A CABAC based HEVCc video steganography algorithm without bitrate increase

Article *in* Journal of Computational Information Systems · March 2015

DOI: 10.12733/jcis13774

CITATIONS
20

READS
1,216

3 authors, including:

Gaobo Yang
Hunan University
125 PUBLICATIONS **2,285** CITATIONS

SEE PROFILE

Journal of Computational Information Systems 11: 6 (2015) 2121–2130
Available at http://www.Jofcis.com

A CABAC Based HEVC Video Steganography Algorithm without Bitrate Increase ⋆

Bo JIANG [1], Gaobo YANG [1,*], Weibin CHEN [2]

[1] *School of Information Science and Engineering, Hunan University, Changsha 410082, China*

[2] *School of Electronics and Communication Engineering, Changsha Institute, Changsha 410005, China*

Abstract

Video steganography is an important tool for secure communication, which should be closely related with video coding standard. By investigating the details of context-adaptive binary arithmetic coding (CABAC) in the entropy coding of the latest high efficiency video coding (HEVC) standard, a novel concept of "constant bitrate information bit" (CBIB) is presented. For the motion vector difference (MVD) in HEVC, there are a large amounts of constant-bitrate information bits which can exploited for video steganography. By designing a codeword reservation and substitution rule for the encoding of MVDs, a CABAC-based video steganography algorithm is proposed for HEVC video. Video steganography is conducted in the stage of entropy coding, which is the last step of video compression. It is efficient because no complex operations such as rate distortion optimization (RDO) or full decoding are involved. Moreover, since it is based on bit substitution with constant bitrate, there is not any bitrate increase caused by video steganography. Experimental results on several test sequences prove the effectiveness of the proposed video steganography approach specifically designed for HEVC.

Keywords: Video Steganography; Entropy Coding; CABAC; Bit Substitution

1 Introduction

Steganography is the art to hide secret information behind text, audio, and video. There are massive information amount of digital video signals, which makes video steganography an important tool for secure communication and copyright protection [1]. Meanwhile, efficient video coding is needed to make video communication feasible. Thus, most digital video contents are encoded and stored in the form of compressed bit-stream. In general, video steganography algorithms are usually combined with video encoder. According to the embedding stage of compression or the location of syntax elements, video steganography algorithms can be divided into three categories. That is, secret information in embedded in the stages of transform coding, inter/intra mode prediction and entropy coding. The syntax elements for data hiding are discrete sine transform (DCT) coefficients [2, 3], prediction modes [4, 5], and motion vectors [6], respectively.

⋆Project supported by the National Nature Science Foundation of China (No. 61379143, No. 61072122).

*Corresponding author.
Email address: yanggaobo@hnu.edu.cn (Gaobo YANG).

1553–9105 / Copyright © 2015 Binary Information Press
DOI: 10.12733/jcis13774
March 15, 2015

Up to present, most video steganography are designed for MPEG-x and H.26x. The most representative works are summarized as follows. To improve the visual effect of video, a data hiding algorithm is proposed for H.264/AVC video streams without intra- frame distortion drift [2]. Several paired-coefficients of a 4x4 DCT block are exploited to accumulate the embedding induced distortion. The directions of intra-frame prediction are utilized to avert the distortion drift. We also present an information hiding algorithm based on intra-prediction modes and matrix coding for H.264/AVC video stream [5]. Intra-4x4 coded blocks (I4-blocks) are divided into groups and two watermark bits are mapped to every three I4-blocks by matrix coding to map between watermark bit and intra-prediction modes. It can guarantee a high PSNR and slight bitrate increase after data hiding. To further enhance the security, embedding position template is utilized to select candidate I4-blocks.

