Computers, Materials & Continua CMC, vol.63, no.2, pp.911-922, 2020

Reversible Data Hiding Based on Run-Level Coding in H.264/AVC Video Streams

Yi Chen[1], Hongxia Wang[2,*] and Xuyun Zhang[3]

Abstract: This paper presents a reversible data hiding (RDH) method, which is designed by combining histogram modification (HM) with run-level coding in H.264/advanced video coding (AVC). In this scheme, the run-level is changed for embedding data into H.264/AVC video sequences. In order to guarantee the reversibility of the proposed scheme, the last nonzero quantized discrete cosine transform (DCT) coefficients in embeddable 4×4 blocks are shifted by the technology of histogram modification. The proposed scheme is realized after quantization and before entropy coding of H.264/AVC compression standard. Therefore, the embedded information can be correctly extracted at the decoding side. Peak-signal-noise-to-ratio (PSNR) and Structure similarity index (SSIM), embedding payload and bit-rate variation are exploited to measure the performance of the proposed scheme. Experimental results have shown that the proposed scheme leads to less SSIM variation and bit-rate increase.

Keywords: Reversible data hiding, run-level coding, visual quality, bit-rate increase, H.264/AVC video streams.

1 Introduction

Reversible data hiding (RDH) has attracted much attention from many researchers because it can embed additional information into digital media, such as images and videos, and recover the original media content after extracting the embedded information [Ni, Shi, Ansari et al. (2006); Tian (2003)]. In the past two decades, RDH in images has been rapidly developed that leads to make many research achievements [Shi, Li, Zhang et al. (2016)]. For instance, Du et al. [Du, Yin and Zhang (2018)] proposed a RDH method with high payload for JPEG images. In Du et al.'s method [Du, Yin and Zhang (2018)], they changed the JPEG file header to preserve the image quality and sorted the run/length values to make the marked JPEG image less in terms of filesize increase. However, there still exists much research room to develop the technology of RDH, even DH, in videos.

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu, 611756, China.

[2] College of Cybersecurity, Sichuan University, Chengdu, 610064, China.

[3] Department of Electrical, Computer and Software Engineering, University of Auckland, Auckland, 1023, New Zealand.

[*] Corresponding Author: Hongxia Wang. Email: hxwang@scu.edu.cn.

Received: 22 July 2019; Accepted: 15 September 2019.

With the development of video compression standard, such H.264/advanced video coding (AVC) and high efficiency video coding (HEVC) [Sullivan, Ohm, Han et al. (2012); Wiegand, Sullivan, Bjontegaard et al. (2003)], it is necessary to develop RDH or data hiding (DH) in videos for different applications purposes [Tew and Wong (2014)]. Currently, there have been some DH schemes like Nie et al. [Nie, Weng, Feng et al. (2018); Chen, Wang, Wu et al. (2018a); Chen, Wang, Wu et al. (2018b)] and RDH schemes such as Kim et al. [Kim and Kang (2018)] are reported. In Nie et al. [Nie, Weng, Xu et al. (2018)], Nie et al. gave a mapping rule to modify the intra prediction modes to propose a video DH method. In this method, they first made an analysis of changing intra prediction modes and then they gave a mapping rule combined with the analysis. Based on this, they defined a distortion cost function to exploit syndrome-trellis code [Filler, Judas and Fridrich (2011)] to embed data. Therefore, this method can obtain good visual quality and better security in covert communication. In addition, Chen et al. [Chen, Wang, Wu et al. (2018b)] also proposed a video DH with little bit-rate growth by changing middle and high frequency coefficients combined with exploiting modification direction (EMD) [Zhang and Wang (2006)]. Afterward, Chen et al. [Chen, Wang, Wu et al. (2018a)] gave several assumptions to analyze distortion spread and then combined with matrix embedding to propose a video DH scheme. Based on these assumptions and matrix embedding, their scheme can obtain relatively good visual quality and slight bit-rate variation. However, these mentioned schemes cannot recover the original compression videos. Therefore, this has motivated us to propose a novel RDH scheme in this paper. The proposed scheme brings the ideal of EMD [Zhang and Wang (2006)] to combine with one dimensional histogram modification (HM) and a novel mapping rule. Therefore, the compression video sequences with the hiding data keeps less increase in terms of bit-rate and provides good visual quality.

