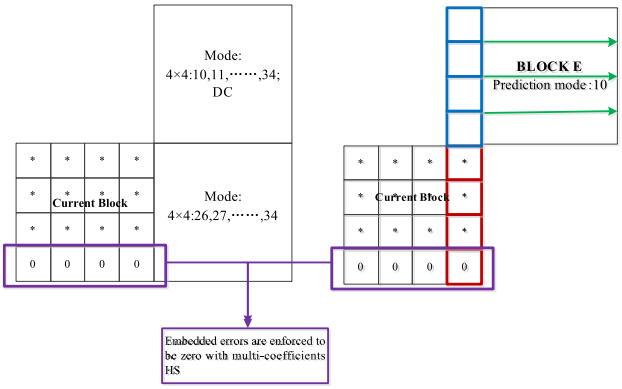


FIGURE 5. Current prediction block meets Group 2.

right-block in the future. That is, if the lowest-row embedded errors of the current block are also zero (as depicted with purple lines in Fig. 5), the embedding-induced errors will not propagate to all the adjacent prediction blocks. As shown in Fig. 5(b), if the prediction mode of top-right-block E is 10, the directional orientation of 10 is horizontally to the left. During the prediction process of block E, the reference samples are the coefficients marked as blue instead of the column furthest to the right coefficients of current block marked as red. That is, if we embed data into current block, the embedding errors on the column furthest to the right will not be used as reference samples in the future prediction process of its top-right-block E.

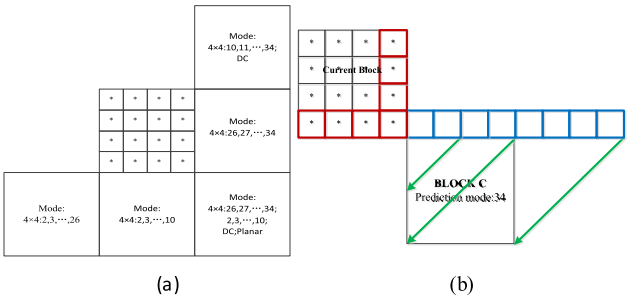


FIGURE 6. Current prediction block meets Group 3.

Group 3: The prediction modes of left-down-block $\in \{2, 3, \ldots, 26\}_{4\times4}$, down-block $\in \{2, 3, \ldots, 10\}_{4\times4}$, top-right-block $\in \{10, 11, \ldots, 34, DC\}_{4\times4}$, right-block $\in \{26, 27, \ldots, 34\}_{4\times4}$, and right-down-block $\in \{2, 3, \ldots, 10, 26, 27, \ldots, 34, DC, planar\}$. It means that left-down-block has the prediction modes 2, 3, ..., 26, and down-block has the prediction modes 2, 3, ..., 10, Top-right-block has the prediction modes 10, 11, ..., 34 or DC, and right-block has the prediction modes 26, 27, ..., 34 and right-down-block has the prediction modes 2, 3, ..., 34, DC or planar. These are defined as Group 3, as shown in Fig. 6(a). If a prediction block meets Group 3, the residual coefficients on the lowest row and the column furthest to the right will not be used for the intra-frame prediction of all adjacent blocks in the future. That is, the embedding-induced errors will not propagate to all the adjacent prediction blocks, regardless of whether the embedded induced errors in the lowest row or in the column furthest to the right column are zero. As shown in Fig. 6(b), if the prediction mode of right-down-block (block C) is 34, the directional orientation of 34 is marked as green lines. During the prediction process of block C, the reference samples are the coefficients marked as blue instead of the corner coefficients of current block marked as red. That is, if we embed data into current block, the embedding errors in the current block will not be used as reference samples in the future prediction process of its right-down-block.

B. PROPOSED ERROR ANALYSIS & MULTI-COEFFICIENTS

1) ERROR ANALYSIS OF DST COEFFICIENTS

During data hiding procedures, the data is embedded into the *DST* coefficients as shown in (6), where the embedded data matrix $\Delta = (\delta_{i,j})_{4\times4.}$ is added to the DST coefficient matrix $\tilde{Y}$.

$$\tilde{Y}' = \tilde{Y} + \Delta \tag{8}$$

After the inverse DST steps, the residual sample matrix X'' can be obtained with equation (9) as follows:

$$\begin{aligned} X'' &= H^T \tilde{Y}' H \\ &= H^T(\tilde{Y}. \times Qstep. \times 2^{6-shift})H \\ &\quad + H^T(\Delta. \times Qstep. \times 2^{6-shift})H \end{aligned} \tag{9}$$

Then error matrix of the residual sample E can be calculated from equations (7) and (9) as follows:

$$\begin{aligned} E &= X'' - X' \\ &= H^T(\tilde{Y}. \times Qstep. \times 2^{6-shift})H \\ &\quad + H^T(\Delta. \times Qstep. \times 2^{6-shift})H \\ &\quad - H^T(\tilde{Y}. \times Qstep. \times 2^{6-shift})H \end{aligned} \tag{10}$$

And in conclusion with equation (10), the error residual sample matrix E introduced by embedding data can be represented as:

$$E = H^T(\Delta. \times Qstep. \times 2^{6-shift})H \tag{11}$$

2) MULTI-COEFFICIENTS

The method in this paper embeds data into luminance coefficients of 4×4 prediction blocks since human eyes are more sensitive to the changes of blocks with larger size ($\geq 8 \times 8$). To avoid intra-frame distortion drift, the embedded errors of the 4×4 block on the column furthest to the right or on the lowest row should be enforced to be zero when the current block meets Group 1 or 2, as described in Section 3.1. Similarly, when the current prediction block meets Group 3, the embedded errors will not be propagated.

The multi-coefficients can be defined as three quantized DST coefficients (Y_1, Y_2, Y_3). Y_1 is used for data embedding, while Y_2 and Y_3 are used for compensation. The procedure for enforcing the embedded error of the lowest row or column furthest to the right to be zero in the matrix E is defined

as distortion compensation. There are two sets of multi-coefficients proposed in this paper:

Vertical Set (***VS***) = (Y_{00}, Y_{02}, Y_{03}), (Y_{10}, Y_{12}, Y_{13}), (Y_{20}, Y_{22}, Y_{23}), (Y_{30}, Y_{32}, Y_{33}).

Horizontal Set (***HS***) = (Y_{00}, Y_{20}, Y_{30}), (Y_{01}, Y_{21}, Y_{31}), (Y_{02}, Y_{22}, Y_{32}), (Y_{03}, Y_{23}, Y_{33}).

Proposition 1: If $e_{i3} = 0 (i = 0, 1, 2, 3)$, then

$$\Delta = \begin{bmatrix} \delta_{00} & 0 & -\delta_{00} & \delta_{00} \\ \delta_{10} & 0 & -\delta_{10} & \delta_{10} \\ \delta_{20} & 0 & -\delta_{20} & \delta_{20} \\ \delta_{30} & 0 & -\delta_{30} & \delta_{30} \end{bmatrix}$$

where $\delta_{i,j} \in Z (i, j = 0, 1, 2, 3)$.

