

Data Hiding in H.264/AVC Video Files Using the Coded Block Pattern

Hong Zhang^{1,2(✉)}, Yun Cao^{1,2}, Xianfeng Zhao^{1,2},
Haibo Yu^{1,2}, and Changjun Liu^{1,2}

¹ State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences,
No. 89A, Minzhuang Road, Beijing 100093, China
{zhanghong, caoyun, zhaoxianfeng, yuhaibo, liuchangjun}@iie.ac.cn

² University of Chinese Academy of Sciences,
No. 19A, Yuquan Road, Beijing 100049, China

Abstract. This paper presents a novel data hiding method for H.264 which employs the coded block pattern (CBP) as the data carrier. The CBP is defined in the H.264/AVC standard, and indicates which luma and chroma blocks in a macroblock contain non-zero transform coefficients. To the best of our knowledge, little research has been conducted on exploiting the CBP for information hiding. In the proposed method, the four least significant bits of each CBP are mapped to a single bit to form the embedding channel. In addition, an embedding distortion function is specifically designed to evaluate the impacts of CBP manipulation on visual quality and compression efficiency. Combined with the suitably defined distortion function, the syndrome-trellis codes are employed to minimize the overall embedding impact. The experimental results show that only a slight reduction in visual quality would be caused, even when large payload is embedded. In addition, the compression efficiency could be maintained with insignificant fluctuation in bitrates.

Keywords: Data hiding · Coded block pattern · H.264/AVC · Compressed video

1 Introduction

In recent years, with the rapid development of broadband Internet services and practical video compression techniques, high-quality digital video can be conveniently downloaded and broadcast through social networking services such as YouTube, Facebook and Weibo. Therefore, digital video has become one of the most influential media in the entertainment industry. With the increasing popularity of digital video, copyright protection for video grows in importance and receives a lot of attention. Information hiding is one of the possible solutions for protecting copyrighted video from illegal copying and distribution.

Information hiding refers to the process of inserting information into a host to achieve certain features or serve specific purposes. Information hiding in video

can be divided into two main categories according to the domain where data embedding takes place. One category is spatial domain-based information hiding. Under this embedding scheme, raw pixel values are directly modified using those conventional information hiding methods designed for digital image, such as Least Significant Bit (LSB) Matching [1] and Spread Spectrum (SS) [2, 3]. However, researchers refrain from taking these approaches due to the inevitable loss of hidden data caused by video compression. The other category is usually referred to as compressed domain-based information hiding. Methods of this type manipulate entities of compressed video to realize data embedding. Commonly used entities include motion vectors (MVs) [4–8], intra prediction modes [9–11], inter prediction modes [12, 13], quantized DCT coefficients (DCTCs) [14–16], quantization parameters [17, 18], reference frame indices [19] and variable length codes [20, 21].

Prior to H.264/AVC [22], there are numerous information hiding methods proposed for MPEG-1/2 standards. However, the data carriers (hiding venues) are restricted to DCTCs and MVs, mainly because other entities of MPEG-1/2 compressed video are not quite suitable for information hiding purposes. With the advent of the H.264/AVC standard in 2003, information hiding in video has then received much attention due to the introduction of various new technical features in H.264/AVC, which provides plenty of opportunities for information hiding. For instance, in Hu et al.'s work [9], 4×4 intra prediction modes (I4-modes) were exploited for data embedding, which is based on a predefined mapping rule between I4-modes and hiding bits. Kapotas and Skodras [12] proposed forcing the encoder to choose a particular inter prediction mode according to the information to be embedded. In [21], a specific type of codeword in CAVLC (Context-based Adaptive Variable Length Coding) was employed as the data carrier to realize information hiding.

In the H.264/AVC standard, the coded block pattern (CBP) is a syntax element contained in the Macroblock Layer¹, and indicates which luma and chroma blocks within a macroblock contain non-zero transform coefficients. To the best of our knowledge, the CBP is never utilized as the data carrier for information hiding, contrasting with other entities of H.264/AVC compressed video. Nevertheless, the CBP has the potential to be exploited for data embedding, and the reasons are listed as follows. First, CBPs are losslessly coded to form an integral part of compressed bitstreams. Thus it is reasonably practicable to employ the CBP as the data carrier for information hiding. Second, for CBP-based data embedding, any crucial process in video compression would not be influenced except for entropy coding. Consequently, only a slight reduction in compression efficiency and visual quality would be caused. Last but not least, compressed video usually contains a substantial quantity of CBPs, which ensures the high embedding capacities that CBP-based data hiding methods can have.

