

A Novel Video Reversible Data Hiding Algorithm Using Motion Vector for H.264/AVC

Ke Niu, Xiaoyuan Yang*, and Yingnan Zhang

Abstract: A novel reversible data hiding algorithm for H.264/AVC videos is proposed. Histogram Shifting (HS) of motion vector values is used efficiently in reversible data hiding to embed data. By designating a specific decoded reference frame, the distortion accumulation effects due to modification of the motion vectors is overcome. All the extracted information can be recovered without loss of the original compressed video carrier. The experimental results show that the proposed algorithm is simple, has higher capacity and invisibility than other existing schemes, and can adjust the relationship between capacity and invisibility according to embedded load.

Key words: reversible video data hiding; H.264/AVC; motion vector; histogram modification; modified reference frame

1 Introduction

According to the relationship between carrier and embedded message, information hiding can be divided into covert communication and digital watermarking. The basic principle of the two is the same, where digital information is embedded into a digital carrier, and both require good invisibility. But covert communication is concerned with the information used for embedding, and the digital carrier is not important since the carrier becomes useless after the information is extracted; on the contrary, digital watermarking requires the embedded watermark be suitable for the carrier. In general, the watermark needs to reflect some characteristics of the carrier.

The embedding of information, in most cases, will cause the carrier to lose part of the data, so the carrier cannot be completely recovered after extracting. In some special applications, such as in the fields of

medicine, military, and law, false positives would be caused even by slight distortion of a digital image. Therefore, any irreversible loss of carrier data is not allowed^[1]. Reversible Data Hiding (RDH) can recover the original data completely without any loss of information^[2], and is thus a lively research direction in this field. Since the first reversible data hiding algorithm was proposed for image authentication in 1997^[3], various RDH techniques have been widely used, ranging from image authentication^[3,4], medical image processing^[5], vector restoration in CAD^[6], cloud data coloring^[7], and so on. In recent years, a large number of RDH schemes have been proposed^[8]. Of these, the RDH algorithms based on Difference Expansion (DE)^[9] and Histogram Shifting (HS)^[10] have become the most representative two, which can promote the development of steganography technology. Many of the existing reversible schemes are based on the structure of these two algorithms^[8]. At present, most of the RDH algorithms are based on digital images hidden in the spatial domain or compressed domain. But with the continuous development of network and computer technology, digital video is becoming popular media data in both offline and online environments, and digital video acquisition is more convenient than before^[11]. It is important to consider ways to protect the videos from malicious use, efficient ways to search the desired

• Ke Niu, Xiaoyuan Yang, and Yingnan Zhang are with Key Laboratory of Network & Information Security of PAP, Engineering College of PAP, Xi'an 710086, China. E-mail: niuke@163.com; xyangwj@126.com; zyn583@163.com.

* To whom correspondence should be addressed.

Manuscript received: 2016-11-10; revised: 2017-03-04; accepted: 2017-03-07

videos in the database, reliable method to recover errors in video transmission, etc., for the future of digital video. To accomplish such tasks, video reversible data hiding provides one of the solutions^[12].

The H.264/AVC video coding standard has high compression efficiency. Because it is suitable for video data network transmission and other characteristics, it has been widely used. The RDH scheme based on H.264 video has attracted the attention of researchers^[11–13]. In Ref. [11], data are coded using BCH code and then embedded into the DCT coefficients of a 4×4 block of I-frame. This algorithm is more focused on the performance robustness. In Ref. [13], in order to improve the embedding capacity, randomly select two DCT coefficients that satisfy condition in 4×4 luma blocks. Then the information is embedded by 2-D histogram modification, and hiding capacity is obtained with the good invisibility. In Ref. [14], motion compensation is used to embed information by inter-frame data error expansion (EP), which achieves high embedding capacity while minimizing video distortion. These RDH algorithms are embedded in the DCT coefficients of the compressed video without using the motion vectors. Because the H.264 compression standard supports more flexible and smaller blocks (from 4×4 to 16×16), there are more motion vectors than in previous video compression standards. Meanwhile, for luma and chroma components in the H.264 compression standard, the vectors are estimated with $\frac{1}{4}$ and $\frac{1}{8}$ precision, respectively, this makes motion estimation more accurate. But the impact of the motion vector on the video quality is also greater than in previous standards. Therefore, in H.264 video, the motion vector is an ideal reversible watermarking modulation point, but at the same time, maintaining a large hiding capacity alongside the high visual quality is difficult.

In this paper, a new method of embedding information in motion vectors based on the histogram migration technique is proposed. The proposed algorithm has low computational cost, higher capacity, better imperceptibility, and higher security compared with the previous algorithms.

2 Preliminaries

2.1 Motion compensation and motion vectors

The H.264 standard supports motion vectors of

different block sizes and finer subpixels (the luminance component is $\frac{1}{4}$ pixel resolution) compared to earlier compression standards. Each macroblock can be divided into four modes: 16×16, 16×8, 8×16, and 8×8. If the macroblock is divided into 8×8 mode, then each 8×8 sub-macroblock can be divided into four modes: 8×8, 8×4, 4×8, and 4×4, as shown in Fig. 1^[15]. Each block is individually motion-compensated, and contains a motion vector.