This rest of this paper is organized as follows. The proposed video RDH method is present in Section 2. And then some experimental results are given and analyzed in Section 3. Finally, we draw some conclusions in Section 4.

2 Proposed scheme

This section contains three subsections, i.e., run-level coding, data hiding, data extracting and video recovery.

2.1 Run-level coding

As a part of H.264/AVC, run-level coding plays an important role in this compression coding standard. Therefore, we first review the procedure of run-level coding. To better describe run-level coding, we take an example combined with zig-zag scanning order (shown in Fig. 1) since 4×4 block is regarded as a basic operation unit in a macroblock of H.264/AVC.

As shown in Fig. 1, run-level coding is used in the current 4×4 block to encode quantized coefficients and compactly represent nonzero quantized DCT coefficients and the large number of zero quantized DCT coefficients. For example, according to zig-zag scanning order the input vector from low frequency area to high frequency area is represented as [0, -3, 1, 0, 0, 0, 1, 0, 0, -1, 0, 1, 0, 0, -1, 0]. After run-level coding, it can be denoted as (1, -3),

(0, 1), (3, 1), (2, -1), (1, 1) and (2, -1). Here, each (*x*, *y*) is called as run-level pair. *x* denotes the number of zeroes preceding a nonzero coefficient and *y* represents the value of the nonzero coefficients. Based on this, furthermore, all run-level pairs are denoted as two vectors, i.e., L=[-3, 1, 1, -1, 1, 1] and R=[1, 0, 3, 2, 1, 2], in order. Our proposed scheme is based on the modifications of R and L of luma component in macroblocks with intra frame 4×4 prediction modes in intra frames.

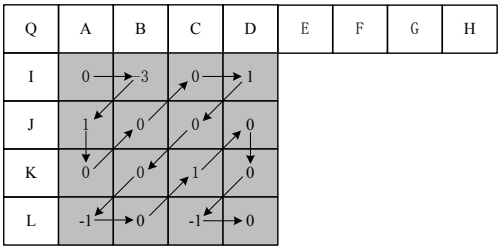

Figure 1: A luminance block containing 16 quantized DCT coefficients in zig-zag scanning order and some coefficients of its adjacent blocks

2.2 Data hiding

In the scheme, the macroblocks with intra frame 4×4 prediction modes in intra frames are selected as embeddable blocks. Furthermore, a 4×4 block of luma component in an embeddable macroblock is regarded as embeddable if the number of nonzero quantized DCT coefficients is not less 7 and the value of the last quantized DCT coefficient (shown in Fig. 1) is 0. Based on the run-level coding, the procedure of data hiding is described in details as follows.

At the encoding side, the data-hider first counts the number of embeddable 4×4 blocks of luma component in the current embeddable macroblock, and after that according to it the data-hider can know how many information bits the current macroblock can carry.

In an embeddable 4×4 block, the last nonzero quantized DCT coefficient '*P*' will be shifted by

$$P' = \begin{cases} P+1, \textit{if } P > 0 \\ P-1, \textit{if } P < 0 \end{cases} \tag{1}$$

The 4×4 block cannot be exploited to carry data and is changed just for restoring the original videos at the decoding side. *P'* is corresponding modification of *P*. When the location of *P* is not AC_{15}, none or one zero quantized DCT coefficient of the current 4×4 block may be changed to carry message by using the given rule in the following.

As the above-mentioned, the number of to-be-embedded information bits is firstly calculated according to the number of embeddable 4×4 blocks in a macroblock. After that, part or all of embeddable 4×4 blocks in the macroblock are selected to carry message. For the embeddable 4×4 blocks selected for data hiding, one of the zero quantized DCT coefficients, which are closest to the last nonzero quantized DCT coefficient according to the zig-zag scanning order in 4×4 block, is modified for carrying additional data. If there are (or is) *j* ($1 \leqslant j \leqslant 16$) embeddable 4×4 blocks and we can combine a mapping rule with

j zero quantized DCT coefficients for embedding data. For instance, the to-be-embedded data is embedded by Eq. (2) when j=1.