In this paper, we propose a novel data hiding method for H.264/AVC which exploits the CBP as the data carrier. In the proposed embedding scheme, the

¹ The Macroblock Layer contains all the syntax elements necessary to decode a single macroblock.

four LSBs of each CBP are mapped to a single bit to form the embedding channel. In addition, an embedding distortion function is specifically designed to evaluate the impacts of CBP manipulation on visual quality and compression efficiency. With the suitably defined distortion function utilized, the Syndrome-Trellis Codes (STC) [23] are further exploited to minimize the overall embedding impact. The experimental results show that the proposed method is capable of performing data hiding at the expense of a fairly slight reduction in visual quality, even when large payload is embedded. More importantly, the problem of bitrate overhead caused by data embedding could be considerably alleviated, enabling an efficient use of transmission and storage resources. As a result, the proposed method is suitable to be used in practice.

The rest of this paper is organized as follows. In Sect. 2, the basic concepts of H.264/AVC syntax and the CBP are introduced. In Sect. 3, the embedding distortion caused by modification of CBPs is analyzed, and our CBP-based data hiding method is presented. In addition, the practical implementation of the proposed method is discussed. In Sect. 4, comparative experiments are conducted to demonstrate the effectiveness of our method. Finally we draw conclusions in Sect. 5.

2 The Coded Block Pattern in H.264/AVC

H.264/AVC syntax is defined in the H.264/AVC standard and specifies the exact structure of an H.264/AVC-compliant bitstream in terms of syntax elements.

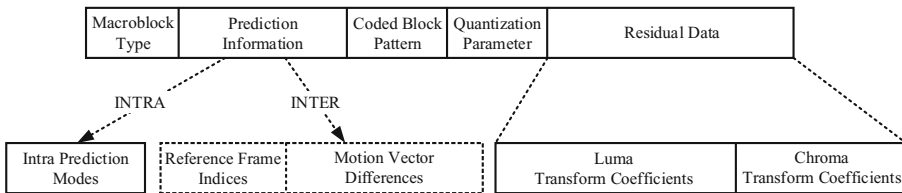


Fig. 1. The basic structure of the Macroblock Layer in the H.264/AVC standard.

According to H.264/AVC syntax specifications, the CBP is a syntax element for macroblocks that are not coded using 16×16 intra prediction modes. Contained in the Macroblock Layer (Fig. 1), the CBP indicates which luma and chroma blocks in a macroblock contain non-zero transform coefficients. In addition, the CBP can be expressed as a 6-bit binary number, and can take values between 0 and 47.

As shown in Fig. 2, given a CBP represented by $\overline{b_5b_4b_3b_2b_1b_0}$ where b_i ($i = 0, 1, \dots, 5$) denotes a binary digit, it is constructed as follows:

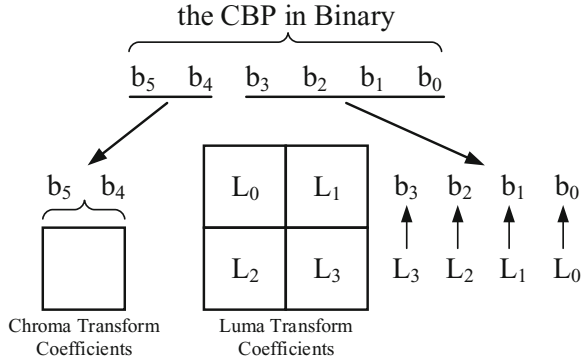


Fig. 2. The relationship between the CBP and the corresponding transform coefficients. b_i ($i = 0, 1, 2, 3$) indicates whether there are non-zero coefficients present in L_i , one of the four 8×8 luma blocks. b_5 and b_4 closely correlate with the presence of non-zero chroma transform coefficients.

1. Each of the four LSBs of the CBP, i.e. b_i ($i = 0, 1, 2, 3$), indicates whether there are one or more non-zero transform coefficients within the corresponding 8×8 luma block (denoted by L_i in Fig. 2). Specifically, the value of b_i is given by

$$b_i = \begin{cases} 1 & \text{non-zero coefficient(s) present} \\ 0 & \text{no non-zero coefficients present} \end{cases} \quad (1)$$

where $i = 0, 1, 2, 3$.

