# Adaptive Video Data Hiding through Cost Assignment and STCs

Yanli Chen, Hongxia Wang, Hanzhou Wu, Zhiqiang Wu, Tao Li and Asad Malik

**Abstract**—With the increasing popularity of digital video communication, video data hiding has become an active research topic in covert communication and privacy protection. Traditional video data hiding methods often use quantized discrete cosine transform (QDCT) coefficients to carry a sufficient payload. However, since QDCT coefficients expose texture features and motion characteristics of the present video frame heavily, data embedding with QDCT coefficients may lead to significant intra-frame distortion and inter-frame distortion drift. To avoid obvious visual artifacts and keep bit-rate within a satisfactory level of the marked video, data embedding in QDCT coefficients should take into account both the intra-frame and inter-frame distortion impacts. It motivates the authors to propose an efficient cost assignment-based video data hiding method in this paper. The proposed cost assignment method aims to accurately evaluate the data embedding distortion. Specifically, the proposed scheme considers intra-frame changes and intra-frame distortion drift, for which the texture and motion changes of frames can be measured. The frame position is also used to reflect a cumulative distortion difference of multiple frames. For data embedding, syndrome-trellis code (STC) is adopted to minimize the overall distortion. Experimental results show that the proposed method significantly outperforms existing works in terms of payload-distortion performance.

**Index Terms**—Video data hiding, Cost assignment, Minimal embedding distortion, Syndrome-trellis code.

## 1 Introduction

Modern data hiding is the art and science of covert communication which hides secret messages in the cover media such as text, audio, video, image and so on. In recent years, with the rapid development of portable terminals technologies, broadband internet service, and multimedia processing technologies, the video is becoming a popular media on the internet, especially in social network and entertainment industry. Meanwhile, with the wide application of digital video, copyright and privacy protection for videos attract a lot of attention where data hiding is one of the most promising solutions.

Video sequence can be considered as multi-images in chronological order, thus, some image data hiding methods have been planted to video data hiding as well, such as least significant bit matching methods [1] [2], spread spectrum methods [3], histogram shifting methods [4], parity check methods [5] [6], reversible data hiding methods [7] [4] [8] [9] [10], [11], [12], syndrome-trellis codes(STCs) [13] [14], and digraph matching [15]. However, comparing with image data hiding, in video data hiding we need to consider not only the spatial feature in video frames but also the temporal features among frames.

*Yanli Chen is with School of Big data and Computer Science, Guizhou Normal University, Guiyang 550025, Guizhou, China, and also with he school of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, Sichuan, China.*
*Hongxia Wang and Tao Li are with College of Cybersecurity, Sichuan University, Chengdu ,610064, Sichuan, China.*
*Hanzhou Wu is with Shanghai University, and also with Shanghai Institute for Advanced Communication and Data Science, Shanghai 200444, China.*
*Zhiqiang Wu is with Tibet University, Lhasa 850000, China, and also with Wright State University, Dayton, OH, USA 45435.*
*Asad Malik is with School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, Sichuan, China.*
*Corresponding author: Hongxia Wang, Email:hxwang@scu.edu.cn*

Usually, data hiding techniques embed data into carriers to achieve specific features or serve specific purposes. There are two categories of video data hiding schemes. Firstly, data hiding is performed in the spatial domain, and data is embedded in the pixel values directly (e.g., least significant bit matching). However, modifications in these methods may be overridden during video compression procedure. Secondly, data is embedded in the compressed domain, and the modifications are realized in the entities of compressed video. Generally, the entities include motion vectors(MVs) [16] [17] [18], prediction modes [19] [20], quantized discrete cosine transformation(QDCT) coefficients [21] [22] and variable length codes [23] [24].

Goals of most data hiding methods are to embed data with minimal embedding impact. As both spatial and temporal features exist in the video, the error-tolerance degree of the human visual system (HVS) in a video is higher than that in an image. Nevertheless, the entities provided by compression standard include motion residual, MVs, prediction modes and so on. Their values are always low, then, small embedding distortion may have a great impact. Embedding distortion includes additive distortion and nonadditive distortion. In [13] [14], the additive distortion was considered in image steganography [25], and authors pointed out that the data hiding method using STCs could minimize embedding distortion. For images, there are spatial correlations between neighbor pixels, and changes on adjacent pixels will interact with each other. In [26], joint distortion for adaptive stenography was introduced where blocks with two or more pixels were considered as super-pixels, and then the joint distortion of a super-pixels was calculated. In [27], a cost assignment method in the image using controversial pixels prior (CPP) rule was proposed. It revealed that controversial pixels are not sensitive to

modification, and consequently, data was embedded into controversial pixels with minimal embedding costs. In [28], a distortion metric was proposed and it was calculated by a correlation of inter- and intra-block, and distortions were calculated based on this distortion metric. Overall, these distortions are considered in the image data hiding schemes, and they cannot be applied to video data hiding directly.

In the video, previous frames are reference frames of subsequence frames. I-frame is the basic frame in a group of pictures (GOP), and distortion in it would produce more error accumulation. In a GOP with IPPP encoding mode, every P-frame is the reference of next P frames, and they are encoded in inter-frame mode. Therefore, changes in previous frames would have an impact on the next frames. In [29], the inter-frame distortion drift was introduced, and it declared that the distortion in previous frames can be accumulated to the next frames and it also provided methods to prevent inter-frame distortion drift. In [30], authors proposed a video data hiding method using low distortion transform, and embedded message into prediction errors that were produced within two transformations. In [5], a high capacity data hiding method using parity checking was proposed. The message was embedded into the position of last non-zero (LNZ) coefficient for every $4 \times 4$ block, and capacity was adjustable by changing a threshold. In [9], a 3D RDH was proposed to embed the message into three coefficients of $4\times4$ blocks in a macro-block (MB) which were selected randomly. In summary, most of the existing data hiding schemes focus on discussing embedding distortion, and little research is about the relationship between embedding modification and its cost which is also very important for data hiding.

In this paper, we propose an adaptive video data hiding through cost assignment and STCs which aims to minimize the embedding distortion and can be used in the application scenarios that need low embedding distortion, especially for covert communication. In the video sequence, since distortion accumulation exists, distortion caused by embedding in the current frame is measured by the impact on both current frame and subsequent frames which are considered as intra-frame distortion and inter-frame distortion drift respectively. For intra-frame distortion, since modification in high-frequency coefficients is not sensitive to HVS, motion and texture features are important factors which have a visual impact on marked video quality. For inter-frame distortion, it is accumulated from prior frames to subsequence frames, then, frame position can reflect the distortion accumulation well. Consequently, we consider these three factors in the cost assignment function. Based on the proposed cost assignment function, embedding operation is implemented by adjusting the embedding order in frames. So, we select different coefficients as cover in different blocks and embed message in subsequent frames with higher priority to reflect the adaptive of proposed scheme.

The rest of this paper is organized as follows. In Section 2, the motivation of this paper is described. The main idea of content-adaptive data hiding through cost assignment and STCs is introduced in Section 3. In Section 4, experimental results are introduced. At last, the paper is concluded in Section 5.

TABLE 1
symbol and their meaning

| symbol | Description |
|---|---|
| $B_{cur}, R, B_{ref}$ | pixels block |
| $f_i, \hat{f}_i, r_i$ | i-th video frame |
| $C_1$ | DCT coefficients set |
| $Ac_{i,j}, \rho_{i,j}$ | DCT coefficient and spatial frequency |
| $w_h, w_v$ | sizes of a pixel |
| $N_4, N_{16}, N_{cover}$ | block numbers, cover number |
| $f_{cost}$ | cost assignment function |
| $\phi_l, \phi_{mv}, \phi_f$ | frame position,MV and texture factor |
| De(i),Ds(i),$\bar{De}$, $\hat{De}$ | The embedding distortion |
| $\alpha, \beta$ | distortion constant and scaling factor |
| $Mv_x, Mv_y$ | components of MV |
| block1 | one dimension DCT coefficients of a block |
| fz(i),fz | frequency zone and their set |
| $Wf_i$ | texture degree |
| Sumf | the coefficients sum |
| e | cover element |
| fnum | frame number of a GOP |
| $\eta$ | distortion increment factor |
| m | message |
| $x_i, y_i$ | location of pixel window |
| C | cover |
| MC | marked cover |
| $\mu_{x_i}, \mu_{y_i}$ | mean values |
| $\sigma_{x_i}, \sigma_{y_i}$ | standard deviation |

## 2 MOTIVATION

In a video, every frame can be considered as an image, and great spatial correlation among pixels exist in it. Texture feature, which makes the masking effect in visual, is one of its important features. Meanwhile, a great temporal correlation exists in the video which is considered as motion property. Based on the spatial and temporal correlations, we propose a cost assignment which is used to find minimal distortion for data hiding. The main motivations of the proposed scheme are as follows. We also summarize our notation in Table1.

### 2.1 Embedding Method

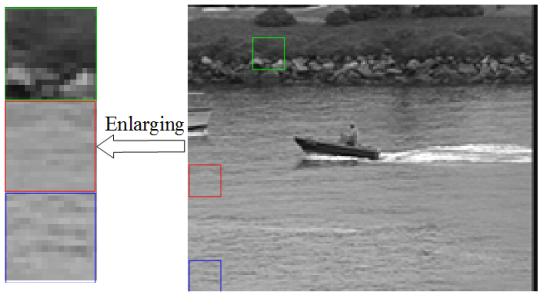
Most data hiding methods try to embed message with minimum distortion in the covers. In [14], authors showed that the embedding method with syndrome-trellis codes (STCs) may minimize embedding impact through finding the optimal path in syndrome trellis. For data hiding schemes with STCs, the cost is considered as an additive distortion, and embedding with the optimal path leads to a minimal distortion. The main idea of this paper is to assign embedding cost for every cover element and minimize distortion. Therefore, we embedded the message using STCs.

### 2.2 Embedding Capacity and Texture Feature

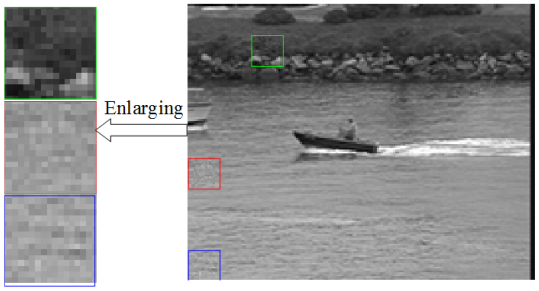
In [31], authors indicated that HVS was less sensitive to the high-frequency component of spatial frequency, and nonzero high-frequency coefficients existed in the abundant texture areas. Naturally, the error-tolerance rate of the human visual system (HVS) is related to the texture feature for images and videos. In other words, the areas with abundant texture make HVS less sensitive to high-frequency components. Thus, coefficients in texture areas can carry more bits than that in plain areas. Then, we embed message with different size into blocks adaptively according to the texture features.

To explain the error-tolerance performance, we use three blocks in a frame with different texture features and use

the 3rd frame of sequence *coastguard* as the test frame. The blocks with size $16 \times 16$ marked by red, blue, and green color edging shown in Fig.1(a) are located in plain, edge and texture areas, respectively, and these blocks are enlarged in the left. To illustrate the impact of data hiding on different texture areas, we embed the same message into the three blocks by substituting 4 least significant bits of the cover pixels, and the results are shown in Fig.1(b) in which the three blocks are also enlarged in the left. The figure shows that blocks with red edging are distinguishable mostly, and the difference of blocks with green edging is inconspicuous compared with the other blocks. This example clearly illustrates that the texture areas can tolerant more errors, and texture feature can influence embedding cost and capacity of a block.




(a) original frame




(b) marked frame

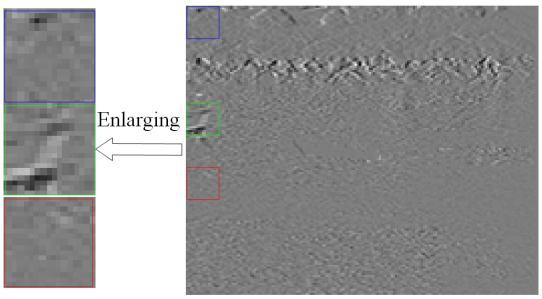
Fig. 1. The visual difference in different texture area

### 2.3 Motion Property

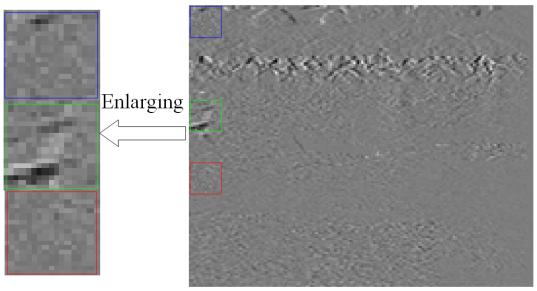
Motion property is a typical feature of video and one of the key properties which are different from images. For activity blocks, they have several characteristics. First, activity blocks attract more attention, modifications in them may generate obvious visual errors. Second, activity blocks always have greater residuals which need to be encoded and transmitted. Assuming $B_{ref}$, $B_{cur}$, and $R$ denote the reference, current and their residual, then, their correlation can be described as:

$$B_{cur} = R + B_{ref} \tag{1}$$

Residual is an important part of the activity frame/block, and more modification on them would impact much on visual quality. For a block, if the percentage of residual in higher activity blocks is higher, then, its embedding cost is also higher. We used frame motion residual and three blocks with color edges in Fig.2(a), in which the values of residual decreased with blocks in green, blue and red edging, respectively, to verify the relationship between the embedding impact and motion property. Similarly, we enlarged the three blocks in Fig.2(a). In order to explain the impact of embedding data in different motion property blocks, we embedded the same message with 1024 bits into the three blocks with the same method in Section 2.2, and the results are shown in Fig.2(b) in which the three marked blocks are enlarged to display the difference clearly. Comparing with the block residuals in Fig.2, we can get the truth that the impact of embedding is inversely proportional to their residual values.