$$Z' = \begin{cases} 0, & if\ B = '0' \\ -1, & if\ B = '1' \\ not\ used, & otherwise \end{cases} \tag{2}$$

where Z'=(z_1, z_2, ..., z_k) consists of k ($1 \leqslant k \leqslant j$, here, k=j=1) zero quantized DCT coefficients and B=(b_1, b_2, ..., b_i) is the to-be-embedded data with i ($1 \leqslant i \leqslant 5$, here i=1) bits. z_j corresponds to the j^{th} embeddable 4×4 block in a given macroblock and P_j, which corresponds to z_j and is the last nonzero quantized DCT coefficient in the same 4×4 block, is modified by Eq. (1). When j=2 or 3, we can embed two information bits according to Eq. (3) as follows.

$$Z' = \begin{cases} (0,0,*), & if\ B = '00' \\ (1,0,*), & if\ B = '01' \\ (-1,0,*), & if\ B = '10' \\ (0,1,*), & if\ B = '11' \\ not\ used, & otherwise \end{cases} \tag{3}$$

where '*' denotes there exist (s) embeddable zero quantized DCT coefficients but no need to change them. For instance, one of the two zero coefficients may be changed for DH when j=2. That is to say, there is on more zero quantized DCT coefficients to leave. For j=3, there exist 1 coefficient and it is needless to change it for DH. Furthermore, it is unnecessary to change the last zero coefficient corresponding to '*' when j=3. In fact, although we can use and change '*' according to Eq. (2) to embed 1 information bit, the main idea of our proposed scheme is at most 1 zero coefficient in an embeddable macroblcok is changed that results in nothing is done for other zero coefficients. Therefore, we just need select two embeddable 4×4 blocks so that two information bits can be carried. In our experimental simulation, according to the order of coding 4×4 block in a macroblock we select enough embeddable 4×4 blocks for DH. Similarly, we select embeddable 4×4 blocks to carry message for other cases.

When j=4, 5, 6 or 7, three information bits can be carried by just changing 1 zero coefficient (using Eq. (4)).

$$Z' = \begin{cases} (0,0,0,0,*), & if\ B = '000' \\ (1,0,0,0,*), & if\ B = '001' \\ (-1,0,0,0,*), & if\ B = '010' \\ (0,1,0,0,*), & if\ B = '011' \\ (0,-1,0,0,*), & if\ B = '100' \\ (0,0,1,0,*), & if\ B = '101' \\ (0,0,-1,0,*), & if\ B = '110' \\ (0,0,0,1,*), & if\ B = '111' \\ not\ used, & otherwise \end{cases} \tag{4}$$

Likewise, when j=8, 9, 10, 11, 12, 13, 14 or 15, four information bits can be carried by

using similar embedding rule like j=1, 2, ..., 7. In the best case, there exist 16 embeddable 4×4 blocks in a macroblock and five information bits can be embedded into the macroblock by using similar embedding rule like j=1, 2, ..., 7. For each macroblock of videos, marked videos can be obtained by the embedding rule.

2.3 Data extracting and video recovery

The procedure of data extraction and recovery is also very simple and it is described in the following.

For each 4×4 block of macroblocks with intra-frame 4×4 prediction mode in I frames, if the number of nonzero quantized DCT coefficients is not less than 7 and the absolute value of AC_{15} is not larger than 1, the current 4×4 block is selected as an embeddable 4×4 block. Similarly, the data-extractor first counts the number of embeddable 4×4 blocks in a macroblock denoted as j and then according to it to extract the embedded information bits. At the decoder side, the embedded information bits are extracted according to the mapping rule exploited at the encoder side. They are respectively described as follows.

For j=1,

$$B' = \begin{cases} '0', & if\ Z' = 0 \\ '1', & if\ Z' = -1 \end{cases} \tag{5}$$

where B'=(b_1, b_2, ..., b_i)($1 \leqslant i \leqslant 5$, here, i=1) is the extracted information bits, Z'=(z'_1, z'_2, ..., z'_k)($1 \leqslant k \leqslant j \leqslant 16$, here, k=j=1) consists of coefficients with embedded information bits and z'_j corresponds to the j^{th} embeddable 4×4 block in a given macroblock.

For j=2 or 3,

$$B' = \begin{cases} '00', & if\ Z' = (0,0,*) \\ '01', & if\ Z' = (1,0,*) \\ '10', & if\ Z' = (-1,0,*) \\ '11', & if\ Z' = (0,1,*) \end{cases} \tag{6}$$

analogously, where '*' denotes there exists embeddable coefficients for data hiding but no need to change them, which corresponds to the encoder side.