Proof: In order to avoid intra-frame distortion drift, the embedded errors of the 4×4 block on the column furthest to the right should be enforced to be zero when the current block meets Group 1 , the error matrix E can be represented as follows:

$$E = \begin{bmatrix} e_{00} & e_{01} & e_{02} & 0 \\ e_{10} & e_{11} & e_{12} & 0 \\ e_{20} & e_{21} & e_{22} & 0 \\ e_{30} & e_{31} & e_{32} & 0 \end{bmatrix}$$

And according to the embedded error analysis depicted in section 3.2.1, the relevance between the error matrix E and embedded error matrix Δ derived from data hiding can be represented as follow:

$$E = H^T (\Delta . \times Qstep. \times 2^{6-shift}) H$$

If we overwrite equation (11) with matrix, we can obtained the equation as follows:

$$\begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \delta_{00} & \delta_{01} & \delta_{02} & \delta_{03} \\ \delta_{10} & \delta_{11} & \delta_{12} & \delta_{13} \\ \delta_{20} & \delta_{21} & \delta_{22} & \delta_{23} \\ \delta_{30} & \delta_{31} & \delta_{32} & \delta_{33} \end{bmatrix} \times \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix} = \begin{bmatrix} e_{00} & e_{01} & e_{02} & 0 \\ e_{10} & e_{11} & e_{12} & 0 \\ e_{20} & e_{21} & e_{22} & 0 \\ e_{30} & e_{31} & e_{32} & 0 \end{bmatrix}$$

Then we can obtain the computational results of the elements in the column furthest to the right of the above equation as follows:

$$(84\delta_{00} - 74\delta_{01} + 55\delta_{02} - 29\delta_{03}) \begin{bmatrix} 29 \\ 55 \\ 74 \\ 84 \end{bmatrix} + (84\delta_{10} - 74\delta_{11} + 55\delta_{12} - 29\delta_{13}) \begin{bmatrix} 74 \\ 74 \\ 0 \\ -74 \end{bmatrix} + (84\delta_{20} - 74\delta_{21} + 55\delta_{22} - 29\delta_{23}) \begin{bmatrix} 84 \\ -29 \\ -74 \\ 55 \end{bmatrix} + (84\delta_{30} - 74\delta_{31} + 55\delta_{32} - 29\delta_{33}) \begin{bmatrix} 55 \\ -84 \\ 74 \\ -29 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the one- dimensional matrix $[29\ 55\ 74\ 84]^{\mathrm{T}}$, $[74\ 74\ 0\ -74]^{\mathrm{T}}$, $[84\ -29\ -74\ 55]^{\mathrm{T}}$, $[55\ -84\ 74\ -29]^{\mathrm{T}}$ are linearly independent, then we can get the results from above equation as follows:

$$\begin{cases} (84\delta_{00} - 74\delta_{01} + 55\delta_{02} - 29\delta_{03}) = 0 \\ (84\delta_{10} - 74\delta_{11} + 55\delta_{12} - 29\delta_{13}) = 0 \\ (84\delta_{20} - 74\delta_{21} + 55\delta_{22} - 29\delta_{23}) = 0 \\ (84\delta_{30} - 74\delta_{31} + 55\delta_{32} - 29\delta_{33}) = 0 \end{cases} \tag{12}$$

According to (12), we can find one target error matrix Δ, which satisfies equation (12) as follows:

$$\Delta = \begin{bmatrix} \delta_{00} & 0 & -\delta_{00} & \delta_{00} \\ \delta_{10} & 0 & -\delta_{10} & \delta_{10} \\ \delta_{20} & 0 & -\delta_{20} & \delta_{20} \\ \delta_{30} & 0 & -\delta_{30} & \delta_{30} \end{bmatrix}$$

Proposition 2: If $e_{3j} = 0 (j = 0, 1, 2, 3)$, then

$$\Delta = \begin{bmatrix} \delta_{00} & \delta_{01} & \delta_{02} & \delta_{03} \\ 0 & 0 & 0 & 0 \\ -\delta_{00} & -\delta_{01} & -\delta_{02} & -\delta_{03} \\ \delta_{00} & \delta_{01} & \delta_{02} & \delta_{03} \end{bmatrix}$$

where $\delta_{i,j} \in Z (i, j = 0, 1, 2, 3)$.

Proof: In order to avoid intra-frame distortion drift, the embedded errors of the 4×4 block on the lowest-row should be enforced to be zero when the current block meets Group 2 , the error matrix E can be represented as follows:

$$E = \begin{bmatrix} e_{00} & e_{01} & e_{02} & e_{03} \\ e_{10} & e_{11} & e_{12} & e_{13} \\ e_{20} & e_{21} & e_{22} & e_{23} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

And according to the embedded error analysis depicted in section 3.2.1, the relevance between the error matrix E and embedded error matrix Δ derived from data hiding can be represented as follow:

$$E = H^T (\Delta . \times Qstep. \times 2^{6-shift}) H$$

If we overwrite equation (11) with matrix, we can obtained the equation as follows:

$$\begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \delta_{00} & \delta_{01} & \delta_{02} & \delta_{03} \\ \delta_{10} & \delta_{11} & \delta_{12} & \delta_{13} \\ \delta_{20} & \delta_{21} & \delta_{22} & \delta_{23} \\ \delta_{30} & \delta_{31} & \delta_{32} & \delta_{33} \end{bmatrix} \times \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix} = \begin{bmatrix} e_{00} & e_{01} & e_{02} & e_{03} \\ e_{10} & e_{11} & e_{12} & e_{13} \\ e_{20} & e_{21} & e_{22} & e_{23} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

