Inter-frame distortion drift analysis for reversible data hiding in encrypted H.264/AVC video bitstreams

Yuanzhi Yao, Weiming Zhang, Nenghai Yu *

Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China

A R T I C L E   I N F O

Article history:
Received 12 December 2015
Received in revised form 1 May 2016
Accepted 2 May 2016
Available online 3 May 2016

Keywords:
Reversible data hiding
Video encryption
Inter-frame distortion drift
Coefficient selection
Histogram shifting

A B S T R A C T

Due to the privacy-preserving requirement for cloud data management, it is necessary to perform data hiding in the encrypted domain. This paper proposes an effective scheme for reversible data hiding in encrypted H.264/AVC video bitstreams. In the encryption phase, the codewords of intra prediction modes, motion vector differences and partial residual coefficients are encrypted without video bit rate increment for preserving the confidentiality of video content. In the data hiding phase, we first present a theoretical analysis of the picture distortion caused by data embedding and the subsequent inter-frame distortion drift. Based on the analysis, we estimate the embedding distortions caused by modifying different residual coefficients and embed data into residual coefficients with different priorities for decreasing the inter-frame distortion drift. The data embedding is implemented by the histogram shifting technique. If the receiver decrypts the encrypted video bitstream and extracts the embedded data, the original video bitstream can be perfectly recovered. In addition, if the receiver decrypts the encrypted video bitstream without extracting the embedded data, the bitstream can still be decoded to obtain the reconstructed video with good quality.

1. Introduction

In recent years, the rapid development of cloud computing and network technology brings about novel storage ways for digital videos. More and more users prefer to store their compressed video data in cloud servers for saving local storage resources and distributing video data efficiently. Given that cloud services are vulnerable to malicious attacks and untrustworthy administrators, it is necessary to store video bitstreams in the encrypted domain for avoiding privacy leakage. However, it is difficult for cloud servers to manage encrypted video bitstreams. One feasible solution is to embed additional authentication data into the encrypted video bitstream for access control and media annotation. Therefore, the joint scheme for video encryption and video data hiding has attracted considerable research interest.

Differing from traditional text/binary data encryption [1], video encryption usually requires that the encryption scheme should have low computing complexity and the encrypted bitstream should be format-compliant. Therefore, it is not practical to encrypt video data completely with traditional ciphers due to high computational cost. Alternatively, we can selectively encrypt a fraction of video data to improve efficiency and keep format compliant. Since H.264/AVC is one of the most widely used video coding standards [2], some schemes have been proposed to encrypt H.264/AVC videos. The schemes proposed in [3,4] encrypt videos by scrambling the intra prediction modes (IPM) of intra-coded macroblocks. The encryption is performed by using bitwise XOR (exclusive-OR) operation on the IPMs. An improved scheme in [5] encrypts not only the intra prediction modes but also the motion vector differences (MVD). Another scheme in [6] encrypts the intra prediction modes, the motion vector differences and the residual coefficients. In addition, the scheme in [7] proposes to encrypt other coding parameters such as the sequence parameter set and the picture parameter set. In the view of perturbing macroblock prediction, the intra prediction modes, the motion vector differences and the residual coefficients are key coding parameters which need to be encrypted. By scrambling these coding parameters, the decoded reconstructed video will be severely distorted so that the confidentiality of video content can be preserved.

Reversible data hiding (RDH) is a technique which embeds secret messages into a cover in the reversible manner. Reversibility means that the original cover can be losslessly recovered after the embedded messages are extracted. Reversible data hiding can be utilized for media annotation and integrity authentication. Recently, some effective methods about reversible data hiding in encrypted images have been proposed [8–12]. These methods
cannot be applied to reversible data hiding in encrypted videos directly. The major problem for video data hiding is how to decrease the distortion drift caused by data embedding, while this problem does not need to be discussed in reversible image data hiding. The distortion drift can severely degrade the reconstructed video quality. There exist two types of distortion drifts in video data hiding, which are the intra-frame distortion drift and the inter-frame distortion drift. To avert the intra-frame distortion drift, Ma et al. [13] proposed to use pairs of quantized DCT coefficients for data embedding so that the distortion in current cover block will not propagate to its neighboring blocks. A compensation method proposed in [14] eliminates the intra-frame distortion drift by obtaining the difference between the original and modified reconstructed quantized DCT coefficients and compensating for the quantized DCT coefficients. Two reversible video data hiding methods which discuss data embedding in inter-coded frames can be found in [15,16]. Gujjunoori et al. [15] proposed to embed two bits into the zero quantized DCT coefficients in the fixed position. Bouchama et al. [16] brought modifications to Gujjunoori et al.’s method [15] for improving the embedding capacity and the video quality. Bouchama et al.’s method [16] can embed three bits into a set of quantized DCT coefficients according to the predefined mapping rule. However, these methods [15,16] do not consider decreasing the inter-frame distortion drift. To the best of our knowledge, almost no encrypted video data hiding method focuses on decreasing the inter-frame distortion drift in the open literature. Besides, the inter-frame distortion drift is more influential than the intra-frame distortion drift because the inter prediction is usually more frequently used than the intra prediction in video coding. Therefore, how to decrease the inter-frame distortion drift is a key and unresolved problem in encrypted video data hiding. Because functionalities of video encryption and video data hiding are different, the joint scheme for video encryption and video data hiding should combine these two functionalities together to protect the confidentiality and authenticate the identification. Therefore, the joint scheme should support data embedding/extraction operation in the encrypted domain and keep the video bitstream format-compliant. For example, data can be embedded into the encrypted video bitstream without knowing the encryption key, or data can be extracted from the encrypted video bitstream without complete video decoding process. Lian et al. [17] proposed to encrypt the intra prediction modes, the signs of motion vector differences and the quantized DCT coefficients using stream ciphers and embed data in DCT coefficients. The quantization embedding is used in this scheme so that data embedding cannot be performed directly on the quantized DCT coefficients. In the scheme proposed by Park et al. [18], the intra prediction modes, the signs of motion vector differences and the signs of DCT coefficients are encrypted and data are embedded in intra prediction modes. However, the stego video bitstream is not fully format-compliant. Recently, Xu et al. [19] presented the scheme for data hiding in encrypted H.264/AVC video bitstreams by analyzing the codewords of context-adaptive variable length coding (CAVLC). In this scheme, video encryption and data embedding are performed using the codeword substitution technique. The major advantage of this scheme is that the video bitstream size is strictly preserved even after video encryption and data embedding.

Although the schemes [17–19] achieve video encryption and video data hiding together, their common drawback is that reversible data hiding is not applied. If the receiver decrypts the encrypted video and extracts the embedded data, he/she can only obtain the reconstructed video with degraded quality. Xu et al. proposed a scheme for reversible data hiding in the encrypted video bitstream [20] using the histogram shifting technique [21]. Differing from the encryption scheme in [19], the scheme in [20] encrypts the signs of all non-zero quantized DCT coefficients. However, if the sign of one non-zero quantized DCT coefficient is encrypted and its original suffixLength is 0, its codeword structure may change. Therefore, the scheme in [20] may cause the video bit rate increment during the encryption process. Moreover, in some application scenarios, the receiver may download the video from the cloud server and view the decrypted reconstructed video using the encryption key. The decrypted video still contains the embedded authentication data, which is convenient for the cloud server to manage the video. At this time, the distortion drift caused by data embedding will degrade the reconstructed video quality. The scheme in [20] does not explore this problem like other existing schemes [15,16]. Hence, new reversible video data hiding schemes which are capable of decreasing the inter-frame distortion drift should be sought.