In recent years, HEVC has been developed as a new video coding standard mainly focusing on the coding of Ultra High Definition (UHD) videos as the high resolution and high quality videos are getting more popular. As a successor to H.264/AVC standard, HEVC can achieve outstanding compression performance for HD and UHD video [7]. HEVC is expected to become the mainstream video coding standard in the next 5 to 10 years. Apparently, video steganography for HEVC video is worthy of investigation because of both theoretical value and practical application potential. Since HEVC is a newly developed video coding standard, the video steganography algorithms specifically designed for digital video encoded by HEVC is still in scarcity. To the best of our knowledge, there are only two approaches in the literature. An error propagation free data hiding algorithm is proposed for HEVC intra-coded frames [8]. Since HEVC framework adopts both DCT and discrete sine transform (DST) for transform coding, it is actually extending the idea of DCT coefficients-based data hiding approach for H.264/AVC [2] to DCT/DST coefficients-based data hiding approach for HEVC. In addition, a large-capacity information hiding method is proposed for HEVC Video [9], it follows the idea of modulating intra-prediction mode for data hiding [5]. The mapping rule between prediction modes and secret information is established by utilizing the probability distribution of the statistically optimal and suboptimal prediction modes.

In this paper, a novel video steganography algorithm without bit-rate increase is presented for HEVC video. Secret information in embedded in the stage of entropy coding by exploiting the new features of CABAC for HEVC. The contributions of the proposed approach are two-folds. First, by analyzing the syntax of CABAC, a novel concept of "constant bitrate information bit" (CBIB) is introduced. CBIB implies that the mutual substitution between 0 and 1 remains the bitstream syntax compliance but does not change the bitrate after bit substation. Second, a replacement rule is specified for the MVD codeword to achieve data hiding. Since data hiding is performed in the stage of entropy coding, which is the last step of video compression, the proposed approach does not need any computationally intensive processing such as RDO and full-decoding of input video stream and can achieve real-time performance. Moreover, since data hiding is realized by utilizing the CBIBs of CABAC syntax elements, it does not have any bitrate increase. It is particularly suitable for broadcast monitoring because no re-packetizing is required for video stream.

The rest paper is organized as follows: Section 2 discusses in detail the CABAC entropy coding for HEVC, and the concept of CBIB is introduced; Section 3 present the proposed video steganography approach for HEVC video using the CBIBs; Experimental results and performance analysis are reported in Section 4, and we conclude this paper in Section 5.

2 CABAC Entropy Coding of HEVC

Entropy coding is a lossless compression step at the last stage of video encoding (and the first stage of video decoding). By previous steps such as intra/inter prediction, transform coding and quantization, video data has been reduced to a series of syntax elements. These syntax elements include prediction modes, prediction residue and motion vectors. They are needed to be further compressed by entropy coding to be used at the decoder for frame reconstruction. CABAC is an entropy coding method firstly presented in H.264/AVC to improve coding efficiency. Actually, it maps the symbols (i.e., syntax elements) to codewords with a non-integer number of bits. For the latest video coding standard HEVC, CABAC is also adopted as the base entropy coding. Specifically, CABAC involves three main functions: binarization, context modeling and arithmetic coding. Binarization maps the syntax elements to binarize symbols (bins). Context modeling estimates the probability of the bins. Finally, arithmetic coding is used to compress the bins into bits based on the estimated probability (context coded) or equal probability of 0.5 (bypass coded). Fig. 1 shows the schematic diagram of CABAC in HEVC, which highlights the main functions and the data dependencies. The updated context is fed back for recursive interval division and accurate probability estimation, and there is no influence on bypass encoding. Therefore, any changes will make the coding stream structure change, and the encoded video cannot playback correctly [10]. From the above analysis, we can conclude that binarization is the only outlet available for video steganography during the CABAC encoding process.