(a) original frame residual




(b) marked frame residual

Fig. 2. The visual difference in different motion residual

## 3 Proposed Scheme

We propose an adaptive data hiding scheme through cost assignment and STCs in which cost assignment function is constructed according to features of the video sequence, such as texture feature, motion property, and frame position. Texture feature and motion property make embedding capacity different in distinct blocks, and frame position makes the current P-frame have to accept the distortion from previous frames, which is called inter-frame distortion drift. To reduce the inter-frame distortion drift, embedding locations are selected with a principle of minimal distortion. First, embedding locations should be selected to prevent the inter-frame drift. Second, the embedding cost of every embedding location should be assigned according to the texture feature, motion property of current blocks, and also frame position. At last, the message is hidden into cover with STCs method adaptively. In this paper, we consider video as a carrier, and we construct cover by selecting QDCT coefficients from video carrier.

### 3.1 Cover Generation

In [29], inter-frame distortion drift for every QDCT coefficient in a $4 \times 4$ block was discussed, and we conclude that distortion drift cannot be avoided, but embedding in

the coefficients in $C_1$ can achieve the least distortion drift. Fig.3 shows the coefficients of a $4 \times 4$ block where the red positions are coefficients in $C_1$. Thus, we selected all the alternating current (AC) coefficients in $C_1$ [29] as the embedding locations.

$$C_1 = \{AC_{2,2}, AC_{2,4}, AC_{4,2}, AC_{4,4}\} \tag{2}$$

In [32], authors revealed that the sensitivity of HVS in middle frequency is higher than that of others, and it is less sensitive in the low frequency than that in high frequency. That means we can use the texture in blocks, which may have different sensitivity of HVS, to conceal the embedding distortions. Since elements in $C_1$ are the middle and high-frequency coefficients, considering the texture feature, the modification in high-frequency coefficients of texture area would not catch much attention. Conversely, the mid-frequency coefficients in the plain area represent the texture feature, and the distortion in them is lower than that of modification in the high-frequency coefficients with zero values.

In general, data hiding schemes always embed message by modifying cover elements, and it have impact on the texture feature of the whole block. So, although the cover element is a single coefficient, the texture feature factors should be assigned in the whole frequency zone. In Fig. 3, the locations around the coefficients in $C_1$ with same colors are considered as the same frequency zone. We divide DCT coefficients in a block into low-frequency, median-frequency, median-high frequency, and high-frequency zones. The low frequency zone concludes the coefficients $AC_{1,2}, AC_{2,1}, AC_{1,3}$ and $AC_{2,2}$, and coefficients $AC_{3,1}, AC_{4,1}, AC_{3,2}, AC_{2,3}, AC_{1,4}$ and $AC_{2,4}$ belong to median-frequency zone, and $AC_{3,3}, AC_{4,2}$ belong to median-high-frequency zone, then $AC_{4,3}, AC_{3,4}$and $AC_{4,4}$ are in the high frequency zone.

Based on these frequency zones and texture feature, the cost assignment should consider both the impact of modification on the single element and the whole frequency zone. To calculate the cost conveniently, we consider the proportion of the frequency zone in the whole block as *texture degree*. For example, if $AC_{2,2}$ is one of the cover elements, the texture degree is the ratio of absolute sum of coefficients $AC_{1,2}, AC_{2,1}, AC_{1,3}, AC_{2,2}$ to the absolute sum of all AC coefficients in the block. According to the principle of minimizing embedding impact on video in visual, the selection of cover elements is important, and the cover elements are generated as follows:

**step 1)** Searching the edge pixels in a frame before encoding, and then classifying blocks into texture, edge and plain areas according to edge pixels number.

**step 2)** Selecting the elements in $C_1$ as cover elements after the DCT coefficients are quantized:

- For plain blocks, HVS is less sensitive to the low-frequency coefficients, then, $AC_{2,2}$ is selected as a cover element.
- There are nonzero median-high-frequency coefficients for edge blocks, and also there are two elements in $C_1$ belonging to the media-high-frequency. Coefficients $AC_{4,2}, AC_{2,4}$ are chosen as cover elements.

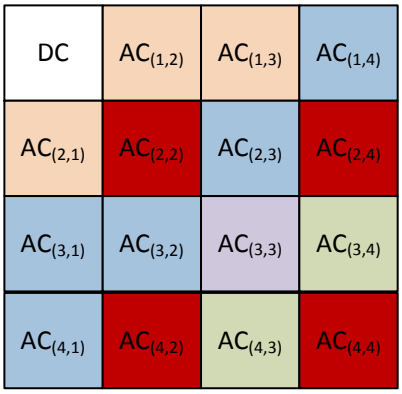



Fig. 3. DCT coefficients of a $4 \times 4$ block

- In texture area, only a few high-frequency coefficients are zeros. These abundant textures make modification unnoticeable. So, we select coefficient $AC_{4,4}$ as the cover element in the texture area.

Now, assuming $N_4$ is the number of $4 \times 4$ blocks in a frame, the number of cover elements is dynamic:

$$N_4 \leq N_{cover} \leq 2N_4 \tag{3}$$

Since we implement the block classification before video encoding, there may be some difference between the decoded and original video sequence, and it may result to incorrect block classification and error bit rate of the extracted data. Actually, in this scheme, it may reduce the probability of incorrect block classification that all the $4 \times 4$ blocks in an MB share the block classification results.

### 3.2 Cost Assignment Function

As we have shown in Section 2, cost assignment is related to texture feature, motion property, and frame position. For $P$ frames, prior $P$ frames are the reference of the subsequent frames, both the motion property and frame position can result in distortion accumulation, and frequency distribute characteristic (also called texture feature in the spatial domain) is an important factor which may make distortion noticeable in visual. The cost assignment function is determined by all the three factors, and it can be written as

$$f_{cost} = f(\phi_l, \phi_{Mv}, \phi_f) \tag{4}$$

where $\phi_l, \phi_{Mv}, \phi_f$ indicate the factors of frame position, motion property, and texture feature.

#### 3.2.1 *Frame Position Factor*

In a video sequence, the distortion in prior frames is accumulated in the subsequent frames. We assume there are $fnum$ frames in a GOP, and the current frame is $f_i$. In [29], authors illustrated that the distortion in the current frame consists of the distortion accumulated from the previous frame and the embedding distortion in the current frame. We rewrite it as

$$\begin{cases} D_s(2) = & \alpha D_s(1) + D_e(2) \\ D_s(3) = & \alpha D_s(2) + D_e(3) \\ \quad \vdots & \\ D_s(i) = & \alpha D_s(i-1) + D_e(i) \end{cases} \tag{5}$$

where $\alpha \in (0,1)$ is a constant related to the video content, and $D_e(i), D_s(i)$ are the embedding distortion and total distortion in the current frame. As shown in Eq. (5), for a fixed frame, the embedding distortion is diffused to subsequence frames, and the proportion is increased monotonously with $\alpha^{i-1}$. However, since $\alpha$ is within the range of $(0,1)$, diffused distortion of a fixed frame to subsequence frames is inversely proportional to the position index $i$. In other words, the frame factor is inversely proportional to the frame index, and the linear function can keep the original relationship between the independent variable and dependent variable and then the frame position factor can be expressed in a simple way in Eq. (6).

$$factor_l = \beta / i \tag{6}$$

where $\beta$ is the scaling factor which can adjust the distortion with the frame position. It determines the inter-frame distortion drift in every $P$ frames and $\beta$ is determined by many factors, such as activity, residual and so on. To make the function simple, we set it to be 1.

### 3.2.2 Motion Feature Factor

The motion vector and motion residual are two measurements of the motion feature, and they have positive correlations with each other. Compared with motion residual, motion vectors can express the motion feature more directly. Motion vectors consist of magnitudes and phases. The magnitude indicates the magnitude of activity, and the phase indicates the motion direction. Assuming $f_0, f_1, f_2, \cdots, f_n$ denote the I-frame and n P-frames in a GOP, and $r_1, r_2, \cdots, r_n$ denote the residual of P-frames. Their relationship can be written as

$$\begin{cases} f_1 = & f_0 + r_1 \\ f_2 = & f_0 + r_1 + r_2 \\ \quad \vdots & \\ f_n = & f_0 + r_1 + r_2 + \cdots + r_n \end{cases} \tag{7}$$

Eq. (7) shows that every frame can be expressed as the summation of I-frame and residual in the prior frame. Thus, residual is an important part of the current frame. Compared with static blocks, residual of activity blocks is greater, and it results in a high percentage of residual in the original block values and has much impact on the original blocks. Since motion vectors reflect the residual, we can consider that motion vector is directly proportional to modification cost and its magnitude can describe the activity of a block. Let $Mv_x, Mv_y$ denote the horizontal and vertical components of MV, thus, the motion feature factor can be written as the magnitude of motion vectors.

$$\phi_{Mv} = \sqrt{Mv_x^2 + Mv_y^2} \tag{8}$$

### 3.2.3 Frequency Distribution Factor

We reorder the QDCT coefficients in Eq. (9).

$$\begin{aligned} block1 = \{&DC, AC_{2,1}, AC_{1,2}, AC_{1,3}, AC_{2,2}, AC_{3,1}, \\ &AC_{4,1}, AC_{3,2}, AC_{2,3}, AC_{1,4}, AC_{2,4}, AC_{3,3}, \\ &AC_{4,2}, AC_{4,3}, AC_{3,4}, AC_{4,4}\} \end{aligned} \tag{9}$$

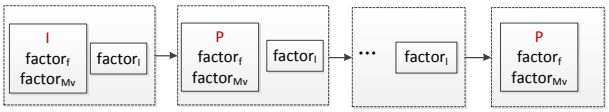



Fig. 4. The model of the factors in a video

where $DC$ is the direct current (DC) coefficient. As we know, $AC_{2,2}, AC_{2,4}, AC_{4,2}, AC_{4,4}$ are coefficients with the lowest distortion drift, and they are considered as cover elements. Since they are located in different spatial frequency zones, we can rewrite the four frequency zones as follows.

$$\begin{aligned} fz &= \{fz(1), fz(2), fz(3), fz(4)\} \\ &= \{\{AC_{2,1}, AC_{1,2}, AC_{1,3}, AC_{2,2}\}, \\ &\quad \{AC_{3,1}, AC_{4,1}, AC_{3,2}, AC_{2,3}, AC_{1,4}, AC_{2,4}\}, \\ &\quad \{AC_{3,3}, AC_{4,2}\}, \{AC_{4,3}, AC_{3,4}, AC_{4,4}\}\} \end{aligned} \tag{10}$$

The frequency in four frequency zones in Eq. (10) decreases roughly, and we let $Wf_1, Wf_2, Wf_3, Wf_4$ denote the four texture degrees, and their values are proportional to the texture feature in the corresponding frequency zone.

$$\begin{cases} Wf_1 = & \dfrac{\sum\limits_{e \in fz(1)} |e|}{\sum\limits_{i=4,j=4}^{i=1,j=1} |AC_{i,j}|} \\ Wf_2 = & \dfrac{\sum\limits_{e \in fz(2)} |e|}{\sum\limits_{i=4,j=4}^{i=1,j=1} |AC_{i,j}|} \\ Wf_3 = & \dfrac{\sum\limits_{e \in fz(3)} |e|}{\sum\limits_{i=4,j=4}^{i=1,j=1} |AC_{i,j}|} \\ Wf_4 = & \dfrac{\sum\limits_{e \in fz(4)} |e|}{\sum\limits_{i=4,j=4}^{i=1,j=1} |AC_{i,j}|} \end{cases} \tag{11}$$

Generally, embedding in the area with abundant texture is imperceptible, and it implies that embedding cost is low. As we know, Eq. 11 shows $0 \leq Wf_i \leq 1$, thus, the frequency factor can be expressed as

$$\phi_f = 1 - Wf_i, i = 1, 2, 3, 4 \tag{12}$$

### 3.2.4 Discussion on Expression of Cost Assignment Function

Based on the above factors, we can express the cost assignment function according to two aspects.