We propose an effective scheme for reversible data hiding in encrypted H.264/AVC video bitstreams in this paper. We mainly concentrate on decreasing the inter-frame distortion drift caused by data embedding. In the encryption phase, three types of key coding parameters including the intra prediction modes, the motion vector differences and the quantized DCT coefficients are selectively encrypted using stream ciphers without video bit rate increment. In the data hiding phase, to explore the fundamental principle of the video quality degradation caused by data embedding, we present a theoretical analysis of the inter-frame distortion drift, which is the central element of our proposed scheme. Based on the analysis, we estimate the embedding distortions caused by modifying different residual coefficients and embed data into residual coefficients with different priorities for decreasing the inter-frame distortion drift. The data embedding is implemented by the histogram shifting technique. In conclusion, the highlights of this paper can be summarized as follows.

- Video encryption can keep the bitstream format-compliant and the video bit rate constant.
- A theoretical analysis of the inter-frame distortion drift is presented.
- The proposed scheme can embed data in the video bitstream with low distortion based on the inter-frame distortion drift analysis.
- The video bitstream can be losslessly recovered after video decryption and data extraction.

The rest of the paper is organized as follows. In Section 2, we present the scheme for reversible data hiding in encrypted H.264/AVC video bitstreams. The experimental results are shown in Section 3. Finally, Section 4 concludes the paper.

2. Proposed scheme for reversible data hiding in encrypted H.264/AVC video bitstreams

In this section, the scheme for reversible data hiding in encrypted H.264/AVC video bitstreams is elaborated. The framework of the proposed scheme is shown in Fig. 1. The scheme is composed of three components: video encryption, data embedding in encrypted video, and data extraction and video recovery. In the encryption phase, the content owner encrypts the original H.264/AVC video bitstream using stream ciphers to obtain the encrypted video bitstream. In the data hiding phase, the data hidee (e.g., a database manager or a cloud server) can embed some additional authentication data into the encrypted video bitstream without

1 The suffixLength is a syntax element for encoding the suffix length of the codeword of Level.
Knowing the content of original video. At the receiver end, the content owner himself or an authorized third party can extract the embedded data either in the encrypted domain or in the decrypted domain. If the receiver decrypts the encrypted video bitstream and extracts the embedded data, the original video bitstream can be perfectly recovered, as depicted in Scenario 1 in Fig. 1. In addition, if the receiver decrypts the encrypted video bitstream without extracting the embedded data, the bitstream can still be decoded to obtain the reconstructed video with good quality, as depicted in Scenario 2 in Fig. 1.

2.1. H.264/AVC video encryption

Selective encryption is one of the most practical techniques for video data. Because it can decrease computing complexity and have adequate perceptual scrambling effect by encrypting some key coding parameters. Besides, selective encryption can keep the bitstream format-compliant to allow the video decoder to identify bitstream structure without decryption. In [20], it is suggested that the intra prediction modes (IPM), the signs of motion vector differences and the signs of quantized DCT coefficients are encrypted using the standard stream cipher RC4. However, as pointed out in [22], cryptographic algorithms get weaker over time, including RC4. Therefore, to encrypt these coding parameters, we use the novel stream cipher Spritz [22] in the proposed scheme. Spritz attempts to remedy the weakness in RC4 and remain its general design principles. Differing from [20], the signs of quantized DCT coefficients are selectively encrypted to keep video bit rate constant after encryption.

2.1.1. Intra prediction mode (IPM) encryption

In H.264/AVC, four types of intra prediction modes are supported, including Intra_4 × 4, Intra_16 × 16, Intra_chroma and L_PCM [2]. Because Intra_4 × 4 and Intra_16 × 16 are frequently used in intra prediction coding, we selectively encrypt these two modes to scramble the intra prediction of blocks. When utilizing the Intra_16 × 16 mode, the whole luma component of a macroblock is predicted. There are four different prediction modes for an Intra_16 × 16 block [23]. The last bit of each codeword for Intra_16 × 16 mode is encrypted by applying the bitwise XOR operation with a pseudo-random bit to keep the codeword length constant [4]. Pseudo-random bits are generated by an encryption key $E_{Key}$ using the stream cipher Spritz. When utilizing the Intra_4 × 4 mode, each 4 × 4 block is predicted from its spatially neighboring samples. There are nine different prediction modes for an Intra_4 × 4 block, ranging from mode 0 to mode 8 [23]. The predictive coding technique is applied for efficiently compressing the prediction mode. If the prediction mode Mode of the currently considered block is equal to the most probable mode $MPM$, the codeword for Intra_4 × 4 mode should keep unchanged. Otherwise, three bits of fixed-length code in each codeword for Intra_4 × 4 mode are encrypted by applying the bitwise XOR operation with a pseudo-random sequence [6]. The pseudo-random sequence is generated by an encryption key $E_{Key}$ using the stream cipher Spritz. According to H.264/AVC standard [23], the length of the encrypted codeword for intra prediction mode can be the same as the original one.

2.1.2. Motion vector difference (MVD) encryption

In order to scramble the inter prediction of blocks, the motion vectors should also be encrypted. To improve the compression efficiency, motion vector prediction is further performed on the motion vector to obtain the motion vector difference (MVD). In H.264/AVC baseline profile, Exp-Golomb entropy coding is used to encode MVDs [23]. The structure of Exp-Golomb codeword is $|Mzeros|1||INFO$, in which $|Mzeros|$ are M leading bits and $|INFO|$ is an $N$-bit data carrying information. The codeword length is $(2M + 1)$. The last bit of each codeword is encrypted by applying the bitwise XOR operation with a pseudo-random bit. Pseudo-random bits are generated by an encryption key $E_{Key}$ using the stream cipher Spritz. When the value of MVD is equal to 0, its corresponding codeword “1” should keep unchanged during the encryption process. Encrypting the last bit of each codeword may change the sign of MVD, but does not affect the codeword length [17].

2.1.3. Residual coefficient encryption

To further protect the confidentiality of video content, the residual coefficients of each frame should also be encrypted. In H.264/AVC baseline profile, the context-adaptive variable length coding (CAVLC) is used to encode the quantized DCT coefficients of each residual block [23]. To decrease computing complexity and keep format compliant, not all syntax elements need to be encrypted. Therefore, the signs of quantized DCT coefficients are scrambled by encrypting corresponding codewords of $trailing\_ones\_sign\_flag$ and $Level$.