2. The two MSBs (Most Significant Bit) of the CBP, b_5 and b_4 , are associated with chroma transform coefficients, and can take the values specified by

$$\overline{b_5 b_4} = \begin{cases} 00_2 & \text{no non-zero chroma coefficients present} \\ 01_2 & \text{non-zero chroma DC coefficient(s) present,} \\ & \text{no non-zero chroma AC coefficients present} \\ 10_2 & \text{non-zero chroma AC coefficient(s) present} \end{cases} \quad (2)$$

Residual data for a macroblock is transmitted according to the corresponding CBP. If the CBP indicates a block contains no non-zero coefficients, that block would be bypassed. In addition, since the CBP is losslessly entropy coded and inserted into H.264/AVC bitstreams, the decoding of the residual data for a macroblock could be facilitated by referring to the corresponding CBP.

3 The Proposed Method

As introduced in Sect. 2, the CBP indicates which luma and chroma blocks in a macroblock contain non-zero transform coefficients. In addition, residual data for a macroblock is transmitted according to the corresponding CBP.

In this section, we propose a data hiding method for H.264/AVC which utilizes the CBP as the data carrier. An embedding distortion function is designed to evaluate the reduction in visual quality and compression efficiency caused by modification of CBPs. In addition, combined with the suitably defined distortion function, the STC [23] is adopted in the embedding channel construction to minimize the overall embedding impact. Consequently, the proposed method can embed the payload with very limited embedding distortion introduced.

3.1 The Embedding Channel Construction

Given the cover with N CBPs denoted by $\mathbf{C} = (C_1, C_2, \dots, C_N)$, the embedding channel construction is carried out by exploiting the STC as follows.

Each CBP C_i ($i = 1, 2, \dots, N$) is mapped to a single bit according to the parity check function given by

$$P(C) = \oplus_{i=0}^3 c_i \quad (3)$$

where c_i denotes one of the four LSBs of C and satisfies $\overline{c_3c_2c_1c_0} = C \%_0^2 16$. The symbol \oplus represents the bitwise exclusive OR operation. Based on the parity check function P , the embedding channel $\mathbf{p} = (p_1, p_2, \dots, p_N)$ can then be obtained, where $p_i = P(C_i)$ ($i = 1, 2, \dots, N$).

Assume that modifications of CBPs are mutually independent and let p_i ($i = 1, 2, \dots, N$) be assigned a positive scalar γ_i expressing the impact of making an embedding change at the corresponding CBP C_i , the STC is adopted to minimize the overall embedding impact $D(\mathbf{p}, \mathbf{p}') = \sum_{i=1}^N \gamma_i [p_i \neq p'_i]$ ³, where $\mathbf{p}' = (p'_1, p'_2, \dots, p'_N)$ denotes the embedding channel with hidden data. Given an αN -bit message \mathbf{m} to be embedded, where α denotes the embedding payload, the STC-based data embedding and extraction can be formulated as

$$\text{Emb}_{\text{STC}}(\mathbf{p}, \mathbf{\Gamma}, \mathbf{m}) = \arg \min_{\mathbf{p}' \in \mathcal{C}(\mathbf{m})} D(\mathbf{p}, \mathbf{p}') = \tilde{\mathbf{p}}, \quad (4)$$

$$\text{Ext}_{\text{STC}}(\tilde{\mathbf{p}}) = \tilde{\mathbf{p}} \mathbf{H}_{\text{STC}}^T = \mathbf{m}. \quad (5)$$

Here, $\mathbf{\Gamma} = (\gamma_1, \gamma_2, \dots, \gamma_N)$ denotes the distortion scalar vector, $\mathcal{C}(\mathbf{m})$ represents the coset corresponding to the syndrome \mathbf{m} , and $\mathbf{H}_{\text{STC}} \in \{0, 1\}^{\alpha N \times N}$ is the parity check matrix of the STC which should be shared between communication parties. For detailed information about the STC, please refer to [23].

The design of embedding distortion is an essential part of the distortion minimization framework, and could directly affect the performance of the embedding algorithm. Given a CBP C and one of its four LSBs c , the embedding distortion caused by modification of c is analyzed as follows. If c is modified from 1 to 0, as mentioned in Sect. 2, those non-zero transform coefficients within the corresponding 8×8 luma block (Fig. 2) would be bypassed, leading to a decrease in bitrates

² $\%_0^2$ denotes the modulo operator, i.e., $a \%_0^2 b$ denotes the remainder of a divided by b .