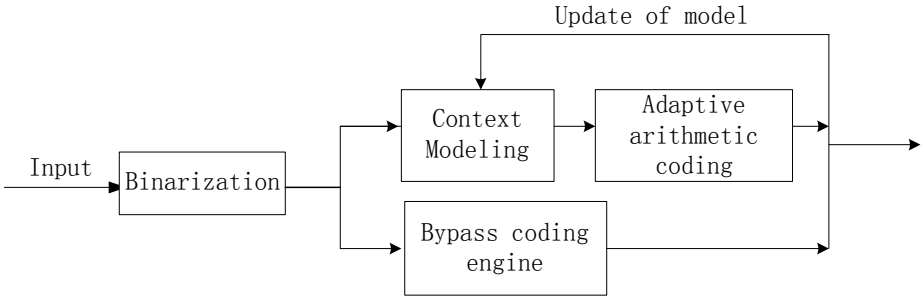


Fig. 1: CABAC schematic diagram

Though both H.264/AVC and HEVC adopt CABAC for entropy coding, there are some differences between them. Compared with H.264/AVC, the maximum number of context-coded bins is reduced by 8 [11] in HEVC. The reduction of bins mainly comes from the binarization process. For HEVC, four different binarization processes are used, including unary (U), truncated unary (TU), fixed length (FL), k-th-order Exp-Golomb (EGK). Let N be a symbol to be encoded which has 8 possible values from 0 to 7. An example is provided in Table 1 to show the difference among these four binarization schemes. Unary coding (U) is quite simple because it uses the first N bits "1" as the first bin and the last bin "0" as the terminator. Truncated unary coding (TU) is similar with U. The only difference is that when $N + 1 = cMax$, all bins are "1". The decoder searches for a bin with a series of "0" up to cMax to determine when the syntax element is completed. For FL, it uses a fixed length of bins. That is, ceil (log2 $(cMax + 1)$). For EGK, it is a little complex because it is made up of three parts:

$$\underbrace{\{1, 1, 1...\}}_{M} 1[Info] \tag{1}$$

Where M=floor $(log2N + 1)$ and $Info = N + 1 - 2N$. The length of $Info$ is M, and each EGK coding including $(2M + 1)$ dates (Table 1).

Table 1: Binarization used in the CABAC of HEVC

N	$U(cMax = 7)$	$TU(cMax = 7)$	FL	$EK0$
0	0	0	000	1
1	10	10	001	010
2	110	110	010	00100
3	1110	1110	011	00101
4	11110	11110	100	00110
5	111110	111110	101	00111
6	1111110	1111110	110	0001000
7	11111110	11111111	111	0001001

Four binarization methods can be divided into two categories according to the last bit. The first class includes U and TU whose last bits determine when a syntax element is complete. Specifically, if a "1" bit is replaced by "0", the decoder will stop searching the bins. On contrary, if the last bit "0" is replaced by "1", the decoder will not stop searching unless a bit "1" is found. It is no doubt that this will introduce errors. FL and EGK belongs the second category because even when there is bit substitution between 0 and 1 for the last bit, the bitstream still remains syntax compliance and it can be decoded by standard decoder. Furthermore, its bit-rate keeps unchanged and the impact of bit substitution is within 1. Therefore, a novel concept of "bit substitution with constant bitrate" (CBIB) can be introduced. That is, the last bit of elements encoded by the second category entropy coding is defined as CBIB. For HEVC, almost all "Flag" syntax elements are encoded with FL. If these "flag" information are modified during encoding or steganograpy, it will incur unpredictable errors in the decoder. In summary, those CBIBs, which are suitable for video steganography, are produced by EGK.

Table 2: Codeword of EGK in H.264/AVC and HEVC

N	H.264/AVC	HEVC
0	0	$abs\_mvd\_greater0\_flag$
1	10	$abs\_mvd\_greater1\_flag$
2	110	0**0**
3	1110	0**1**
-	-	-
9	111111111	11001**1**
10	1111111110000	11010**0**

For both HEVC and H.264/AVC, EGK is mainly utilized for the entropy coding of motion vector differences (MVDs) which have relatively big amplitude. In HEVC, the number of con-

text coded bins is significantly reduced for MVDs. Only the first two bins are context coded ($abs\_mvd\_greater0\_flag, abs\_mvd\_greater1\_flag$), and it is followed by bypass coded first-order Exp-Golomb bins (abs_mvd_minus2). However, the first 9 bins of MVD are context coded TU bins in H.264/AVC, followed by bypass coded third-order Exp-Golomb bins. This kind of adjustment is beneficial for video steganography. Compared with H.264/AVC, there are much more EGK-coded bins in the bitstream for HEVC. Apparently, more EGK-coded bins mean that there will have more CBIBs, which implies a larger capacity for video steganography. Table 2 makes a comparison of codeword encoded by EGK between HEVC and H.264/AVC, where CBIB is highlighted in bold. It is clear that every EGK word in HEVC provides an extra CBIB when its absolute value is between 2 and 9. Fig. 2 shows the distribution of absMVD, which is the absolute value of MVD. For these four video sequences, the values of most absMVDs are distributed between 0 and 8. It also illustrates that HEVC can offer more CBIBs for video steganography than H.264/AVC.