The one-bit codeword of $trailing\_ones\_sign\_flag$ is encrypted by applying the bitwise XOR operation using the stream cipher Spritz, which is determined by an encryption key $E_{Key}$. The codeword of each $Level$ is made up of a prefix level_prefix and a suffix...
level_suffix as [level_prefix][level_suffix]. The last bit of each codeword is encrypted by applying the bitwise XOR operation with a pseudo-random bit. Pseudo-random bits are generated by an encryption key E_key using the stream cipher Spritz. It should be noted that when suffixLength is equal to 0, the corresponding codeword of Level should not be encrypted to keep the codeword length constant [19].

2.2. Data embedding in encrypted video

The data hider can embed some additional authentication data into the encrypted video bitstream without knowing the content of original video for access control and media annotation. Reversible data hiding can be used to recover the video bitstream losslessly after video decryption and data extraction. When the receiver views the decrypted reconstructed video without extracting the embedded data, the distortion drift caused by data embedding will degrade the reconstructed video quality. Therefore, we should endeavor to decrease the inter-frame distortion drift caused by data embedding.

2.2.1. Inter-frame distortion drift analysis

We first present a theoretical analysis of the picture distortion caused by data embedding and the subsequent inter-frame distortion drift. The theoretical analysis is inspired by the source and channel rate-distortion analysis in [24]. To analyze the inter-frame distortion drift and its influence on the visual quality of the reconstructed video, we consider the video in the decrypted domain. Besides, we only focus on the distortion drift of P-frames, because B-frames do not cause distortion drift. Let \( F(n, i) \) be the original luma value of pixel \( i \) in the \( n \)th video frame and \( \hat{F}(n, i) \) be the corresponding reconstructed luma value of pixel \( i \) in the feedback loop at the encoder. We denote the reconstructed luma value of pixel \( i \) at the receiver end as \( \tilde{F}(n, i) \). \( F(n, i) \) may be different from \( F(n, i) \) due to data embedding. For inter-coded macroblocks, let \( r(n, i) \) be the motion compensation residue at the encoder. Let \( \tilde{F}(n, i) \) and \( \hat{F}(n, i) \) be the corresponding reconstructed values with embedded data and without embedded data respectively. Because the positions and the number of embedded message bits can be random, \( F(n, i) \) and \( \hat{F}(n, i) \) are actually random variables.

Therefore, we can model and analyze the overall distortion \( D(n) \) of the \( n \)th video frame at the receiver end which is given by

\[
D(n) = E\{[\tilde{F}(n, i) - F(n, i)]^2\}
\]

where \( E\{x(n, i)\} \) is denoted as the average (over all pixels) expected value of the random variable \( x(n, i) \). The source distortion \( D_s(n) \) caused by lossy compression and the embedding distortion \( D_e(n) \) caused by embedding data in quantized DCT coefficients of the \( n \)th video frame are respectively given by

\[
D_s(n) = E\{[\hat{F}(n, i) - F(n, i)]^2\}
\]

\[
D_e(n) = E\{[\hat{F}(n, i) - \tilde{F}(n, i)]^2\}
\]

In this paper, we assume that \( D_s(n) \) and \( D_e(n) \) are uncorrelated with each other. That is to say

\[
D(n) = D_s(n) + D_e(n)
\]

To justify this assumption, we encode four QCIF video sequences\(^7\) [25] which are Carphone, Foreman, Hall and Salesman at frame rate 30 fps with the H.264/AVC reference software JM 10.2 [26]. The GOP (Group of Picture) structure is IPPP. The data embedding starts in the first P-frame in the GOP and then proceeds to subsequent P-frames. We use the histogram shifting technique proposed in [21] to embed message bits in the quantized luma DCT coefficients of each P-frame. We denote \( T_p \) and \( T_o \) as the two highest bins in the histogram of quantized DCT coefficients. (Zero coefficients are not used for data embedding.) Bellinieme et al. [27] had pointed out that the 2D DCT coefficients of the differential signal tend to be less correlated if the motion estimation is used. Thus, the distribution of DCT coefficients can be well modeled with the Laplacian probability distribution [28]. This proposition reveals that \( T_p = 1 \) and \( T_o = -1 \) respectively. When we choose bin \( T_p \) and bin \( T_o \) as the actual data embedding zone, the embedding capacity is the sum of the number of bin \( T_p \) and the number of bin \( T_o \). The embedding rate \( \gamma \) can be defined as the ratio of the number of actual embedded message bits \( m \) to the embedding capacity. In Fig. 2, we plot \( D(n) \) and \( D_s(n) + D_e(n) \) for each frame of QCIF video sequence Foreman at embedding rate 10% using QP 28. It can be seen that \( D(n) \) is approximately equal to \( D_s(n) + D_e(n) \). The average relative difference \( e_D \) between \( D(n) \) and \( D_s(n) + D_e(n) \) is defined by

\[
e_D = \frac{1}{N} \sum_{n=1}^{N} \frac{|D_s(n) + D_e(n) - D(n)|}{D(n)} \times 100\%
\]

where \( N \) is the total number of video frames. We perform similar tests on other video sequences and embedding rates. The average relative difference for each test is listed in Table 1. It can be seen that \( e_D \) is very small. This implies that it is quite reasonable to assume that the source distortion \( D_s(n) \) and the embedding distortion \( D_e(n) \) are uncorrelated with each other and that \( D(n) = D_s(n) + D_e(n) \). Given the original video bitstream, \( D_s(n) \) is constant. Therefore, to decrease the overall distortion \( D(n) \) and improve the decoded reconstructed video quality, the only thing left is to estimate \( D_e(n) \) and endeavor to decrease it during data embedding.

We suppose that the GOP structure is IPPP, in which only the first frame is encoded as an I-frame and the remaining frames are encoded as P-frames. For a pixel in the inter-coded macroblock of the \( n \)th frame \((n \geq 2)\) at the receiver end, in case of no embedded data, its reconstructed value is \( \tilde{F}(n - 1, j) + \hat{F}(n, i) \) where pixel \( j \) is
the motion prediction of pixel $i$. (If the sub-pixel motion estimation is used, $j$ could point to a sub-pixel position.) If the macro-block contains embedded data, its reconstructed value is $\hat{F}(n-1, j) + \hat{r}(n, i)$. Therefore, the expected embedding distortion is

$$D_r(n) = E\{\hat{F}(n, i) - \hat{r}(n, i)^2\} = (1-p)E\{|\hat{F}(n-1, j) + \hat{r}(n, i) - \hat{F}(n, i)^2\} + pE\{|\hat{F}(n-1, j) + \hat{r}(n, i) - \hat{F}(n, i)^2\}$$

where $p$ is the probability of embedding data in the $n$th frame. It should be noted that the fifth identity in Eq. (6) is based on the assumption that $E\{|\hat{F}(n-1, j) - \hat{F}(n-1, j)|^2\} = 0$. The proof of the assumption in Eq. (6) is presented in Appendix A. If we assume that

$$E\{|\hat{F}(n-1, j) - \hat{F}(n-1, j)|^2\} = \alpha D_r(n-1)$$

where $\alpha$ is a constant relating to the video content, we can rewrite Eq. (6) as

$$D_r(n) = (1-p)\alpha + p\alpha D_r(n-1) + pE\{|\hat{F}(n, i) - \hat{r}(n, i)^2\} + pE\{|\hat{F}(n, i) - \hat{r}(n, i)^2\}$$

(8)

It can be seen from Eq. (8) that the embedding distortion $D_r(n)$ of the $n$th frame is the sum of $\alpha D_r(n-1)$ and $pE\{|\hat{F}(n, i) - \hat{r}(n, i)^2\}$. We can obtain two propositions from Eq. (8). One proposition is that we should endeavor to decrease $E\{|\hat{F}(n, i) - \hat{r}(n, i)^2\}$ for decreasing the embedding distortion in the $n$th frame. Another proposition is that the embedding distortion of each frame can accumulate as the frame index increases. This is defined as the inter-frame distortion drift.