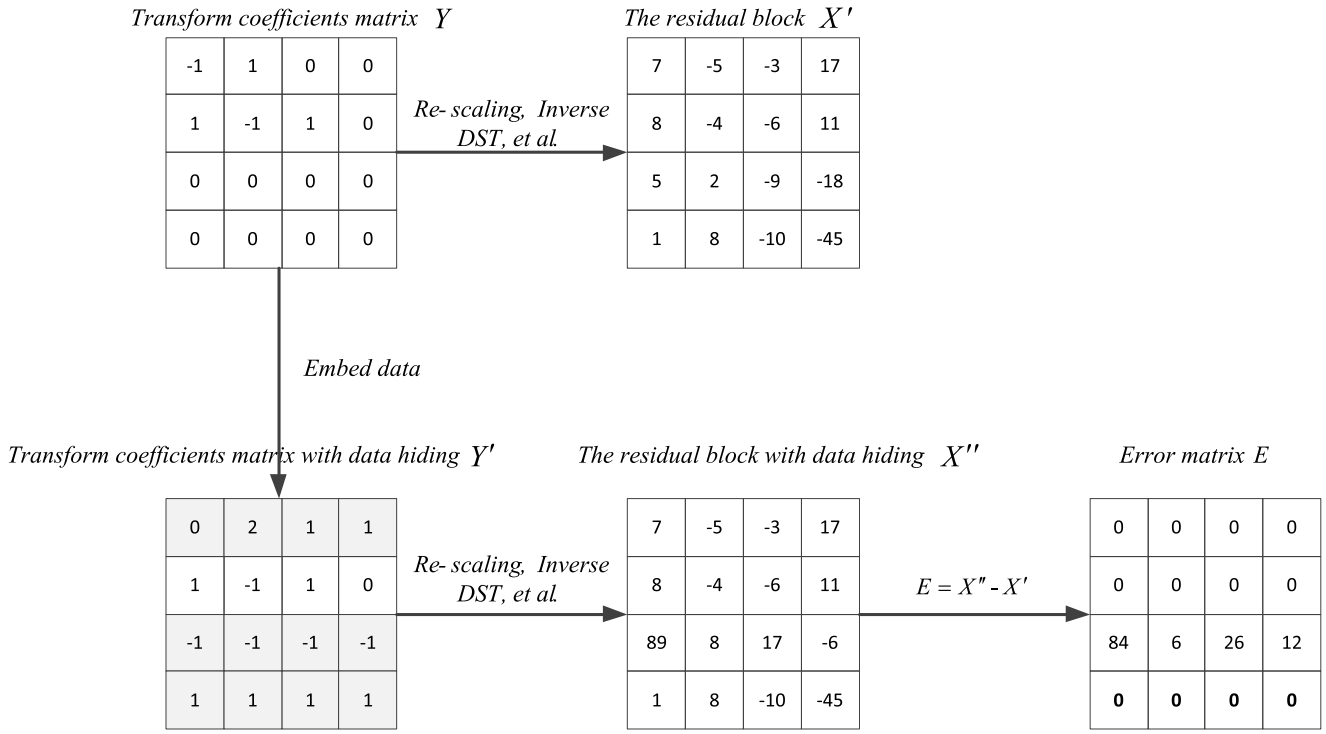


FIGURE 7. 7 A 4×4 block from Blowing Bubbles with an embedding HS multi-coefficient.

Then we can obtain the computational results of the elements in the lowest-row of the above equation as follows:

$$
\begin{aligned}
&(84\delta_{00} - 74\delta_{10} + 55\delta_{20} - 29\delta_{30}) \begin{bmatrix} 29 \\ 55 \\ 74 \\ 84 \end{bmatrix} \\
&+ (84\delta_{01} - 74\delta_{11} + 55\delta_{21} - 29\delta_{31}) \begin{bmatrix} 74 \\ 74 \\ 0 \\ -74 \end{bmatrix} \\
&+ (84\delta_{02} - 74\delta_{12} + 55\delta_{22} - 29\delta_{32}) \begin{bmatrix} 84 \\ -29 \\ -74 \\ 55 \end{bmatrix} \\
&+ (84\delta_{03} - 74\delta_{13} + 55\delta_{23} - 29\delta_{33}) \begin{bmatrix} 55 \\ -84 \\ 74 \\ -29 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}
$$

Since the one- dimensional matrix $[29\ 55\ 74\ 84]^{\mathrm{T}}$, $[74\ 74\ 0\ -74]^{\mathrm{T}}$, $[84\ -29\ -74\ 55]^{\mathrm{T}}$, $[55\ -84\ 74\ -29]^{\mathrm{T}}$ are linearly independent, then we can get the results from above equation as follows:

$$
\begin{cases}
(84\delta_{00} - 74\delta_{10} + 55\delta_{20} - 29\delta_{30}) = 0 \\
(84\delta_{01} - 74\delta_{11} + 55\delta_{21} - 29\delta_{31}) = 0 \\
(84\delta_{02} - 74\delta_{12} + 55\delta_{22} - 29\delta_{32}) = 0 \\
(84\delta_{03} - 74\delta_{13} + 55\delta_{23} - 29\delta_{33}) = 0
\end{cases} \tag{13}
$$

According to (13), we can find one target error matrix Δ, which satisfies equation (13) as follows:

$$
\Delta = \begin{bmatrix}
\delta_{00} & \delta_{01} & \delta_{02} & \delta_{03} \\
0 & 0 & 0 & 0 \\
-\delta_{00} & -\delta_{01} & -\delta_{02} & -\delta_{03} \\
\delta_{00} & \delta_{01} & \delta_{02} & \delta_{03}
\end{bmatrix}
$$

To clarify multi-coefficients, we provide a 4×4 block as an example to justify the HS multi-coefficients (Y_{00}, Y_{20}, Y_{30}), (Y_{01}, Y_{21}, Y_{31}), (Y_{02}, Y_{22}, Y_{32}), and (Y_{03}, Y_{23}, Y_{33}) in Fig. 7. This residual block is located at the 184th 4×4 block from the 1st coding tree block (CTB) in the 0th frame of BlowingBubbles. The residual transform coefficient matrix $\tilde{Y}$ is as shown in Fig.7. During the decoding process without data hiding, after the re-scaling and inverse DST transform, we can obtain the residual block X'. For the decoding process with data hiding, we embed data (1, 1, 1, 1) into (Y_{00}, Y_{01}, Y_{02}, Y_{03}) and compensate coefficients (Y_{20}, Y_{30}; Y_{21}, Y_{31}; Y_{22}, Y_{32}; Y_{23}, Y_{33}) according to the multi-coefficient set *HS*. After re-scaling and inverting *DST* transform, we can obtain the residual block X' and the error matrix E. The results show that the sample values on the lowest row of matrix E are all zero.

C. ARCHITECTURE OF THE PROPOSED METHOD

This proposed method embeds data into the *H.265/HEVC* transform and quantization coefficients. Fig. 8 depicts the process structure of the proposed method.

1) DATA EMBEDDING

The embedding operation is shown in Fig. 8 (a). To improve the robustness of the data hiding method, the original data

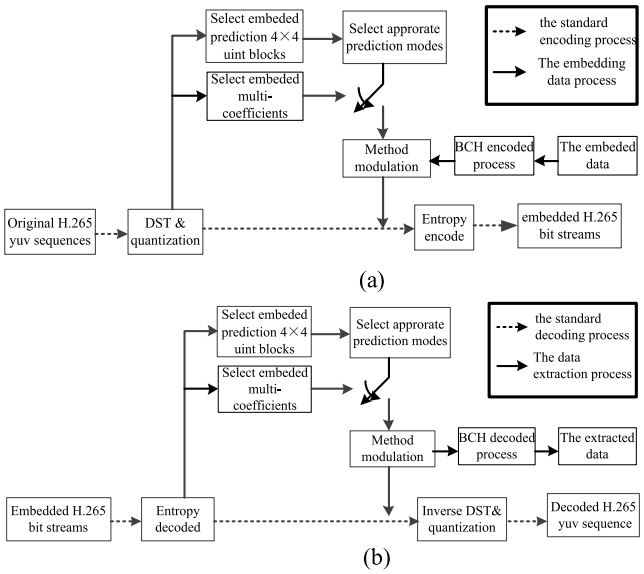


FIGURE 8. **Proposed data hiding method diagram (a) Embedding. (b) Extraction.**

should be encoded by using (n, k, t) *BCH* code before embedded modulation. To accurately extract the embedded data in the future, the 4×4 block of coefficients that are not all zero could be selected, since a block with all zero coefficients will not be transmitted to the bitstream. The original *YUV* video is first compressed with all possible encoded parameters to determine the optimal intra-frame prediction modes and quantized *DST* coefficients. Then, the 4×4 blocks with the multi-coefficients for embedding are selected. The *BCH* encoded data is embedded into multi-coefficients based on Modulation Method 1 or 2. Finally, all the quantized *DST* coefficients are entropy encoded to obtain the embedded video bitstream.

To more clearly make the embedding modulation procedure, multi-coefficients $(\tilde{Y}_{00}, \tilde{Y}_{02}, \tilde{Y}_{03})$ and $(\tilde{Y}_{00}, \tilde{Y}_{20}, \tilde{Y}_{30})$ are exploited for the data hiding steps as an example.

Step 1: Encode the original data with (n, k, t) *BCH* code.

The original data is first encoded to target *BCH* encoded data by using *BCH* with parameters (n, k, t).