First, since frame position exists in the whole GOP and has an impact on the quality of P-frames, we consider a series of video sequences as a 4D signal $f(\phi_f, \phi_{Mv}, \phi_l)$. For every frame position $\phi_l$, the signal is a frame which can be considered as a 3D signal in which the frame height and width are two-dimensional coordinate axes, and the pixel values are the dependent values. Also, factors of texture feature and motion property are characteristics of the 3D signal and exist in every frame. The impact of these three factors can be modeled as in Fig.4.

Second, embedding cost is generally a nonnegative number, then, the cost function is a function of the three factors and satisfies $f_{cost} \geq 0$. Furthermore, to define the cost assignment function accurately, we can discuss the relationship between the three factors with embedding cost.

- Motion and texture feature are important characteristics and exist in every frame. Motion feature reflects the relative changes between neighboring frames, and texture feature is inherent for a frame. Both of them can be affected by inter-frame distortion drift and embedding operations.
- Since the cover elements are QDCT coefficients, and they can reflect texture feature of the current block, we assume embedding cost caused by weakening texture feature is higher than that by strengthening. In order to illustrate this, we let $Sum_f$ denote the sum of the coefficients in current frequency zone, and $e$ denote the cover element, then the embedding cost in $e$ with $+1$ and $-1$ is correlated to the sign of $Sum_f$. No matter what the sign of $e$ is, if $Sum_f > 0$, the cost for embedding with $+1$ is lower than that with $-1$, conversely, the cost for embedding with $+1$ is higher than that with $-1$.
- Section 2 illustrates that cost assignment function is proportional to the three factors, and we consider three functions which are monotonically increasing: linear, exponential and logarithmic functions. For a linear function, the sign of the dependent variable equals the sign of independent variable, therefore it does not meet our demand. For logarithmic functions, as the independent variable must be positive, it does not meet our demand either. Thus, we take the exponential function as the cost function and it can be written as

$$f_{cost} = exp(\phi_{Mv} + sign(Sum_f) \cdot \phi_f \cdot \phi_l) \tag{13}$$

where $sign(\cdot)$ is a sign function.

### 3.3 Data Hiding Scheme

In this paper, the message is embedded using STCs which can minimize embedding distortion. The embedding cost of every cover element is assigned according to the cost assignment function in Section 3.2. Since the cover is generated from the QDCT coefficients, and large modification may produce notable distortion, thus the $\pm 1$ embedding is used. Meanwhile, data hiding is implemented GOP by GOP independently using STC.

In [29], it indicated that embedding distortion of the current frame in Eq.(5) include the summation of embedding distortion in the current and distortion drifting from the previous frame. In [14], it illustrated that there are two patterns of embedding with minimal distortion, the payload-limited sender (PLS) and distortion-limited sender (DLS). For PLS, no matter how many bits message includes and distortion are, the message is embedded completely. However, for DLS, the embedding is implemented within the constraint of a given distortion.

In a GOP, to avoid distortion accumulation, distortion in the prior frames should be lower. Generally, for every P-frame, if the message size is less than the capacity of a frame carries, they are embedded into the last P-frame in a GOP. Otherwise, we first embed the message into frames averagely with PLS, then, if the distortion is higher than the target distortion, we embed message again with DLS until the message is embedded into the whole GOP completely. The embedding procedure in a frame is described using pseudocode in Fig. 5.

**Input**: C, m, fnum, Nc, cost

fnum : number of frames
C = ( C(1), C(2), ..., C(fnum) ) : cover
Nc : number of embedded message
m = ( m(1), m(1), ..., m(i), )
cost : cost of every element in C(i)

**Output** : S,$\overline{De}$

S= ( S(1), S (2), ..., S (fnum) ) : stego
$\overline{De}$: embedding distortion

```
for i = 2: fnum-1   % first frame is I-frame
    [De(i), S(i)] = STCsPL ( C(i), m( Nc+1: ( length (m) -Nc) / ( fnum-i+1 )), cost);
    If i==2
      De(i − 1) = η*De(i)/fnum; %targe distortion
    end
      De(i)= De(i − 1)+ η*( De(i) -Dê (i)) / ( fnum-i )
    If  De(i) >De(i)
        [De(i), S(i)] = STCsDL ( C(i), m( Nc: length ( m)), cost);%embed with DLS
    end
end
[De(i) (fnum), S(fnum) ] = STCsPL ( cover ( fnum),m ( Nc+1: length ( m)), cost );
```

Fig. 5. Pseudocode of the embedding with STCs

Assume that message size is large enough, let $\bar{De}$ indicate the distortion caused by embedding all the rest message bits into the rest of frames with PLS averagely, and $\hat{De}(i)$ denote the target distortion in the $i$-th frame. If $\bar{De} > \hat{De}(i)$, the embedding operation in the current frame can be rejected, and the message would be embedded in the current frame with DLS with the limited distortion $\hat{De}(i)$. Otherwise, the embedding operation is accepted. However, for the rest of the frames in the GOP, they must carry the rest of the message bits. we define parameter $\hat{De}(i)$ as follows. $\eta > 0$ is the scaling factor which is used to reflect the distortion increment between two neighboring frames, and it is related to the distortion and capacity of every frame. Its value will be discussed in the next section. Fig. 5 describes the processing of data hiding.

$$\hat{De}(i) = \hat{De}(i-1) + \eta \cdot \frac{\bar{De} - \hat{De}(i-1)}{f_{num} - i}, 0 < \eta \leq \lceil \frac{f_{num}}{2} \rceil \tag{14}$$

After the target distortion is assigned, the message is embedded in the cover using STCs. We denote the cover and marked cover as **C** and **MC**, and message to be embedded in the $i-th$ frame as **m**, then, the embedding operation using STCs is as follows:

$$Emb_{STC}(\mathbf{C}, \mathbf{m}, \hat{De}) = \arg\min_{\mathbf{MC} \in \mathcal{C}(\mathbf{m})} D(\mathbf{C}, \mathbf{MC}) = \tilde{\mathbf{MC}} \tag{15}$$

$$Ext_{STC}(\tilde{\mathbf{MC}}) = \mathcal{H}\tilde{\mathbf{MC}} = \mathbf{m} \tag{16}$$

where $\mathcal{H} \in \{0,1\}^{m \times n}$ is a parity-check matrix of STC which should be shared between receivers and sender,

$\mathcal{C}(m) = \{z \in \{0,1\}^n \mid \mathcal{H}\mathbf{z} = \mathbf{m}\}$ is the coset corresponding to $\mathbf{m}$, $\tilde{\mathbf{MC}}$ is the marked cover in the receiver side. For more information about STCs, please refer to [14] and [13].

## 4 Experimental Results

In this section, we have implemented experiments to demonstrate the performance of the proposed scheme which is mainly about the capacity-distortion performance of marked video.

### 4.1 Experimental Setup

In our experiments, we used H.264 [33] [34], a highly practical compression standard, to provide the compressed video samples. For each compressed video sample, the quantization parameter was set to be 28. In addition to the proposed scheme, we also implemented schemes from Chen et al.'s [21], Zhao et al.'s [9] and Mehdi et al's [5] for comparison. Since scheme in [21] embedded message into I-frames which is different from other schemes, we measured embedding capacity with the message that was embedded in a GOP. Six different video sequences (i.e., foreman, coastguard, hall, akiyo, grandma, silent) in QCIF format were used, and every 10 frames were considered as a GOP with structure $IPPP$, in which only the first frame in a GOP was encoded as I-frame and others were encoded as P-frames.

Generally, the performance in a GOP is similar to the performance of the whole video, and we used three videos with different size to further explain this. The results are presented in Table 2, and Table 3. Table 2 shows the means values of the results, and Table 3 shows their standard deviations.

In Table 2,"O", "brg" and "M" mean performance values for reconstructed frames, bit growth ratio, and performance values of the reconstructed frame with marked data respectively, and all values are calculated by comparing the pixel values between reconstructed/marked frames and original frames. In Table 3, "O-Std" and "M-Std" indicate the standard deviations(variances) of performance values for original and marked video sequences. Meanwhile, the metrics values are average values of that in a GOP, and capacity is message size that was embedded in a GOP. They are the same in other tables.

Both the two tables show that for a video sequence, the performance with video length 10, 20 or 30 is similar, such as peak signal noise ratio (PSNR) [35], structural similarity index (SSIM), mean-square error (MSE) [35] and their standard deviations, and their change rules are similar. Therefore, to reduce the computational efficiency, 10 frames were used for every video sequence in these experiments.

The performance values in Table2 and Table3 were calculated as follows:

$$PSNR(f_k, \hat{f}_k) = 10 \lg \frac{255^2}{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(f_k^{i,j} - \hat{f}_k^{\,i,j})^2} \quad (17)$$

$$MSE(f_k, \hat{f}_k) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(f_k^{i,j} - \hat{f}_k^{\,i,j})^2 \quad (18)$$

$$SSIM(f_k, \hat{f}_k) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} ssim(x_i, y_j) \quad (19)$$

where $m, n$ are size of a frame, $M, N$ are the number of windows in a frame, $(x_i, y_j)$ are the locations of the window, $f_k, \hat{f}_k$ are the $k$-th frame of the video, and function $ssim(x_i, y_j)$ [36] is the SSIM value of a window, and it is described as

$$ssim(x_i, y_j) = \frac{(2\mu_{x_i}\mu_{y_i} + C_1)(2\sigma_{x_i}\sigma_{y_i} + C_2)}{(\mu_{x_i}^2 + \mu_{y_i}^2 + C_1)(\sigma_{x_i}^2 + \sigma_{y_i}^2 + C_2)} \quad (20)$$

where $\mu_{x_i}, \mu_{y_i}$ are mean values of the pixels in a window, $\sigma_{x_i}, \sigma_{y_i}$ are the standard deviations.

### 4.2 Discussion on Scaling Factor $\eta$

In Section 3.3, Eq. (15) is the target distortion for every frame and it is assigned according to distortion in the prior frame, the size of rest message and frames. The scaling factor is used to measure the increment of target distortion and has an impact on the quality of the marked video. In this subsection, we discuss the impact of $\eta$ on video quality. First, we analyze target distortion and rewrite Eq.(15) as

$$\begin{aligned} \hat{De}_i = & \quad \hat{De}_{i-1} + \eta \cdot \frac{\bar{De} - \hat{De}_{i-1}}{f_{num} - i} \\ = & \quad \hat{De}_{i-1} + \frac{\eta}{f_{num} - i} \cdot (\bar{De} - \hat{De}_{i-1}) \end{aligned}$$

where $f_{num}$ indicates the total number of frames in a GOP. If $\frac{\eta}{f_{nym}-i} = 1$, then $\hat{De}_i = \bar{De}$, which means that it could satisfy distortion demand that the rest of message bits are embedded into the rest of frames averagely, and it may happen when the frame number in a GOP is big enough. Meanwhile, $\frac{\eta}{f_{num}-i} = 1$ means $f_{num} = \eta + i$, and greater $\eta$ indicates small $i$. Generally, in a GOP, considering to make inter-frame distortion drift lower, distortion in first frames should be lower than that in subsequent frames, then, we need a big $i$ value. Without loss of generality, we set $0 < \eta < \frac{\lceil f_{num} \rceil}{2}$. In this paper, our experiments applied $\eta = 1, 2, 3, 4$ to select a suitable $\eta$ value. Since the change ratio of marked video performance is invisible for low capacity, we embedded message with 18000 bits in a GOP which is the capacity limitation. The experimental results are listed in Table 4 and Table 5.

Table 4 shows that performance is slightly different with different scaling factors, and it shows the most suitable scaling factor is different for different videos and 3 is a suitable scaling factor for most of the videos. In Table 5, it shows that there is little difference in the standard deviations. Since there are strong temporal correlation and fast broadcasts speed, if the standard deviations are small, the quality of marked video sequences cannot be influenced. Then, we selected 3 as the scaling factor in our experiments.

### 4.3 Visual Quality

PSNR is a metric to measure the video quality degradation of marked video produced in the proposed scheme. High PSNR values indicate imperceptible modification. Fig.6 shows that the visual artifacts between the cover frame and marked frame are invisible to HVS with capacity size 5400 bits in a GOP.