³ The Iverson bracket $[I]$ is defined to be 1 if the logical expression I is true and 0 otherwise.

but a reduction in reconstructed visual quality. On the other hand, if c is modified from 0 to 1, although reconstructed visual quality could be maintained, bitrate overhead would be caused inevitably because 64 extra zero coefficients have to be entropy coded and transmitted. In addition, the number of bits required for representing C could also be affected if c is modified. Consequently, both visual quality and compression efficiency could be influenced by modification of c . The distortion caused by modification of c (denoted by $\Psi(C, c)$) can then be defined in terms of the fluctuation in visual quality and bitrates, i.e.

$$\Psi(C, c) = |\mathcal{J}(\mathbf{S}, \mathbf{S}_{\text{rec}}, R) - \mathcal{J}(\mathbf{S}, \mathbf{S}'_{\text{rec}}, R')|. \quad (6)$$

Here, \mathbf{S} denotes the original macroblock associated with C , \mathbf{S}_{rec} and \mathbf{S}'_{rec} represent the corresponding reconstructed (decoded) macroblock with c unaltered and that with c modified, respectively. R denotes the number of bits required for coding \mathbf{S} when c is unaltered, and R' modified. \mathcal{J} denotes the Lagrangian cost function given by

$$\mathcal{J}(\mathbf{A}, \mathbf{B}, R) = D(\mathbf{A}, \mathbf{B}) + \lambda R \quad (7)$$

where D represents the visual distortion measured as SSD (Sum of Squared Differences), and λ is the Lagrange multiplier determined by the employed H.264 encoder that is used for controlling the tradeoff between D and R .

According to the analysis above, for a CBP C , the corresponding embedding distortion scalar γ can be intuitively defined as

$$\gamma = \begin{cases} \min \{\Psi(C, c_i) \mid i = 0, 1, 2, 3\} & \mathbf{S} \text{ is inter-coded.} \\ \infty & \text{else} \end{cases} \quad (8)$$

where \mathbf{S} denotes the macroblock associated with C . c_i ($i = 0, 1, 2, 3$) represents one of the four LSBs of C , and satisfies $\overline{c_3c_2c_1c_0} = C \% 16$.

3.2 Practical Implementation

In practice, almost all available H.264/AVC CODECs can be customized to implement the proposed method. Typically, I-frames are not used for the data embedding, and secret message bits are embedded into P- or B-frames in a frame-by-frame manner. Without loss of generality, the processes of embedding and extraction with one single P- or B-frame are described as follows.

1. Data Embedding

First, the raw frame is subdivided into macroblocks to be fed into the customized encoder. For each macroblock not coded using 16×16 intra-prediction modes, record its associated CBP determined by the encoder, and calculate the corresponding distortion scalar according to (8). Afterwards, the cover with N CBPs $\mathbf{C} = (C_1, C_2, \dots, C_N)$ and the distortion scalar vector $\mathbf{\Gamma} = (\gamma_1, \gamma_2, \dots, \gamma_N)$ are obtained.

Second, the embedding channel $\mathbf{p} = (p_1, p_2, \dots, p_N)$ is constructed according to the parity check function (3). As described in (4), the STC is subsequently exploited to embed an αN -bit message \mathbf{m} by turning \mathbf{p} into $\tilde{\mathbf{p}}$, where $\text{Emb}_{\text{STC}}(\mathbf{p}, \mathbf{\Gamma}, \mathbf{m}) = \tilde{\mathbf{p}} = (\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_N)$.

Third, the raw frame is fed into the encoder again. Each CBP C_i in \mathbf{C} is then subject to the possible modification controlled by \tilde{p}_i . Specifically, if $\tilde{p}_i = p_i$, C_i is left unaltered. In case $\tilde{p}_i \neq p_i$, C_i is modified by flipping one of its four LSBs \tilde{c} that satisfies

$$\tilde{c} = \arg \min_{c \in \mathcal{X}} \Psi(C_i, c). \quad (9)$$

where \mathcal{X} denotes the set of the four LSBs of C_i .

Finally, the compressed frame with an αN -bit message embedded is inserted into the compressed bitstream.

2. Data Extraction

Compared with the embedding process, data extraction is much easier. Received by the recipient, the compressed frame can be decoded by any H.264 decoder to parse a total of N CBPs from the Macroblock Layer (Fig. 1). With $\tilde{\mathbf{p}}$ reconstructed, the message can be extracted by computing $\mathbf{m} = \tilde{\mathbf{p}}\mathbf{H}_{\text{STC}}^T$.

