Multimed Tools Appl (2018) 77:11979–12001
DOI 10.1007/s11042-017-4844-1

An efficient information hiding method based on motion vector space encoding for HEVC

Jie Yang[1,2] **· Songbin Li**[1,2]

Received: 18 November 2016 / Revised: 18 April 2017 / Accepted: 17 May 2017 /
Published online: 3 June 2017

Abstract As the newest video coding standard, high efficiency video coding (HEVC) has great potential as a new information hiding carrier. This paper proposes an efficient information hiding method based on motion vector space encoding for HEVC encoding process. In this method, the mapping relationship between motion vector set and the points in the motion vector space is defined. The motion vector components from the $N/2$ prediction units (PUs) with smallest size in a coding tree unit (CTU) are selected as the secret information carriers. Each N secret bits are converted to a $2N + 1$-ary number. By modifying at most one element in the set of N motion vector components, the mapping value of the set in the motion vector space can be equal to the $2N + 1$-ary number. In this way, information hiding is realized. Since at most one element is changed and the $N/2$ PUs with smallest size are selected, this method contributes to excellent transparency of steganography and anti-steganalysis performance with high embedding efficiency. To the best of our knowledge, this is the first information hiding method based on motion vector for HEVC. Experimental results verify that the proposed method is practicable and has better performance than two typical embedding rules of information hiding based on motion vector.

Keywords HEVC · Information hiding · Motion vector · Space encoding · Video coding

✉ Songbin Li
lisb@dsp.ac.cn

Jie Yang
yangjie13@mails.ucas.ac.cn

1 Institute of Acoustics, Chinese Academy of Sciences, Beijing, 100190, China

2 University of Chinese Academy of Sciences, Beijing, 100049, China

1 Introduction

Information hiding technology utilizes the redundancy of human sense organs on digital signals to embed important information in digital media carriers such as audio, images, video, etc., by certain algorithms. It can be divided into two major categories digital watermarking and steganography according to different application scenarios. The former embeds copyright information into the carriers to express the legitimate rights and interests of the content owner, and prevents the content from being tampered. It also can enable the authentication of captured contents to trace back information leakage. While the later embeds secret information in the carriers for covert communication. Video information hiding technology plays a very important role, because it can prevent video content being illegally spread, protect information integrity, monitor and record the video transmission and distribution.

There are two major categories of video information hiding methods based on different embedding occasions of secret information. One is embedding before encoding, which regards the video as a sequence of motion pictures, and hides the secret information into the video frame pixels according to certain intensity. Although this method is easy to implement, the embedded information may be lost after compression encoding. The other category is embedding during encoding, which embeds secret information in the coding process of image or video. Through combining certain aspects of the characteristics of coding standards and finding the appropriate information embedding position, we can change some intermediate results of the encoding process to achieve secret information embedding. The encoding embedding method has greater application value and has gained more attention because the video is usually transmitted or stored after compression coding. At present, the popular video coding standards H.26X and MPEG-X both have high compression ratio, and the video data redundancy has been removed to a great extent after compression coding, which makes it more difficult to embed more data into the compressed video stream. In the existing literatures, most of the second category combining with certain aspects of the compression characteristics utilizes coding process to hide information, such as intra prediction, DCT transform, entropy coding, and motion estimation etc.

There are many methods utilizing intra prediction for information hiding [12, 22, 34, 38, 39, 45]. Although the implementations are different, these methods essentially hide information by modifying the intra encoding prediction mode. Information can only be embedded in I frames. But the proportion of I frames in a video sequence is low, which makes the embedding capacity of this kind of method limited. The papers [7, 18, 21] utilizing DCT transform to hide secret information usually choose large non-zero coefficients after the integer transform and quantization. Since the quantity of these coefficients is usually less, the embedding capacity is not very high. The entropy coding methods modifying code elements based on CAVLC or CABAC [19, 23, 25, 27, 33, 35, 46] may cause large distortion and even lead to decoding failure. Motion estimation is used to hide information [2, 5, 8, 11, 14, 26, 36, 40–42, 44] by modifying motion vector or adjusting the motion vector search process. Since the distortion of motion estimation will be encoded and transferred in the compensation process and the motion estimation has large error, the information hiding algorithms based on motion estimation will contribution little additional distortion to the reconstructed frames. At the same time, the P frames and B frames are the major frames in the video sequences. As a result there are a lot of motion vectors and we can achieve a large embedding capacity. In view of this, the information hiding technology based on motion vector modification has been widely studied.