TABLE 2
The embedding performance (mean values) of videos with different frame number

| video | Capacity (bits) | length | PSNR | | SSIM | | MSE | | brg |
|---|---|---|---|---|---|---|---|---|---|
| | | | O | M | O | M | O | M | |
| akiyo | 10800 | 10 | 39.3581 | 37.9844(↓1.3737) | 0.7591 | 0.7184(↓0.0407) | 7.5412 | 10.4019(↑2.8607) | 0.0381 |
| | | 20 | 39.2176 | 37.8783(↓ 1.3393) | 0.7574 | 0.7159(↓ 0.0415) | 7.9159 | 10.6876(↑ 2.7717) | 0.0209 |
| | | 30 | 38.8924 | 37.5189(↓ 1.2735) | 0.7547 | 0.7119(↓ 0.0428) | 10.9839 | 13.8512(↑ 2.8673) | 0.0209 |
| foreman | 10800 | 10 | 37.8684 | 36.4729(↓ 1.3955) | 0.8296 | 0.8033(↓ 0.0263) | 10.6305 | 14.7620(↑ 4.1315) | 0.0251 |
| | | 20 | 37.8396 | 36.2017(↓ 1.6389) | 0.8297 | 0.7991(↓ 0.0306) | 10.7011 | 16.0345(↑ 5.3234) | 0.0142 |
| | | 30 | 37.7987 | 36.2322(↓ 1.5665) | 0.8274 | 0.7989(↓ 0.0385) | 10.8060 | 15.8443(↑ 5.0403) | 0.0101 |
| silent | 10800 | 10 | 37.1734 | 36.1657(↓ 1.0077) | 0.8897 | 0.8666(↓ 0.0231) | 12.4748 | 15.7502(↑ 3.2734) | 0.0104 |
| | | 20 | 37.1546 | 36.1200(↓ 1.0336) | 0.8893 | 0.8664(↓ 0.0229) | 12.5198 | 15.9553(↑ 3.3255) | 0.0101 |
| | | 30 | 37.1565 | 36.0633(↓ 1.0932) | 0.8894 | 0.8660(↓ 0.0234) | 12.5242 | 16.2039(↑ 3.6783) | 0.0089 |

TABLE 3
The embedding performance (standard deviations) of videos with different frame number

| video | (bits) | video length | PSNR | | SSIM | | MSE | |
|---|---|---|---|---|---|---|---|---|
| | | | O-Std | M-Std | O-Std | M-Std | O-Std | M-Std |
| akiyo | 10800 | 10 | 0.1387 | 0.4598(↑ 0.3211) | 0.0064 | 0.0207(↑ 0.0143) | 0.2137 | 1.0152(↑ 0.8015) |
| | | 20 | 0.1536 | 0.4870(↑ 0.3334) | 0.0048 | 0.0205(↑ 0.0157) | 0.2603 | 0.9727(↑ 0.7124) |
| | | 30 | 0.1639 | 0.4740(↑ 0.3101) | 0.0043 | 0.0207(↑ 0.0164) | 0.2994 | 1.0336(↑ 0.7342) |
| foreman | 10800 | 10 | 0.1868 | 0.5156(↑ 0.3287) | 0.0035 | 0.0080(↑ 0.0045) | 0.4265 | 1.5379(↑ 1.1114) |
| | | 20 | 0.1619 | 0.9806(↑ 0.8187) | 0.0045 | 0.0138(↑ 0.0093) | 0.4124 | 4.6506(↑ 4.2382) |
| | | 30 | 0.1747 | 0.8968(↑ 0.7221) | 0.0054 | 0.0141(↑ 0.0087) | 0.4352 | 4.1256(↑ 3.6904) |
| silent | 10800 | 10 | 0.1646 | 0.2515(↑ 0.0869) | 0.0023 | 0.0104(↑ 0.0081) | 0.4988 | 0.9159(↑ 0.4171) |
| | | 20 | 0.1841 | 0.4949(↑ 0.3108) | 0.0020 | 0.0096(↑ 0.0076) | 0.4919 | 1.4727(↑ 0.9808) |
| | | 30 | 0.1734 | 0.4957(↑ 0.3223) | 0.0022 | 0.0103(↑ 0.0081) | 0.5321 | 1.7952(↑ 1.2631) |

TABLE 4
The embedding performance (mean velues) with different scaling factors

| video sequence | Capacity (bits) | scaling factor $\eta$ | PSNR | | SSIM | | MSE | | best selection |
|---|---|---|---|---|---|---|---|---|---|
| | | | O | M | O | M | O | M | |
| suzie | 18000 | 1 | 38.3180 | 36.4106(↓ 1.9074) | 0.7815 | 0.7280(↓ 0.0535) | 9.5904 | 15.2610(↑ 5.6706) | 3 |
| | | 2 | 39.1663 | 37.0171(↓ 2.1492) | 0.7448 | 0.6776(↓ 0.0672) | 7.883 | 13.2271(↑ 5.3441) | |
| | | 3 | 39.1465 | 37.0587(↓ 2.0878) | 0.7466 | 0.6809(↓ 0.0657) | 7.9192 | 12.9983(↑ 5.0811) | |
| | | 4 | 39.1090 | 36.9752(↓ 2.1338) | 0.7434 | 0.6758(↓ 0.0676) | 7.9886 | 13.2572(↑ 5.2706) | |
| foreman | 18000 | 1 | 37.7348 | 35.8124(↓ 1.9214) | 0.8257 | 0.7857(↓ 0.0400) | 10.9773 | 17.5817(↑ 6.6044) | 3 |
| | | 2 | 37.7570 | 34.7496(↓ 2.0074) | 0.8267 | 0.7847(↓ 0.0420) | 10.9124 | 17.6657(↑ 6.7533) | |
| | | 3 | 37.7313 | 35.8164(↓ 1.9149) | 0.8271 | 0.7858(↓ 0.0413) | 10.9753 | 17.2477(↑ 6.2724) | |
| | | 4 | 37.7354 | 35.7686(↓ 1.9668) | 0.8281 | 0.7861(↓ 0.0420) | 10.9618 | 17.4327(↑ 6.4709) | |
| akiyo | 18000 | 1 | 39.1792 | 37.0419(↓ 2.1373) | 0.7501 | 0.6834(↓ 0.0667) | 7.8653 | 13.4583(↑ 5.5930) | 3 |
| | | 2 | 39.1663 | 37.0171(↓ 2.1492) | 0.7448 | 0.6776(↓ 0.672) | 7.8837 | 13.2271(↑ 5.3434) | |
| | | 3 | 39.1465 | 37.0587(↓ 2.0878) | 0.7466 | 0.6809(↓ 0.0657) | 7.9172 | 12.9983(↑ 5.0811) | |
| | | 4 | 39.1090 | 36.9752(↓ 2.1338) | 0.7434 | 0.6758(↓ 0.0676) | 7.9886 | 13.2572(↑ 5.2686) | |
| coastguard | 18000 | 1 | 36.3507 | 34.8066(↓ 1.5441) | 0.9052 | 0.8736(↓ 0.0316) | 15.0738 | 21.8281(↑ 6.7543) | 3 |
| | | 2 | 36.3485 | 34.7905(↓ 1.5580) | 0.9047 | 0.8729(↓ 0.0318) | 15.0807 | 21.7818(↑ 6.7011) | |
| | | 3 | 36.3508 | 34.8374(↓ 1.5134) | 0.9052 | 0.8744(↓ 0.0308) | 15.0717 | 21.4964(↑ 6.4257) | |
| | | 4 | 36.3357 | 34.8046(↓ 1.9668) | 0.9052 | 0.8741(↓ 0.0420) | 15.1243 | 21.6606(↑ 6.4709) | |
| hall | 18000 | 1 | 37.9085 | 36.0309(↓ 1.8776) | 0.8021 | 0.7518(↓ 0.0503) | 10.5422 | 16.7839(↑ 6.2417) | 4 |
| | | 2 | 37.8549 | 36.0342(↓ 1.8207) | 0.7997 | 0.7523(↓ 0.0474) | 10.6726 | 16.5336(↑ 5.8610) | |
| | | 3 | 37.8480 | 36.0194(↓ 1.8286) | 0.8015 | 0.7513(↓ 0.0502) | 10.6874 | 16.5203(↑ 5.8329) | |
| | | 4 | 37.8344 | 36.0205(↓ 1.8139) | 0.8005 | 0.7524(↓ 0.0481) | 10.7207 | 16.4897(↑ 5.7690) | |
| silent | 18000 | 1 | 37.1400 | 35.6280(↓ 1.5120) | 0.8884 | 0.8533(↓ 0.0351) | 12.5697 | 18.1009(↑ 5.5312) | 1 |
| | | 2 | 37.1784 | 35.5758(↓ 1.6026) | 0.8888 | 0.8537(↓ 0.0351) | 12.4612 | 18.2047(↑ 5.7429) | |
| | | 3 | 37.1853 | 35.5959(↓ 1.5894) | 0.8873 | 0.8530(↓ 0.0343) | 12.4414 | 18.0869(↑ 5.6455) | |
| | | 4 | 37.1653 | 35.5851(↓ 1.5792) | 0.8887 | 0.8547(↓ 0.0340) | 12.4981 | 19.0997(↑ 6.6016) | |

Table 6 shows the PSNR, SSIM, and MSE values with embedding 5400 bits, 9000bits and 10800 bits in a GOP, and Table 7 shows their standard deviations. There are several illustrations of the two tables. First, be influenced by the system, there is little difference in performance for the same original reconstructed video with different capacity. Second, for a video sequence, we estimate its performance by comparing metrics of the original video and marked video. When capacity grows the metrics change few, such as embedding capacity increases from 5400 bits to 10800 bits, PSNR value increases from 0.9 to 1.05. Third, the distortion performance values could reflect data hiding performance, and their standard deviations reflect the stabilization of the performance listed in Table 7 which shows the standard deviations are increased after data hiding, and all the standard deviations are small.

In additional, we show maximum embedding capacity(MEC) and its corresponding distortion in Table 8. In this table, cover size, MEC and PSNR indicate the number of embedding locations, maximum embedding capacity and the PSNR values with maximum embedding capacity in a GOP. Generally, the higher PSNR value indicates the lower distortion, and it shows that the minimum PSNR value with acceptable video quality should be between 30dB and 50dB in [37]. The table 8 shows the PSNR values with MEC in our scheme are near 35dB. So, we believe the video quality with the maximum embedding capacity in Table 8 is acceptable.

TABLE 5
The embedding performance (standard deviations) with different scaling factors

| video sequence | Capacity (bits) | scaling factor $\eta$ | PSNR O | PSNR M | SSIM O | SSIM M | MSE O | MSE M |
|---|---|---|---|---|---|---|---|---|
| suzie | 18000 | 1 | 0.1684 | 0.7177(↑ 0.5493) | 0.0111 | 0.0287(↑ 0.0176) | 0.3793 | 2.1939(↑ 1.8146) |
| | | 2 | 0.1725 | 0.6834(↑ 0.5109) | 0.0112 | 0.0286(↑ 0.0174) | 0.4234 | 2.2741(↑ 1.8507) |
| | | 3 | 0.1581 | 0.6693(↑ 0.5112) | 0.0091 | 0.0271(↑ 0180) | 0.3766 | 2.0216(↑ 2.6450) |
| | | 4 | 0.1539 | 0.6370(↑ 0.4831) | 0.0098 | 0.0260(↑ 0.0162) | 0.3701 | 2.0160(↑ 1.6459) |
| foreman | 18000 | 1 | 0.2184 | 0.7856(↑ 0.5672) | 0.0027 | 0.0153(↑ 0.0126) | 0.5107 | 2.8241(↑ 2.3134) |
| | | 2 | 0.2008 | 0.8279(↑ 0.6271) | 0.0024 | 0.0155(↑ 0.0131) | 0.5062 | 2.9585(↑ 2.4523) |
| | | 3 | 0.2311 | 0.8029(↑ 0.5718) | 0.0023 | 0.0147(↑ 0.0124) | 0.5101 | 2.9197(↑ 2.4096) |
| | | 4 | 0.2125 | 0.7528(↑ 0.5403) | 0.0018 | 0.0141(↑ 0.0123) | 0.5086 | 2.6741(↑ 2.1655) |
| akiyo | 18000 | 1 | 0.1558 | 1.0155(↑ 0.8597) | 0.0121 | 0.0360(↑ 0.0239) | 0.1949 | 2.5177(↑ 2.3228) |
| | | 2 | 0.1268 | 0.8646(↑ 0.7378) | 0.0159 | 0.0378(↑ 0.0219) | 0.2464 | 2.3801(↑ 2.1337) |
| | | 3 | 0.2447 | 0.9753(↑ 0.7305) | 0.0138 | 0.0361(↑ 0.0223) | 0.2440 | 2.6700(↑ 2.4260) |
| | | 4 | 0.0920 | 0.8309(↑ 0.7389) | 0.0149 | 0.0348(↑ 0.0199) | 0.2133 | 2.3758(↑ 2.1625) |
| coastguard | 18000 | 1 | 0.1294 | 0.6169(↑ 0.4875) | 0.0030 | 0.0117(↑ 0.0087) | 0.4758 | 2.8027(↑ 2.3269) |
| | | 2 | 0.1307 | 0.6289(↑ 0.4982) | 0.0027 | 0.0116(↑ 0.0089) | 0.4438 | 2.8286(↑ 2.3848) |
| | | 3 | 0.1272 | 0.5272(↑ 0.4000) | 0.0029 | 0.0118(↑ 0.0089) | 0.4758 | 3.0498(↑ 2.5740) |
| | | 4 | 0.1218 | 0.5911(↑ 0.4693) | 0.0029 | 0.0099(↑ 0.0070) | 0.4242 | 2.5807(↑ 2.1565) |
| hall | 18000 | 1 | 0.2326 | 0.9113(↑ 0.6787) | 0.0089 | 0.0308(↑ 0.0219) | 0.6429 | 3.1056(↑ 2.4627) |
| | | 2 | 0.2491 | 0.8691(↑ 0.6200) | 0.0095 | 0.0307(↑ 0.0212) | 0.5714 | 2.8558(↑ 2.2844) |
| | | 3 | 0.2528 | 0.8820(↑ 0.6292) | 0.0089 | 0.0271(↑ 0.0181) | 0.5845 | 2.7251(↑ 2.1406) |
| | | 4 | 0.2362 | 0.8458(↑ 0.6095) | 0.0090 | 0.0292(↑ 0.0202) | 0.5136 | 2.7621(↑ 2.2485) |
| silent | 18000 | 1 | 0.1543 | 0.5451(↑ 0.3908) | 0.0023 | 0.0177(↑ 0.0154) | 0.4835 | 2.9981(↑ 2.5146) |
| | | 2 | 0.1792 | 0.6484(↑ 0.4692) | 0.0028 | 0.0154(↑ 0.0126) | 0.5037 | 1.8816(↑ 1.3779) |
| | | 3 | 0.1729 | 0.4393(↑ 0.2664) | 0.0029 | 0.0165(↑ 0.0136) | 0.4760 | 2.1069(↑ 1.6309) |
| | | 4 | 0.1547 | 0.6226(↑ 0.4679) | 0.0021 | 0.0139(↑ 0.0118) | 0.5086 | 2.6498(↑ 2.1412) |