4 Experiments

In this section, experiments are conducted to demonstrate that the proposed method is capable of embedding the payload with very limited embedding distortion introduced.

4.1 Experimental Setup

In the experiments, x264 [24], a highly practical H.264 encoder, is customized to prepare the compressed video samples, with only the *Baseline Profile* adopted. For each compressed video sample to be prepared, the quantization parameter (QP) is set as 20 or 28. In addition to the proposed method, Kapotas and Skodras [12] and Aly's [6] methods are also implemented for comparison, and are referred



Fig. 3. The raw sequences used for performance evaluation.

Table 1. The impacts of data embedding on visual quality and compression efficiency under different experimental settings. RC_{PSNR} defined in (10) is used for evaluating the impacts on visual quality, and RC_{BR} given by (11) is employed for evaluating the impacts on compression efficiency. (NF (the Number of Frames), IM (Impact Measure)).

Sequence	NF	IM	QP = 20			QP = 28		
			Ours	ALG1	ALG2	Ours	ALG1	ALG2
Bus	150	RC_{PSNR}	-0.3947	0.0529	-0.0312	-0.3252	-0.0343	-0.1454
		RC_{BR}	-1.4219	6.3199	2.4008	-1.4552	11.6714	5.5127
City	300	RC_{PSNR}	-0.2203	0.0652	-0.0024	-0.3662	-0.0314	-0.0428
		RC_{BR}	-1.4853	14.2124	6.6469	-2.1621	19.3939	16.3481
Coastguard	300	RC_{PSNR}	-0.5229	0.0290	-0.0169	-0.3103	-0.0575	-0.1120
		RC_{BR}	-1.5379	4.8642	3.8725	-1.4821	8.7027	7.7539
Crew	300	RC_{PSNR}	-0.4400	-0.0210	-0.0397	-0.3900	-0.0910	-0.0908
		RC_{BR}	-1.1542	2.4761	5.6439	-1.5194	2.9023	11.1871
Flower	250	RC_{PSNR}	-0.8790	0.2352	-0.0306	-0.4450	0.2469	-0.0229
		RC_{BR}	-1.2293	9.1775	2.9747	-1.2171	16.1361	5.3261
Football	260	RC_{PSNR}	-0.4943	-0.0165	-0.0306	-0.3418	-0.0605	-0.0932
		RC_{BR}	-0.9673	1.7062	3.3846	-0.9351	2.3373	6.8543
Foreman	300	RC_{PSNR}	-0.2437	0.0572	-0.0238	-0.4193	-0.1266	-0.0989
		RC_{BR}	-1.6034	9.7714	6.5952	-2.1021	17.7053	18.2170
Harbour	300	RC_{PSNR}	-0.5503	0.0483	-0.0145	-0.2770	-0.0058	0.0029
		RC_{BR}	-1.5228	3.9207	2.0061	-1.4535	7.2136	4.6024
Ice	240	RC_{PSNR}	-0.2561	0.0358	-0.0893	-0.5677	-0.0782	-0.1632
		RC_{BR}	-0.8086	9.4638	15.1377	0.4275	11.8305	26.7857
Mobile	300	RC_{PSNR}	-0.3518	0.1210	0.0169	-0.3083	0.1328	0.0206
		RC_{BR}	-0.9311	5.5549	1.8686	-1.5534	11.9526	4.0792
Soccer	300	RC_{PSNR}	-0.2269	-0.0119	-0.0237	-0.3597	-0.0830	-0.1270
		RC_{BR}	-1.0695	3.9515	5.0369	-1.3647	4.9959	11.9280
Stefan	90	RC_{PSNR}	-0.5985	0.1132	0.0306	-0.3304	-0.0420	-0.0503
		RC_{BR}	-1.3692	6.1065	2.6727	-1.4838	13.8216	7.0889
Tempete	260	RC_{PSNR}	-0.3252	0.1052	-0.0310	-0.3141	0.0344	-0.0258
		RC_{BR}	-1.0634	4.7759	2.1870	-1.5454	10.1810	5.6177
Walk	376	RC_{PSNR}	-0.4400	0.0230	-0.0391	-0.4569	-0.0877	-0.1113
		RC_{BR}	-0.9860	4.2180	4.0630	-1.4641	7.4054	10.2806

to as ALG1 and ALG2 respectively. Since these three methods are not applicable to I-frames, the embedding strength is measured by average embedded bits per inter-coded frame (bpf), and is fixed at 190 bpf in the experiments. As shown in Fig. 3, 14 CIF-resolution raw test sequences are used in the experiments to evaluate the impacts of data embedding on visual quality and compression efficiency. All of them are progressively scanned and with YUV 4:2:0 color sampling.