Tables 1 and 2 show the motion vector statistics of the YUV video sequence with 150 frames in qcif (176×144) format. Luminance of the frames is encoded with H.264 basic level and MPEG2, respectively. In the tables, mvx and mvy denote the horizontal and vertical components of the motion vector, respectively. In the encoding process, the first frame is I-mode and the

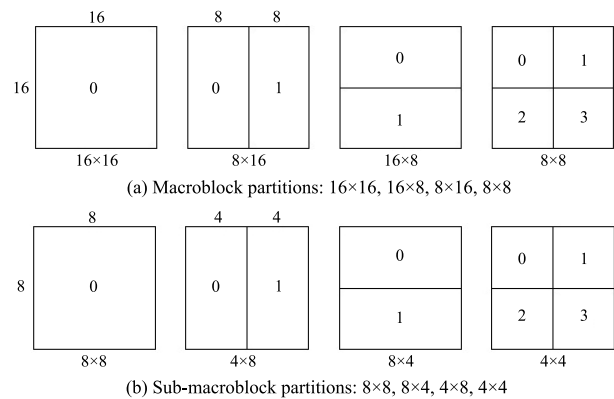


Fig. 1 The partition modes of macro-block and sub-macro-block in H.264.

Table 1 The number of motion vectors in MPEG2 videos.

Video sequence	News	Foreman	Mobile	Salesman
mvx=0	13 973	9791	14 692	14 296
mvy=0	13 498	7949	12 976	13 921
mvx=±1	440	3183	44	405
mvy=±1	794	4200	1701	526
Other mvx	338	1777	15	50
Other mvy	459	2602	74	304
Sum	14 751	14 751	14 751	14 751

Table 2 The number of motion vectors in H.264 videos.

Video sequence	News	Foreman	Mobile	Salesman
mvx=0	17 750	19 988	9077	11 650
mvy=0	17 313	19 517	9469	8439
4mvx=±1	4278	3546	9300	14 527
4mvy=±1	3127	3784	7940	12 566
Other mvx	2821	2291	14 344	6108
Other mvy	4409	2524	15 312	11 283
Sum	24 849	25 825	32 721	32 285

remaining frames are all P-mode. It can be seen that the H.264 encoding scheme contains more motion vectors, and that the number of motion vectors varies with video content. The videos which have rich texture contain more blocks and more motion vectors. If the motion vector is used as the modulation point to embed the information, an H.264 compressed video with complex texture, rich details, and fast motion can obtain a higher hiding capacity.

2.2 Motion vector histogram migration

According to Section 2.1, the H.264 compressed bit stream has more motion vectors than previous compression standards, so it is suitable to use motion vectors as embedding cover of reversible watermark. Ni et al.^[10] proposed an RDH scheme for HS to perform a wide range of applications. A large number of RDH schemes based on HS are proposed. These extend from the spatial domain to the transform domain and from image carrier to video carrier, and the performance and speed of the algorithm are improved as well^[11, 13, 14, 16, 17]. As for the compressed video, embedded information gained by modifying the motion vector allows for a great distortion of the contents, so most of algorithms are based on the histogram change of DCT coefficients.

For the vast majority of video sequences, most of the motion vectors are 0, which is the peak of the vector histogram. Figures 2 and 3 show the vector histograms for the foreman and news video clips, respectively, with each encoded by MPEG2 and H.264. Since H.264 supports the motion vector of the luminance component

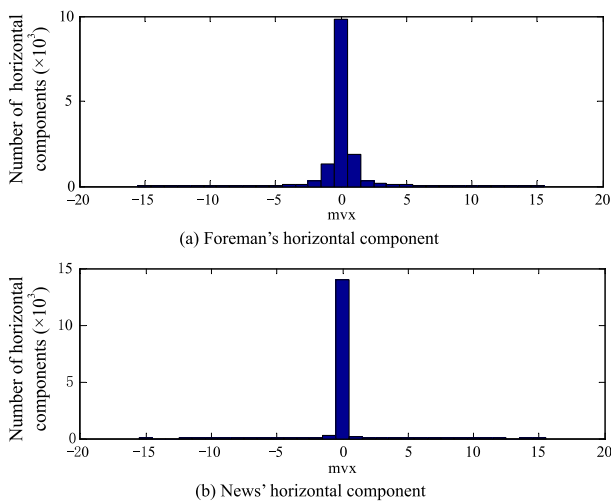


Fig. 2 Motion vector horizontal component histograms of foremans and news in MPEG2.

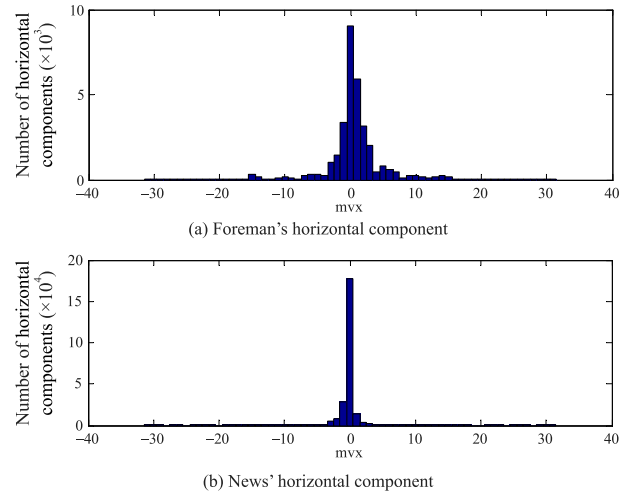


Fig. 3 Motion vector horizontal component histograms of foreman and news in H.264.

at $\frac{1}{4}$ pixel precision, its histogram is more smooth than the MPEG2 vector histogram. That is to say, compared with MPEG2, H.264's vector histogram has more ± 1 and less 0 points. But in general, most of the motion vector amplitude (H.264 is 4 times that of the motion vector amplitude) are concentrated at 0 and ± 1 .