For j=4, 5, 6 or 7,

$$B' = \begin{cases} '000', & if\ Z' = (0,0,0,0,*) \\ '001', & if\ Z' = (1,0,0,0,*) \\ '010', & if\ Z' = (-1,0,0,0,*) \\ '011', & if\ Z' = (0,1,0,0,*) \\ '100', & if\ Z' = (0,-1,0,0,*) \\ '101', & if\ Z' = (0,0,1,0,*) \\ '110', & if\ Z' = (0,0,-1,0,*) \\ '111', & if\ Z' = (0,0,0,1,*) \end{cases} \tag{7}$$

where three information bits can be extracted according to this equation. Likewise, four information bits can be extracted when j=8, 9, 10, 11, 12, 13, 14 or 15 and five embedded

information bits can be extracted when j=16. By doing so, all embedded information can be extracted out. In the following, we will describe the process of recovering the original compressed videos. For each 4×4 block of embeddable macroblocks, the number of nonzero quantized DCT coefficients is not less than 7 and the absolute value of the last nonzero coefficient is 1 and it is set to 0. After that, if the absolute value of the last nonzero quantized DCT coefficient is larger than 1 and then this coefficient is modified by

$$P'' = \begin{cases} P' - 1, & if\ P' > 0 \\ P' + 1, & if\ P' < 0 \end{cases} \tag{8}$$

By doing this for each macroblock, the original compressed videos can be restored.

3 Experimental results

Table 1: Embedding payloads (bits) under different QP values

Sequences	QP=24	QP=25	QP=26	QP=27	QP=28	QP=29	QP=30
Akiyo	6704	6314	5654	5245	4806	4264	3777
Bus	12589	12177	11195	10450	9687	8691	7717
Coastguard	23194	21797	19196	17359	15611	13123	11238
Container	12364	11760	10815	10273	9807	9135	8530
Flower	15948	16105	15947	15861	15628	15352	14944
Foreman	14054	13264	11712	10760	9705	8425	7199
Mobile	26333	26367	25951	25713	25363	24754	24000
News	10727	10331	9527	8986	8386	7561	6928
Silent	17442	16146	13922	12429	10979	8688	6935
Stefan	6660	6596	6307	6080	5867	5536	5296
Tempete	21197	20759	19928	19137	18224	16867	15527
Waterfall	23388	22066	19257	17130	14791	11634	9148
Average	15883	15806	14117	13285	12404	11167	10103

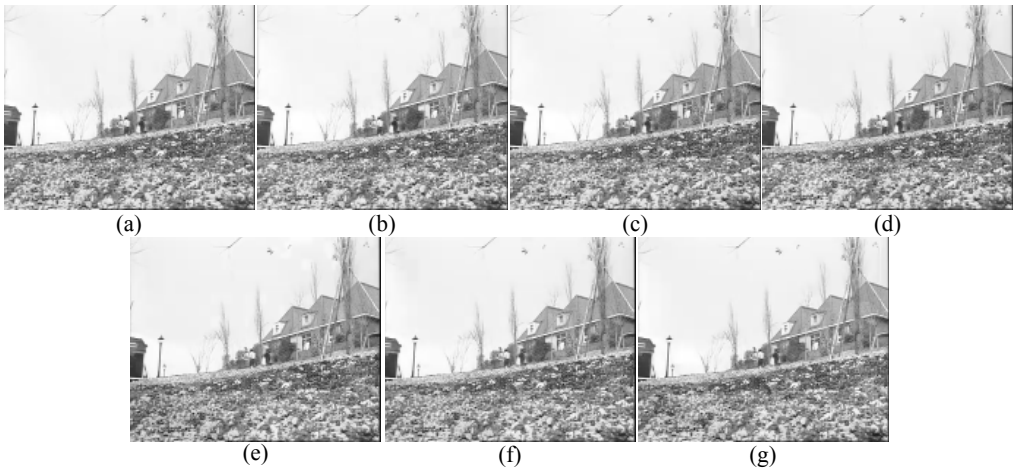

Figure 2: 129th marked frame of Flower. (a) QP=24. (b) QP=25. (c) QP=26. (d) QP=27

(e) QP=28. (f) QP=29. (g) QP=30

Experiments are implemented by H.264/AVC test model, i.e., Joint Model 12.0 (JM 12.0) with Main profile. 12 video sequences, i.e., Akiyo, Bus, Coastguard, Container, Flower, Foreman, Mobile, News, Silent, Stefan, Tempete and Waterfall, with the resolution of 352×288 are selected for the experiments. The group of picture is IBPBPBPBPBPBPBPB and Frame Rate is 30 per second. Except for the mentioned, other coding parameters keep their default values.