In the earlier work, motion vector was used to hide information usually by directly modifying motion estimation, which is relatively simple. The main idea of these methods is that they make the amplitude or phase angle of modified motion vector satisfy certain rules. Recently, Aly [2] proposed a novel method based on previous work. This method used motion vector prediction error as the basis for selecting motion vector carriers and embedded information by directly replacing the least significant bit (LSB) of motion vector which met the threshold condition. Guo et al. [11] proposed a method to embed secret information by adjusting the parity of the horizontal component and the vertical component. Besides, there were some novel information hiding methods by modifying the search range of motion in the process of video encoding. For instance, Zhu et al. [44] and Swaraja et al. [26] proposed two video steganography algorithms. They divided the search points into two sets to represent the secret information 0 and 1 respectively, and hided information in the process of motion search with 1/4 pixel accuracy. Yao et al. [40] designed a distortion function by exploiting the spatial-emporal correlation to minimize the embedding impact on motion vectors, and utilized the syndrome-trellis codes (STCs) [10] to implemente their method. Zhang et al. [41] proposed a steganographic method based on motion vector's local optimality and used the STCs to implement embedding.

In addition, there are some other methods not belonging to these four categories such as information hiding based on IPCM macroblocks [1, 15], information hiding based on coding block size [16, 17, 20], and information hiding integrating several methods [9, 29].

As we know, the application of high-definition video becomes more and more popular in daily life, and the market gradually introduces the beyond-HD formats with higher resolution (e.g., 4k×2k, 8k×4k). Meanwhile, the demand of efficient video transmission is also growing because of the development of network and video-on-demand and other services, which leads to a higher performance requirement for video coding clarity, frame rate and compression ratio. The popular video coding standard H.264/AVC cannot satisfy the requirements. The ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group officially jointly released H.265/HEVC as the next-generation video coding standard in February 2013. With a good parallel processing architecture, HEVC has largely improved the high-resolution processing ability. Besides, HEVC has higher compression ratio and supports data loss recovery, which reduces the difficulty of transmission. Compared with H.264/AVC, HEVC can adapt to the increasingly diverse high-definition network video services. It can be expected that HEVC will replace H.264/AVC in the future and become the new mainstream video coding standard. That brings the question how to effectively protect the copyright of HEVC encoded video. Therefore, it is significant to study the information hiding methods based on the standard, which will be helpful to promote the application of HEVC.

As H.26X and MEPG-X, HEVC also follows the hybrid video coding approach, which integrates intra prediction, motion estimation and compensation, transform, quantization, encoding and other sectors into the encoder [24]. So the idea of the video information hiding methods described above can also be applied to HEVC, that is, we can combine certain aspects of the characteristics of the encoder to embed secret information in the coding process. However HEVC makes large improvement and innovation in the technical details. For example, the macroblock in H.264/AVC has been replaced by transform unit (TU), prediction unit (PU) and coding unit (CU) and each unit has a flexible size. HEVC introduces merge mode for motion estimation in PUs. These new characteristics determine the aforementioned video information hiding methods cannot be directly applied to HEVC. Therefore, it is necessary to design appropriate information hiding method for the coding properties of HEVC. Since HEVC has been proposed for a short time and is very complicated, the information hiding technique based on HEVC video coding standard is in the beginning stages. The existing related information hiding methods [4, 6, 13, 28, 30–32, 37, 43] implement embedding during the encoding.

In the aspect of intra prediction, Wang et al. utilized the probability distribution of the statistical optimal prediction mode and suboptimal prediction mode and modified the mapping relationship between prediction modes and secret information to achieve information hiding [30]. They also utilized standard array of block code to achieve that three secret information bits are embedded in four successive 4×4 luminance blocks by modifying 1.25 prediction modes on average [31], and then reduced the bit rate increase and video quality degradation with Hamming+1 [37]. In addition, they embedded secret information by establishing a mapping relationship table between the angle differences of intra prediction and secret information and modifying the mapping relationship [32]. In the aspect of DCT transform, the DCT coefficients and DST coefficients are modified according to certain rules for information hiding [4, 6, 28, 43]. In the aspect of entropy coding, Jiang et al. [13] presented the concept of "constant bitrate information bit" (CBIB) and exploited the CBIBs of the motion vector difference for information hiding.

Currently, this is no related research of information hiding in the aspect of motion estimation. Additionally, in the main video information hiding approaches described above, the information hiding method based on motion vector has a larger advantage. So we combine the video coding features to propose an efficient video information hiding method for HEVC. In this paper, the motion vector is selected as the carrier in the HEVC video coding process. We create N-dimensional motion vector space with a mapping relationship and realize that a $2N + 1$-ary number can be embedded by modifying at most one element in a set of N motion vector components, which has high embedding efficiency. In addition, we use a secret key to complete the embedding and extracting for secret information, and only choose the motion vectors of PUs with smallest size as the carriers, which makes the proposed algorithm have good transparency.