Step 2: Select embeddable blocks and appropriate multi-coefficients.

If the current block meets Group 1, one bit is embedded into $(\tilde{Y}_{00}, \tilde{Y}_{02}, \tilde{Y}_{03})$, as described in Modulation Method 1; if the current block meets Group 2, one bit is embedded into $(\tilde{Y}_{00}, \tilde{Y}_{20}, \tilde{Y}_{30})$, as described in Modulation Method 1; and if the current block meets Group 3, sixteen bits are embedded into all of coefficients $Y_{i,j}$, $(i, j = 0, 1, 2, 3)$, as described in Modulation Method 2.

Otherwise, we go to Step 2 to select the next embeddable block to embed the *BCH* encoded data.

Modulation Method 1:

Assume that the selected multi-coefficients are (Y_1, Y_2, Y_3).

(1) If the embedded bit is 1, (Y_1, Y_2, Y_3) are modified as follows.

Case 1: $Y_1 \bmod 2 = 0$ *and* $Y_1 \geq 0$, then $Y_1 = Y_1 + 1$, $Y_2 = Y_2 - 1$, $Y_3 = Y_3 + 1$.

Case 2: $Y_1 \bmod 2 = 0$ *and* $Y_1 < 0$, then $Y_1 = Y_1 - 1$, $Y_2 = Y_2 + 1$, $Y_3 = Y_3 - 1$.

Case 3: $Y_1 \bmod 2 \neq 0$, then $Y_1 = Y_1$, $Y_2 = Y_2$, $Y_3 = Y_3$.

(2) If the embedded bit is 0, (Y_1, Y_2, Y_3) are modified as follows.

Case 1: $Y_1 \bmod 2 \neq 0$ *and* $Y_1 \geq 0$, then $Y_1 = Y_1 + 1$, $Y_2 = Y_2 - 1$, $Y_3 = Y_3 + 1$.

Case 2: $Y_1 \bmod 2 \neq 0$ *and* $Y_1 < 0$, then $Y_1 = Y_1 - 1$, $Y_2 = Y_2 + 1$, $Y_3 = Y_3 - 1$.

Case 3: $Y_1 \bmod 2 = 0$, then $Y_1 = Y_1$, $Y_2 = Y_2$, $Y_3 = Y_3$.

Modulation Method 2:

Assume that the embedded coefficient is Y_1.

(1) If the embedded bit is 1, Y_1 is modified as follows.

Case 1: $Y_1 \bmod 2 = 0$, then $Y_1 = \begin{cases} Y_1 + 1, \textit{if } Y_1 \geq 0 \\ Y_1 - 1, \textit{if } Y_1 < 0 \end{cases}$.

Case 2: $Y_1 \bmod 2 \neq 0$, then $Y_1 = Y_1$.

(2) If the embedded bit is 0, Y_1 is modified as follows.

Case 1: $Y_1 \bmod 2 \neq 0$, then $Y_1 = \begin{cases} Y_1 + 1, \textit{if } Y_1 \geq 0 \\ Y_1 - 1, \textit{if } Y_1 < 0 \end{cases}$.

Case 2: $Y_1 \bmod 2 = 0$, then $Y_1 = Y_1$.

2) DATA EXTRACTION

The extraction operation is shown in Fig. 8 (b). After entropy decoding of the H.265 video, based on the appropriate prediction modes of the adjacent prediction blocks, the extracted blocks are selected and the hidden data m_i is extracted as follows:

Assuming that the extracted coefficient is Y_1:

$$m_i = \begin{cases} 1, & \textit{if } Y_1 \bmod 2 \\ & = 1 \text{ and current block meets condition 1, 2 or 3} \\ 0, & \textit{if } Y_1 \bmod 2 \\ & = 0 \text{ and current block meets condition 1, 2 or 3} \end{cases}$$

IV. EXPERIMENTAL RESULTS

The proposed method was implemented in the *H.265/HEVC* reference software version *HM16.0* provided by Fraunhofer Heinrich Hertz Institute. The encoder used a fixed quantization parameter of 32 for I frames. The test video sequences of *Basketball Pass, Keiba*and*Blowing Bubbles*, were used for the experiments, and the detailed resolution information is shown in Table 2. The test video sequences were encoded into 20 frames at 30 frame/s and with the Main profile encoded.

The *PSNR1* (Peak Signal-to-Noise Ratio) in this paper is calculated by a comparison between the original YUV files and the corresponding decoded *YUV* files. The *PSNR2* in this paper is calculated by the comparison between the original *YUV* files and the corresponding decoded *YUV* files of data hiding. The survival rate in this paper is the proportion of correct extracted data compared to the original embedded data. The quantization parameter (*QP*) is the parameter of re-quantization when the test videos with embedded data are re-encoded again. The error-bit-rate is the probability of error

TABLE 2. Detailed resolution of the test videos.

Test video sequences	Resolution
BasketballPass	416×240
Keiba	416×240
BlowingBubbles	416×240
BQSquare	416×240
RaceHorses	416×240
BQMall	832×480
BasketballDrill	832×480
PartyScene	832×480
KristenAndSara	1280×720
FourPeople	1280×720
ParkScene	1920×1080

into one bit, and the data hiding embedding capacity is the sum of the bits embedded in the twenty I frames.

A. EXPERIMENTAL PERFORMANCE OF VISUAL QUALITY

To prove the performance of visual quality, the visual quality of the proposed method is compared by hiding data in the quantization multi-coefficients without distortion drift and by embedding data with distortion drift. In addition, a 4×4 prediction block with Group 2 and the parameters of *BCH* (7, 4, 1) are used, and data is embedded into *HS* (Y_{01}, Y_{21}, Y_{31}), (Y_{02}, Y_{22}, Y_{32}), (Y_{03}, Y_{23}, Y_{33})} as an example (the data embedded into Y_{00}, Y_{01}, Y_{02}, Y_{03} and Y_{20}, Y_{30}, Y_{21}, Y_{31}, Y_{22}, Y_{32}, Y_{23}, Y_{33} are compensated according to Modulation Method 1) to compare the values of PSNR for every frame. Fig. 9 presents the experimental results of PSNR1 and PSNR2 when using different resolution test video sequences when the 4×4 block meets Group2. We embed data into *HS* {(Y_{00}, Y_{20}, Y_{30}), (Y_{01}, Y_{21}, Y_{31}), (Y_{02}, Y_{22}, Y_{32}), (Y_{03}, Y_{23}, Y_{33})}. The average PSNR1 is 40.56 dB; PSNR2 without and with distortion drift is 39.37dB, 35.81 dB, respectively.

The maximum discrepancy between *PSNR1* and *PSNR2* with the proposed method is 2.16 dB, and the minimum discrepancy is 0.12 dB. However, the maximum between *PSNR1* and *PSNR2* with distortion drift is 7.41 dB, and the minimum discrepancy is 1.98 dB, as described in Fig. 9.

Table 3 provides the experimental results with and without distortion drift for every frame when a prediction block meets Group 2. In Table 3, *POC* represents the display order of a specified video sequence. For the test video sequences BasketballPass and Keiba, the average discrepancy between *PSNR1* and *PSNR2* without distortion drift is 0.28 dB and 0.12 dB, respectively, and the average discrepancy between *PSNR1* and *PSNR2* with distortion drift is 10.05 dB and 8.8 dB, respectively.