If the peak value and the zero point in the motion vector histogram of the video are searched for using as the algorithm in Ni et al.^[10], the modified motion vector will be too much, which will cause great distortion of the video content. Analysis of Fig. 3 shows that the H.264 histogram distribution is relatively smooth while in the histogram occupies a larger proportion. Considering the hiding capacity and invisibility, the motion vectors in point of the sub-peak (± 1 point) in Fig. 3 are embedded. The histogram is moved as the algorithm of Ni et al.^[10] For example, the horizontal component is modified as shown in Eq. (1).

$$mvx' = \begin{cases} mvx, & \text{if } mvx = 0; \\ \frac{4mvx + \text{sign}(mvx)}{4}, & \text{if } |4mvx| > 1; \\ \frac{4mvx + \text{sign}(mvx)b}{4}, & \text{if } |4mvx| = 1 \end{cases} \quad (1)$$

In the equation, mvx is the horizontal component of the motion vector, mvx' is the horizontal component of the modified motion vector, and b is the binary message to be embedded.

Figure 4 shows the histogram after vector's removal. The embedding capacity of the watermark depends on the number of motion vectors at the ± 1 point. The more complex and fast moving videos, there will be larger numbers of motion vectors within each ± 1 point and be

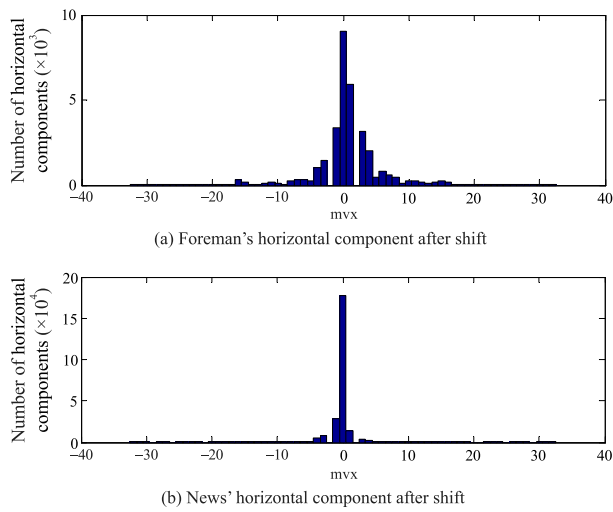


Fig. 4 Motion vector horizontal component histograms' shift of foreman and news in H.264.

greater embedding capacity.

2.3 Reference frame transformation

By modifying the ± 1 point, the change of the 0 vector is avoided, and it occupies the largest proportion in the histogram. However, in the H.264 compression scheme, decoding of the current frame depends on the previous one. The modification of the motion vector will lead to error accumulation since the first frame uses I-mode coding and the rest use P-mode. With the increase in the number of P-mode frames, the visual quality of the video frames will continue to decline, as shown in Fig. 5.

The encoded image in H.264 is stored in the reference buffer of the encoder, and the decoder has a corresponding reference image list for motion-compensated inter-frame prediction^[15]. Unlike the previous standards which only have a single reference frame, it uses prediction techniques on multiple reference frames which allows one or two frames to be selected out of up to 15 frames. In order to effectively prevent error propagation and accumulation caused by motion vector modification, the selected reference frame is not embedded information; that is, its original motion vector value is maintained by using the selectable characteristics of the reference frame. The reference frame is used for several subsequent frame decodings. Since these particular frames and their reference frames are clean, the accumulation of errors is truncated. With a change of the reference frame, the error is limited to the embedded frame, which effectively controls the quality of the video.

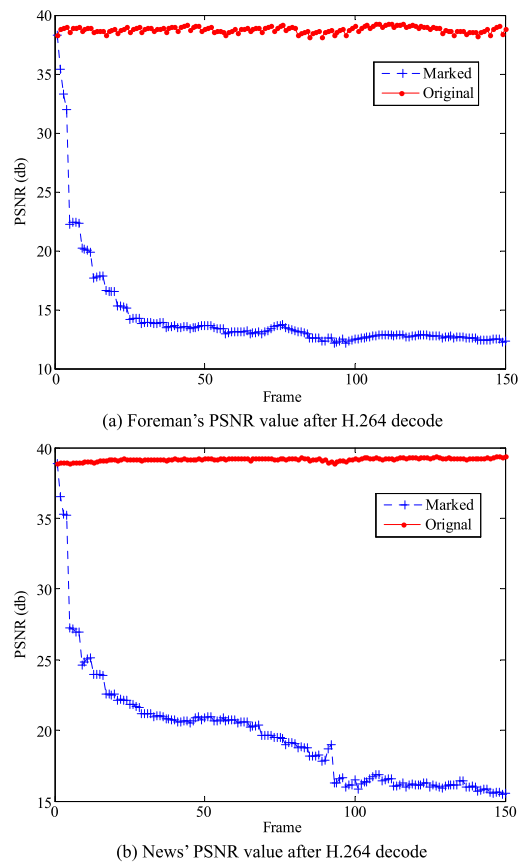


Fig. 5 PSNR values of original and marked motion vectors (foreman and news).