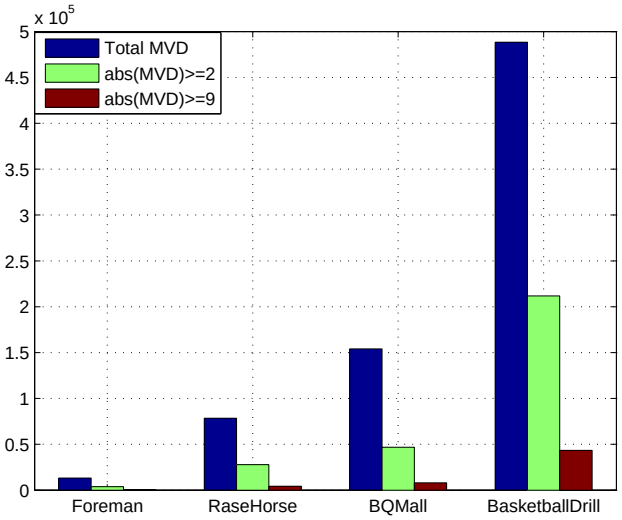


Fig. 2: Distribution of absMVD

3 The Proposed Video Steganography Algorithm for HEVC

To enhance the algorithm security, a chaos-based encryption technique is used for pre-processing of secret data to be hidden. Logistic map is a simple non-linear model, which is defined as follows:

$$x_{k+1} = \mu x_k(1 - x_k) \quad (|\ (x_k)\ | < 1) \tag{2}$$

Where μ is referred to be the bifurcation parameter ($1 < \mu < 4$). The dynamics of this system changes dramatically with different μ, exhibiting periodicity or chaos. For $3.569945 < \mu \leq 4$, the resulted sequence is in a chaotic state. It generates a large number of uncorrelated, random-like and non-periodic signals. Without knowing μ and the initial conditions x_0, it is very difficult for anyone to recover the chaotic sequences. That is, the sensitive dependence improves the security of secret data.

As discussed above, the CBIBs of EGK-encoded MVD is suitable for video steganography. Thus, video steganography turns into an issue of defining a mapping rule, which determines whether a CBIB of MVD will be changed or not. To reduce the possible side effect on visual quality, a threshold T is configured in video encoder. It controls which CBIBs will be used for video steganograpy. The steps for video steganography are summarized as follows.

Step 1 Generate a chaotic sequence $\{B_i, i = 1, 2, 3 \ldots\}$ using Logistic map.

Step 2 When it meets $absMVD \geq T$ and B_i, the encoder will hide a secret bit W. If W does not equal the CBIB, the bins will be changed as mapping rule, which is defined as follows.

$$bins' = \begin{cases} bins \mid (\sim 1) & CBIB = 1, w = 0 \\ bins \wedge 1 & CBIB = 0, w = 1 \end{cases} \tag{3}$$

where "$bins$" and "$bins'$" represents the binarize symbols of a $absMVD$ syntax elements before and after steganography. If a secret bit W is not equals to the CBIB, the CBIB will turn to W, else the CBIB will remain the same.

Step 3 Encode the next codeword and then go to Step 2 for video steganography until there are no more bits to be hidden.

Apparently, there are no complex operations for video steganography. It does not need any decoding or re-encoding process. It simply makes three judgments and at most one bit substitution in the process of entropy coding. This can guarantee the real-time performance of video steganography.

For data extraction, it is simultaneously performed during the entropy decoding of MVD. It needs the same logistic sequence as data hiding. Only two simple judgments are needed for data extraction, and the whole process of data extraction is summarized as follows.

Step 1 Generate the same logistic sequence B_i, which is the same with that used in video steganograpy process.

Step 2 After parsing the codeword of $absMVD$ during entropy decoding, the decoder reads a logistic bit B_i, and judge if meets $absMVD \geq T$ and $B_i = 1$. If yes, the secret bit W is restored as follows.

$$W = bins' \mid 1. \tag{4}$$

Step 3 Repeat Step 2 until all the MVDs are decoded.