The actual visual quality of the video sequences *Basketball Pass* and *Keiba* is shown in Fig. 10; the discrepancy of pixels between *BasketballPass* and *Keiba* is shown in Fig. 11. Fig. 10 (a) and (d) are the original frames.

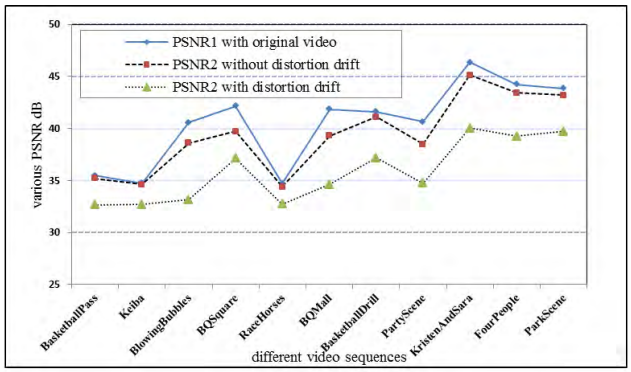


FIGURE 9. Comparison of the average *PSNR* of test video sequences with and without distortion drift when meets Group 2.

TABLE 3. Values of *PSNR1* and *PSNR2* with and without distortion drift for every frame when meeting Group 2.

Video sequences	*POC*	*PSNR1*	*PSNR2* without distortion drift	*PSNR2* with distortion drift
BasketballPass	0	35.42	35.13	22.4
	1	35.35	35.02	30.64
	2	35.4	**35.19**	24.35
	3	35.4	35.12	21.99
	4	35.41	35.12	27.33
Keiba	0	35.33	**35.23**	29.53
	1	35.15	34.96	21.61
	2	34.95	34.87	28.58
	3	34.96	34.8	24.7
	4	34.96	34.86	26.93

Fig. 10 (b) and (e) are the embedded frames without distortion drift. Fig. 10 (c) and (f) are the embedded frames with distortion drift. The visual distortion in the frames Fig. 10 (c) and (f) is obvious and intolerable for data hiding. For example, there is obvious distortion on the bottom of frame (c) and on the middle of frame (f), including the foot of the player and the structure of the railing, as indicated by the green arrows. Fig. 11 (a) is the pixel discrepancy between Fig. 10 (a) and between Fig. 10 (d) and Fig. 10 (e), and Fig. 11 (d) is the pixel discrepancy between Fig. 10 (d) and Fig. 10 (f). In Fig. 11, the white represents different pixel values of the two compared videos; the black represents the identical pixel values of the two compared videos. We can see that the discrepancy distribution introduced by data hiding without distortion drift is dotted; however, the distribution of discrepancy with distortion drift is striped.

Table 4 presents a comparison of *PSNR* between the proposed method and the literatures [26], [29]. In Table 4, *BCH* parameters (63, 7, 15) and Group 2 are considered in the proposed method. The average *PSNR2* in the proposed method is 36.54 dB, in [29] is 34.50 dB, in [26] is 33.57 dB, respectively. It is clear that the visual quality of the proposed method is better than the methods in [26] and [29]. Moreover, the corresponding embedding capacity is larger than the method [26] but less than method in [29] because of the redundancy for BCH coding with parameter (63, 7, 15).

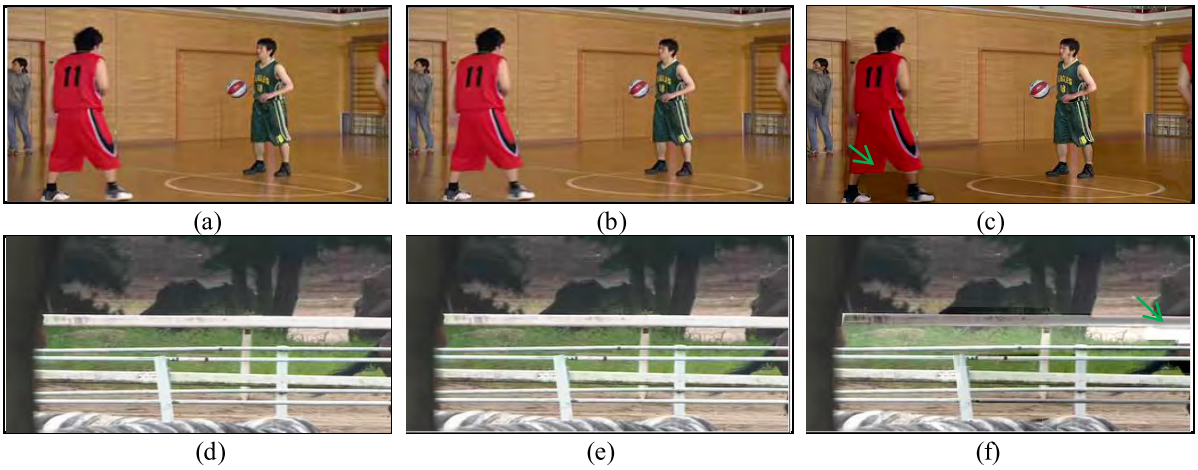


FIGURE 10. Original and embedded frames of Basketball Pass and Keiba data hiding without and with distortion drift at POC = 0.

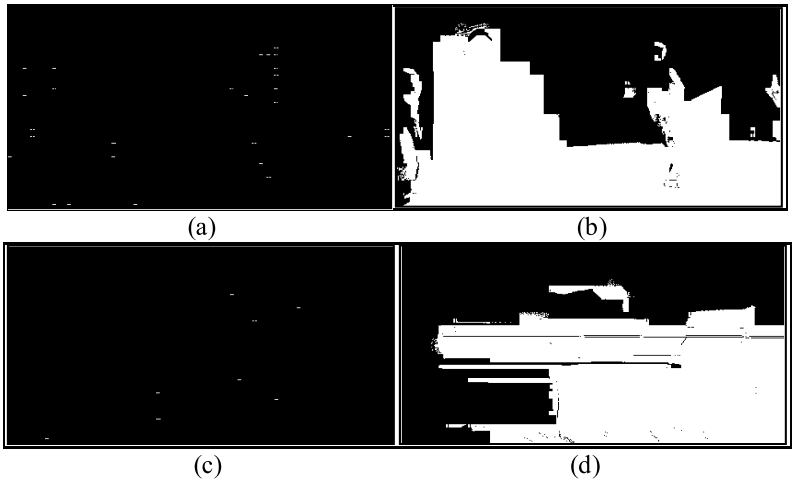


FIGURE 11. Discrepancy on pixels between Basketball Pass and Keiba data hiding without and with distortion drift at POC = 0.

TABLE 4. Comparison *PSNR* with the proposed method and the literature [26], [29] when meeting Group 2.

Video sequence	The Proposed Method		The Literature [29]		the Literature [26]	
	PSNR 2(dB)	*Capacity(bits)*	*PSNR 2(dB)*	*Capacity(bits)*	*PSNR 2(dB)*	*Capacity(bits)*
BasketballPass	**35.34**	362	**32.67**	1424	**34.22**	318
Keiba	**34.58**	238	**32.68**	1024	**32.4**	213
Blowing Bubbles	**39.7**	946	**38.17**	2188	**34.09**	792

It can be concluded from the average *PSNR*, the actual visual quality of every frame or the comparison of *PSNR* that data hiding without intra-frame distortion drift can achieve much better visual quality.