Figure 6 is a schematic diagram of the reference frame before and after the change. The normal coding will be the smallest residual frame as a reference frame codec, and to effectively control the error diffusion, the provisions of the reference frame transformation are used.

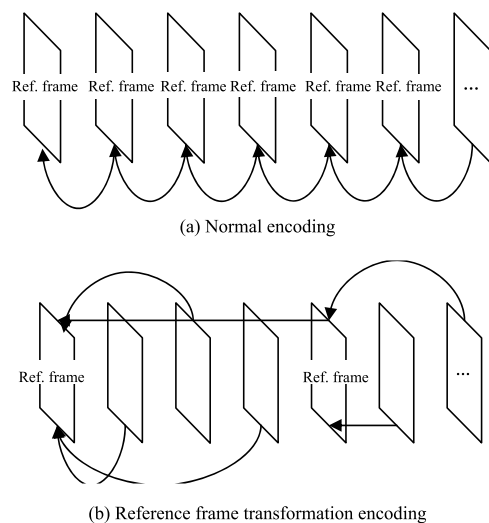


Fig. 6 Reference frames before and after the change.

Figure 7 shows the scenario where a specific reference frame is selected for every four frames, and then the information is embedded into the horizontal or vertical components of the remaining frames. It can be seen in Fig. 7 that using this error accumulation control method greatly improves the videos' quality, more so than methods that modify the motion vector directly. Due to fast motion and rich texture videos having more motion vectors with values of ± 1 , the average PSNR is lower than that in slow motion and simple texture videos. This is consistent with the experimental results.

3 Reversible Data Hiding Algorithm

A novel video steganography scheme proposed in this paper embeds the information in the H.264 video stream of the modified reference frame. The embedded video can be decoded, and it can be reconstructed in lossless format using the secret key. This sketch is shown in Fig. 8. The H.264 code stream of the reference frame can be obtained using the raw video coding, or the existing H.264 video can be recoded using the lossless H.264 coding method^[18, 19].

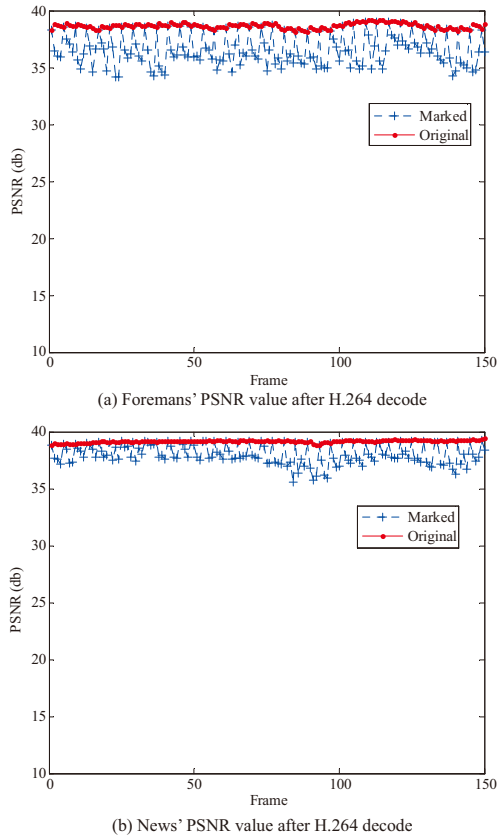


Fig. 7 PSNR values of original and marked motion vectors with reference frame transformation (foreman and news).

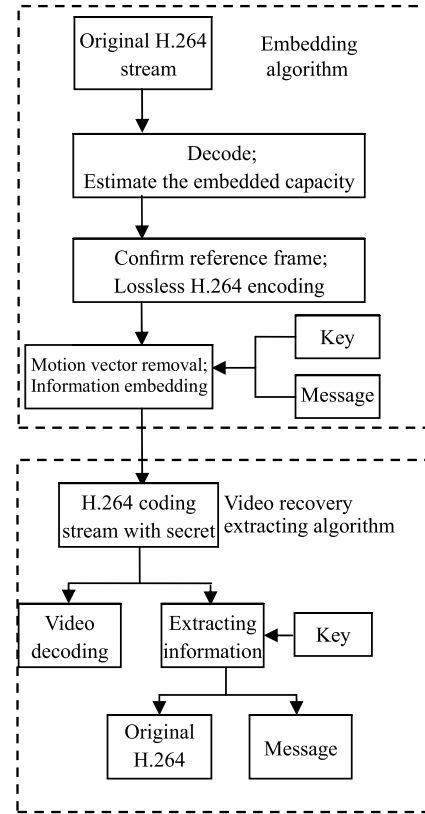


Fig. 8 Framework of our algorithm.

3.1 Embedding algorithm

Step 1 The algorithm embeds the information in the H.264 compressed video in which reference frames are specified. There are two ways to preprocess the video:

- (1) Encode the YUV sequence by H.264 according to the required video quality.
- (2) Decode the H.264 code stream and then encode by the lossless H.264.

In the coding process, the reference frame interval K is determined, and the embedding capacity C is estimated in the coding process. Compare the number of bits B to be embedded and the capacity C , and then adjust the size of K . If $\text{mod}(i, K) = 0$, then f_i is the reference frame of f_{i+1} to f_{i+K} .

Step 2 Randomly generate a binary key “Key”, with the length of “Key” being equal to the number of embedded frames N .