Table 2: Comparisons of PSNR value (dB) under different QP values

Sequences	Original				Proposed			
	24	26	28	30	24	26	28	30
Akiyo	42.50	41.47	40.50	39.32	36.44	35.43	34.13	32.95
Bus	36.68	35.63	34.60	33.46	30.38	29.06	27.83	27.83
Coastguard	36.75	35.61	34.54	33.37	28.68	27.68	26.67	26.48
Container	38.84	37.54	36.41	35.21	31.57	29.40	29.87	27.90
Flower	36.75	35.59	34.42	33.08	32.48	30.96	29.93	28.17
Foreman	38.59	37.55	36.58	35.50	30.78	29.85	30.77	28.57
Mobile	36.21	35.05	33.90	32.59	30.80	29.39	27.81	26.47
News	40.96	39.81	38.70	37.43	33.91	32.32	30.89	30.29
Silent	38.79	37.50	36.37	35.12	29.84	27.36	28.34	27.36
Stefan	37.32	36. 30	35. 26	34.08	29.80	27.71	26.38	25.90
Tempete	36.80	35.75	34.70	33.54	30.17	29.01	28.02	27.06
Waterfal	38.01	36.69	35.50	34.20	29.80	27.96	27.35	28.49
Average	38.18	37.04	35.95	34.74	31.22	29.67	28.99	28.12

3.1 Embedding payload

In this subsection, embedding payloads on 12 video sequences under different values of QP are shown in Tab. 1. In the following, according to Tab. 1 some analysis of embedding payload will be addressed in details.

As shown in Tab. 1, for each video sequence, the embedding payloads are decreasing with the increase of QP value. For instance, the embedding payloads on Akiyo are 6704, 6314, 5654, 5245, 4806, 4267 and 3777 bits corresponding to the QP values from 24 to 30. Clearly, each video sequence can provide similar embedding payloads like this. Actually, the average values of embedding payload meets this. What's more, from Tab. 1 we can observe that the more embedding payloads the video sequence has, more significant details it has. For example, Mobile obtains the embedding payloads 26333, 26367, 25951, 25713, 25363, 24754 and 24000 bits when QP is set from 24 to 30. Moreover, Waterfall also obtains large embedding payloads, i.e., 23388, 22066, 19257, 17130, 14791, 11634 and 9148 bits when setting the same QP values. In summary, the

video sequences with more texture can provide more embedding payload and with the increasing of QP values the embedding payloads on each video sequence is decreasing.

3.2 Visual quality

In this subsection, 7 marked frames of Flower are first given as Fig. 2 to subjectively evaluate the visual quality of marked video sequences. They are set to different QP values from 24 to 30. Obviously, no significant visual distortion can been observed, although there must exist distortion compared with the original video sequence. Next, PSNR and

Table 3: SSIM values after and before data hiding under different QP values

Sequences	Original				Proposed			
	24	26	28	30	24	26	28	30
Akiyo	0.98	0.98	0.97	0.97	0.97	0.97	0.96	0.95
Bus	0.97	0.96	0.95	0.94	0.95	0.93	0.92	0.91
Coastguard	0.96	0.95	0.93	0.91	0.92	0.90	0.88	0.87
Container	0.95	0.94	0.92	0.91	0.94	0.92	0.91	0.89
Flower	0.99	0.98	0.98	0.97	0.98	0.97	0.97	0.96
Foreman	0.96	0.95	0.94	0.93	0.94	0.93	0.92	0.91
Mobile	0.98	0.98	0.97	0.97	0.96	0.96	0.94	0.92
News	0.98	0.97	0.97	0.96	0.96	0.95	0.95	0.94
Silent	0.96	0.95	0.94	0.93	0.93	0.90	0.90	0.89
Stefan	0.98	0.98	0.98	0.97	0.95	0.94	0.93	0.93
Tempete	0.98	0.98	0.97	0.96	0.95	0.94	0.93	0.92
Waterfall	0.97	0.96	0.95	0.93	0.97	0.89	0.88	0.88
Average	0.97	0.96	0.95	0.94	0.94	0.93	0.92	0.91

SSIM are exploited to objectively measure visual quality of video sequences. Both PSNR and SSIM before and after embedding data into video sequences are given as Tabs. 2 and 3. It is noted that the value of QP is respectively set to 24, 26, 28 and 30 in this subsection, which is corresponding to Tab. 1.