B. EXPERIMENTAL PERFORMANCE OF ROBUSTNESS

To compare the robustness quality of the proposed method and results from the literatures [26], [29], we focus on the unreliable transmission situations of error bits and re- quantization attacks. The multi-coefficients *HS* $\{(Y_{00}, Y_{20}, Y_{30}), (Y_{01}, Y_{21}, Y_{31}), (Y_{02}, Y_{22}, Y_{32}), (Y_{03}, Y_{23}, Y_{33})\}$ are chosen with a 4×4 block when a prediction mode meets Group 2 with a series of attack parameters, such as the error-bit-rate in error bits and the value of *QPs* in re-quantization scenes.

Embedded data error bits may occur with the error-bit probability when transmitted with illegal modification. Fig. 12 presents the experiment results of the survival rate when the embedded data meet different error-bit rates. For *Basketball Pass*, the average survival rate is 97.95% with *BCH* (7, 4, 1), 99.58% with *BCH* (15, 5, 3) and 100% with *BCH* (63, 7, 15). However, the average survival rate with [29] is 93.91%, with [26] is 94.14%. It can be seen from the results that the survival rate with *BCH* (63, 7, 15) can achieve the best robustness among the parameters of *BCH* from (7, 4, 1) to (63, 7, 15), since *BCH* (63, 7, 15) can correct 15 error bits with the segment of 63 embedded bits. Based on the above results, we use the parameter of *BCH* (63, 7, 15) as an example to compare the robustness of the proposed method and [26] and [29] during a re-quantization attack.

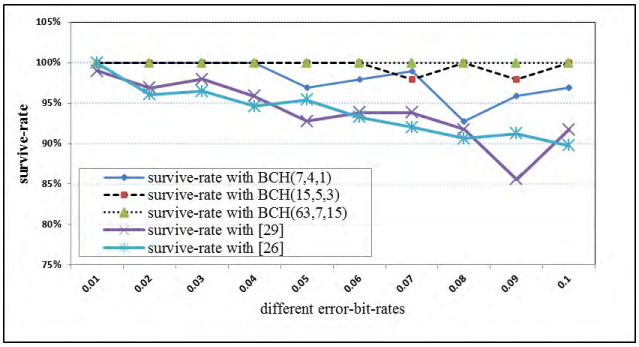


FIGURE 12. Survival rate of BasketballPass when different error bit rates are met.

TABLE 5. Comparison of survival rate between the proposed method and that in [26] and [29].

Video sequence	QP	Survival rate with BCH	Survival rate [29]	Survival rate [26]
BasketballPass	29	77.50%	62.88%	61.53%
	30	92.50%	64.94%	77.77%
	31	**100%**	78.35%	79.29%
	32	**100%**	**87.13%**	82.75%
	33	**95%**	61.85%	**84.61%**
	34	79.18%	61.85%	53.84%
	35	60%	42.26%	22.22%
Keiba	29	89.35%	59.79%	58.85%
	30	100%	62.88%	82.35%
	31	**100.00%**	55.67%	80.32%
	32	**100.00%**	**70.10%**	**81.50%**
	33	**100.00%**	46.39%	84.61%
	34	67.05%	44.32%	38.46%
	35	52.94%	42.26%	30.76%

Table 5 presents the experimental results of survival rate for *BasketballPass* and*Keiba* when the re-quantization *QPs* range from 29 to 35. The average survival rates of the two video sequences with the proposed method are 86.31% and 87.05%. However, the average survival rates with [29] are 65.61% and 54.49%, with [26] are 66.00% and 65.26%, respectively. In particular, for a fixed *QP* of 32 in the embedding procedure, the average survival rate with *BCH* can reach 100% when the *QP* of the re-quantization attack during the re-encoding procedure is 32; however, the best survival rate of [29] reaches 87.13%, of [26] reaches 82.75% in the same situation.

Similarity *(Sim)* and bit error rate *(BER)* have also been used to evaluate the robustness performance of data hiding [34], [35]. The *Sim* and *BER* can be defined as follows:

$$Sim = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}[M(i,j)\times\hat{M}(i,j)]}{\sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b}M(i,j)^2}\times\sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b}\hat{M}(i,j)^2}} \quad (14)$$

$$BER = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}[M(i,j)\oplus\hat{M}(i,j)]}{a\times b} \quad (15)$$

where $M(i,j)$ and $\hat{M}(i,j)$ are the original and obtained data, $a \times b$ is the size of the embedded data. Table 6 shows the robustness of the proposed method against different re-quantization attacks. For Basketball, the highest robustness can be achieved when the re-quantization *QP* are 32, 31, the maximum *Sim* and minimum *BER* can be 1 and 0 with the proposed method, 0.86 and 15.06% with [29], 0.85 and 17.24% with [26], respectively.

TABLE 6. Comparison of *Sim* and *BER* performance of the proposed method with [26] and [29].

Type of Attack		The Proposed Method		The Method in [29]		The Method in [26]	
		Sim	*BER(%)*	*Sim*	*BER(%)*	*Sim*	*BER(%)*
No attacks		1	0	1	0	1	0
Re-quantization	QP=30	0.94	6	0.62	38.35	0.8	22.42
	QP=31	1	0	0.8	22.34	0.81	20.7
	QP=32	**1**	**0**	**0.86**	**15.06**	**0.85**	**17.24**
	QP=33	0.96	4	0.58	42.55	0.85	16.38
	QP=34	0.8	22.23	0.58	42.55	0.5	45.98

The error-bit and re-quantization attack are used as examples to compare the robustness performances of the proposed method and the method in [26] and [29]. It can be concluded from the experimental results that the proposed method can attain higher robustness than [26] and [29], and the proposed method performs better than [26] and [29], especially when the data hiding meets error bits or a re-quantization attack during network transmission.

C. EXPERIMENTAL PERFORMANCE OF THE STATISTICAL CHARACTERISTICS

In this section, we discuss the performance of various statistical characteristics of the proposed method, as presented in Fig. 13 and Tables 7 to 9.

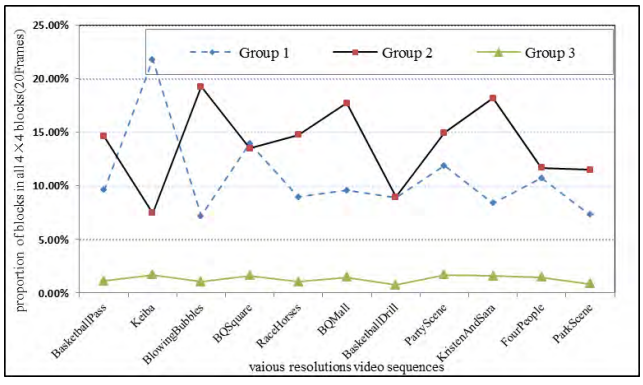


FIGURE 13. Comparison of the proportion of embedding blocks with Group 1, Group 2 and Group 3.

Fig. 13 presents the distributed proportions of suitable blocks with meeting Groups 1, 2 or 3 compared to all the

4×4 prediction blocks. The average number proportions of 4×4 blocks that meet Groups 1, 2 and 3 are 10.74%, 13.86% and 1.32%, respectively.