Step 3 Let \mathbf{mv}_{ij} be the j -th motion vector of the i -th frame, \mathbf{mv}_{xij} its horizontal component, and \mathbf{mv}_{yij} its vertical component. The current operation frame is not I-frame, and $b \in \{0, 1\}$ is the message bit we want to hide. Embedding of the message by vector histogram removal is achieved using Eqs. (2) and (3).

$$mvx'_{ij} = \begin{cases} mvx_{ij}, & \text{if } (mvx_{ij} = 0, \text{ or } \\ & \text{mod}(i, K) = 0); \\ \frac{4mvx_{ij} + \text{sign}(mvx_{ij})}{4}, & \text{if } (|4mvx_{ij}| > 1 \text{ and } \\ & \text{Key}(i) = 1); \\ \frac{4mvx_{ij} + \text{sign}(mvx_{ij})b}{4}, & \text{if } (|4mvx_{ij}| = 1 \text{ and } \\ & \text{Key}(i) = 1) \end{cases} \quad (2)$$

$$mvy'_{ij} = \begin{cases} mvy_{ij}, & \text{if } (mvy_{ij} = 0 \text{ or } \\ & \text{mod}(i, K) = 0); \\ \frac{4mvy_{ij} + \text{sign}(mvy_{ij})}{4}, & \text{if } (|4mvy_{ij}| > 1 \\ & \text{and Key}(i) = 0); \\ \frac{4mvy_{ij} + \text{sign}(mvy_{ij})b}{4}, & \text{if } (|4mvy_{ij}| = 1 \\ & \text{and Key}(i) = 0) \end{cases} \quad (3)$$

Step 4 Encode the motion vector after embedding the message to form the H.264 stream.

3.2 Extracting algorithm and video recovery

Step 1 Read the H.264 current decoding frame. If it is not I-frame, then jump to Step 2; otherwise, continue to the next frame.

Step 2 Extract the message according to Eq. (4), mark mv'_{ij} the j -th motion vector of the i -th frame, with mvx'_{ij} as its horizontal component, and mvy'_{ij} as its vertical component. b' is the current extracted bit.

$$b' = \begin{cases} 0, & \text{if } (((\text{Key}(i) = 1 \text{ and } |4mvx'| = 1) \\ & \text{or } (\text{Key}(i) = 0 \text{ and } |4mvy'| = 1)) \\ & \text{and } (\text{mod}(i, K) \neq 0)); \\ 1, & \text{if } (((\text{Key}(i) = 1 \text{ and } |4mvx'| = 2) \\ & \text{or } (\text{Key}(i) = 0 \text{ and } |4mvy'| = 2)) \\ & \text{and } (\text{mod}(i, K) \neq 0)) \end{cases} \quad (4)$$

Step 3 Restore the original video, using mv''_{ij} for resumption of the j -th motion vector of the i -th frame, mvx''_{ij} for its horizontal component, and mvy''_{ij} is its vertical component. Operate according to Eqs. (5) and (6).

$$mvx''_{ij} = \begin{cases} mvx'_{ij}, & \text{if } (mvx'_{ij} = 0 \\ & \text{or Key}(i) = 0 \text{ or } \\ & \text{mod}(i, K) = 0 \text{ or } \\ & |4mvx'_{ij}| = 1); \\ \frac{4mvx'_{ij} - \text{sign}(mvx'_{ij})}{4}, & \text{if } (|4mvx'_{ij}| \geq 2 \\ & \text{and Key}(i) = 1) \end{cases} \quad (5)$$

$$mvy''_{ij} = \begin{cases} mvy'_{ij}, & \text{if } (mvy'_{ij} = 0 \\ & \text{or Key}(i) = 1 \text{ or } \\ & \text{mod}(i, K) = 0, \text{ or } \\ & |4mvy'_{ij}| = 1); \\ \frac{4mvy'_{ij} - \text{sign}(mvy'_{ij})}{4}, & \text{if } (|4mvy'_{ij}| \geq 2 \\ & \text{and Key}(i) = 0) \end{cases} \quad (6)$$

Step 4 Restore the original video stream using the extracted vector mv''_{ij} .

4 Experimental Results and Analysis

Experiments were performed using the MATLAB H.264 codec as described in Refs. [20, 21]. The carrier video will be obtained by recoding lossless H.264 compression, or by encoding the YUV video sequence as required. In order to fully reflect the performance of the algorithm proposed in this paper, the YUV video sequence is compressed directly to get the cover H.264 video.

In the experiment, we take 8 QCIF (176×144) video sequences: city, news, carphone, mobile, foreman, container, salesman, and coastguard. For each video sequence we take 150 frames and operate using the luminance components of each. Except for the first I-frame, the rest are P-frame mode, and the quality factor QP is 27. MATLAB 2012 is used for simulation. The embedding message is a binary sequence, and the secret key Key is a pseudorandom binary sequence whose length is equal to the number of P-frames which will be embedded. The reference frame interval is K . Hidden capacity is shown in Table 3, and PSNR values are shown in Table 3 and Fig. 9.

Table 3 shows the embedding capacity and the PSNR values when the algorithm proposed in this paper takes different K values. The embedding capacity here shows significant improvement compared with Liu et al.'s scheme^[11]. It can be seen that the videos with faster motion and complex textures have larger PSNR decreases, because they contain more nonzero motion vectors, such as in coastguard and mobile; on the contrary, videos such as container and salesman have lower PSNR value decreases.