4.2 Impacts on Visual Quality

PSNR (Peak Signal-to-Noise Ratio) is a commonly used objective measurement of picture quality. In YUV domain, human visual system is more sensitive to luminance (Y) than to chrominance (U or V). Thus, the picture quality in the experiments is measured by PSNR to luminance (PSNR_Y). Accordingly, given a stego video sequence V_s and the corresponding cover V_c , the impacts of data embedding on visual quality can be evaluated by computing the relative change between the average PSNR_Y values of V_s and V_c , i.e.

$$\text{RC}_{\text{PSNR}}(V_s, V_c) = \frac{\text{PSNR}_Y(V_s) - \text{PSNR}_Y(V_c)}{\text{PSNR}_Y(V_c)} \times 100 \quad (10)$$

where $\text{PSNR}_Y(V)$ denotes the average PSNR_Y value of the sequence V .

It can be observed from Table 1 that, our proposed method always has an adverse impact on visual quality. The reasons are explained as follows. As described in Sect. 3, the modification of a given CBP is performed by flipping one of its four LSBs. If that bit is modified from 1 to 0, those non-zero residual coefficients within the corresponding 8×8 luma block would be bypassed, which could easily result in a reduction in visual quality. If that bit is modified from 0 to 1, the 64 zero coefficients in the corresponding 8×8 luma block need to be entropy coded and transmitted. In this case, only compression efficiency would be negatively influenced. Therefore, when the embedding payload is large enough, a reduction in visual quality would always be caused by our proposed method.

Nevertheless, for our proposed method, visual quality is regarded as a crucial factor in evaluating the embedding distortion. In addition, by virtue of the STC, a practical distortion minimization framework, our proposed method could embed the payload with very limited embedding distortion introduced. Thus, although the impacts on visual quality could always be made by our method, it could be considerably alleviated and reduced to a fairly low level. As shown in Table 1, for the proposed method, the minimum value of RC_{PSNR} is given by -0.8790 for $\text{QP} = 20$, and -0.5677 for $\text{QP} = 20$.

As can be seen in Table 1, compared with our method, both ALG1 and ALG2 seem to cause a minor reduction in visual quality. However, this is achieved at the expense of significant bitrate overhead, leading to poor rate-distortion performance in general.

As shown in Fig. 4, we take a closer look at the specific sequence named “stefan”, and RC_{PSNR} is computed for each frame. It can be observed that, for the proposed method, although a reduction in visual quality is caused, it can always be maintained at a low level. In contrast, with the increase in QP, the impact of data embedding on visual quality becomes more obvious and severe for ALG1 and ALG2.

4.3 Impacts on Compression Efficiency

Compression efficiency, also known as “compression capability”, is the most fundamental driving force behind the adoption of modern video compression tech-

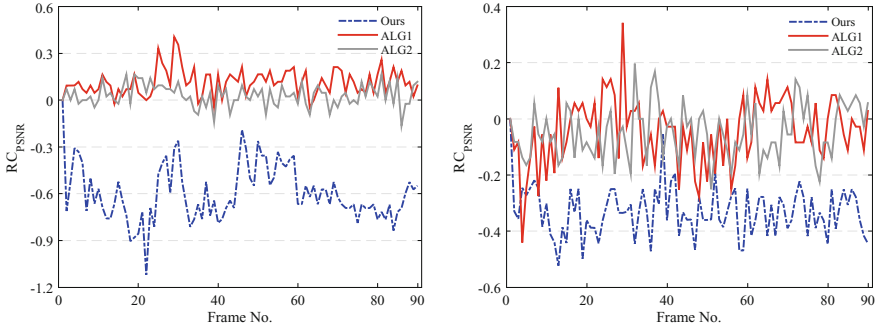


Fig. 4. Impacts on visual quality are evaluated in a frame-by-frame manner for sequence “stefan”, with QP set as 20 (left) and 28 (right).

niques. Since compression efficiency is a crucial factor in assessing the performance of video encoders, the impacts of data embedding on compression efficiency could suggest whether or not the corresponding data hiding method is suitable to be used in situations where limited transmission and storage resources are available.