TABLE 6
The embedding capacity and distortion performance(mean values)

| video sequence | Capacity (bits) | PSNR(dB) O | PSNR(dB) M | SSIM O | SSIM M | MSE O | MSE M | brg |
|---|---|---|---|---|---|---|---|---|
| suzie | 5400 | 38.83 | 37.97(↓ 0.90) | 0.79 | 0.77(↓ 0.02) | 9.03 | 10.39(↑ 1.36) | 0.0204 |
| | 9000 | 38.53 | 37.61(↓ 0.92) | 0.79 | 0.76(↓ 0.03) | 9.14 | 11.29(↑ 2.15) | 0.0334 |
| | 10800 | 38.47 | 37.42(↓ 1.05) | 0.79 | 0.76(↓ 0.03) | 9.26 | 11.81(↑ 2.55) | 0.0472 |
| foreman | 5400 | 37.98 | 37.10(↓ 0.88) | 0.83 | 0.82(↓ 0.01) | 10.36 | 12.73(↑ 2.37) | 0.0218 |
| | 9000 | 37.91 | 36.85(↓ 1.06) | 0.83 | 0.81(↓ 0.02) | 10.54 | 13.51(↑ 2.97) | 0.0402 |
| | 10800 | 37.88 | 36.61(↓ 1.17) | 0.83 | 0.81(↓ 0.02) | 10.61 | 14.31(↑ 3.7) | 0.0520 |
| akiyo | 5400 | 39.42 | 38.84(↓ 0.58) | 0.77 | 0.75(↓ 0.02) | 7.44 | 8.50(↑ 1.06) | 0.0234 |
| | 9000 | 39.40 | 38.43(↓ 0.97) | 0.76 | 0.73(↓ 0.03) | 7.48 | 9.37(↑ 1.89) | 0.0368 |
| | 10800 | 39.38 | 38.21(↓ 1.17) | 0.76 | 0.72(↓ 0.04) | 7.50 | 9.88(↑ 2.38) | 0.0496 |
| coastguard | 5400 | 36.48 | 35.95(↓ 0.53) | 0.91 | 0.90(↓ 0.01) | 14.62 | 16.53(↑ 1.91) | 0.0027 |
| | 9000 | 36.44 | 35.67(↓ 0.77) | 0.91 | 0.89(↓ 0.02) | 14.78 | 17.67(↑ 2.89) | 0.0040 |
| | 10800 | 36.39 | 35.43(↓ 0.96) | 0.91 | 0.89(↓ 0.02) | 14.94 | 18.70(↑ 3.76) | 0.0120 |
| hall | 5400 | 38.09 | 37.44(↓ 0.65) | 0.81 | 0.79(↓ 0.02) | 10.10 | 11.78(↑ 1.68) | 0.0170 |
| | 9000 | 38.04 | 37.04(↓ 1.00) | 0.81 | 0.78(↓ 0.03) | 10.31 | 12.95(↑ 2.64) | 0.0295 |
| | 10800 | 37.98 | 36.84(↓ 1.14) | 0.80 | 0.77(↓ 0.03) | 10.37 | 13.60(↑ 3.23) | 0.0252 |
| silent | 5400 | 37.18 | 36.61(↓ 0.57) | 0.89 | 0.88(↓ 0.01) | 12.47 | 14.26(↑ 1.79) | 0 |
| | 9000 | 37.18 | 36.51(↓ 0.67) | 0.89 | 0.87(↓ 0.02) | 12.46 | 14.53(↑ 2.07) | 0.0194 |
| | 10800 | 37.17 | 36.34(↓ 0.83) | 0.89 | 0.87(↓ 0.02) | 12.49 | 15.15(↑ 2.66) | 0.0276 |

TABLE 7
The embedding capacity and distortion performance(variances)

| video sequence | Capacity (bits) | PSNR(dB) O | PSNR(dB) M | SSIM O | SSIM M | MSE O | MSE M |
|---|---|---|---|---|---|---|---|
| suzie | 5400 | 0.2516 | 0.1456(↓ 0.1060) | 0.0025 | 0.0114(↑ 0.0089) | 0.6109 | 0.3207(↓ 0.2902) |
| | 9000 | 0.2194 | 0.2717(↑ 0.0523) | 0.0034 | 0.0162(↑ 0.0128) | 0.5035 | 0.7445(↑ 0.2410) |
| | 10800 | 0.1994 | 0.3737(↑ 0.1743) | 0.0053 | 0.0197(↑ 0.0143) | 0.4498 | 1.0119(↑ 0.5621) |
| foreman | 5400 | 0.1704 | 0.2597(↑ 0.0893) | 0.0041 | 0.0032(↓ 0.0009) | 0.4113 | 0.8231(↑ 0.4118) |
| | 9000 | 0.1757 | 0.4784(↑ 0.3027) | 0.0043 | 0.0078(↑ 0.0035) | 0.3949 | 1.2931(↑ 0.8982) |
| | 10800 | 0.1868 | 0.5156(↓ 0.3287) | 0.0035 | 0.0080(↑ 0.0045) | 0.4265 | 1.5379(↑ 1.1113) |
| akiyo | 5400 | 0.1628 | 0.2922(↑ 0.1293) | 0.0022 | 0.0102(↑ 0.0080) | 0.3023 | 0.4715(↑ 0.1692) |
| | 9000 | 0.1588 | 0.3413(↑ 0.1825) | 0.0039 | 0.0173(↑ 0.0134) | 0.2652 | 0.8357(↑ 0.5705) |
| | 10800 | 0.1387 | 0.4598(↑ 0.3211) | 0.0064 | 0.0207(↑ 0.0143) | 0.2137 | 1.0152(↑ 0.8015) |
| coastguard | 5400 | 0.1506 | 0.2145(↑ 0.0.0639) | 0.0039 | 0.0019(↓ −0.0020) | 0.5514 | 0.4466(↓ 0.1048) |
| | 9000 | 0.1428 | 0.3351(↑ 0.1923) | 0.0034 | 0.0044(↑ 0.0010) | 0.4913 | 1.1184(↑ 0.6271) |
| | 10800 | 0.1321 | 0.3923(↑ 0.0.2602) | 0.0031 | 0.0053(↑ 0.0022) | 0.4899 | 1.4023(↑ 3.76) |
| hall | 5400 | 0.0603 | 0.3370(↑ 0.2767) | 0.0032 | 0.0109(↑ 0.0077) | 0.1436 | 0.8214(↑ 0.6778) |
| | 9000 | 0.1067 | 0.4881(↑ 0.3814) | 0.0040 | 0.0172(↑ 0.0132) | 0.2814 | 1.3646(↑ 1.0832) |
| | 10800 | 0.1262 | 0.5970(↑ 0.4708) | 0.0066 | 0.0219(↑ 0.0153) | 0.4220 | 1.8126(↑ 1.3906) |
| silent | 5400 | 0.1678 | 0.4177(↑ 0.2499) | 0.0018 | 0.0070(↑ 0.0052) | 0.5102 | 1.8988(↑ 1.3886) |
| | 9000 | 0.1812 | 0.4900(↑ 0.3088) | 0.0025 | 0.0103(↑ 0.0078) | 0.4983 | 1.9193(↑ 1.4210) |
| | 10800 | 0.1646 | 0.2515(↑ 0.0.0869) | 0.0023 | 0.0104(↑ 0.0081) | 0.4988 | 0.9159(↑ 0.4171) |

Fig. 6. Comparison of video original frames with the marked frames with size $144 \times 176$: $a-e$ and $a^{'}-e^{'}$,$b-j$ and $b^{'}-j^{'}$, $k-o$ and $k^{'}-o^{'}$, $p-t$ and $p^{'}-t^{'}$ are the original and marked 2nd, 4th, 5th, 8th, 10th frames in *foreman*, *akiyo*, *coastguard*, *suzie* with embedding 5400 bits in a GOP

TABLE 8
Maximum embedding capacity under acceptable distortion

| | suize | akiyo | foreman | coastguard | hall | grandma |
|---|---|---|---|---|---|---|
| cover size | 18255 | 18376 | 18177 | 19323 | 18457 | 17622 |
| MEC | 24748 | 25198 | 23940 | 26009 | 24743 | 24117 |
| PSNR | 35.4014 | 35.8680 | 35.0530 | 34.1527 | 35.1451 | 35.1850 |

### 4.4 Comparison with Other Schemes

For video data hiding methods, capacity and distortion are important performance factors. In this subsection, video data hiding schemes in [21], [9] and [5] are used as comparison schemes. For scheme in [21], it embedded two secret $(2n+1)$-ary numbers into $2n$ nonzero QDCT coefficients in an MB for I-frame with low distortion. However, there is only one I-frame, and most of the QDCT coefficients are zeros, so embedding capacity is limited. In [9], a 3D RDH was proposed and the message was embedded in P-frame. Carrier was coefficient triples which were selected from AC coefficients of the embeddable $4 \times 4$ blocks. The embeddable blocks were selected randomly when their DC values were greater than the threshold. So, the capacity is limited by the number of embeddable blocks. In [5], when the value of position for LNZ was higher than the given threshold, the message was embedded in the LNZ coefficient for every $4 \times 4$ block. Usually, if there are not enough high-frequency components in a block, the block is not embeddable, then capacity is decided by the high-frequency components of the frame.

For the four schemes, the capacities are different. The capacity in scheme [5] was tunable, but in the other three schemes, it was fixed. For the scheme in [5], if we used the same threshold value, when we implemented the comparison, the capacity was too few to compare the performance with other schemes. Thus, we set it to be 5, and it could reach the same capacity as the scheme in [21]. Now, the capacity declined monotonously with scheme in proposed, [9], [5] and [21]. In our experiments, we embed a different size of the message to compare the performance.

We first embedded message with the size of capacity in [21] into four schemes. The results are shown in Table 9 and Table 10. In Table 9, the performance values are mean values of metrics for test frames, and Table 10 shows their standard deviations. We can get some conclusion from the two tables.

First, for the mean values of metrics, Table 9 shows the capacity is much lower than that in Table 4 and Table 6. Expect video sequence *suzie*, most performances in the proposed scheme are better than that in other schemes. For the much low embedding capacity, there is a tiny difference between different schemes. For scheme in [21], the message was embedded in nonzero QDCT coefficients of I frame, the capacity was limited by the GOP numbers and the video content, and the embedding distortion in I frame could result in the inter-frame distortion drift. However, for other two schemes, the message was hidden into the selected blocks, and the capacity was limited too, especially for the scheme in [5], for video *silent* and *akiyo*, it could embed the same message only when the threshold was set to be 2. Table 4 and Table 6 shows it is different from other schemes that we could embed more message in the proposed scheme with low inter-frame distortion. However, for the limited capacity in other schemes, the message was usually embedded in the first P frames which may cause inter-frame distortion drift. Fig.7 shows the PSNR values decreasing of the four schemes with the capacity in Table 9.

Second, for the standard deviations of metrics, Table10 shows the proposed scheme nearly could keep all the lowest standard deviations.

We can also get some conclusions the experimental results in Fig.7:

- Compared with other schemes, the proposed embedding method did not cause inter-frame distortion because the message in the proposed scheme was embedded in the last frames when there was less message to be embedded.
- Besides video *akiyo* and *silent*, degradation of PSNR values for most videos in the proposed scheme was lower.
- Though we couldn't find the inter-frame distortion directly, for each P frame, both the inter-frame distortion and embedding distortion exists and affects the frame quality.