4 Experimental Results and Discussion

To show the performance of the proposed approach, we have done experiments on standard test sequences. The proposed approach is integrated into the reference software of HEVC codec (H-M10.0, low delay mode). The experimental platforms are Intel Core 2 CPU with 1GB RAM, and the operation system is Linux. All test sequences have 80 frames, and the details are summarized in Table 3. Apparently, they have different motion complexities. These five sequences are encoded with a GOP as IPPP. That is, there is one I frame every four frames. The values of quantization parameters (QP) are set with 17, 22 and 27, respectively. In the following, we will make analysis about the capacity, visual quality, bitrate change and real-time performance.

Table 3: Standard test sequences

Class	resolutions	Bit-rate	videos	Description
A	2560x1600	30	*PeopleOnStreet*	Moderate motion
B1	1920x1080	24	*Kimono*	Low motion
B2	1920x1080	50	*BasketballDrive*	Complex motion
C	832x480	60	*BQMall*	Moderate motion
D	416x240	30	*RaceHorses*	Complex motion

Table 4: Embedding capacity, SSIM and bit-rate increasing (T=4)

Class	Sequence	QP	Capacity	SSIM			Δ Bit rate
				Original	Stegoed	Δ SSIM	
A	*PeopleOnStreet*	17	3634	0.98724	0.98146	-0.00578	0
		22	3337	0.97301	0.96612	-0.00689	0
		27	2859	0.95228	0.94525	-0.00703	0
B1	*Kimono*	17	659	0.97130	0.96393	-0.00737	0
		22	379	0.96309	0.95351	-0.00958	0
		27	283	0.95192	0.94113	-0.01079	0
B2	*BasketballDrive*	17	1038	0.96692	0.95709	0.00983	0
		22	530	0.93875	0.90798	-0.03077	0
		27	380	0.91974	0.90810	-0.01164	0
C	*BQmall*	17	226	0.97911	0.96743	-0.01168	0
		22	139	0.96982	0.95928	-0.01054	0
		27	117	0.95342	0.94602	-0.00740	0
C	*RaceHorses*	17	104	0.98831	0.95984	-0.02847	0
		22	97	0.97641	0.94536	-0.03105	0
		27	78	0.94836	0.92353	-0.02483	0

4.1 Capacity

For the proposed video steganography approach, there are three factors which have influences on its capacity: the number of CBIB, the number of "1" in B_i and the threshold T. Table 4 shows the embedding capacity every frame for the test sequences. Apparently, *PeopleOnStreet* sequence provides higher capacity that the rest sequences because of the biggest resolution and relatively complex movement.

Fig. 3 shows the improvement of capacity. Comparison is made between the proposed approach and the method in [12], which embeds secret data using MVD encoded by EG3 in H.264/AVC video. It is obvious that there is great improvement of embedding capacity as what claimed in Section 2.

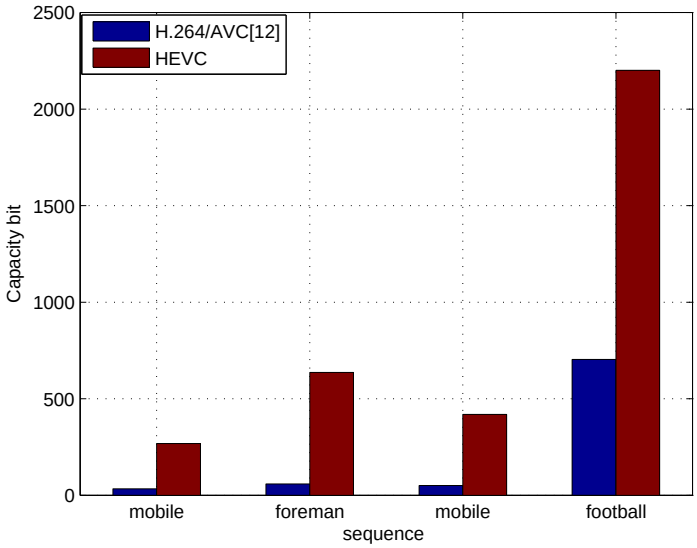
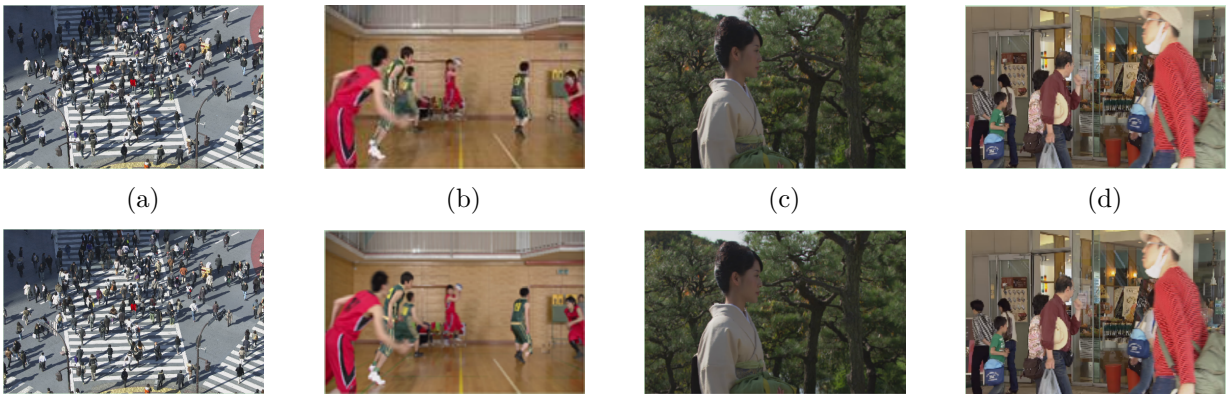