### 2.2.2. Inter-frame distortion drift prevention

Based on the inter-frame distortion drift analysis, we investigate the discrete cosine transform in H.264/AVC standard for decreasing the inter-frame distortion drift. In H.264/AVC, integer $4 \times 4$ transform can be computed exactly in arithmetic and can avoid mismatch problem in inverse transform [23]. The $4 \times 4$ transform is depicted in Eq. (9).

$$Y = C_j R_{C_j} U$$

(9)

where

$$C_j = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}$$

In Eq. (9), $Y$ is the matrix of original unscaled luma DCT coefficients corresponding to the residual $4 \times 4$ block $R$. Then, the post-scaling and quantization step is given in Eq. (10).

$$Z = \text{round}(Y \times PF \times Qstep)$$

(10)

where $\times$ and $\div$ are denoted as the element-wise product operator of matrices and the element-wise division operator of matrices respectively. In Eq. (10), $Z = (z_{ij})_{4 \times 4}$ is the quantized DCT coefficients in the $4 \times 4$ block and $Qstep$ is the quantization step size which is determined by $Q$. ($Q$ is the quantization parameter.) $PF$ is the quantization matrix which is given by

$$PF = \begin{pmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{pmatrix} \ 

a = \frac{1}{2}, \ b = \sqrt{\frac{2}{5}}$$

Then, the quantized DCT coefficients will undergo the entropy encoding.

At the feedback loop at the encoder, after inverse quantization step in Eq. (11), the reconstructed residual block $\hat{R} = (\hat{r}_{ij})_{4 \times 4}$ without embedded data can be obtained in Eq. (12).

$$\hat{Y} = Z \times Qstep \times PF \times 64$$

(11)

$$\hat{R} = \text{round}([C_j^t \hat{Y}C_j]_{4 \times 4}) = \text{round}([C_j^t (Z \times Qstep \times PF) C_j]_{4 \times 4})$$

(12)

where

$$C_j = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1/2 & -1/2 & -1 \\ 1/2 & -1 & 1 & 1/2 \\ 1/2 & -1 & 1 & 1/2 \end{pmatrix}$$

After data embedding, the message bits are embedded in the quantized DCT coefficients as depicted in Eq. (13), where $\Delta = (\delta_{ij})_{4 \times 4}$ is the error matrix added to the quantized DCT coefficients in the $4 \times 4$ block as follows.

$$Z' = Z + \Delta$$

(13)

After inverse quantization step in Eq. (14), the reconstructed residual block $\hat{R} = (\hat{r}_{ij})_{4 \times 4}$ with embedded data can be obtained in Eq. (15).

$$\hat{Y} = Z' \times Qstep \times PF \times 64$$

$$= Z' \times Qstep \times PF \times 64 + \Delta \times Qstep \times PF \times 64$$

(14)

$$\hat{R} = \text{round}([C_j^t (\hat{Y}C_j)]_{4 \times 4})$$

$$= \text{round}([C_j^t (Z' \times Qstep \times PF) C_j]_{4 \times 4})$$

(15)

The error matrix $E$ of the pixel luma value between the reconstructed residual block $\hat{R}$ with embedded data and the reconstructed residual block $\hat{R}$ without embedded data can be calculated as depicted in Eq. (16), where $E = (e_{ij})_{4 \times 4}$.

<table>
<thead>
<tr>
<th>Video sequence</th>
<th>Quantization parameter</th>
<th>Embedding rate (%)</th>
<th>Average relative difference $e_d$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carphone</td>
<td>28</td>
<td>10</td>
<td>0.0240</td>
</tr>
<tr>
<td>Carphone</td>
<td>24</td>
<td>20</td>
<td>0.0491</td>
</tr>
<tr>
<td>Foreman</td>
<td>28</td>
<td>10</td>
<td>0.0319</td>
</tr>
<tr>
<td>Foreman</td>
<td>24</td>
<td>20</td>
<td>0.0595</td>
</tr>
<tr>
<td>Hall</td>
<td>28</td>
<td>10</td>
<td>0.0224</td>
</tr>
<tr>
<td>Hall</td>
<td>24</td>
<td>20</td>
<td>0.0016</td>
</tr>
<tr>
<td>Salesman</td>
<td>28</td>
<td>10</td>
<td>0.0326</td>
</tr>
<tr>
<td>Salesman</td>
<td>24</td>
<td>20</td>
<td>0.0253</td>
</tr>
</tbody>
</table>

**Table 1**

Average relative difference $e_d$ between $D_r(n)$ and $D_r(n) + D_e(n)$.
\[ E = \hat{R} - \tilde{R} \]
\[ = \text{round}(C_i^T(Z \times \text{Qstep} \times \text{PF})C_i + C_i^T(\Delta \times \text{Qstep} \times \text{PF})C_i) \]
\[ - \text{round}(C_i^T(Z \times \text{Qstep} \times \text{PF})C_i) \]  

(16)

Let \( z_{ij} \) be the matrix of quantized DCT coefficients in the \( 4 \times 4 \) block obtained from the matrix \( Z \) whose \( \delta_i \) has been added into the \((i, j)\)th coefficient \( z_{ij} \). Without considering the rounding operator on coefficients, we can estimate the independent embedding distortion \( \rho_i(Z, z_{ij}Z_{-,ij}) \) caused by modifying the \((i, j)\)th coefficient in the \( 4 \times 4 \) block using the MSE (Mean Squared Error) measure in Eq. (17).

\[
\rho_i(Z, z_{ij}Z_{-,ij}) = \frac{1}{4 \times 4} \sum_{i=1}^{4} \sum_{j=1}^{4} (\hat{r}_{ij} - \tilde{r}_{ij})^2 = \begin{cases} 
\frac{25}{1024} \text{Qstep}^2 b_i^2 \delta_i^2 & \text{if } (i, j) \in C_1 \\
\frac{5}{32} \text{Qstep}^2 a_i b_i ^2 \delta_i^2 & \text{if } (i, j) \in C_2 \\
\text{Qstep}^2 a_i^2 \delta_i^2 & \text{if } (i, j) \in C_3 
\end{cases} 
\]  