Table 9 and Table 10 shows the mean values and variances for each metrics respectively. We use T-test, which use T-distribution to compare whether the differences between the two samples are significant, to verify the improvement of our scheme, and the results are listed in Table 11. In the table, "Pro- [9]" means the comparison between the proposed scheme and the scheme in [9], and "Sig.", "CL(%)" are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2932983, IEEE Transactions on Dependable and Secure Computing



TABLE 9
The embedding performance (mean values) with capacity in [21]

| video sequence | Capacity (bits) | scheme | PSNR | | SSIM | | MSE | |
|---|---|---|---|---|---|---|---|---|
| | | | O | M | O | M | O | M |
| suzie | 72 | [21] | 38.6483 | 38.6350(↓ 0.0133) | 0.0.7971 | 0.7970(↓ 0.0001) | 8.8979 | 8.9303(↑ 0.0324) |
| | | [9] | 38.6451 | 38.5523(↓ 0.0928) | 0.7973 | 0.7962(↓ 0.0011) | 8.9046 | 9.0956(↑ 0.1910) |
| | | [5] | 38.6524 | 38.5662(↓ 0.0862) | 0.7968 | 0.7961(↓ 0.0007) | 8.8900 | 9.0662(↑ 0.1762) |
| | | proposed | 38.6483 | 38.6283(↓ 0.0200) | 0.7971 | 0.7969(↓ 0.0002) | 8.8979 | 8.9374(↑ 0.0395) |
| foreman | 159 | [21] | 38.0967 | 38.0510(↓ 0.0457) | 0.8347 | 0.8345(↓ 0.0001) | 10.0898 | 10.2170(↑ 0.1272) |
| | | [9] | 38.0607 | 37.6344(↓ 0.4263) | 0.8335 | 0.8288(↓ 0.0037) | 10.1728 | 11.2179(↑ 1.0451) |
| | | [5] | 38.0539 | 37.4168(↓ 0.6371) | 0.8345 | 0.8311(↓ 0.0034) | 10.1881 | 11.8122(↑ 1.6241) |
| | | proposed | 38.0967 | 38.0490(↓ 0.0477) | 0.8347 | 0.8341(↓ 0.0006) | 10.0898 | 10.2057(↑ 0.1159) |
| akiyo | 112 | [21] | 39.4330 | 39.3762(↓ 0.0568) | 0.7662 | 0.7660(↓ 0.0002) | 7.5282 | 7.4153(↑ 0.1129) |
| | | [9] | 39.4047 | 39.2788(↓ 0.1259) | 0.7651 | 0.7635(↓ 0.0016) | 7.4626 | 7.6830(↑ 0.2204) |
| | | [5] | 39.4043 | 39.2514(↓ 0.1529) | 0.7659 | 0.7652(↓ 0.0007) | 7.4631 | 7.7302(↑ 0.2671) |
| | | proposed | 39.4330 | 39.3994(↓ 0.0336) | 0.7662 | 0.7656(↓ 0.0006) | 7.4153 | 7.4721(↑ 0.0568) |
| coastguard | 212 | [21] | 36.5480 | 36.1318(↓ 0.4161) | 0.9090 | 0.9018(↓ 0.0071) | 14.4091 | 15.8618(↑ 1.4527) |
| | | [9] | 36.5336 | 36.2753(↓ 0.2582) | 0.9084 | 0.9058(↓ 0.0026) | 14.4563 | 15.3348(↑ 0.8786) |
| | | [5] | 36.5496 | 36.2961(↓ 0.2535) | 0.9088 | 0.9064(↓ 0.0024) | 14.4038 | 15.2639(↑ 0.8601) |
| | | proposed | 36.5480 | 36.5248(↓ 0.0231) | 0.9090 | 0.9085(↓ 0.0004) | 14.4091 | 14.4866(↑ 0.0775) |
| hall | 137 | [21] | 38.1905 | 38.0771(↓ 0.1135) | 0.8114 | 0.8096(↓ 0.0018) | 9.8636 | 10.0079(↑ 0.1443) |
| | | [9] | 38.1806 | 37.9853(↓ 0.1753) | 0.8114 | 0.8081(↓ 0.0032) | 9.9319 | 10.3432(↑ 0.4113) |
| | | [5] | 38.1745 | 38.0312(↓ 0.1433) | 0.8124 | 0.8104(↓ 0.0020) | 9.9001 | 10.2334(↑ 0.3333) |
| | | proposed | 38.1905 | 38.1619(↓ 0.0286) | 0.8114 | 0.8109(↓ 0.0005) | 9.8636 | 9.9393(↑ 0.0667) |
| silent | 159 | [21] | 37.2059 | 37.1600(↓ 0.0459) | 0.8934 | 0.8932(↓ 0.0002) | 12.3855 | 112.5370(↑ 0.1515) |
| | | [9] | 37.1808 | 37.0637(↓ 0.1171) | 0.8926 | 0.8909(↓ 0.0017) | 12.4551 | 12.7955(↑ 0.3404) |
| | | [5] | 37.1797 | 36.9128(↓ 0.2669) | 0.8932 | 0.8921(↓ 0.0012) | 12.4585 | 13.2886(↑ 0.8301) |
| | | proposed | 37.2059 | 37.0496(↓ 0.1563) | 0.8932 | 0.8919(↓ 0.0013) | 12.3855 | 12.8995(↑ 0.5140) |

TABLE 10
The embedding performance (variances) with capacity in [21]

| video sequence | Capacity (bits) | scheme | PSNR | | SSIM | | MSE | |
|---|---|---|---|---|---|---|---|---|
| | | | O-Std | M-Std | O-Std | M-Std | O-Std | M-Std |
| suzie | 72 | [21] | 0.3104 | 0.3450(↑ 0.0344) | 0.0040 | 0.0.0042(↑ 0.0002) | 0.6719 | 0,7586(↑ 0.0867) |
| | | [9] | 0.3118 | 0.3037(↓ 0.0081) | 0.0043 | 0.0052(↑ 0.0009) | 0.6742 | 0.6595(↓ 0.0147) |
| | | [5] | 0.3148 | 0.3012(↓ 0.0317) | 0.0039 | 0.0037(↓ 0.0002) | 0.6801 | 0.6533(↓ 0.0268) |
| | | proposed | 0.3104 | 0.2983(↓ 0.0121) | 0.0040 | 0.0038(↓ 0.0002) | 0.6719 | 0.6485(↓ 0.0234) |
| foreman | 159 | [21] | 0.2106 | 0.3494(↑ 0.1387) | 0.0070 | 0.0074(↑ 0.0004) | 0.5163 | 0.9051(↑ 0.3887) |
| | | [9] | 0.1988 | 0.1602(↓ 0.0387) | 0.0066 | 0.0065(↓ 0.0001) | 0.4890 | 0.4123(↓ 0.0766) |
| | | [5] | 0.1940 | 0.2978(↑ 0.1038) | 0.0068 | 0.0054(↓ 0.0014) | 0.4784 | 0.8211(↑ 0.3427) |
| | | proposed | 0.2106 | 0.2502(↑ 0.0396) | 0.0070 | 0.0062(↓ 0.0008) | 0.6105 | 0.5163(↓ 0.0942) |
| akiyo | 112 | [21] | 0.1826 | 0.3361(↑ 0.1535) | 0.0031 | 0.0035(↑ 0.0004) | 0.3183 | 0.6292(↑ 0.0.3109) |
| | | [9] | 0.1629 | 0.1770(↑ 0.0140) | 0.0.0025 | 0.0040(↑ 0.0015) | 0.2860 | 0.3175(↑ 0.0315) |
| | | [5] | 0.1602 | 1551(↓ 0.0051) | 0.0028 | 0.0025(↓ 0.0003) | 0.2817 | 0.2767(↓ 0.0050) |
| | | proposed | 0.1826 | 0.1718(↓ 0.0108) | 0.0031 | 0.0031(↓ 0.0000) | 0.3183 | 0.3004(↓ 0.0.0179) |
| coastguard | 212 | [21] | 0.1814 | 0.2085(↑ 0.0.0271) | 0.0048 | 0.0.0030(↓ 0.0018) | 0.6304 | 0.7765(↑ 0.1461) |
| | | [9] | 0.1768 | 0.1128(↓ 0.0640) | 0.0044 | 0.0043(↓ 0.0001) | 0.6149 | 0.4018(↓ 0.2131) |
| | | [5] | 0.1817 | 0.1368(↓ 0.0449) | 0.0047 | 0.0039(↓ 0.0008) | 0.6313 | 0.4838(↓ 0.1475) |
| | | proposed | 0.1814 | 0.1867(↑ 0.0052) | 0.0048 | 0.0046(↓ 0.0002) | 0.6464 | 0.6304(↓ 0.0160) |
| hall | 137 | [21] | 0.0259 | 0.1855(↑ 0.1596) | 0.0026 | 0.0019(↓ 0.0007) | 0.0588 | 0.4971(↑ 0.4384) |
| | | [9] | 0.0298 | 0.1005(↑ 0.0707) | 0.0028 | 0.0040(↑ 0.0012) | 0.0682 | 0.2398(↑ 0.1716) |
| | | [5] | 0.0212 | 0.0734(↑ 0.0522) | 0.0025 | 0.0030(↑ 0.0015) | 0.0482 | 0.1727(↑ 0.1244) |
| | | proposed | 0.0259 | 0.0840(↑ 0.0581) | 0.0026 | 0.0029(↑ 0.0003) | 0.1960 | 0588(↓ 0.1372) |
| silent | 159 | [21] | 0.2019 | 0.3217(↑ 0.1198) | 0.0.0032 | 0.0.0037(↑ 0.0005) | 0.5860 | 0.9898(↑ 0.4038) |
| | | [9] | 0.1822 | 0.1820(↓ 0.0002) | 0.0.0028 | 0.0041(↓ 0.0012) | 0.5323 | 0.5434(↑ 0.0.0111) |
| | | [5] | 0.1848 | 0.3929(↑ 0.2081) | 0.0031 | 0.0026(↓ 0.0012) | 0.5394 | 0.1.2995(↑ 0.7601) |
| | | proposed | 0.2019 | 0.4706(↑ 0.2687) | 0.0031 | 0.0023(↓ 0.0008) | 0.5860 | 1.5449(↑ 0.9589) |

TABLE 11
T-test results for improvement in proposed scheme with capacity in [21]

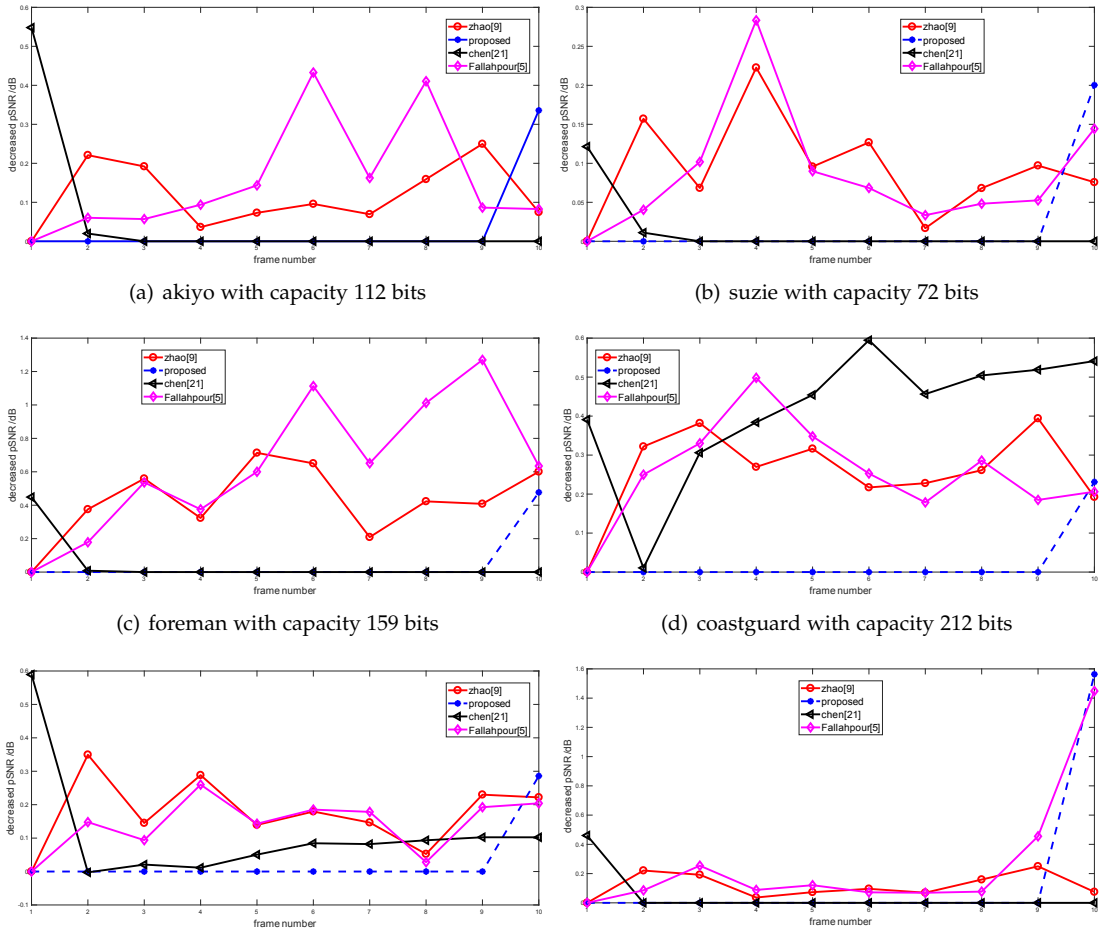
| video | | Leneve test for variances | | | T-test for Mean values | | |
|---|---|---|---|---|---|---|---|
| | | pro[9] | pro[5] | pro[21] | pro[9] | pro[5] | pro[21] |
| silent | sig. | 0.111 | 0.964 | 0.145 | 0.81 | 0.602 | 0.513 |
| | CL(%) | 88.9 | 3.6 | 85.5 | 19 | 39.8 | 48.7 |
| akiyo | sig. | 0.732 | 0.237 | 0.471 | 0.067 | 0.055 | 0.722 |
| | CL(%) | 26.8 | 76.3 | 52.9 | 93.3 | 94.5 | 27.8 |
| suzie | sig. | 0.604 | 0.438 | 0.451 | 0.021 | 0.056 | 0.776 |
| | CL(%) | 39.6 | 56.2 | 54.9 | 97.9 | 94.4 | 22.4 |
| fore- | sig. | 0.184 | 0.026 | 0.921 | 0.000 | 0.001 | 0.976 |
| man | CL(%) | 81.6 | 97.4 | 7.9 | 100 | 99.9 | 2.4 |
| hall | sig. | 0.383 | 0.715 | 0.390 | 0.004 | 0.008 | 0.189 |
| | CL(%) | 61.7 | 28.5 | 61 | 99.6 | 99.2 | 81.1 |
| coast- | sig. | 0.233 | 0.172 | 0.087 | 0.000 | 0.000 | 0.000 |
| guard | CL(%) | 76.7 | 82.8 | 91.3 | 100 | 100 | 100 |