With compression efficiency represented as bitrates, given a a stego video sample V_s and the corresponding cover V_c , the impacts of data embedding on compression efficiency can then be evaluated by computing the relative change between the bitrates of V_s and the V_c , i.e.

$$RC_{BR} = \frac{BR(V_s) - BR(V_c)}{BR(V_c)} \times 100 \quad (11)$$

where $BR(V)$ denotes the bitrate of the sequence V .

As described in Sect. 3, a distortion function is defined in terms of the fluctuation in visual quality and compression efficiency. In addition, by using the STC, our method can embed the payload while minimizing that suitably defined distortion function. Therefore, compared with traditional video data hiding methods, the proposed approach can perform data embedding with minor fluctuation in bitrates, even when large payload (190 bpf) is embedded.

As can be seen in Table 1, for our proposed method, the problem of bitrate overhead is considerably alleviated, enabling more efficient use of transmission and storage resources. Therefore, our method is capable of maintaining compression efficiency at the cost of a fairly slight reduction in visual quality. As expected, both ALG1 and ALG2 are subject to marked bitrate overhead, leading to reduced compression efficiency.

Given a single frame F_s of a stego video sequence, the impacts on compression efficiency could be evaluated as

$$RC_{frmSize} = \frac{\text{size}(F_s) - \text{size}(F_c)}{\text{size}(F_c)} \quad (12)$$

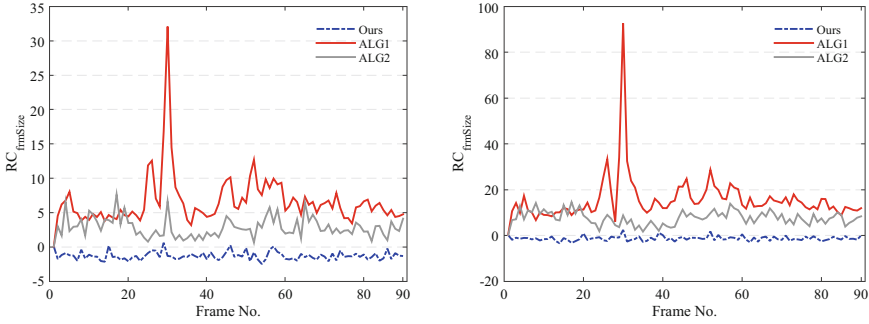


Fig. 5. Impacts on compression efficiency are evaluated in a frame-by-frame manner for sequence “stefan”, with QP set as 20 (left) and 28 (right).

where F_c denotes the frame of the cover video sequence which corresponds to F_s , and $\text{size}(F)$ represents the size of the frame measured in bytes.

As shown in Fig. 5, our proposed method can always provide well maintained compression efficiency, but, by contrast, significant bitrate overhead could be caused by ALG1 and ALG2. Thus, compared with ALG1 and ALG2, our method is capable of performing data embedding with rate-distortion efficiency well maintained, even when large payload is embedded.

5 Conclusion

In this paper, we propose a novel data hiding method which utilizes the CBP as the data carrier. A reasonable embedding distortion is designed by considering the reduction in visual quality and compression efficiency caused by CBP manipulation. Combined with the distortion function, the STC is employed to minimize the overall embedding impact. Thus, the proposed method is capable of conducting data embedding with limited distortion introduced.

The experimental results demonstrate that, compared with traditional video data hiding methods, the proposed approach is capable of embedding large payload with only a slight reduction in visual quality produced. In addition, the problem of bitrate overhead could be considerably alleviated, thus well maintaining the compression efficiency. Therefore, our method enables more efficient use of transmission and storage resources, and is suitable to be used in practice.

Acknowledgments. This work was supported by the NSFC under 61303259 and U1536105, Strategic Priority Research Program of CAS under XDA06030600, National Key Technology R&D Program under 2014BAH41B01, and Key Project of Institute of Information Engineering, CAS, under Y5Z0131201.