2 HEVC data partitioning and motion estimation

HEVC abandons the macroblock concept in H.264/AVC, but uses CTU, CU, PU and TU to partition and process the video frame data. The biggest processing unit is CTU, similar to the macroblock in H.264/AVC. Each CTU can be recursively partitioned into CUs, and

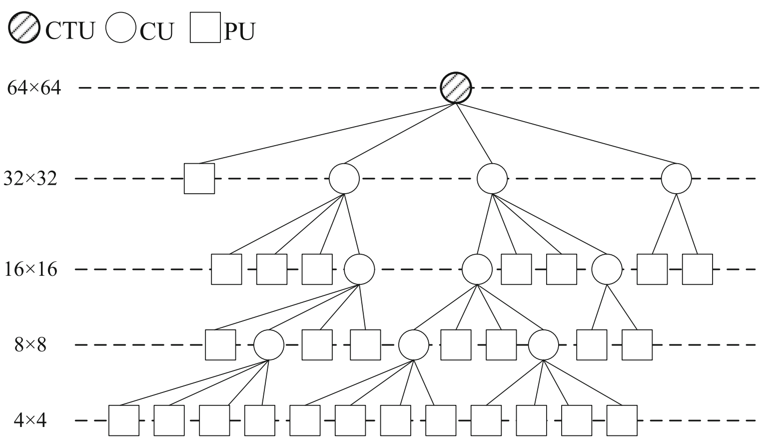


Fig. 1 Subdivision of a CTU into CUs [and PUs]

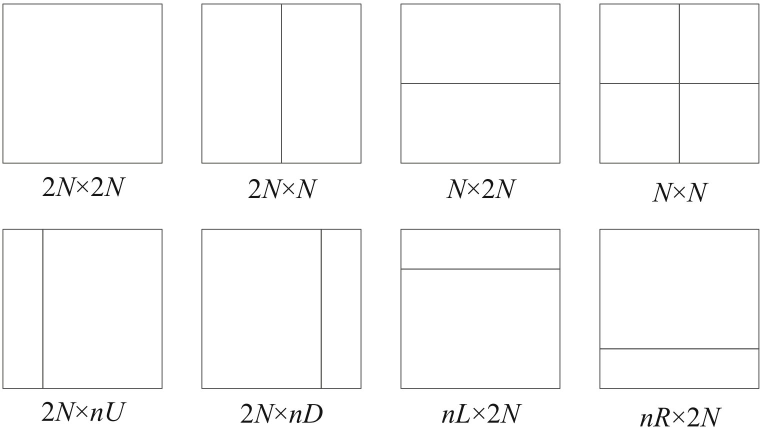


Fig. 2 Eight modes for partitioning a CU into PUs

each CU can be partitioned into PUs when implementing intra or inter prediction. Each CU can also be partitioned into TUs when implementing transformation and quantization. Each processing unit has more abundant and flexible division modes than H.264/AVC.

A CTU is partitioned into one or more CUs according to the quadtree structure. Whether using intra or inter prediction mode for each frame in HEVC is determined at the CU level. According to the decision of the prediction mode, the CU will be further split into PUs and performs the prediction process in PUs. Figure 1 shows an example of the subdivision of a CTU into CUs and PUs in the form of a tree. The root of the tree is a CTU with the size of 64×64 which is split into smaller pieces of CUs and PUs gradually. The PUs located at the leaf node have different size blocks.

The inter prediction of HEVC supports four symmetric partition modes: PART_$2N \times 2N$, PART_$2N \times N$, PART_$N \times 2N$ and PART_$N \times N$. It also supports four asymmetric partition modes: PART_$2N \times nU$, PART_$2N \times nD$, PART_ $nL \times 2N$ and PART_$nR \times 2N$. All the partition modes are shown as in Fig. 2.

Motion estimation and compensation technology is widely used in video compression of inter prediction for eliminating video data redundancy in temporal domain to improve coding compression ratio. Motion estimation divides the images which need to inter prediction in a video sequence into a plurality of non-overlapping blocks, and then searches for its most similar matching block in a certain range of the reference frame. The displacement from the matching block to the current block is the motion vector. Figure 3 shows the motion vector $d = (x, y)$ with horizontal component x and vertical component y. Assume

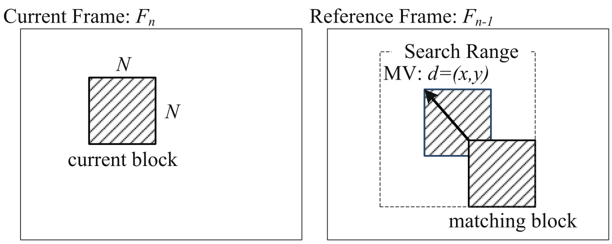


Fig. 3 Motion vector expresses displacement from matching block to current block

the matching block image value with pixel coordinate (i, j) is $f_m(i, j)$ and the current block image value is $f_c(i, j)$. We can gain the prediction value of the current block after motion compensation with vector $d = (x, y)$, and the prediction is $f_c(i, j) = f_m(i + x, j + y)$. The inter prediction process of video coding can be described as following: firstly, calculate the prediction motion vector dp; secondly, make the prediction motion vector as the starting point and search for the best matching block to obtain the motion vector d in the reference frame; finally, in order to reduce the code stream, only the residual date Δd between pd and d should be encoded and written in the coding stream.