two parameters which indicate significance and confidence level, other tables are similar. While, the smaller value of "Sig." means more significant improvement in proposed scheme, and other tables are similar. For the results in Table 11, compared with the schemes in [9] and [5], the improvement of the proposed scheme is significant. Because for the schemes in [21] and this paper, few messages are embedded in one frame, and one for I-frame, and another for the last frame. For the results in Table 10, we embedded few messages in the last frame preferentially, and the table shows the fluctuation of PSNR differences between original and marked frames is not so significant.

To make the comparison more significance, we embedded more messages in a video sequence. Since the schemes in [21] and [5] couldn't carry more message than the capacity in Table 9, and the capacity limitation of the proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2019.2932983, IEEE Transactions on Dependable and Secure Computing






(a) akiyo with capacity 112 bits

(b) suzie with capacity 72 bits

(c) foreman with capacity 159 bits

(d) coastguard with capacity 212 bits

(e) hall with capacity 137 bits

(f) silent with capacity 159 bits

Fig. 7. The PSNR decreasing comparison with capacity in Table 9

scheme is higher than that in [9], we only embedded message that scheme in [9] could carry in schemes [9] and the proposed scheme. The results are shown in Table 12 and Table 13.

From Table12, we can see that the average quality decreasing with metrics for marked video sequence in the proposed scheme is better than that in the scheme in [9]. And in Table 13, most of the variances for marked video sequence in proposed scheme are higher than that in the scheme in [9] for marked data was embedded with giving priority to the last frames to reduce the inter-frame distortion drift.

The performance degradation is inconspicuous in Table 12, and we also use Fig.8 to show the PSNR degradation. Fig.8 shows that there is PSNR degradation in every P-frame, and that means the message was distributed in all the P frames for the scheme in [9], and distortion existed in every P-frame. Because each frame could carry different size message, it is difficult to find inter-frame distortion drift between neighbor frames for different capacity in each P frame. For the proposed scheme, because cover element was selected based on minimizing inter-frame distortion drift, and posterior frames had priority to carry the message, so, PSNR values in first P frames keep consistent with the original decoded frames, and the last two or three frames carrying message make the more PSNR degradation than that in [9]. We can say it is an effective method to prevent inter-frame distortion drift, and it also was the expression of content adaptive based data hiding scheme.

TABLE 14
T-test results for improvement in proposed scheme with capacity in [9]

| video sequence | Leneve test for variances (confidence level:%) | T-test for mean values (confidence level:%) |
|---|---|---|
| silent | 0.306(69.4) | 0.644(35.6) |
| akiyo | 0.002(99.8) | 0.375(62.5) |
| suzie | 0.002(99.8) | 0.466(53.4) |
| foreman | 0.054(94.6) | 0.128(87.2) |
| hall | 0.030(97) | 0.122(87.8) |
| coastguard | 0.050(95) | 0.069(93.1) |

Table 12 and Table 13 show that the performance difference between different schemes are more significant, and obviously, the PSNR degradation in proposed scheme is lower than that in [9], but for the variances it is contrary. Even so, we couldn't declare the proposed scheme improve the performance essentially from the video samples with 10 frames. we use T-test to calculate the significance between them, and Table 14 shows the results. The table shows that for both variance and mean values it is more significant for the high embedding capacity. comparing values in Table 14 with that in Table 11, we can declare that more message embedding makes the improvement significance, and this is also the advantage of the proposed scheme.

TABLE 12
The embedding performance (mean values) with capacity in [9]

| video sequence | Capacity (bits) | scheme | PSNR | | SSIM | | MSE | |
|---|---|---|---|---|---|---|---|---|
| | | | O | M | O | M | O | M |
| suzie | 1513 | [9] | 38.6215 | 38.2728(↓ 0.3487) | 0.7945 | 0.7840(↓ 0.0106) | 8.9516 | 9.6892(↑ 0.7376) |
| | | proposed | 38.6483 | 38.6283(↓ 0.0200) | 0.7971 | 0.7969(↓ 0.0002) | 8.8979 | 8.9374(↑ 0.0395) |
| foreman | 1420 | [9] | 38.0340 | 37.4988(↓ 0.5352) | 0.8330 | 0.8242(↓ 0.0088) | 10.2341 | 11.5694(↑ 1.3353) |
| | | proposed | 38.0852 | 37.8391(↓ 0.2461) | 0.8343 | 0.8303(↓ 0.0040) | 10.1167 | 10.7776(↑ 0.6609) |
| akiyo | 1513 | [9] | 39.4093 | 39.0208(↓ 0.3885) | 0.7643 | 0.7532(↓ 0.0111) | 7.4547 | 8.1515(↑ 0.6967) |
| | | proposed | 39.4353 | 39.1772(↓ 0.2581) | 0.7662 | 0.7602(↓ 0.0060) | 7.4116 | 7.8792(↑ 0.4676) |
| coastguard | 1180 | [9] | 36.5044 | 36.1337(↓ 0.3707) | 0.9082 | 0.9029(↓ 0.0053) | 14.5520 | 15.8408(↑ 1.2888) |
| | | proposed | 38.1823 | 38.0297(↓ 0.1526) | 0.8109 | 0.8069(↓ 0.0040) | 9.8822 | 10.2624(↑ 0.3801) |
| hall | 1270 | [9] | 38.1167 | 37.7534(↓ 0.3633) | 0.8090 | 0.7994(↓ 0.0096) | 10.0326 | 10.9141(↑ 0.8813) |
| | | proposed | 38.1844 | 38.0169(↓ 0.1675) | 0.8109 | 0.8061(↓ 0.0048) | 9.8776 | 10.2977(↑ 0.4201) |
| silent | 1250 | [9] | 37.2019 | 36.8189(↓ 0.3830) | 0.8928 | 0.8863(↓ 0.0065) | 12.3964 | 13.5804(↑ 1.1840) |
| | | proposed | 37.2029 | 36.9331(↓ 0.2698) | 0.8932 | 0.8891(↓ 0.0041) | 12.3939 | 13.2875(↑ 0.8936) |

TABLE 13
The embedding performance (variances) with capacity in [9]

| video sequence | Capacity (bits) | scheme | PSNR | | SSIM | | MSE | |
|---|---|---|---|---|---|---|---|---|
| | | | O-Std | M-Std | O-Std | M-Std | O-Std | M-Std |
| suzie | 1513 | [9] | 0.3001 | 0.2162(↓ 0.0839) | 0027 | 0.0031(↑ 0.0004) | 0.6515 | 0.4857(↓ 0.1658) |
| | | proposed | 0.3047 | 0.3741(↑ 0.0694) | 0.0036 | 0.0132(↑ 0.0095) | 0.6613 | 0.8310(↑ 0.1697) |
| foreman | 1420 | [9] | 0.1858 | 0.1032(↓ 0.0826) | 0.0058 | 0.0043(↓ 0.0015) | 0.4598 | 0.2751(↓ 0.1847) |
| | | proposed | 0.2112 | 0.5535(↑ 0.3423) | 0.0065 | 0.0068(↑ 0.0003) | 0.5170 | 1.4887(↑ 0.9717) |
| akiyo | 1513 | [9] | 0.1641 | 0.1522(↓ 0.0119) | 0.0022 | 0.0043(↓ 0.0021) | 0.2880 | 0.2850(↓ 0.0.0030) |
| | | proposed | 0.1854 | 0.3272(↑ 0.1418) | 0.0030 | 0.0089(↑ 0.0059) | 0.3227 | 0.6020(↑ 0.2792) |
| coastguard | 1180 | [9] | 0.1620 | 0.0803(↓ 0.0817) | 0.0043 | 0.0034(↓ 0.0009) | 0.5672 | 0.2925(↓ 0.2747) |
| | | proposed | 0.0287 | 0.3256(↑ 0.2969) | 0.0030 | 0.0108(↑ 0.0079) | 0.0654 | 0.8105(↑ 0.7451) |
| hall | 1270 | [9] | 0.0384 | 0.1559(↑ 0.1175) | 0.0029 | 0.0056(↑ 0.0027) | 0.0888 | 0.3808(↑ 0.2920) |
| | | proposed | 0.0258 | 0.3539(↑ 0.3281) | 0.0030 | 0.0124(↑ 0.0095) | 0.0585 | 0.8889(↑ 0.0.8304) |
| silent | 1250 | [9] | 0.1953 | 0.3971(↑ 0.2018) | 0.0028 | 0.0031(↑ 0.0003) | 5681 | 1.3411(↑ 0.7730) |
| | | proposed | 0.1990 | 5728(↑ 0.3738) | 0.0029 | 0.0081(↑ 0.0051) | 0.5782 | 1.9711(↑ 0.1.3929) |

## 5 Conclusion

Adaptive data hiding scheme is to embed the message into frames based on minimizing embedding distortion and maximizing embedding capacity according to message size and video content. In this paper, we proposed an adaptive video data hiding scheme by cost assignment and STCs, which could minimize embedding distortion. The embedding cost was calculated by the proposed cost function in which the factors which had an impact on the embedding distortion were considered, such as texture feature, motion property, frame position and so on. Texture feature and motion property may have an impact on the frame residual, and inter-frame distortion drift is reflected by the frame position. Based on these characteristics, the message was embedded adaptively. In order to get better performance, STCs was used to embed the message. Experimental results show that the combination of proposed cost assignment function and STCs could improve the capacity-distortion performance, and we also used T-test to verify the improvement. In our future works, we think that applying the proposed scheme in high definition videos would be an interesting direction.

## Acknowledgments

This work is supported in part by National Key Research and Development Program of China (Grant No. 2016YFB0800600), the National Natural Science Foundation of China (NSFC) under grant No. U1536110, the National Science Foundation under Grant 1748494, the Ohio Federal Research Network, and the Fundamental Research Funds for the Central Universities under the grant No.YJ201881.

## References

[1] Jarno Mielikainen. Lsb matching revisited. *IEEE Signal Process. Lett.*, 13(5):285–287, 2006.

[2] Nan-I Wu and Min-Shiang Hwang. A novel lsb data hiding scheme with the lowest distortion. *The Imaging Science Journal*, 65(6):371–378, 2017.

[3] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, Dec 1997.

[4] H. Z. Wu, Y. Q. Shi, H. X. Wang, and L. N. Zhou. Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(8):1620–1631, Aug 2017.

[5] Mehdi Fallahpour, Shervin Shirmohammadi, and Mohammed Ghanbari. A high capacity data hiding algorithm for H.264/AVC video. *Security and Communication Networks*, 8:2947–2955, 03 2015.

[6] Yanli Chen, Hongxia Wang, Hanzhou Wu, and Xingming Sun. A video error concealment method using data hiding based on compressed sensing over lossy channel. *Telecommunication Systems*, 68(2):337–349, Jun 2018.

[7] H. Z. Wu, H. X. Wang, and Y. Q. Shi. Dynamic content selection-and-prediction framework applied to reversible data hiding. In *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Dec 2016.