(17)

where

\[
C_1 = \{(2, 2), (4, 2), (2, 4), (4, 4)\} \\
C_2 = \{(1, 2), (2, 1), (1, 4), (2, 3), (3, 2), (4, 1), (3, 4), (4, 3)\} \\
C_3 = \{(1, 1), (1, 3), (1, 3), (3, 3)\}
\]

It can be known that \( \frac{25}{1024} \text{Qstep}^2 b_i^2 \delta_i^2 < \frac{5}{32} \text{Qstep}^2 a_i b_i ^2 \delta_i^2 < \text{Qstep}^2 a_i^2 \delta_i^2 \) by sorting independent embedding distortions. Therefore, we can classify coefficients based on their corresponding independent embedding distortions. In Eq. (17), \( C_1 \), \( C_2 \) and \( C_3 \) represent the coefficient classification sets. According to the linear property of a \( 4 \times 4 \) transform, it is easy to verify that the embedding distortion \( D_e(\hat{R}, \tilde{R}) \) in the \( 4 \times 4 \) block can be captured by the sum of the independent embedding distortion \( \rho_i(Z, z_{ij}Z_{-,ij}) \) in the form

\[
D_e(\hat{R}, \tilde{R}) = E\{ (\hat{r}_{ij} - \tilde{r}_{ij})^2 \} = \frac{4 \times 4}{4 \times 4} \sum_{i=1}^{4} \sum_{j=1}^{4} \rho_i(Z, z_{ij}Z_{-,ij}) 
\]  

(18)

From Eqs. (17) and (18) we can infer that it is necessary to embed data using coefficient selection, which means that data should be embedded into the coefficients in \( C_1 \) with highest priority, the coefficients in \( C_2 \) with medium priority and the coefficients in \( C_3 \) with lowest priority. The coefficient classification illustration in the \( 4 \times 4 \) block is given in Fig. 3. The coefficient selection aims to decrease the embedding distortion in the current frame.

To justify the proposition, we encode QCIF video sequence Foreman [25] at frame rate 30 fps using H.264/AVC reference software JM 10.2 [26]. The first 100 frames are encoded and the GOP structure is IPPP. The data embedding starts in the first P-frame in the GOP and then proceeds to subsequent P-frames. We use the histogram shifting technique proposed in [21] to embed message bits in the quantized luma DCT coefficients of each P-frame. We choose \( T_p = 1 \) and \( T_n = -1 \) as the actual data embedding zone, where \( T_p \) and \( T_n \) are the two highest bins in the coefficient histogram. (Zero coefficients are not used for data embedding.) The embedding capacity \( C \) is the sum of the number of bin \( T_p \) and the number of bin \( T_n \) as depicted in Eq. (19).

\[
C = \text{num}(T_p) + \text{num}(T_n) 
\]  

(19)

During data embedding process, the proposed inter-frame distortion drift prevention scheme should be used. We suppose that the total frame number of the video is \( N \) and only the first frame is encoded as an I-frame. The GOP number is 1 and its structure is IPPP. We denote the to-be-embedded data as a binary message sequence \( b = (b_1, b_2, ..., b_m) \) with length \( m \). The message sequence can be firstly encrypted by applying the bitwise XOR operation with a pseudo-random sequence to ensure security. The pseudo-random sequence is generated by a data hiding key using the stream cipher Spritz. The data embedding algorithm is described as follows.

- **Case 1:** If \( z_{ij} > T_p \) or \( z_{ij} < T_n \), shift \( z_{ij} \) by 1 unit to the right or the left respectively.
  \[
  z_{ij} = \begin{cases} 
z_{ij} + 1 & \text{if } z_{ij} > T_p \\
z_{ij} - 1 & \text{if } z_{ij} < T_n 
\end{cases} 
\]  

(20)

where \( z_{ij} \) is the \((i, j)\)th original encrypted quantized DCT coefficient in the \( 4 \times 4 \) block and \( z'_{ij} \) is the corresponding stego coefficient.

- **Case 2:** If \( z_{ij} = T_p \) or \( z_{ij} = T_n \), modify \( z_{ij} \) according to the to-be-embedded message bit.
2.3. Data extraction and video recovery

In the proposed scheme, the embedded data can be extracted either in the encrypted domain or in the decrypted domain. The order of video decryption and data extraction implies two different application scenarios.

2.3.1. Data extraction in the encrypted domain

In this application scenario, in order to protect the confidentiality of video content, the cloud server does not have sufficient authority to access video content due to absence of the encryption key. Therefore, it is necessary for the cloud server to extract the embedded authentication data in the encrypted domain to manage the video.

Before data extraction in the encrypted domain, partial decoding is performed on the encrypted video bitstream to obtain the encrypted quantized luma DCT coefficients. The embedded data can be extracted from two zones, which are $Z_p = \{ T_p, T_p + 1 \}$ and $Z_n = \{ T_n - 1, T_n \}$. The data extraction is the inverse process of data embedding. We can extract the embedded message sequence $b = (b_1, b_2, \ldots, b_m)$ as depicted in Eq. (23).

$$
\begin{aligned}
    b_k = \begin{cases} 
        0 & \text{if } z_{ij} \in Z_p \text{ and } \mod(z_{ij} - T_p, 2) = 0 \\
        1 & \text{if } z_{ij} \in Z_p \text{ and } \mod(z_{ij} - T_p, 2) = 1 \\
        0 & \text{if } z_{ij} \in Z_n \text{ and } \mod(T_n - z_{ij}, 2) = 0 \\
        1 & \text{if } z_{ij} \in Z_n \text{ and } \mod(T_n - z_{ij}, 2) = 1 
    \end{cases}
\end{aligned}
$$

where $z_{ij}$ is the $(ij)$th stego quantized DCT coefficient in the $4 \times 4$ block. The original encrypted quantized DCT coefficient recovery is described as follows.

- **Case 1**: If $z_{ij} \in \{ T_p, T_p + 1 \}$ or $z_{ij} \in \{ T_n - 1, T_n \}$, the original encrypted quantized DCT coefficient can be losslessly recovered by

$$
\begin{aligned}
    z_{ij} = \begin{cases} 
        z_{ij} & \text{if } z_{ij} \in Z_p \text{ and } \mod(z_{ij} - T_p, 2) = 0 \\
        z_{ij} - 1 & \text{if } z_{ij} \in Z_p \text{ and } \mod(z_{ij} - T_p, 2) = 1 \\
        z_{ij} & \text{if } z_{ij} \in Z_n \text{ and } \mod(T_n - z_{ij}, 2) = 0 \\
        z_{ij} + 1 & \text{if } z_{ij} \in Z_n \text{ and } \mod(T_n - z_{ij}, 2) = 1 
    \end{cases}
\end{aligned}
$$

- **Case 2**: If $z_{ij} > T_p + 1$ or $z_{ij} < T_n - 1$, shift $z_{ij}$ by 1 unit to the center.

$$
\begin{aligned}
    z_{ij} = \begin{cases} 
        z_{ij} - 1 & \text{if } z_{ij} > T_p + 1 \\
        z_{ij} + 1 & \text{if } z_{ij} < T_n - 1 
    \end{cases}
\end{aligned}
$$

- **Case 3**: If $T_n < z_{ij} < T_p$, $z_{ij}$ is equal to $z_{ij}$.

$$
\begin{aligned}
    z_{ij} = z_{ij}
\end{aligned}
$$

Since the data embedding process is reversible, the content owner can decrypt the encrypted video to obtain the original video with the encryption key.