Motion search process of HEVC supports quarter-pixel precision luminance blocks as H.264/AVC video coding standard. For each block, firstly, search for the integer-pixel position; secondly, search for the half-pixel position around the best integer-pixel position; finally, search for the quarter-pixel position around the half-pixel position. In the searching process, the best matching pixel position is used to calculate the motion vector. Since motion searching process needs large calculation cost in the entire coding process, motion merge is introduced into HEVC to reduce the calculation. The motion vector of PU with a merge mode can be derived from its neighboring PUs without motion searching. From the above we can see that the motion estimation is itself an imprecise process, that is, the motion vector obtained by the motion estimation is an estimated value, which provides a good opportunity for information hiding.

3 Motion vector space encoding

Assuming that a CTU with inter prediction mode in HEVC has m motion vectors, and each motion vector includes a horizontal component h and a vertical component v, expressed as $d_i = (h_i, v_i)$, where $i \in \{1, 2, ..., m\}$. The range of the components is associated with the motion search parameters. All motion vectors components of the CTU can be described as: $MV = (h_1, v_1, h_2, v_2, ..., h_m, v_m)$, from which we can take out $N(N \leq 2m)$ elements to construct a motion vector component N-tuple described as: $C = (c_1, c_2, ..., c_N)$. Then we can obtain another N-tuple τ of $2N + 1$-ary, where $\tau = (x_1, x_2, ..., x_i, ..., x_N)$ and x_i is computed as:

$$x_i = \begin{cases} c_i \bmod (2N+1) & c_i \geq 0 \\ (c_i \bmod (2N+1) + 2N + 1) \bmod (2N+1) & c_i < 0 \end{cases} \tag{1}$$

An N-dimensional space can be constructed based on τ, and x_i is the corresponding i-th dimensional coordinate of the space. The space is called motion vector space and described as Γ. Obviously, arbitrary N-tuple τ can be mapped as a point in Γ, and the mapping is defined as $f(x_1, x_2, ..., x_i, .., x_N)$. All the points in Γ can construct an N-dimensional space lattice, in which each dimension coordinate has $2N + 1$ values with the range of {0,1,...,$2N$}. The assignment process of the points in Γ is described as following:

Step 1. Assign the points of the 1-dimensional axis:

$$f(x_1) = x_1 \bmod (2N+1) \tag{2}$$

Step 2. Spread the 1-dimensional points to 2-dimensional plane:

$$f(x_1, x_2) = (f(x_1) + 2x_2) \bmod (2N+1) \tag{3}$$

Step 3. Spread the 2-dimensional plane points to 3-dimensional space:

$$f(x_1, x_2, x_3) = (f(x_1, x_2) + 3x_3) \bmod (2N+1) \tag{4}$$

Step N. Spread the $N-1$-dimensional space points to N-dimensional space:

$$f(x_1, x_2, ..., x_N) = (f(x_1, x_2, ..., x_{N-1}) + Nx_N) \bmod (2N+1) \tag{5}$$

The above assignment process of the points in motion vector space is called motion vector space encoding. Figure 4 shows the process with $N = 3$.

Firstly, while N is equal to 3, the range of the space coordinate is {0, 1, 2, 3, 4, 5, 6} according to (1). Secondly, according to (2) and the space coordinate range, we can obtain the points values on the 1-dimensional axis, shown in Fig. 4a. Thirdly, the 2-dimensional lattice can be computed by spreading the 1-dimensional points to the 2-dimensional plane according to (3), shown in Fig. 4b. Finally, establish a 3-dimenonal coordinate system and spread the 2-dimensional plane points to 3-dimensional space, shown in Fig. 4c. Obviously, each point value in Γ is $2N+1$-ary. Here we will prove the space Γ has the following important property:

Property 1 The computing process of each point value in N-dimensional space lattice with motion vector space encoding can be simplified as:

$$f(x_1, x_2, ..., x_N) = (x_1 + 2x_2 + ... + Nx_N) \bmod (2N+1) \tag{6}$$

According to the motion vector space encoding method and (2) to (5), we can obtain:

$$\begin{aligned} & f(x_1, x_2, ..., x_N) \\ & = (f(x_1, x_2, ..., x_{N-1}) + Nx_N) \bmod (2N+1) \\ & = ... \\ & = ((x_1 \bmod (2N+1) + 2x_2) \bmod (2N+1) + ... + Nx_N) \bmod (2N+1) \\ & = ((x_1 + 2x_2) \bmod (2N+1) + ... + Nx_N) \bmod (2N+1) \\ & = ... \\ & = (x_1 + 2x_2 + ... + Nx_N) \bmod (2N+1) \end{aligned} \tag{7}$$