The PSNR of each frame in Fig. 9 varies with the period of the reference frame, and the erroneous accumulation effect of the motion vector is effectively eliminated by designating the reference frame. Compared with Liu et al.'s scheme^[11], the

Table 3 The embedding capacity and PSNR values of our algorithm.

Video sequence	Algorithm		PSNR (db)	PSNR decrease (%)	Capacity (bit)
News	Original H264		39.15	–	
	Proposed scheme	$K = 2$	38.27	2.25	1125
		$K = 3$	38.24	2.32	2882
		$K = 4$	38.04	2.84	5130
		$K = 5$	37.80	3.45	7712
		$K = 6$	37.76	3.55	10346
	Ref. [11]		35.48	9.37	261–1274
Mobile	Original H264		36.58	–	
	Proposed scheme	$K = 2$	33.86	7.44	5604
		$K = 3$	32.68	10.66	11747
		$K = 4$	32.08	12.30	17521
		$K = 5$	31.61	13.59	23067
		$K = 6$	31.28	14.49	28273
	Ref. [11]		30.11	17.69	288–1771
City	Original H264		38.02	–	
	Proposed scheme	$K = 2$	37.47	1.45	1803
		$K = 3$	36.77	3.29	4331
		$K = 4$	36.28	4.58	6288
		$K = 5$	35.99	5.34	8518
		$K = 6$	35.79	5.87	10670
	Ref. [11]		33.86	10.94	253–1455
Carphone	Original H264		39.47	–	
	Proposed scheme	$K = 2$	38.70	1.95	3238
		$K = 3$	38.05	3.60	7617
		$K = 4$	37.83	4.16	12207
		$K = 5$	37.59	4.76	16747
		$K = 6$	37.48	5.04	21466
	Ref. [11]		36.13	8.45	298–1753
Salesman	Original H264		37.79	–	
	Proposed scheme	$K = 2$	37.76	0.08	1691
		$K = 3$	37.58	0.56	4056
		$K = 4$	37.42	0.98	6862
		$K = 5$	37.35	1.16	9953
		$K = 6$	37.19	1.59	13421
	Ref. [11]		35.33	6.51	447–1414
Coastguard	Original H264		37.00	–	
	Proposed scheme	$K = 2$	35.95	2.84	3188
		$K = 3$	35.26	4.70	7273
		$K = 4$	34.91	5.65	12156
		$K = 5$	34.63	6.41	16913
		$K = 6$	34.27	7.38	22207
	Ref. [11]		32.72	11.57	304–1152
Container	Original H264		38.46	–	
	Proposed scheme	$K = 2$	38.27	0.49	2136
		$K = 3$	38.04	1.09	5543
		$K = 4$	37.78	1.77	9549
		$K = 5$	37.63	2.16	13530
		$K = 6$	37.49	2.52	17608
	Ref. [11]		35.61	7.41	241–1173
Foreman	Original H264		38.57	–	
	Proposed scheme	$K = 2$	37.59	2.54	3752
		$K = 3$	36.89	4.36	8210
		$K = 4$	36.49	5.39	12679
		$K = 5$	36.24	6.04	16940
		$K = 6$	36.04	6.56	21093
	Ref. [11]		35.42	8.17	332–1325

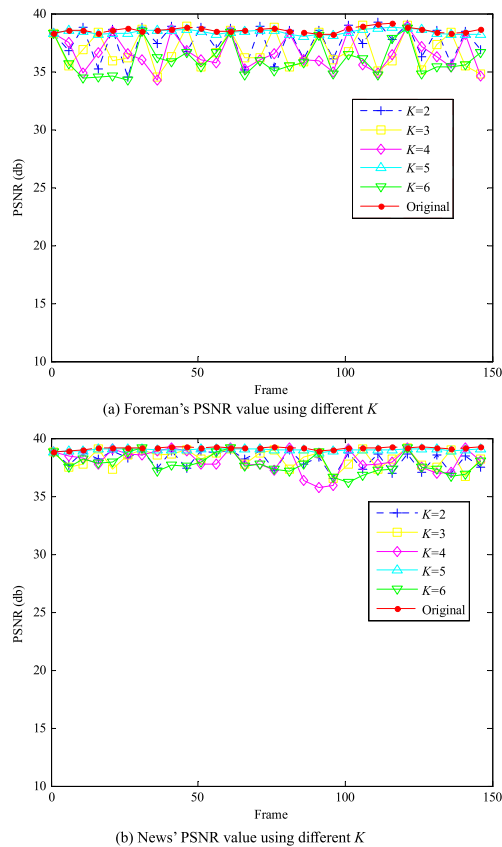


Fig. 9 PSNR values of news and foreman using different K .

invisibility slightly improves.

The security of the embedded message is controlled by the secret key, and only the correct key can extract the message to restore the original video sequence. Otherwise, the operation cannot be completed. Table 4 shows the PSNR value of recover video sequence. When the incorrect secret key is restored, the videos recovered PSNR values are significantly reduced; the video is only recovered completely when the correct secret key is employed. The low PSNR value error is determined by the H.264 compression, so it does not affect the video quality. Figure 10 shows the PSNR for each frame after recovering the video with the correct secret key and the incorrect one, respectively. It can be seen that only the correct secret key can fully recover the video. Table 5 shows the relevance of the extracted message to the original message with using correct key and the incorrect one, respectively. When the information is extracted with the incorrect key, the message is almost completely unrelated to the original.