Fig. 3: Capacity comparison between the proposed approach and the method in [12]

4.2 Visual quality of stego-video

The visual quality of stego-video is expected to be equivalent or very close to that of the original video. Fig. 4 shows the visual qualities of original video and stego-videos. From left to right, they are $PeopleOnStreet, BasketballDrive, KimonoandBQmall$, respectively. There are no visible artifacts or visually noticeable degradation for the stego-videos.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4: Original video frames and frames with hidden data, (a)-(d) original videos and (e)-(h) videos with hidden data

Moreover, objective comparison is made between the original video and stego-video. Structural similarity index (SSIM) is utilized to evaluate the visual quality because it has better capability of perceptual quality. SSIM is within the range of [0, 1], where 1 indicates two frames in comparison are identical. The results are reported in Table 4. From it, we can know that the greatest difference between the original video and stego-video is less than 0.03105dB, and the smallest difference is only 0.00578dB. Therefore, there is only negligible degradation of visual quality for the stego-videos.

4.3 Bitrate

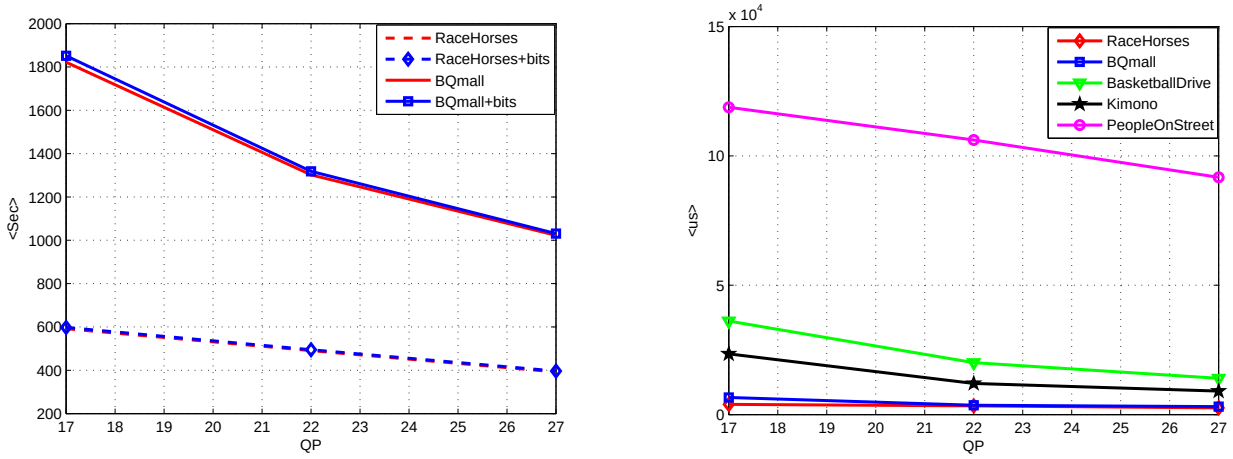
Bitrate increase is another concern for video steganography. The ratio of bitrate increase is defined as follows:

$$\mu = \frac{m-n}{n} \times 100\%. \tag{5}$$

Where m and n are the bitrates before and after steganography, respectively. The experimental result is reported in Table 4 as well. There is not any bitrate increase for stego-video. It is benefited from the mechanism of data hiding, because the CBIBs are fully exploited for video steganography. Actually, this is the greatest advantage of the proposed approach.