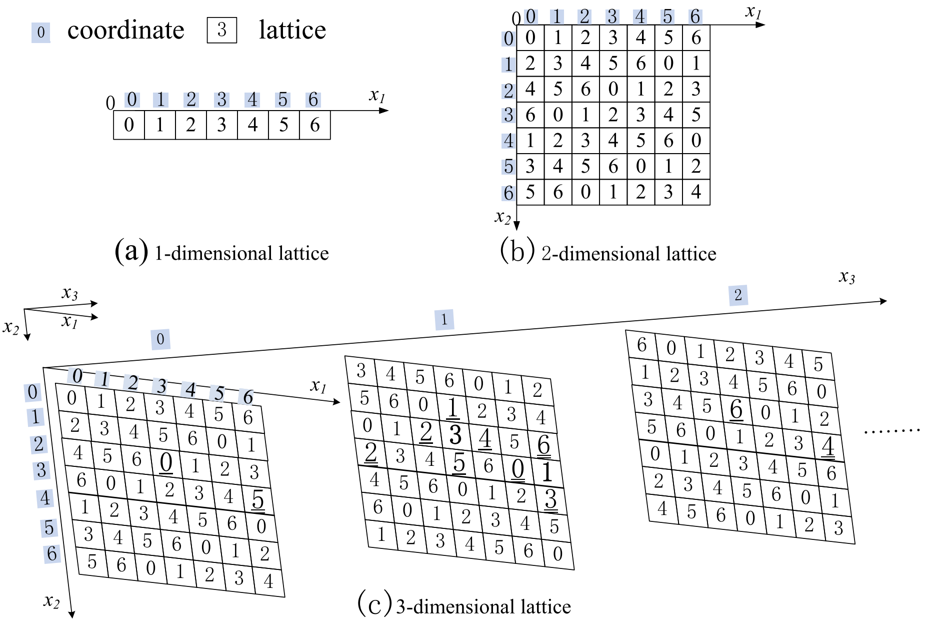


Fig. 4 The constructing process of 3-dimensional space lattice

For each point $P = (x_1, x_2, ...x_i, ..., x_N)$ in N-dimensional motion vector space Γ, assume x_i is the i-th dimension coordinate, then we call B_i^+ and B_i^- neighbor nodes of the point P in the i-the dimension, where $B_i^+ = (x_1, x_2, ..., x_i^+, ..., x_N)$ and $B_i^- = (x_1, x_2, ..., x_i^-, ..., x_N)$. x_i^+ and x_i^- are computed as:

$$\begin{aligned} x_i^+ &= (x_i + 1) \bmod (2N + 1) \\ x_i^- &= (x_i - 1) \bmod (2N + 1) \end{aligned} \tag{8}$$

Obviously, there are $2N$ neighbor nodes of P. We will cite an example with the point value bold 3 of $P = (3, 2, 1)$ in Fig. 4c. The neighbor nodes in the first dimension are 2 and 4; the second dimensional neighbor nodes are 1 and 5; and the third dimensional neighbor nodes are 0 and 6. These neighbor nodes are marked in single underline in the figure. Similarly, the point value 1 with the coordinate (6, 3, 1) is marked in bold, whose neighbor nodes marked in double underline are 0 and 2, 6 and 3, 5 and 4 corresponding to three different dimensions respectively. Through observation, we can find that the values of each point and its neighbor nodes are different from each other and the values of these points can form a consecutive integer set of $\{0, 1, ..., 2N\}$.

This phenomenon in space Γ is an important theoretical basis of our proposed method, because it can guarantee that any point value in Γ can be equal to any specified $2N + 1$-ary value by adding or subtracting 1 to at most one dimension coordinate value. It can be proved in theory. According to Property 1, point P can be described as $f(x_1, x_2, ..., x_k, ..., x_N) = (x_1 + 2x_2 + ... + kx_k + ... + Nx_N) \bmod (2N + 1)$. Let $(x_1 + 2x_2 + ... + kx_k + ... + Nx_N) \bmod (2N + 1) = K$, where K is an integer, then the value of the point P is $K \bmod (2N + 1)$. Assume B_i^+ and B_i^- are the neighbor nodes of point P in the i-th dimension, where $i \in \{1, 2, ...N\}$, then the value of B_i^+ is $f(x_1, x_2, ..., x_i^+, ..., x_N) = (x_1 + 2x_2 + ... + ix_i^+ + ... + Nx_N) \bmod (2N + 1) = (x_1 + 2x_2 + ... + i(x_i + 1) + ... + Nx_N) \bmod (2N + 1) = (K + i) \bmod (2N + 1)$. The value of B_i^- can be correspondingly expressed as: $f(x_1, x_2, ..., x_i^-, ..., x_N) = (K - i) \bmod (2N + 1)$. When i changes from 1 to N, all B_i^+ and B_i^- can constitute a set V containing all neighbor nodes of point P. The set can be described as: $V = \{(K - N) \bmod (2N + 1), ..., (K - 1) \bmod (2N + 1), K \bmod (2N + 1), (K + 1) \bmod (2N + 1), ..., (K + N) \bmod (2N + 1)\}$. Obviously, the set V consists of $2N + 1$ elements, and these elements are one-to-one corresponding to the elements in integers $\{0, 1, ..., 2N\}$ after the values of K and N determined. Hence, for any assigned positive integer $D_0 \in \{0, 1, ...2N\}$ and any point P in Γ, if the value of P is not equal to D_0, there must be a point B in the neighbor nodes of P satisfying. Since B is one of the neighbor nodes of P, B can be obtained by one dimensional coordinate of P plus or minus 1.