To further evaluate the bitrate change of the proposed scheme, bitrate variation BR_var caused by data hiding is introduced, as shown in Eq. (7).

Table 4 PSNR value of recovered video sequence.

Video sequence	Key	PSNR value (db)					
		Orig.	$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$
News	Y	39.14	39.07	39.03	39.12	39.05	39.11
	N		38.42	38.46	38.27	38.07	38.18
Mobile	Y	36.58	36.60	36.60	36.59	36.58	36.56
	N		35.40	34.00	34.02	33.03	32.42
City	Y	38.02	38.85	38.43	38.28	37.95	37.91
	N		37.76	37.02	36.73	36.17	35.92
Carphone	Y	39.47	39.74	39.64	39.57	39.51	39.44
	N		38.94	38.60	38.09	38.08	37.72
Salesman	Y	37.79	37.97	37.93	37.85	37.87	37.78
	N		37.83	37.63	37.46	37.47	37.32
Coastguard	Y	37.00	37.14	37.07	37.02	36.95	36.95
	N		36.40	36.00	35.45	35.12	35.27
Container	Y	38.46	38.58	38.56	38.52	38.46	38.46
	N		38.12	38.01	38.00	38.00	37.71
Foreman	Y	38.57	38.83	38.73	38.64	38.56	38.54
	N		37.83	37.49	36.98	36.49	36.92

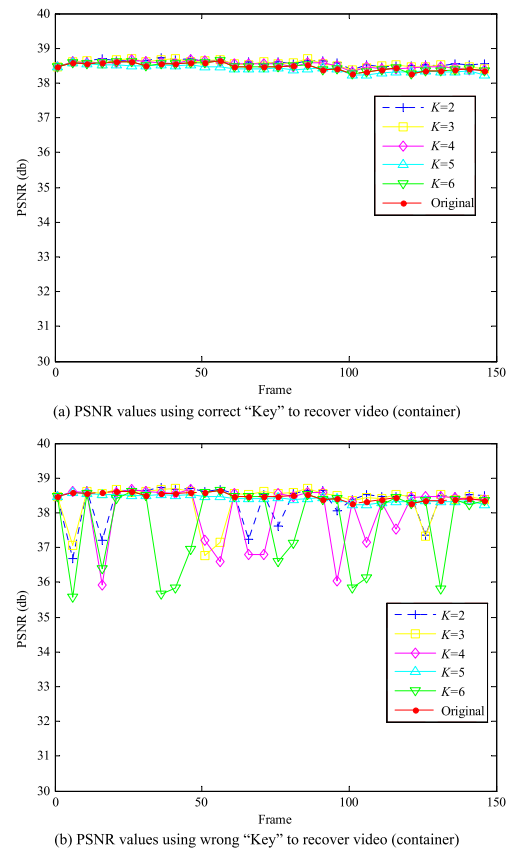


Fig. 10 PSNR values of recovered container video using different K .

$$BR_var = \frac{BR_em - BR_orig}{BR_orig} \times 100\% \quad (7)$$

In this formula, BR_em is the bitrate generated by

Table 5 Correlation coefficient of extracting information and original information.

Video sequence	Key	Correlation coefficient				
		$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$
News	Y	1	1	1	1	1
	N	0.0222	-0.0099	0.0073	0.0157	-0.0003
Mobile	Y	1	1	1	1	1
	N	-0.0071	-0.0236	0.0034	0.0210	-0.0100
City	Y	1	1	1	1	1
	N	0.0053	0.0130	0.0224	0.0025	0.0116
Carphone	Y	1	1	1	1	1
	N	0.0208	0.0160	0.0124	0.0043	0.0049
Salesman	Y	1	1	1	1	1
	N	0.0217	0.0179	-0.0054	0.0016	0.0108
Coastguard	Y	1	1	1	1	1
	N	0.0083	-0.0020	0.0096	-0.0018	-0.0022
Container	Y	1	1	1	1	1
	N	0.0390	0.0217	-0.0028	0.0030	0.0075
Foreman	Y	1	1	1	1	1
	N	0.0143	-0.0117	-0.0038	-0.0097	0.0183

Table 6 Bitrate variation using different K values.

Video sequence	Bitrate increase (%)				
	$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$
News	10.83	18.88	26.10	33.03	35.58
Mobile	5.21	8.82	11.67	13.59	15.69
City	6.57	24.20	38.42	45.11	55.10
Carphone	9.63	16.34	21.80	28.04	31.55
Salesman	7.79	16.92	22.75	28.96	34.43
Coastguard	17.11	30.17	38.71	48.31	53.35
Container	6.01	15.65	22.80	29.98	34.57
Foreman	14.5	26.39	37.00	45.26	51.51

the data embedding encoder, and BR_orig is the bitrate generated by the original encoder. Table 6 shows the bitrate variation using different K values. For the same video, the bitrate increases with the increase of K . Out of all the videos, the fast motion videos, such as mobile and container show smaller bitrate increases than the slow motion videos.

5 Conclusion

The video RDH algorithm proposed in this paper has computational cost, and can be adjusted according to actual needs to balance capacity and invisibility. The security of the algorithm depends on the secret key. Only when the correct secret key is obtained can the message be retrieved, and the video be recovered. The length of the key can be controlled according to need. The maximum length can be equal to the number of embedded information frames, and the security is

high. At present, the algorithm is only applicable to the video steganography of the plaintext domain, and there is a large change in the bitrate of information embedding. The next step is to design a suitable video encryption scheme^[22] to realize the RDH algorithm of the encryption domain, and to optimize the motion vector modification to reduce the bitrate.