2.3.2. Data extraction in the decrypted domain

In this application scenario, the receiver may download the video from the cloud server and view the decrypted reconstructed video using the encryption key. The decrypted video still contains the embedded authentication data, which is convenient for the cloud server to manage the video. If the content owner does not want to upload the video, he/she can extract the embedded data after video decryption. Our proposed inter-frame distortion drift prevention scheme can improve the quality of the reconstructed video containing embedded data. The entire process is composed of the following steps.

- **Step 1**: Generate the marked decrypted video with the encryption key. Due to the symmetry of the bitwise XOR operation, the decryption operation is symmetric to the encryption operation.

- **Step 2**: Extract the embedded data and recover the original video losslessly. Because the video encryption scheme only changes the signs of quantized DCT coefficients, this step can be performed similarly using Eqs. (23)–(26).
Since the data embedding process is reversible, the original video can be obtained with the encryption key.

3. Experimental results

The proposed scheme for reversible data hiding in encrypted H.264/AVC video bitstreams has been integrated into the H.264/AVC reference software JM 10.2 [26]. Besides, the schemes proposed by Gujjunoori et al. [15], Bouchama et al. [16] and Xu et al. [20] are implemented for performance comparison. The histogram shifting technique [21] is used in Xu et al.’s scheme and our proposed scheme for data embedding. As shown in Fig. 5, eight video sequences (i.e., Carphone, Coastguard, Container, Foreman, Hall, Harbor, Mobile, and Salesman) in QCIF format [25] are used in the experiments. The first 100 frames of each video sequence are encoded as an I-frame and the remaining frames are encoded as P-frames.

3.1. Security of the encryption scheme

For the video encryption scheme, the security includes the cryptographic security and the perception security.

Cryptographic security: Cryptographic security means that the encryption scheme can resist cryptographic attacks, which depends on the adopted ciphers. In the encryption scheme, the novel stream cipher Spritz [22], which is an improved variant of the stream cipher RC4, is used to encrypt the key coding parameters and the authentication data. It is estimated that Spritz can produce output with about 24 cycles/byte of computation. Furthermore, it is suggested that about $2^{11}$ bytes of output are needed before reasonably distinguishing Spritz output from random sequence. This is a significant improvement over RC4 [22].

Perception security: Perception security means that the encryption scheme can keep the encrypted video unintelligible. The intra prediction modes (IPM), the motion vector differences (MVD) and the residual coefficients are selectively encrypted in the proposed scheme. In our experiments, QP is set as 28 to generate the video bitstream. The scrambling effect of encrypted video frames is given in Fig. 6. The luma PSNR values of encrypted video frames compared with the original video frames are 11.27 dB, 9.51 dB, 10.79 dB, 10.17 dB, 9.66 dB, 9.54 dB, 9.61 dB and 13.08 dB respectively. Compared with the original video frames in Fig. 5, it is difficult to recognize the contents of encrypted frames. Due to the space limitation, we do not list the encrypted results of all frames. Other frames in each given video sequence have the similar scrambling effect. The scrambling effect can be more obvious if the video content contains intense motion and complex texture. For example, the contents of video sequences Harbor and Mobile are totally distorted. However, there exist a large number of P-Skip mode coded macroblocks in P-frames for encoding video sequences with slight motion. For this coding mode, neither motion vectors nor residual coefficients are transmitted. It means that encrypting motion vectors and residual coefficients has no scrambling effect on the P-Skip macroblocks. For video sequences with slight motion, the perception security of encrypted frames mainly depends on the scrambling effect of corresponding I-frames. Therefore, the scrambling effect of video sequences Hall and Salesman is relatively weaker.

3.2. Visual quality of the stego video

In our experiments, we use commonly adopted measurements PSNR and SSIM [29] to evaluate the objective visual quality of stego videos. The PSNR (dB) and the SSIM are calculated by comparing the decoded reconstructed video sequence with the original video sequence. In the proposed scheme, the embedding capacity $C$ is the sum of the number of bin 1 and the number of bin $-1$, which are the two highest bins in the histogram of quantized DCT coefficients. (Zero coefficients are not used for data embedding.) In Xu et al.’s scheme [20], the scale factor $\beta$ is used for choosing the actual data embedding zone and adjusting the embedding capacity. We set $\beta$ as 0 to guarantee that Xu et al.’s scheme and our proposed scheme have the same embedding capacity at the same coding setting for fair comparison. The embedding rate $\gamma$ and “can be defined as the ratio of the number of actual embedded message bits $m$ to the embedding capacity $C$.

To evaluate the visual quality of the stego videos, we first decrypt the encrypted videos containing embedded data using the encryption key. The comparison of average luma PSNR (dB) and average luma SSIM between Xu et al.’s scheme and our proposed scheme at embedding rate 50% using different QP values is listed in Table 2, in which the embedding capacity of each video is given. The visual quality of cover videos without embedded data is also

![Image](image_url)
presented in Table 2. The bold digits in Table 2 mean that the corresponding scheme can achieve better visual quality of stego videos. The embedding capacity depends on the video content and the QP value. Using the same QP, the embedding capacity of the video with intense motion and complex texture is larger because the number of non-zero quantized DCT coefficients is larger. For instance, the embedding capacity of video sequence Mobile is obviously larger than that of video sequence Salesman. For each given video, the embedding capacity decreases with the increasing of QP value because coarser quantization will generate more zero quantized DCT coefficients. Our proposed scheme focuses on decreasing the inter-frame distortion drift so that it outperforms Xu et al.’s scheme in terms of average luma PSNR and average luma SSIM. Moreover, the superiority on average luma PSNR is more obvious when the QP is smaller. For video sequence Carphone, our proposed scheme outperforms Xu et al.’s scheme in terms of average luma PSNR by 1.30 dB, 1.89 dB and 3.31 dB using QP 32, QP 28 and QP 24 respectively.

To compare the objective visual quality of decrypted stego videos among Gujjunoori et al.’s scheme [15], Boucama et al.’s scheme [16], Xu et al.’s scheme [20] and our proposed scheme, we illustrate the curves of average luma PSNR values at five different embedding rates using QP 28 in Fig. 7. To calibrate the embedding rate among these four schemes, we denote the embedding rate $\gamma$ and as the ratio of the number of actual embedded message bits $m$ to the sum of the number of bin 1 and the number of bin $-1$. Gujjunoori et al.’s

![Encrypted video frames](image)

**Fig. 6.** Encrypted video frames. (a) Carphone, (b) Coastguard, (c) Container, (d) Foreman, (e) Hall, (f) Harbor, (g) Mobile, and (h) Salesman.