Therefore the value corresponding to N-tuple in Γ can be equal to the assigned $2N + 1$-ary number D by modifying at most one element in the N-tuple. It's an important theoretical basis for our embedding algorithm. Because a $2N + 1$-ary number contains $log_2(2N + 1)$ bits, the proposed method can embed $log_2(2N + 1)$ secret information bits by modifying at most one motion vector component.

4 Information embeding and extracting

Motion vectors in HEVC are utilized as the carriers for the proposed information hiding algorithm. The motion vectors of only part of the PUs are chosen to further enhance the security of the algorithm and to more flexibly control the embedding process, and an embedding strength is introduced to control the embedding capacity. The motion vectors chosen as carriers are used for constructing N-tuple C in the proposed method. The mapping value of C in motion vector space is expressed as the secret information.

4.1 The selection and space mapping of motion vector N-tuple

A CTU is utilized for the basic unit of information embedding in our method. This method determines whether the CTU is selected for embedding according to the preset embedding strength before embedding. The embedding strength e is a decimal whose range is 0 to 1. It is the probability of whether the secret information is embedded in CTU. For example, when e is equal to 0.5, there is a probability of 50% for each CTU being selected as a carrier. While embedding information, another decimal r for each CTU is generated by a random decimal generator G. If r is less than e, the CTU is chosen. In order to embed and extract the secret information correctly, the random decimal generator G must be initialized with the same seed while embedding and extracting. The seed is generally used as the secret key K which is informed to the extractor from information embedder by other channels.

If a CTU is selected as a carrier, its corresponding motion vector space dimension N is computed by r while hiding information, that is, the motion vector space is changeable in the embedding process, the purpose of which is to further improve the security of the algorithm. After the space dimension N is computed, this method will count the number of motion vectors which are non-merge and non-skip mode. If the number is bigger than $N/2$, the smallest $N/2$ PUs with their motion vectors will be selected as the carriers. Otherwise, the CTU is still not selected as an embedding carrier. Through the above method, the motion vectors to be modified for steganography are confined in the smallest $N/2$ PUs. Essentially, the image area affected by motion vectors modification is confined, which helps reduce the impact of steganography on the video quality.

The set with N integers of the horizontal and vertical components of the $N/2$ selected PUs is called motion vector N-tuple C, and $C = (c_1, c_2, ..., c_N)$. The N-tuple C can be converted to N-dimensional motion vector space coordinate $(x_1, x_2, ..., x_N)$ in motion vector space Γ according to (1). The space encoding value of the motion vector space coordinate is $f(x_1, x_2, ..., x_N)$ according to Property 1, where $f(x_1, x_2, ..., x_N) \in \{0, 1, ..., 2N\}$. Hence, the motion vector N- tuple is mapped to an integer D ranging from 0 to $2N$, and D is called motion vector mapping value. The embedding purpose of our method is to make the mapping value of N-tuple equal to a $2N + 1$-ary value which will be embedded by modifying one of elements in N-tuple at most. Since the modification only needs D plus or minus 1 operation, the modification degree of the original motion vectors is very small. Moreover, the motion vectors in N-tuple are from the PUs with small area in the current CTU and the steganography only changes one component of one motion vector of the PUs in the CTU. Therefore, the disturbance on video quality introduced by our method is very small while hiding information.