From Tab. 2, we can know that the maximum original PSNR value 42.50 dB is obtained on the video sequence Akiyo when the value of QP is fixed at 24. However, the minimum original PSNR value 32.59 dB is obtained on the video sequence Mobile under such case the value of QP is set to 30. Meanwhile, the video sequence Akiyo also provides the maximum marked PSNR value 36.44 dB when QP=24. However, the video sequence Stefan provides the minimum marked PSNR value when QP=30. Alternatively, as seen from Tab. 3, the maximum original SSIM value is 0.99 and it can be obtained on the video sequence Flower when QP=24. And the minimum original SSIM value 0.91 can be provided by the video sequences Coastguard and Container at the same time. It is noted that under these cases the values of QP are set to 30. For the marked SSIM values, the maximum value and the minimum value are obtained on the video sequences Flower and Coastguard, respectively.

Furthermore, the maximum value and the minimum value are corresponding to QP=24 and QP=30. Combined with Tab. 2, the maximum difference between the original PSNR value and the marked PSNR value at the same QP value is about 10 dB. For SSIM, however, the maximum difference between the original value and the marked value is 0.07 and it can be obtained on the video sequence Waterfall when QP=26 and QP=28. Therefore, although the degradation of PSNR is significant, the degradation of SSIM is less.

3.3 Bit-rate variation

Bit-rate after and before embedding data into video sequences is often used to evaluate the coding efficiency of H.264/AVC codec. Thus, Tab. 4 is given and used for evaluation. As seen in Tab. 4, also corresponding to Tab. 1, different video sequences have different bit-rates. Obviously, under the same condition the significant video sequences lead to much larger bit-rate values using the data hiding encoder. For instance, the video sequences Bus, Flower and Mobile, with more texture, have indeed much larger bit-rate values when compared with the video sequences Akiyo, Container, News and Silent, with more smooth areas. For each video sequence, the bit-rate value become less and less with the increasing of QP since video sequences are compressed more thoroughly when the video encoder has larger QP value. Moreover, the average value of bit-rate increase can be exploited to denote the impact caused by the data hiding. According to Tab. 4, average values of bit-rate increase are respectively 18.20, 15.60, 14.14 and 11.61 corresponding to QP=24, 26, 28 and 30. Given the original bit-rate values, the growth is less. That is to say, the proposed scheme has less impact, in terms of bit-rate variation, on the video encoder.

Table 4: Bit-rate variations under different QP values

Sequences	Original				Proposed			
	24	26	28	30	24	26	28	30
Akiyo	225.41	179.61	148.17	123.31	230.67	184.00	151.89	126.12
Bus	1652.68	1371.54	1165.55	1000.64	1676.80	1392.87	1183.76	1014.59
Coastguard	1548.95	1237.32	999.34	804.74	1573.08	1256.41	1014.29	814.78
Container	508.16	378.10	292.35	226.99	518.96	387.95	301.33	234.86
Flower	2108.11	1791.14	1537.05	1348.75	2125.77	1809.20	1555.32	1366.76
Foreman	765.05	608.14	489.87	410.17	778.53	613.09	499.26	417.08
Mobile	2115.94	1750.95	1468.00	1260.49	2138.95	1774.20	1491.22	1283.01
News	441.72	363.60	307.15	261.31	450.06	371.01	313.19	266.74
Silent	573.52	460.48	380.71	313.88	592.24	474.85	391.56	320.09
Stefan	1627.98	1336.88	1115.24	950.69	1646.41	1355.28	1133.85	969.26
Tempete	1675.58	1368.23	1134.49	958.20	1699.66	1391.07	1155.40	976.17
Waterfall	803.28	612.99	486.23	400.90	833.74	636.19	502.65	409.95
Average	1170.53	954.91	793.67	671.67	1188.73	970.51	807.81	683.28

3.4 Comparisons

The first 6 video sequences above-mentioned are used for comparisons. Kim et al.'s scheme

[Kim and Kang (2018)] is adjusted to compare with our proposed scheme. Here, their scheme is realized in luma component of macroblocks with intra-frame 4×4 prediction modes in intra frames and embeddable 4×4 blocks must satisfy the absolute values of DC quantized DCT coefficients are not less than 3.