References

1. Mielikainen, J.: LSB matching revisited. *IEEE Signal Process. Lett.* **13**(5), 285–287 (2006)
2. Hartung, F., Girod, B.: Watermarking of uncompressed and compressed video. *Signal Process.* **66**(3), 283–301 (1998)
3. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Process.* **6**(12), 1673–1687 (1997)
4. Jordan, F., Kutter, M., Ebrahimi, T.: Proposal of a watermarking technique for hiding/retrieving data in compressed and decompressed video. Tech. rep. M2281, ISO/IEC Document, JTC1/SC29/WG11, Stockholm, Sweden, July 1997
5. Xu, C., Ping, X., Zhang, T.: Steganography in compressed video stream. In: *Proceedings of 1st International Conference on Innovative Computing, Information and Control*, pp. 269–272. Beijing, China, September 2006
6. Aly, H.A.: Data hiding in motion vectors of compressed video based on their associated prediction error. *IEEE Trans. Inf. Forensics Secur.* **6**(1), 14–18 (2011)
7. Cao, Y., Zhang, H., Zhao, X., Yu, H.: Video steganography based on optimized motion estimation perturbation. In: *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pp. 25–31. Portland (June 2015)
8. Zhang, H., Cao, Y., Zhao, X.: Motion vector-based video steganography with preserved local optimality. *Multimed. Tools Appl.* **75**, 13503–13519 (2015)
9. Hu, Y., Zhang, C.: Yuting: information hiding based on intra prediction modes for H.264/AVC. In: *Proceedings of 2007 IEEE International Conference on Multimedia and Expo*, pp. 1231–1234. Beijing, China, July 2007
10. Yang, G., Li, J., He, Y.: Zhiwei: an information hiding algorithm based on intra-prediction modes and matrix coding for H.264/AVC video stream. *AEU Int. J. Electron. Commun.* **65**(4), 331–337 (2011)
11. Xu, D., Wang, R., Wang, J.: Prediction mode modulated data-hiding algorithm for H.264/AVC. *J. Real Time Image Process.* **7**(4), 205–214 (2012)
12. Kapotas, S.K., Skodras, A.N.: A new data hiding scheme for scene change detection in H. 264 encoded video sequences. In: *Proceedings of 2008 IEEE International Conference on Multimedia Expo*, pp. 277–280. Hannover, Germany (2008)
13. Zhang, H., Cao, Y., Zhao, X., Zhang, W., Yu, N.: Video steganography with perturbed macroblock partition. In: *Proceedings of the 2nd ACM Workshop on Information Hiding and Multimedia Security*, pp. 115–122. Salzburg, Austria, June 2014
14. Ma, X., Li, Z., Tu, H., Zhang, B.: A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift. *IEEE Trans. Circ. Syst. Video Technol.* **20**(10), 1320–1330 (2010)
15. Lin, T., Chung, K., Chang, P., Huang, Y., Liao, H.M., Fang, C.: An improved DCT-based perturbation scheme for high capacity data hiding in H.264/AVC intra frames. *J. Syst. Softw.* **86**(3), 604–614 (2013)
16. Chang, P., Chung, K., Chen, J., Lin, C., Lin, T.: A DCT/DST-based error propagation-free data hiding algorithm for hevc intra-coded frames. *J. Vis. Commun. Image Representation* **25**(2), 239–253 (2014)
17. Wong, K., Tanaka, K., Takagi, K., Nakajima, Y.: Complete video quality-preserving data hiding. *IEEE Trans. Circ. Syst. Video Technol.* **19**(10), 1499–1512 (2009)
18. Shanableh, T.: Data hiding in MPEG video files using multivariate regression and flexible macroblock ordering. *IEEE Trans. Inf. Forensics Secur.* **7**(2), 455–464 (2012)

19. Li, J., Liu, H., Huang, J., Zhang, Y.: A robust watermarking scheme for H.264. *Digital Watermarking*, January 2009. http://dx.doi.org/10.1007/978-3-642-04438-0_1
20. Kim, S.M., Kim, S.B., Hong, Y., Won, C.S.: Data hiding on H.264/AVC compressed video. In: Kamel, M., Campilho, A. (eds.) *ICIAR 2007*. LNCS, vol. 4633, pp. 698–707. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-74260-9_62](https://doi.org/10.1007/978-3-540-74260-9_62)
21. Liao, K., Lian, S., Guo, Z., Wang, J.: Efficient information hiding in H.264/AVC video coding. *Telecommun. Syst.* **49**(2), 261–269 (2012)
22. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 560–576 (2003)
23. Filler, T., Judas, J., Fridrich, J.: Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Trans. Inf. Forensics Secur.* **6**(3), 920–935 (2011)
24. VideoLAN-x264, The Best H.264/AVC Encoder. <http://www.videolan.org/developers/x264.html>