TABLE 7. The comparisons of embedding capacity when the proposed method achieve better visual quality than [26] and [29].

video sequences	*BCH*	The Proposed Method		The Method in[29]		The Method in[26]	
		Capacity(bits)	PSNR 2 (dB)	*Capacity(bits)*	PSNR 2 (dB)	*Capacity(bits)*	PSNR 2 (dB)
BasketballPass	(7, 4,1)	**1865**	35.22	**1424**	32.67	**318**	34.22
	(15,5, 3)	1088	35.26				
Keiba	(7, 4,1)	**1248**	34.68	**1024**	32.68	**213**	32.4
	(15,5, 3)	716	34.68				
Blowing Bubbles	(7, 4,1)	**4868**	39.03	**2188**	38.17	**792**	34.09
	(15,5, 3)	2840	39.78				

Table 7 gives embedding capacity performance of BasketballPass, Keiba and Blowing Bubbles. The average embedding capacities of the proposed method with BCH (7, 4, 1) are 2660 bits, with BCH (15, 5, 3) are 1548 bits respectively. The average embedding capacity of [29] is 1545 bits; of [26] is 441 bits, respectively. Moreover, Table 4 also shows the embedding capacity of the proposed method with (63, 7, 15) is 515 bits. However, the proposed method with (63, 7, 15) can obtain better visual quality and robustness as described in Table 4 and 5. Tables 8 also provides the comparison of the bit-rate increase performance between the proposed method and the methods in [26] and [29]. The average bit-rate increase of the proposed method with *BCH* (7, 4, 1) is 0.78%, with *BCH* (15, 5, 3) is 0.81%, with *BCH* (63, 7, 15) are 0.74%. The average bit-rate increase of [29] is 0.83%, of [26] is 0.1%, respectively.

TABLE 8. The comparison of Bit-rate increase when embedding blocks meet Group 2.

video sequences	*BCH*	Bit-rate Increase (%)		
		The proposed method	The Method in[29]	The Method in[26]
BasketballPass	(7,4,1)	1.24	**0.94**	**0.09**
	(15,5,3)	1.26		
	(63,7,15)	**1.12**		
Keiba	(7,4,1)	1.04	**1.35**	**0.09**
	(15,5,3)	1.04		
	(63,7,15)	**0.96**		
BlowingBubbles	(7,4,1)	**0.11**	**0.21**	**0.1**
	(15,5,3)	0.15		
	(63,7,15)	0.15		

Data hiding with stricter block selection condition will get lower embedding capacity and better visual quality [7].

TABLE 9. The comparison of execution time with *BasketballPass*.

Methods	Execution Time (*ms*)	
	Embedding	extraction
The Proposed Method	631	152
The Method in[29]	609	187
The Method in[26]	648	184

The utilization of prediction block selection and multi-coefficients can facilitate the tradeoff between the payload and the visual quality lossless for data hiding. We can improve the embedding capacity by using more *H.265* video sequences since the infinite video sequences of *H.265/HEVC* just meet the redundancy properties of data hiding for the strict visual quality demand scenario. In addition, we also can use alternative parameters of *BCH* to improve the embedding capacity such as *BCH* (7, 4, 1). Thus, we can get the embedding capacity what we want.

All procedures are simple and fast. The data hiding procedure for the video *BasketballPass* with the proposed method costs 631*ms* on the embedding time, 152*ms* on the extraction time; with the method [29] costs 609*ms* on the embedding time, 187*ms* on the extraction time, with the method [26] costs 648*ms* on the embedding time, 184*ms* on the extraction time, respectively.

It can be concluded from the experimental results that the proposed method can achieve sufficient embedding capacity, low bit-rate increase and fast processes on the embedding and extraction time.

V. CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

In this paper, the solution to avoid intra-frame distortion drift and the *BCH* code technique to improve the robustness of data hiding has been investigated in data hiding based on the *H.265* standard. The proposed data hiding procedure includes *BCH* code, data embedding, entropy decoding, data extraction and BCH decoding.

As shown in this work, the combination of multi-coefficients and prediction modes can be a desirable technique to resist the distortion drift, and *BCH* coding is a

powerful technique explored for the robust data hiding method to resist error bits in the embedded data. The experimental results also show that data hiding with the proposed method can achieve more robustness and better visual quality when compared to the state-of-the-art methods [26], [29].

There remain some essential improvements of data hiding for future work. The most prominent one is the temporal distortion drift introduced by data hiding. Even though the combination of the relevant coefficients and prediction modes can solve the problem of intra-distortion drift, inter-distortion drift (the temporal domain), referring to the motion vector, still need to be considered in the future. Another potential area of improvement is the explicit robustness of other smart attacks (e.g., attacks based on modifying the block prediction modes), which will enhance the survive rate of the embedded data in more data hiding applications.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] Y. H. Chen and V. Sze, "A deeply pipelined CABAC decoder for HEVC supporting level 6.2 high-tier applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 5, pp. 856–868, May 2015.

[3] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—Including high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[4] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 59–69, May 2011.

[5] V. Sze and M. Budagavi, "High throughput CABAC entropy coding in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1778–1791, Dec. 2012.

[6] X. Zhang, J. Long, Z. Wang, and H. Cheng, "Lossless and reversible data hiding in encrypted images with public-key cryptography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1622–1631, Sep. 2016.

[7] D. Khanaa and D. Mohanta, "Secure and authenticated reversible data hiding in encrypted images," *Int. J. Eng. Comput. Sci.*, vol. 2, no. 3, pp. 558–568, 2013.

[8] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: Advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.

[9] D. Xu, R. Wang, and Y. Q. Shi, "An improved scheme for data hiding in encrypted H.264/AVC videos," *J. Vis. Commun. Image Represent.*, vol. 36, pp. 229–242, Apr. 2016.

[10] H.-T. Wu, J. Huang, and Y.-Q. Shi, "A reversible data hiding method with contrast enhancement for medical images," *J. Vis. Commun. Image Represent.*, vol. 31, pp. 146–153, Aug. 2015.

[11] D. Xu, R. Wang, and Y. Zhu, "Tunable data hiding in partially encrypted H.264/AVC videos," *J. Vis. Commun. Image Represent.*, vol. 45, pp. 34–45, May 2017.

[12] D. Xiao, Y. Xiang, H. Zheng, and Y. Wang, "Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism," *J. Vis. Commun. Image Represent.*, vol. 45, pp. 1–10, May 2017.

[13] P.-C. Chang, K.-L. Chung, J.-J. Chen, C.-H. Lin, and T.-J. Lin, "A DCT/DST-based error propagation-free data hiding algorithm for HEVC intra-coded frames," *J. Vis. Commun. Image Represent.*, vol. 25, no. 2, pp. 239–253, 2014.

[14] D. Xu, R. Wang, and Y. Q. Shi, "Data hiding in encrypted H.264/AVC video streams by codeword substitution," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 4, pp. 596–606, Apr. 2014.

[15] X. Zhang, "Reversible data hiding with optimal value transfer," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 316–325, Feb. 2013.

[16] Z. Qian, X. Zhang, and S. Wang, "Reversible data hiding in encrypted JPEG bitstream," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1486–1491, Aug. 2014.

[17] Z. Qian and X. Zhang, "Reversible data hiding in encrypted images with distributed source encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 4, pp. 636–646, Apr. 2016.

[18] H. Zhang, Y. Cao, X. Zhao, H. Yu, and C. Liu, "Data hiding in H.264/AVC video files using the coded block pattern," in *Proc. Int. Workshop Digit. Watermarking*, 2016, pp. 588–600.