Figure 5 is an example of a CTU for carrier selection. The PUs in black bold boxes use merge mode. Assume the motion vector space dimension is $N = 8$, then there must be 4 smallest PUs selected from the CTU, and the PUs are non-merge mode and non-skip mode. In Fig. 5, the selected PUs are marked with shadow and their numbers are 10, 11, 13 and 15. All the components of the four PUs' motion vectors will be used to construct the motion

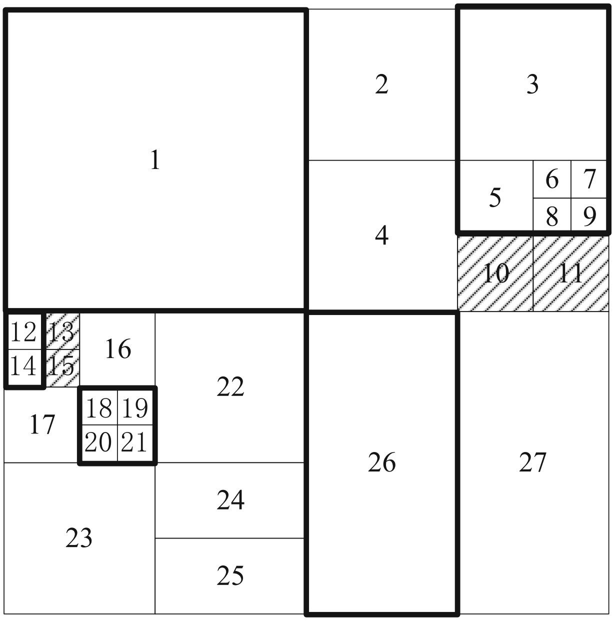


Fig. 5 Subdivision of a CTU and selection of PUs as carriers. PUs in black bold boxes use merge mode. PUs with shadow mark are selected as carriers

vector N-tuple of this CTU. The symbol d denotes the motion vector of the PU marked with shadow; h and v respectively denote horizontal component and vertical component, then these motion vectors can be expressed as: $d_1 = (h_1, v_1)$, $d_2 = (h_2, v_2)$, $d_3 = (h_3, v_3)$, $d_4 = (h_4, v_4)$. The motion vector N-tuple can be expressed as: $C = (h_1, v_1, ..., h_4, v_4)$, rewritten as $C = (c_1, c_2, ..., c_8)$. So the motion vector component set for information hiding is obtained and it can be mapped into a point in Γ according to the aforementioned method.

4.2 Secret information embedding

Secret information embedding is essentially a process that modifies the motion vector N-tuple C to make its mapping value D in Γ equal to the secret information converted to a $2N + 1$-ary integer. Before embedding, the secret information of text, image or other forms needs binarization to obtain secret information bit stream. Denote $l = \lfloor log_2(2N + 1)\rfloor$, then we take out a binary string with the length of l from secret information bit stream for each CTU while embedding. Assume the binary string is S, and $S = (s_l, s_{l-1}, ..., s_1)$, then its corresponding $2N + 1$-ary value F can be computed as:

$$F = \sum_{i=1}^{l} s_i \times 2^{i-1} \tag{9}$$

Algorithm 1 Secret information embedding process

```
Input: Sequence, message bitstream m, secret key K, embedding threshold e;
Output: encoded Sequence with data embedded;
G( ) : random decimal generator;
num : the number of non-merge and non-skip PU with inter prediction mode in
the current CTU;
U: the upper bound of motion vector space dimension;
lower( ): lower limit function;
BEGIN
  initial G( ) with K;
  FOREACH P and B frame in Sequence DO
    FOREACH CTU in the frame DO
      IF ((r = G( )) < e ) THEN
        N = (2 + (r×100)) mod U;
        IF (num > N/2) THEN
          choose the smallest N/2 PUs;
          construct C = (c_1, c_2,..., c_N) from above PUs;
          calculate the mapping value D of C;
          l = lower(Log_2(2N+1));
          exact S from m with length l;
          calculate corresponding value F of S;
          IF (D==F) THEN
            go to next CTU;
          ELSE
            FOREACH c_i in C DO
              modify c_i with variation 1;
              mark modified C as C*;
              calculate D* of C*;
              IF (D* == F) THEN
                go to next CTU;
```

If the mapping value D of N-tuple C is equal to F, no element in C needs to be modified, and end steganography; if D is not equal to F, we will traverse all the elements in C and add or subtract 1 to one of them. Denote the modified motion vector N-tuple is C', and its mapping value is D', then the traversal is over if $D' = F$. The whole information hiding process synthesizing PUs selection process is shown as algorithm 1.

In Fig. 6, we will give an example of secret information embedded in HEVC encoded video sequence using our approach. For each embedding, firstly, select a qualified CTU; secondly, compute the space dimension N; thirdly, extract part of secret information according to N. In this example, the length of the secret information is 10 bits, and N is equal to 2, 8, and 4 in turn for each embedding, so the secret information can be completely embedded into three CTUs for three times. The selected motion vectors are located in the PUs with shadow mark and the black bold boxes indicate the PUs with motion vectors modified. In the case of the second embedding, the motion vector N-tuple is $C = (-1, -2, 30, 16, 0, 0, 11, 19)$, then the corresponding point in motion vector space Γ is (16, 15, 13, 16, 0, 0, 11, 2) according to (1). The point value is $f(16, 15, 13, 16, 0, 0, 11, 2) = 4$ according to property 1, that is, the mapping value of the motion vector N-tuple in current CTU is $D = 4$. The secret information bit string to be embedded is "1100", of which the corresponding $2N + 1$-ary is $F = 12$ according to (9). Through traversal and computation, modify the last element 19 in C to 20 to make the new mapping value D' of the modified motion vector N-tuple equal to 12. Eventually the motion vector N-tuple with embedded secret information is $C' = (-1, -2, 30, 16, 0, 0, 11, 20)$. It can see that the modification degree is small using our approach.