<table>
<thead>
<tr>
<th>Video sequence</th>
<th>Quantization parameter</th>
<th>Embedding capacity</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cover Xu et al. [20] Proposed</td>
<td>Cover Xu et al. [20] Proposed</td>
</tr>
<tr>
<td>Carphone</td>
<td>24</td>
<td>52,355</td>
<td>39.89</td>
<td>33.49</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>22,915</td>
<td>36.95</td>
<td>33.10</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>8934</td>
<td>33.98</td>
<td>31.42</td>
</tr>
<tr>
<td>Coastguard</td>
<td>24</td>
<td>200,428</td>
<td>37.12</td>
<td>26.94</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>92,213</td>
<td>33.94</td>
<td>26.18</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>31,162</td>
<td>30.89</td>
<td>25.38</td>
</tr>
<tr>
<td>Container</td>
<td>24</td>
<td>26,131</td>
<td>38.71</td>
<td>34.91</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>9887</td>
<td>36.05</td>
<td>34.02</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>3389</td>
<td>33.38</td>
<td>32.49</td>
</tr>
<tr>
<td>Foreman</td>
<td>24</td>
<td>60,827</td>
<td>38.86</td>
<td>31.00</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>29,686</td>
<td>36.15</td>
<td>30.17</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>14,221</td>
<td>33.42</td>
<td>28.71</td>
</tr>
<tr>
<td>Hall</td>
<td>24</td>
<td>24,700</td>
<td>39.84</td>
<td>34.51</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>13,461</td>
<td>37.35</td>
<td>32.99</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>7252</td>
<td>34.51</td>
<td>31.50</td>
</tr>
<tr>
<td>Harbor</td>
<td>24</td>
<td>295,230</td>
<td>36.95</td>
<td>27.24</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>160,100</td>
<td>33.65</td>
<td>25.48</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>67,074</td>
<td>30.27</td>
<td>24.54</td>
</tr>
<tr>
<td>Mobile</td>
<td>24</td>
<td>329,938</td>
<td>36.70</td>
<td>24.73</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>169,418</td>
<td>33.13</td>
<td>24.27</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>65,460</td>
<td>29.42</td>
<td>23.96</td>
</tr>
<tr>
<td>Salesman</td>
<td>24</td>
<td>25,661</td>
<td>38.62</td>
<td>33.16</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>12,378</td>
<td>35.60</td>
<td>31.59</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>5600</td>
<td>32.55</td>
<td>30.13</td>
</tr>
</tbody>
</table>
scheme and Bouchama et al.’s scheme cannot achieve high average luma PSNR values because the embedded message bits are largely located in the first few P-frames and the embedding distortion tends to accumulate rapidly. It can be observed that our proposed scheme outperforms other three schemes at five embedding rates. The superiority is more notable if the video content contains intense motion because the inter-frame distortion drift is severer in such video. For video sequence Mobile, our proposed scheme outperforms Gujjunoori et al.’s scheme, Bouchama et al.’s scheme and Xu et al.’s scheme in terms of average luma PSNR by 15.36 dB, 15.11 dB and 5.75 dB respectively at embedding rate 10%. For video sequence Hall, our proposed scheme outperforms Gujjunoori et al.’s scheme, Bouchama et al.’s scheme and Xu et al.’s scheme in terms of average luma PSNR by 9.87 dB, 9.70 dB and 2.47 dB respectively at embedding rate 10%. Through the comparisons in Table 2 and Fig. 7, we can infer that our proposed inter-frame distortion drift prevention scheme is effective in improving the objective visual quality of stego videos.

To explore the effect of our proposed inter-frame distortion drift prevention scheme on the video quality of each frame, we give the comparison of the luma PSNR values of first 100 decoded frames of video sequence Foreman at embedding rate 50% using QP 28 in Fig. 8. It can be seen from Fig. 8 that our proposed scheme can achieve the best visual quality for the following two reasons. Firstly, the coefficient selection in our proposed scheme aims to decrease the embedding distortion in each P-frame. Secondly, data embedding starts in the last frame in each GOP and then proceeds to previous frames one by one to decrease the embedding distortion accumulation in our proposed scheme. Fig. 9 shows the subjective results of the 1st frame, 16th frame, 31st frame, 46th frame, and 61st frame of video sequence Foreman using the same coding setting in Fig. 8. The first frame is an I-frame without containing embedded data. Gujjunoori et al.’s scheme and Bouchama et al.’s scheme cause obvious distortion as depicted in Fig. 9. The subjective visual quality in our proposed scheme is better than that in Gujjunoori et al.’s scheme, Bouchama et al.’s scheme and Xu et al.’s scheme.

3.3. Impacts on the compression efficiency

Data embedding can change the probability distribution of original residual coefficients. As a result, the number of bits used in the stego compressed video will increase compared with that in the cover compressed video. After the cloud server embeds the authentication data into the encrypted video bitstream, the bit rate increment will consume the storage resources. This is not appreciated. To further evaluate the performance of the proposed scheme, the video bit rate increment $BR_{inc}$ caused by data embedding is also compared.
Fig. 7. (continued)

Fig. 8. Comparison of luma PSNR values of first 100 decoded frames of video sequence Foreman at embedding rate 50% using QP 28.
where $BR_{\text{org}}$ is the video bit rate of the original compressed video bitstream and $BR_{\text{stg}}$ is the video bit rate of the stego compressed video bitstream. The comparison of video bit rate increments among Gujjunoori et al.’s scheme [15], Bouchama et al.’s scheme [16], Xu et al.’s scheme [20] and our proposed scheme at five embedding rates using QP 28 is listed in Table 3, in which can be seen that the video bit rate increases as the embedding rate increases. The bold digits in Table 3 mean that the corresponding scheme can achieve the smallest video bit rate increment. In CA VLC-based entropy coding, modifying zero quantized DCT coefficients can considerably increase the video bit rate. To avoid this problem, zero quantized DCT coefficients are not modified during data embedding in Xu et al.’s scheme and our proposed scheme. Therefore, the video bit rate increments in both schemes can maintain in an acceptable range. However, Gujjunoori et al.’s scheme and Bouchama et al.’s scheme modify zero quantized DCT coefficients for data embedding, so the video bit rate increments are larger in most cases.

### 3.4. Discussion of the robustness

The robustness is a beneficial property in video data hiding. With robustness, video data hiding methods can resist some attacks, such as packet loss and recompression. To enhance the robustness of our proposed scheme, error control codes should be used [30,31]. In our experiments, the plain message bits are encoded using BCH $(n, k, t)$ codes [32] in which $n/k = 7/4$ and $t = 1$ to generate the actual to-be-embedded message bits before data embedding. The word plain is associated with the original message bits which need to be transmitted. Given the embedding capacity $C$, a total of $C_p$ plain message bits can be embedded, where $C_p = \lfloor C/k \rfloor$. We use the survival rate to evaluate the robustness of our proposed scheme. The survival rate is defined as the ratio of the number of correctly extracted message bits to the number of actual embedded message bits.

To compare the robustness against packet loss, the stego videos are generated using our proposed scheme with and without BCH $(7, 4, 1)$ codes respectively. Table 4 presents the robustness against packet loss of video sequence Foreman at embedding rate 50%. It can be seen that the survival rate decreases as the packet loss rate increases using certain QP value. The BCH codes can enhance the robustness of our proposed scheme against packet loss.