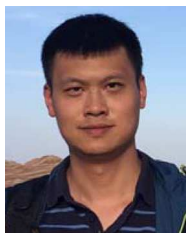
Acknowledgment

This research was supported by the National Natural Science Foundation of China (Nos. 61379152 and 61403417).

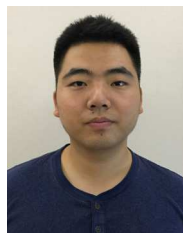
References

- [1] O. M. Al-Qershi and B. E. Khoo, Two-dimensional difference encryption (2d-de) scheme with a characteristics-based threshold, *Signal Processing*, vol. 93, no. 1, pp. 154–162, 2013.
- [2] A. M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.
- [3] J. M. Barton, Method and apparatus for embedding authentication information within digital data, US Patent US5646997, 1997.
- [4] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel, Lossless recovery of an original image containing embedded data, US Patent US6278791, 2001.
- [5] G. Coatrieux, C. L. Guillou, J. M. Cauvin, and C. Roux, Reversible watermarking for knowledge digest embedding and reliability control in medical images, *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 2, pp. 158–165, 2009.
- [6] F. Peng, Y. Z. Lei, M. Long, and X. M. Sun, A reversible watermarking scheme for two-dimensional CAD engineering graphics based on improved difference expansion, *Computer-Aided Design*, vol. 43, no. 8, pp. 1018–1024, 2011.
- [7] K. Hwang and D. Li, Trusted cloud computing with secure resources and data coloring, *IEEE Internet Computing*, vol. 14, no. 5, pp. 14–22, 2010.
- [8] Y. Q. Shi, X. Li, X. Zhang, H. T. Wu, and B. Ma, Reversible data hiding: Advances in the past two decades, *IEEE Access*, vol. 4, pp. 3210–3237, 2016.
- [9] J. Tian, Reversible data embedding using a difference expansion, *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.
- [10] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, Reversible data hiding, *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.
- [11] Y. Liu, L. Ju, M. Hu, X. Ma, and H. Zhao, A robust reversible data hiding scheme for h.264 without distortion drift, *Neurocomputing*, vol. 151, no. 1, pp. 1053–1062, 2015.
- [12] K. S. Wong, K. Tanaka, K. Takagi, and Y. Nakajima, Complete video quality-preserving data hiding, *IEEE*

- Transactions on Circuits & Systems for Video Technology*, Vol. 19, no. 10, pp. 1499–1512, 2009.
- [13] J. Zhao, Z. T. Li, and B. Feng, A novel two-dimensional histogrammodification for reversible data embedding into stereo H.264video, *Multimedia Tools & Applications*, vol. 75, no. 10, pp. 5959–5980, 2016.
- [14] C. Vural and B. Barakli, Reversible video watermarking using motion compensated frame interpolation error expansion, *Signal Image & Video Processing*, vol. 9, no. 7, pp. 1613–1623, 2015.
- [15] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. John Wiley & Sons Ltd., UK, 2003, p. 171.
- [16] M. Fallahpour and M. H. Sedaaghi, High capacity lossless data hiding based on histogram modification, *Ieice Electronics Express*, vol. 4, no. 7, pp. 205–210, 2007.
- [17] S. Lee, Y. Suh, and Y. Ho, Reversible image authentication based on watermarking, in *IEEE International Conference on Multimedia and Expo*, 2006, pp. 1321–1324.
- [18] G. Xuan, Y. Q. Shi, P. Chai, J. Teng, Z. Ni, and X. Tong, Optimum histogram pair based image lossless data embedding, in *International Workshop on Digital Watermarking*, 2007, pp. 264–278.
- [19] S. T. Wei, C. W. Tien, B. D. Liu, and J. F. Yang, Adaptive truncation algorithm for hadamard-transformed H.264/AVC lossless video coding, *International Workshop on Digital Watermarking*, vol. 21, no. 5, pp. 538–549, 2011.
- [20] A. A. Muhit, M. R. Pickering, M. R. Frater, and J. F. Arnold, Video coding using elastic motion model and larger blocks, *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 20, no. 5, pp. 661–672, 2010.
- [21] A. A. Muhit, M. R. Pickering, M. R. Frater, and J. F. Arnold, Video coding using fast geometry-adaptive partitioning and an elastic motion model, *Journal of Visual Communication & Image Representation*, vol. 23, no. 1, pp. 31–41, 2012.
- [22] M. Y. Shabir, A. Iqbal, Z. Mahmood, and A. Ghafoor, Analysis of classical encryption techniques in cloud computing, *Tsinghua Science and Technology*, vol. 21, no. 1, pp. 102–113, 2016.



Ke Niu received the MS degree from Engineering University of CAPF in 2007. He is currently an associate professor with Electronic Technology Department, Engineering University of CAPF. His research interests are information hiding and video processing.



Yingnan Zhang received the MS degree from Engineering University of CAPF in 2014. He is currently a PhD candidate in Key Laboratory of Network&Information Security of CAPF, Department of Electronic Technology in Engineering University of CAPF. His research interests are information hiding

and video processing.



Xiaoyuan Yang received the MS degree from Xidian University in 1991. He is currently a professor, PhD advisor in Engineering University of CAPF. His research interests are cryptography and information security.