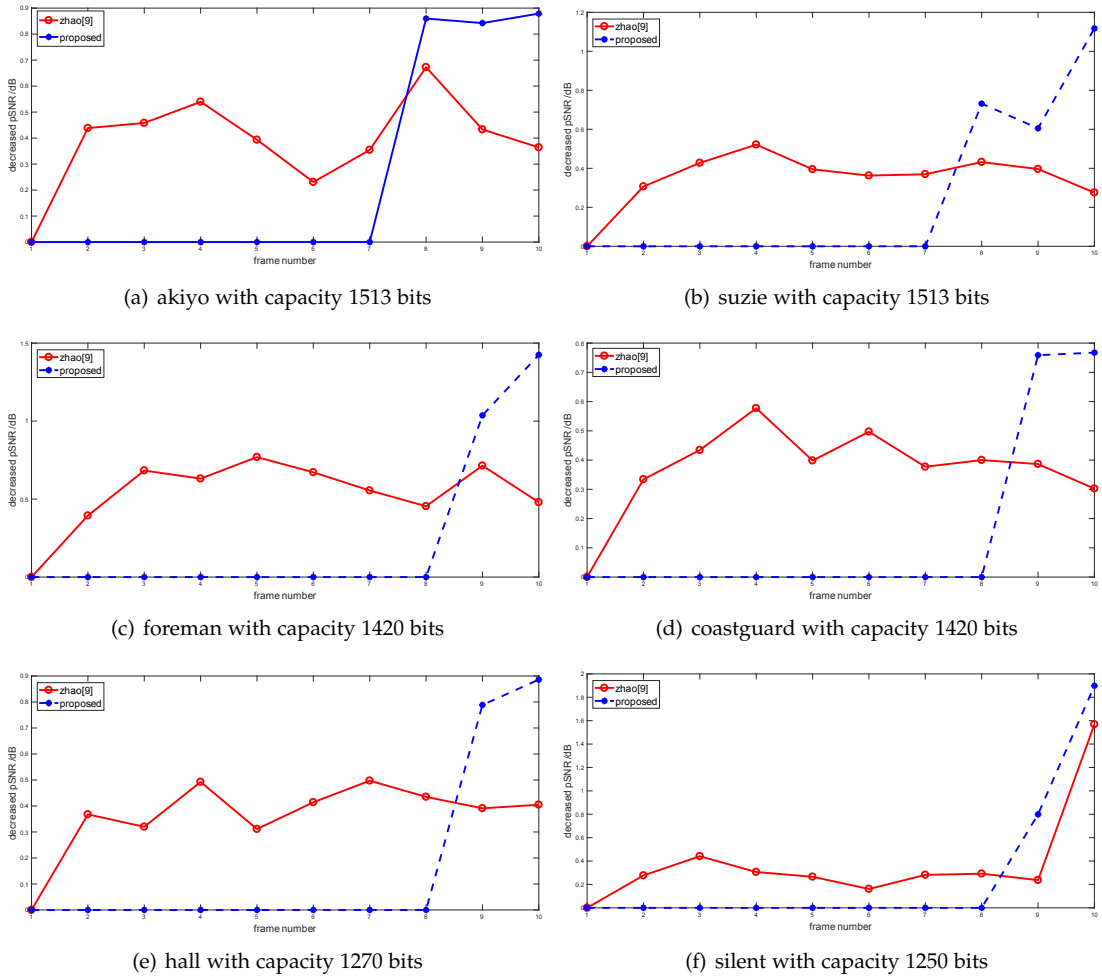
[8] Han-Zhou Wu, Hong-Xia Wang, and Yun-Qing Shi. Ppe-based reversible data hiding. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '16, pages 187–188, New York, NY, USA, 2016. ACM.

[9] Juan Zhao and Zhitang Li. Three-dimensional histogram shifting for reversible data hiding. *Multimedia Systems*, 24(1):95–109, Feb 2018.

[10] Z. Qian, H. Zhou, X. Zhang, , and W. Zhang. Separable reversible data hiding in encrypted jpeg bitstreams. *IEEE Transactions on Dependable and Secure Computing*, 15(6):1055–1067, 2018.

[11] Z. Qian, H. Xu, X. Luo, , and X. Zhang. New framework of reversible data hiding in encrypted jpeg bitstreams. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[12] Z. Qian, X. Zhang, , and S. Wang. Reversible data hiding in encrypted jpeg bitstream. *IEEE Transactions on Multimedia*, 16(5):1486–1491, 2014.




(a) akiyo with capacity 1513 bits

(b) suzie with capacity 1513 bits

(c) foreman with capacity 1420 bits

(d) coastguard with capacity 1420 bits

(e) hall with capacity 1270 bits

(f) silent with capacity 1250 bits

Fig. 8. The PSNR decreasing comparison with capacity in Table 12


[13] Filler Tomas, Judas Jan, and Fridrich Jessica. Minimizing embedding impact in steganography using trellis-coded quantization. In *Proc.SPIE*, volume 7541, pages 7541 – 7541 – 14, 2010.

[14] Tomas Filler, Jan Judas, and Jessica Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. 6:920 – 935, 10 2011.

[15] Hanzhou Wu. Minimizing embedding distortion with weighted bigraph matching in reversible data hiding. *CoRR*, abs/1712.06240, 2017.

[16] H. Zhang, Y. Cao, and X. Zhao. A steganalytic approach to detect motion vector modification using near-perfect estimation for local optimality. *IEEE Transactions on Information Forensics and Security*, 12(2):465–478, Feb 2017.

[17] Y. Cao, H. Zhang, X. Zhao, and H. Yu. Covert communication by compressed videos exploiting the uncertainty of motion estimation. *IEEE Communications Letters*, 19(2):203–206, Feb 2015.

[18] K. Niu, X. Yang, and Y. Zhang. A novel video reversible data hiding algorithm using motion vector for H.264/AVC. *Tsinghua Science and Technology*, 22(5):489–498, September 2017.

[19] Dawen Xu, Rangding Wang, and Jicheng Wang. Prediction mode modulated data-hiding algorithm for H.264/AVC. *Journal of Real-Time Image Processing*, 7(4):205–214, Dec 2012.

[20] Chia-Hsiung Liu and O. T. C. Chen. Data hiding in inter and intra prediction modes of h.264/avc. In *2008 IEEE International Symposium on Circuits and Systems*, pages 3025–3028, May 2008.

[21] Yi Chen, Hongxia Wang, Hanzhou Wu, and Yong Liu. An adaptive data hiding algorithm with low bitrate growth for H.264/AVC video stream. *Multimedia Tools and Applications*, Dec 2017.

[22] 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 Transactions on Circuits and Systems for Video Technology*, 20(10):1320–1330, Oct 2010.

[23] Sung Min Kim, Sang Beom Kim, Youpyo Hong, and Chee Sun Won. Data hiding on H.264/AVC compressed video. In Mohamed Kamel and Aurélio Campilho, editors, *Image Analysis and Recognition*, pages 698–707, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[24] Tian-Qi Wang, Hong-Xia Wang, and Yue Li. Reversible data hiding with low bit-rate growth in H.264/AVC compressed video by adaptive hybrid coding. In Xingming Sun, Alex Liu, Han-Chieh Chao, and Elisa Bertino, editors, *Cloud Computing and Security*, pages 48–62, Cham, 2016. Springer International Publishing.

[25] Z. Wang, Z. Qian, X. Zhang, M.Yang, and D. Ye. On improving distortion functions for JPEG steganography. *IEEE Access*, 6(1):74917–74930, 2018.

[26] W. Zhang, Z. Zhang, L. Zhang, H. Li, and N. Yu. Decomposing joint distortion for adaptive steganography. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(10):2274–2280, Oct 2017.

[27] W. Zhou, W. Zhang, and N. Yu. A new rule for cost reassignment in adaptive steganography. *IEEE Transactions on Information Forensics and Security*, 12(11):2654–2667, Nov 2017.

[28] Chang Wang, Jiangqun Ni, and Chuntao Wang. New distortion metric for efficient JPEG steganography using linear prediction. *J. Signal Process. Syst.*, 81(3):389–400, December 2015.

[29] Yuanzhi Yao, Weiming Zhang, and Nenghai Yu. Inter-frame distortion drift analysis for reversible data hiding in encrypted H.264/AVC video bitstreams. *Signal Processing*, 128:531 – 545, 2016.

[30] M. Jeni and S. Srinivasan. Reversible data hiding in videos using low distortion transform. In *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, pages 121–124, Feb 2013.

[31] Christina A. Burbeck and D. H. Kelly. Spatiotemporal characteristics of visual mechanisms: excitatory-inhibitory model. *J. Opt. Soc. Am.*, 70(9):1121–1126, Sep 1980.

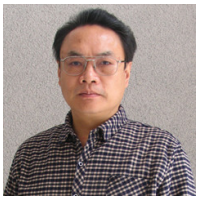
[32] Christian J.van den Branden Lambrecht and Murat Kunt. Characterization of human visual sensitivity for video imaging applications. *Signal Processing*, 67(3):255 – 269, 1998.
[33] Abdullah Muhit, Mark Pickering, Michael R. Frater, and John Arnold. Video coding using elastic motion model and larger blocks. 20:661 – 672, 06 2010.
[34] Abdullah A. Muhit, Mark R. Pickering, Michael R. Frater, and John F. Arnold. Video coding using fast geometry-adaptive partitioning and an elastic motion model. *Journal of Visual Communication and Image Representation*, 23(1):31 – 41, 2012.
[35] Huimin Zhao and Jinchang Ren. Cognitive computation of compressed sensing for watermark signal measurement. *Cognitive Computation*, 8(2):246–260, Apr 2016.
[36] M. C. Hsu, G. L. Wu, and S. Y. Chien. Combination of ssim and jnd with content-transition classification for image quality assessment. In *2012 Visual Communications and Image Processing*, pages 1–6, Nov 2012.
[37] Ozdemir Cetin and A. Turan Ozcerit. A new steganography algorithm based on color histograms for data embedding into raw video streams. *Computers & Security*, 28(7):670–682, 2009.

**Yanli Chen** received B.S. and Ph.D. degree from the Southwest Jiaotong University, Chengdu, China, in 2008 and 2019, respectively. Currently, she is an assistant professor in guizhou Normal University, GuiYang, China. Her research interests include data hiding, information hiding and digital forensics.

**Hongxia Wang** received the B.S. degree from Hebei Normal University, Shijiazhuang, in 1996, and the M.S. and Ph.D. degrees from University of Electronic Science and Technology of China, Chengdu, in 1999 and 2002, respectively. She engaged in postdoctoral research work in Shanghai Jiao Tong University from 2002 to 2004, and she was a professor in Southwest Jiaotong University from 2004 to 2018. From 2013 to 2014, she was a visiting scholar at Computer Science Department, Northern Kentucky University, USA. Currently she is a professor with College of Cybersecurity, Sichuan University, Chengdu. Her research interests include multimedia information security, information hiding, digital watermarking and intelligent information processing. She has published over 100 peer research papers and holds 12 authorized patents.

**Hanzhou Wu** received B.S. and Ph.D. degree from the Southwest Jiaotong University, Chengdu, China, in 2011 and 2017, respectively. He was a visiting scholar in New Jersey Institute of Technology, Newark, USA from 2014 to 2016. He was a research scientist in Institute of Automation, Chinese Academy of Sciences, Beijing, China from 2017 to 2019. Currently, He is an assistant professor in Shanghai University, Shanghai, China. His research interests include watermarking, steganography, graph theory and unsupervised learning. He has published around 20 papers in peer journals and conferences such as IEEE TCSVT, IEEE SPL, ACM, IH and MMSec Workshop, IEEE WIFS, IS and T Electronic Imaging, and so on.

**Zhiqiang Wu** received the B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 1993, the M.S. degree from Peking University, Beijing, China, in 1996, and the Ph.D. degree from Colorado State University, Fort Collins, CO, USA, in 2002, all in electrical engineering. From 2003 to 2005, he was an Assistant Professor with the Department of Electrical Engineering, West Virginia University Institute of Technology. In 2005, he joined the Department of Electrical Engineering, Wright State University, Dayton, OH, USA, where he is currently a Full Professor. Dr. Wu has also held visiting positions at Tibet University, Peking University, Harbin Engineering University and Guizhou Normal University. His research has been supported by NSF, AFRL, AFOSR, NASA, NRL and OFRN.



**Tao Li** received the B.S. and M.S. degrees in computer science and the Ph.D. degree in circuit and system from University of Electronic Science and Technology of China, Chengdu, China, in 1986, 1991,and 1995, respectively. From 1994 to 1995, he was a visiting scholar in University of California at Berkeley for neural networks theory.He is currently a professor with College of Cybersecurity, Sichuan Univrsity, Chengdu. His current research interests include network security, artificial immune theory and disaster recovery application.

**Asad Malik** is currently working as a Ph. D. Scholar at the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan, China since 2015. He has received his Master degree in Computer Application from the Department of Computer Science, Jamia Millia Islamia University, New Delhi, India in the year 2015. His research interests includes in the area of information security, reversible data hiding and image processing. asad@my.swjtu.edu.cn