Fig. 3 shows embedding payloads of the two schemes. Obviously, their embedding payloads on Akiyo, Bus and Container are very close. For Coastguard and Foreman, although embedding payloads are not close when QP values are different, under the same QP value embedding payloads of the two schemes are close.

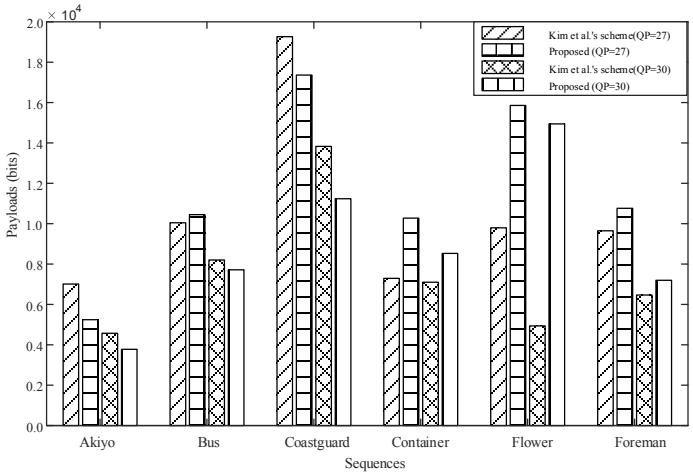

Figure 3: Embedding payloads of Kim et al.'s scheme and proposed scheme

For Flower, Kim et al.'s scheme has a significant difference in terms of embedding payloads when QP values are different. However, the proposed scheme has close embedding payloads. When comparing their embedding payloads, the proposed scheme outperforms Kim et al.'s scheme. In fact, both QP value and the texture of videos have an impact on embedding payloads of the two schemes. According to Kim et al.'s scheme, an embeddable 4×4 block must meet the values of its last 6 quantized DCT coefficients (in zig-zag scanning order, i.e., AC_{11}, AC_{12}, ..., AC_{16}) are zero. Moreover, Kim et al.'s scheme must also meet the absolute value of corresponding DC coefficient is not less than 3 in this paper. For the proposed scheme, the number of nonzero quantized DCT coefficients is not less than 7 in embeddable 4×4 blocks. Flower has many areas with complex texture that leads to easily satisfy the conditions of the proposed scheme. However, it cannot meet the conditions of Kim et al.'s scheme. Therefore, the proposed scheme can obtain larger embedding payload.

In the following, Figs. 4 and 5 show the comparisons of the SSIM and BIR, respectively. Here, BIR is defined by

$$BIR = \frac{Bitrate_{Marked} - Bitrate_{Original}}{Bitrate_{Original}} \times 100\% \tag{9}$$

where, $Bitrate_{Original}$ and $Bitrate_{Marked}$ denote the original bit-rate value and its corresponding bit-rate value caused by RDH. From Fig. 4, observe that although Kim et al.'s scheme has better SSIM values than the proposed scheme, their difference is very slight. For instance,

the maximum difference 0.03 is obtained on Coastguard. For other video sequences, they even have the same SSIM value under the same QP value. Clearly, from Fig. 5, Kim et al.'s scheme can obtain less BIR value on each video sequence than the proposed scheme. However, in fact BIR caused by the proposed scheme are also very slight even though they are larger than that of Kim et al.'s scheme. The maximum value of BIR is not over 3.5% shown in Fig. 5. Combined with Fig. 3, under the same setup, the proposed obtains larger embedding payloads. Therefore, considering novel elements or coding parameter not but motion vector and quantized DCT coefficients in video encoding standard maybe result in good DH/RDH schemes in the future.