4.4 Computational cost

For some applications demanding real-time performance such as video conference and broadcast monitoring, computational cost is an issue to be concerned for video steganography. For the proposed approach, only simple judgments about threshold T and bit substitution are involved for data hiding. For data extraction, there are also two simple judgments. That is, no complex operations such as decoding or re-encoding are involved in the data hiding and data extraction procedures. Fig. 5 shows a comparison between the normal encoding time and the additional processing time for video steganography. It is quite obvious that the proposed approach is efficient.



(a) (b)

Fig. 5: Real-time performance of the proposed approach. (a) encoding time and (b) additional processing time

5 Conclusions

In this paper, a novel concept of bit substitution with constant bitrate (CBIB) is introduced. By investigating the entropy coding mechanism of HEVC, we found that there are much more CBIBs in the HEVC video than H.264/AVC. Therefore, a codeword mapping rule is designed for video steganography, which fully exploiting those CBIBs in the entropy encoding of MVDs by EGK. Experimental results show that the proposed approach has desirable performances of capacity,

visual quality and computational cost. Moreover, it does not lead to any bitrate increase, which is the greatest advantage of the proposed approach.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (61072122, 61379143), the program for New Century Excellent Talents in University (NCET-11-0134), the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) under grant 20120161110014.

References

[1] Sang J, Xiang H, Sang N, et al. Increasing the data hiding capacity and improving the security of a double-random phase-encoding technique based information hiding scheme [J]. Optics Communications, 2009, 282(14): 2713-2721.

[2] Ma X, Li Z, Tu H, et al. A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift [J]. IEEE Trans. on Circuits and Systems for Video Technology, 2010, 20(10): 1320-1330.

[3] Lin T J, Chung K L, Chang P C, et al. An improved DCT-based perturbation scheme for high capacity data hiding in H. 264/AVC intra frames [J]. Journal of Systems and Software, 2013, 86(3): 604-614.

[4] Xu D, Wang R, Wang J. Prediction mode modulated data-hiding algorithm for H. 264/AVC [J]. Journal of Real-Time Image Processing, 2012, 7(4): 205-214.

[5] Yang G, Li J, He Y, et al. An information hiding algorithm based on intra-prediction modes and matrix coding for H. 264/AVC video stream [J]. AEU- International Journal of Electronics and Communications, 2011, 65(4): 331-337.

[6] Wang R, Hu L, Xu D. A watermarking algorithm based on the CABAC entropy coding for H. 264/AVC [J]. Journal of Computer Information System, 2011, 7(6): 2132-2141.

[7] Sullivan G J, Ohm J, Han W J, et al. Overview of the high efficiency video coding (HEVC) standard [J]. IEEE Trans. on Circuits and Systems for Video Technology, 2012, 22(12): 1649-1668.

[8] Chang P C, Chung K L, Chen J J, et al. An error propagation free data hiding algorithm in HEVC intra-coded frames [C]. Asia-Pacific Conference on Signal and Information Processing (APSIPA). IEEE, 2013: 1-9.

[9] Wang J, Wang R, Li W, et al. A Large-capacity Information Hiding Method for HEVC Video [C]. The 3rd International Conference on Computer Science and Service System. Atlantis Press, 2014: 139-142.

[10] Hu L J. Watermarking research based on H.264/AVC video [D]. Ningbo: Ningbo University, 2012.

[11] Sze V, Budagavi M. High throughput CABAC entropy coding in HEVC [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1778-1791.

[12] Zou D, Bloom J A. H. 264 stream replacement watermarking with CABAC encoding [C]. IEEE International Conference on Multimedia and Expo (ICME), 2010, 117-121.