[19] H. Jing, X. He, Q. Han, and X. Niu, "Motion vector based information hiding method for H.264/AVC against motion vector steganalysis," in *Proc. Asian Conf. Intell. Inf. Database Syst.* Berlin, Germany: Springer, 2012, pp. 91–98.

[20] G. Qiu, P. Marziliano, A. T. S. Ho, D. He, and Q. Sun, "A hybrid watermarking scheme for H.264/AVC video," in *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 4, 2004, pp. 865–869.

[21] J. Zhao, Z. Li, and B. Feng, "A novel two-dimensional histogram modification for reversible data embedding into stereo H.264 video," *Multimedia Tools Appl.*, vol. 75, no. 10, pp. 5959–5980, 2016.

[22] X. Ma, Z. Li, J. Lv, and W. Wang, "Data hiding in H.264/AVC streams with limited intra-frame distortion drift," in *Proc. Int. Symp. Comput. Netw. Multimedia Technol.*, 2010, pp. 1–5.

[23] X. Ma, Z. Li, H. Tu, and B. Zhang, "A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1320–1330, Oct. 2010.

[24] M.-J. Hwang, J. Lee, M. Lee, and H.-G. Kang, "SVD-based adaptive QIM watermarking on stereo audio signals," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 45–54, Jan. 2018.

[25] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin, "Robust lossless image data hiding designed for semi-fragile image authentication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 4, pp. 497–509, Apr. 2008.

[26] Y. Liu, S. Jia, M. Hu, Z. J. Jia, L. Chen, and H. G. Zhao, "A reversible data hiding method for H.264 with Shamir's (t, n)-threshold secret sharing," *Neurocomputing*, vol. 188, pp. 63–70, May 2016.

[27] Y. Liu, Z. Li, X. Ma, and J. Liu, "A robust without intra-frame distortion drift data hiding algorithm based on H.264/AVC," *Multimedia Tools Appl.*, vol. 72, no. 1, pp. 613–636, Sep. 2014.

[28] Y. Tew and K. Wong, "Information hiding in HEVC standard using adaptive coding block size decision," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2015, pp. 5502–5506.

[29] S. Swati, K. Hayat, and Z. Shahid, "A watermarking scheme for high efficiency video coding (HEVC)," *PLoS ONE*, vol. 9, no. 8, p. e105613, 2014, doi: 10.1371/journal.Pone.0105613.

[30] Y. Liu, S. Liu, H. Zhao, S. Liu, and C. Feng, "A data hiding method for H.265 without intra-frame distortion drift," in *Proc. Int. Conf. Intell. Comput.*, 2017, pp. 642–650.

[31] Y. Liu, C. Yang, and Q. Sun, "Enhance embedding capacity of generalized exploiting modification directions in data hiding," *IEEE Access*, vol. 6, pp. 5374–5378, Dec. 2017.

[32] B. Bross, W. J. Han, G. J. Sullivan, J. R. Ohm, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 9*, document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct. 2012.

[33] Y. X. Liu, Z. T. Li, X. J. Ma, and J. Liu, "A robust data hiding algorithm for H.264/AVC video streams," *J. Syst. Softw.*, vol. 86, no. 8, pp. 2174–2183, 2013.

[34] R. J. Mstafa and K. M. Elleithy, "A novel video steganography algorithm in the wavelet domain based on the KLT tracking algorithm and BCH codes," in *Proc. Long Island Syst., Appl. Technol.*, May 2015, pp. 1–7.

[35] R. J. Mstafa, K. M. Elleithy, and E. Abdelfattah, "A robust and secure video steganography method in DWT-DCT domains based on multiple object tracking and ECC," *IEEE Access*, vol. 5, pp. 5354–5365, 2017.

[36] A. K. Singh, B. Kumar, S. K. Singh, S. P. Ghrera, and A. Mohan, "Multiple watermarking technique for securing online social network contents using back propagation neural network," *Future Gener. Comput. Syst.*, vol. 86, pp. 926–939, Sep. 2018, doi: 10.1016/j.future.2016.11.023.

[37] C. Rupa, "A digital image steganography using Sierpinski gasket fractal and PLSB," *J. Inst. Eng. India*, vol. 94, no. 3, pp. 147–151, Sep. 2013.

[38] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Inf. Sci.*, vol. 279, pp. 251–272, Sep. 2014.

[39] Y. Tew and K. Wong, "An overview of information hiding in H.264/AVC compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 305–319, Feb. 2014.

[40] R. J. Mstafa and K. M. Elleithy, "A high payload video steganography algorithm in DWT domain based on BCH codes (15, 11)," in *Proc. Wireless Telecommun. Symp. (WTS)*, Apr. 2015, pp. 1–8.

[41] R. J. Mstafa and K. M. Elleithy, "A DCT-based robust video steganographic method using BCH error correcting codes," in *Proc. IEEE Long Island Syst.*, Apr. 2016, pp. 1–6.

[42] H. Yoo, J. Jung, J. Jo, and I.-C. Park, "Area-efficient multimode encoding architecture for long BCH codes," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 12, pp. 872–876, Dec. 2013.

[43] R. J. Mstafa and K. M. Elleithy, "An efficient video steganography algorithm based on BCH codes," in *Proc. Northeast Sect. Conf. Amer. Soc. Eng. Edu. (ASEE)*, 2015, pp. 1–10.

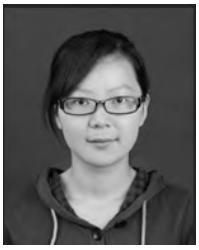
[44] H.-T. Hu and L.-Y. Hsu, "Incorporating spectral shaping filtering into DWT-based vector modulation to improve blind audio watermarking," *Wireless Pers. Commun.*, vol. 94, no. 2, pp. 221–240, 2017.

[45] M. Long, F. Peng, and H.-Y. Li, "Separable reversible data hiding and encryption for HEVC video," *J. Real-Time Image Process.*, vol. 14, no. 1, pp. 171–182, 2018.

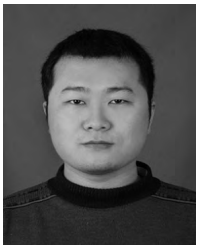
YUNXIA LIU was born in Zhoukou, China, in 1972. She received the B.E. degree from Henan Normal University, Xinxiang, China, in 1996, and the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, 2013. She is currently a Professor with the College of Information Science and Technology, Zhengzhou Normal University. Her research interests include cryptography, network security, and multimedia security.

HONGGUO ZHAO was born in Shanxi, China, in 1990. He received the master's degree from the Huazhong University of Science and Technology, Wuhan, China, 2015. He joined the College of Information Science and Technology, Zhengzhou Normal University, in 2016. His research interest is video steganography and network security.

SHUYANG LIU was born in Zhoukou, China, in 1998. He is currently pursuing the degree with the School of Mathematics and Statistics, Lanzhou University. His research interest is mathematical modeling, scientific computing, and network security.



CONG FENG was born in Shangqiu, China, in 1987. She received the master's degree from Guangxi Normal University, Guangxi, China, in 2013. She joined the College of Information Science and Technology, Zhengzhou Normal University, in 2013. Her research interest is information hiding and artificial intelligence.



SI LIU was born in Xinyang, China, in 1985. He received the master's degree from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2011. He joined the College of Information Science and Technology, Zhengzhou Normal University, in 2011. His research interest is information hiding and artificial intelligence.