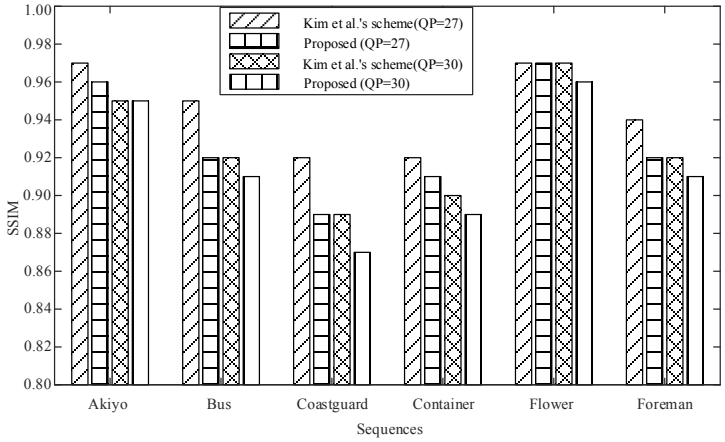

Figure 4: Comparisons of SSIM on six test videos

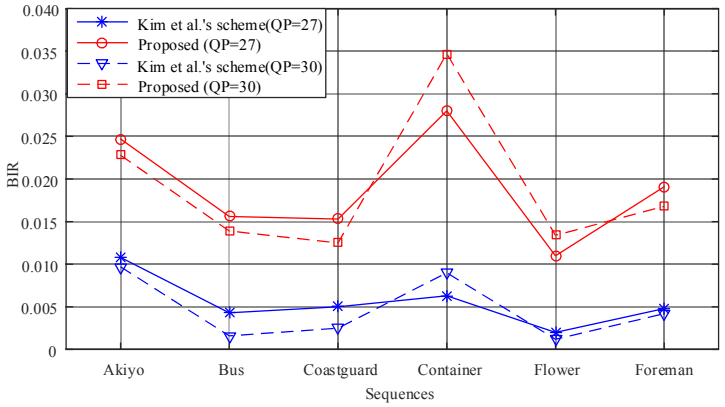

Figure 5: Comparisons of BIR on six test videos

4 Conclusions

This paper first introduces the idea of EMD to reduce the number of changed coefficients. Then, the structure of run-level coding in H.264/AVC is considered to combine with HM to propose a novel RDH method. Therefore, the proposed method can obtain good visual quality in terms of SSIM and result in less bit-rate increase. By the method presented in

this paper, not only motion vectors and coefficients but other coding parameters in video coding standards, such as H.264/AVC and HEVC, can be used for RDH in the future.

Acknowledgement: This work was supported by the National Natural Science Foundation of China (NSFC) under the grant No. 61972269, the Fundamental Research Funds for the Central Universities under the grant No. YJ201881 and Doctoral Innovation Fund Program of Southwest Jiaotong University under the grant No. DCX201824.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Chen, Y.; Wang, H.; Wu, H.; Chen, Y.; Liu, Y. (2018): A data hiding scheme with high quality for H.264/AVC video streams. *International Conference on Cloud Computing and Security*, pp. 99-110.

Chen, Y.; Wang, H.; Wu, H.; Liu, Y. (2018): An adaptive data hiding algorithm with low bitrate growth for H.264/AVC video stream. *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 20157-20175.

Du, Y.; Yin, Z.; Zhang, X. (2018): Improved lossless data hiding for JPEG images based on histogram modification. *Computers, Materials & Continua*, vol. 55, no. 3, pp. 495-507.

Filler, T.; Judas, J.; Fridrich, J. (2011): Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920-935.

Kim, H.; Kang, S. U. (2018): Genuine reversible data hiding technology using compensation for H.264 bitstreams. *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 8043-8060.

Ni, Z.; Shi, Y. Q.; Ansari, N.; Su, W. (2006): Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362.

Shi, Y. Q.; Li, X.; Zhang, X.; Wu, H. T.; Ma, B. (2016): Reversible data hiding: advances in the past two decades. *IEEE Access*, vol. 4, pp. 3210-3237.

Sullivan, G. J.; Ohm, J. R.; Han, W. J.; Wiegand, T. (2012): Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668.

Tew, Y.; Wong, K. (2014): An overview of information hiding in H.264/AVC compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 2, pp. 305-319.

Tian, J. (2003): Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896.

Wiegand, T.; Sullivan, G. J.; Bjontegaard, G.; Luthra, A. (2003): Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576.

Zhang, X.; Wang, S. (2006): Efficient steganographic embedding by exploiting modification direction. *IEEE Communications Letters*, vol. 10, no. 11, pp. 781-783.