To compare the robustness against recompression, we first generate stego videos using our proposed scheme with and without BCH $(7, 4, 1)$ codes respectively when QP is certain. Afterwards, the stego videos are decompressed and recompressed using a range of QP values. Table 5 presents the robustness against recompression of video sequence Foreman at embedding rate 50%. Without BCH codes, the survival rate gets close to 50%. It means that the robustness of our proposed scheme against recompression is very weak without error control. Fortunately, BCH codes can enhance the robustness. With BCH codes, the survival rate can be obviously improved. The BCH codes enhance the robustness at the cost of decreasing the number of embedded plain message bits at given embedding rate. In the actual application scenario, we can adjust the BCH codes and the embedding rate to obtain a tradeoff between the robustness and the number of embedded plain
The bit rate increment for preserving the confidentiality of video content. In the data hiding phase, we present a theoretical analysis of the inter-frame distortion drift for alleviating the video quality degradation due to data embedding. Based on the analysis, we estimate the embedding distortions caused by modifying different residual coefficients and embed data into residual coefficients with different priorities for decreasing the inter-frame distortion drift. The data embedding is implemented by the histogram shifting technique. The embedded data can be extracted either in the encrypted domain or in the decrypted domain for meeting different application scenarios. If the receiver decrypts the encrypted video bitstream and extracts the embedded data, the original video bitstream can be perfectly recovered. In addition, if the receiver decrypts the encrypted video bitstream without extracting the embedded data, the bitstream can still be decoded to obtain the reconstructed video with good quality.

In reversible video data hiding, how to determine the optimal modification of quantized DCT coefficients for obtaining best visual quality and compression efficiency at given embedding rate is still a challenging problem. In the future, a comprehensive reversible video data hiding scheme using rate-distortion optimization deserves further investigation.

### Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61572452 and 61772034, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06030601, and in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06030601.

### 4. Conclusion and future work

In this paper, an effective scheme for reversible data hiding in encrypted H.264/AVC video bitstreams is proposed. In the encryption phase, three types of key coding parameters including the intra prediction modes, the motion vector differences and the quantized DCT coefficients are selectively encrypted without video bit rate increment for preserving the confidentiality of video content. In the data hiding phase, we present a theoretical analysis of the inter-frame distortion drift for alleviating the video quality degradation due to data embedding. Based on the analysis, we estimate the embedding distortions caused by modifying different residual coefficients and embed data into residual coefficients with different priorities for decreasing the inter-frame distortion drift. The data embedding is implemented by the histogram shifting technique. The embedded data can be extracted either in the encrypted domain or in the decrypted domain for meeting different application scenarios. If the receiver decrypts the encrypted video bitstream and extracts the embedded data, the original video bitstream can be perfectly recovered. In addition, if the receiver decrypts the encrypted video bitstream without extracting the embedded data, the bitstream can still be decoded to obtain the reconstructed video with good quality.

In reversible video data hiding, how to determine the optimal modification of quantized DCT coefficients for obtaining best visual quality and compression efficiency at given embedding rate is still a challenging problem. In the future, a comprehensive reversible video data hiding scheme using rate-distortion optimization deserves further investigation.
the China Scholarship Council Program under Grant 201506340006. The authors would like to sincerely thank the editors and the anonymous reviewers for their valuable comments.

Appendix A

This appendix is to verify that the assumption $E[\{\hat{F}(n, i, 1) - \hat{F}(n, 1, 1)\}] = 0$ in Eq. (6), where pixel $j$ is the motion prediction of pixel $i$.

To facilitate understanding, we first provide some explanations for some key variables. Let $F(n, i, 1)$ be the original luma value of pixel $j$ in the $(n-1)$th video frame and $\hat{F}(n, 1, 1)$ be the reconstructed luma value of pixel $j$ in the feedback loop at the encoder. We denote the reconstructed luma value of pixel $j$ at the receiver end as $\tilde{F}(n, 1, 1)$. $\hat{F}(n, 1, 1)$ may be different from $\tilde{F}(n, 1, 1)$ due to data embedding. For inter-coded macroblocks, let $r(n, i)$ be the motion compensation residue at the encoder. Let $r(n, i)$ and $\tilde{r}(n, i)$ be the corresponding reconstructed values with embedded data and without embedded data respectively. According to motion compensation, we can rewrite $\tilde{F}(n, 1, 1) - \hat{F}(n, 1, 1)$ as follows.

$$\tilde{F}(n, 1, 1) - \hat{F}(n, 1, 1) = \left(\tilde{F}(n-2, j_{n-2}) + \tilde{F}(n-1, i_{n-1})\right)$$

$$- \left(\hat{F}(n-2, j_{n-2}) + \hat{F}(n-1, i_{n-1})\right)$$

$$= \left(\tilde{F}(n-2, j_{n-2}) - \hat{F}(n-2, j_{n-2})\right)$$

$$+ \left(\tilde{F}(n-1, i_{n-1}) - \hat{F}(n-1, i_{n-1})\right)$$

$$= \left(\tilde{F}(n-3, j_{n-3}) - \hat{F}(n-3, j_{n-3})\right)$$

$$+ \sum_{k=1}^{n-2} \left[\tilde{F}(n-k, i_{n-k}) - \hat{F}(n-k, i_{n-k})\right]$$

(28)

where pixel $j_{n-1}$ is the motion prediction of pixel $i_{n-1}$. The fifth identity in Eq. (28) is based on the condition that the first-frame is not used for data embedding, i.e., $\hat{F}(1, j_{n-1}) = \tilde{F}(1, j_{n-1})$. After motion compensation, the residues tend to be less correlated [27]. We can rewrite $E[\{\tilde{F}(n-1, 1, 1) - \hat{F}(n-1, 1, 1)\}] = 0$ as follows.

$$E[\{\tilde{F}(n-1, 1, 1) - \hat{F}(n-1, 1, 1)\}] = \sum_{k=1}^{n-2} \left[\tilde{F}(n-k, i_{n-k}) - \hat{F}(n-k, i_{n-k})\right]$$

(29)

Therefore, we just need to verify that the difference between the reconstructed residue $\tilde{r}(n, i)$ with embedded data and the reconstructed residual block $\hat{r}(n, i)$ without embedded data has zero mean, i.e., $E[\tilde{r}(n, i) - \hat{r}(n, i)] = 0$.

We can estimate the error matrix $E$ of the pixel luma value between the reconstructed residual block $\hat{r}(n, i)$ with embedded data and the reconstructed residual block $\tilde{r}(n, i)$ without embedded data in Eq. (16) as follows.

$$E = \hat{r} - \tilde{r}$$

where $\hat{r} = \left[\tilde{r} \times \text{Qstep} \times \text{PF}\right]_C + C(\Delta, \times \text{Qstep} \times \text{PF})_C$

$$\approx C(\Delta, \times \text{Qstep} \times \text{PF})_C$$

(30)

References