After information hiding with our method, the motion vectors with secret information will be encoded and stored or transmitted as code elements in HEVC compressed video stream.

4.3 Secret information extracting

Secret information extracting is essentially a process that extracts motion vectors in HEVC compressed video stream to form an N-tuple, and then compute the mapping value of the N-tuple in space Γ. As the embedding process, we firstly need to judge whether the current CTU has been embedded secret information. Extractor utilizes the random decimal generator G to generate a random decimal r for each CTU. The secret key K provided by embedder is the seed of G. If r is less than the embedding strength e, the CTU will be selected for candidate. Then we compute the space dimension N. If the number of inter prediction PUs with non-merge and non-skip mode in the current CTU is less than $N/2$, the CTU does not carry secret information. If not, we select the smallest $N/2$ PUs of the CTU, and then extract their motion vectors to form motion vector N-tuple $C = (c_1, c_2, ..., c_N)$. We convert C to motion vector space coordinate according to (1) and compute its mapping value D. The mapping value D can be converted a binary string M with length of $l = \lfloor log_2(2N+1) \rfloor$. M is the secret information bit string embedded in the current CTU. After extracting all secret information embedded in CTUs, we can recover the complete embedded secret information by connecting the extracted binary strings in sequence. The detailed secret information extracting process for HEVC compressed video is shown as algorithm 2.

Algorithm 2 Secret information extracting process

```
Input: encoded Sequence, secret key K, embedding threshold e;
Output: message bitstream m;
G( ) : random decimal generator;
num : the number of non-merge and non-skip PU with inter
prediction mode in the current CTU;
U: the upper bound of motion vector space dimension;
lower( ): lower limit function;
BEGIN
  initial G( ) with K;
  initial m as an empty bitstream;
  FOREACH P and B frame in Sequence DO
    FOREACH CTU in the frame DO
      IF ((r = G( )) < e ) THEN
        N = (2 + (r×100)) mod U;
        IF (num > N/2) THEN
          choose the smallest N/2 PUs;
          construct C = (c1, c2,..., cN) from above PUs;
          calculate the mapping value D of C;
          l = lower(Log2(2N+1));
          FOR i ←l to 1 DO
            si = D / 2^(i-1);
            D = D mod 2^(i-1);
          construct S = (s1, s2,..., sl) ;
    m = m + S;
```

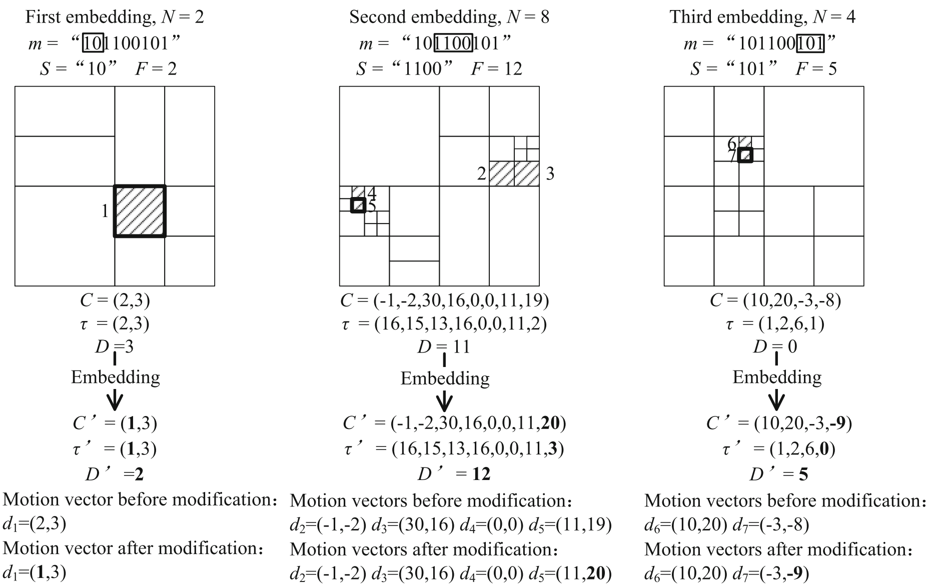


Fig. 6 An example of secret information embedding for CTU

We will give an example of extracting secret information from a video sequence shown as Fig. 7, which is corresponding to the embedding process in Fig. 6. For each CTU which carries secret information, firstly, we compute the space dimension N; secondly, construct

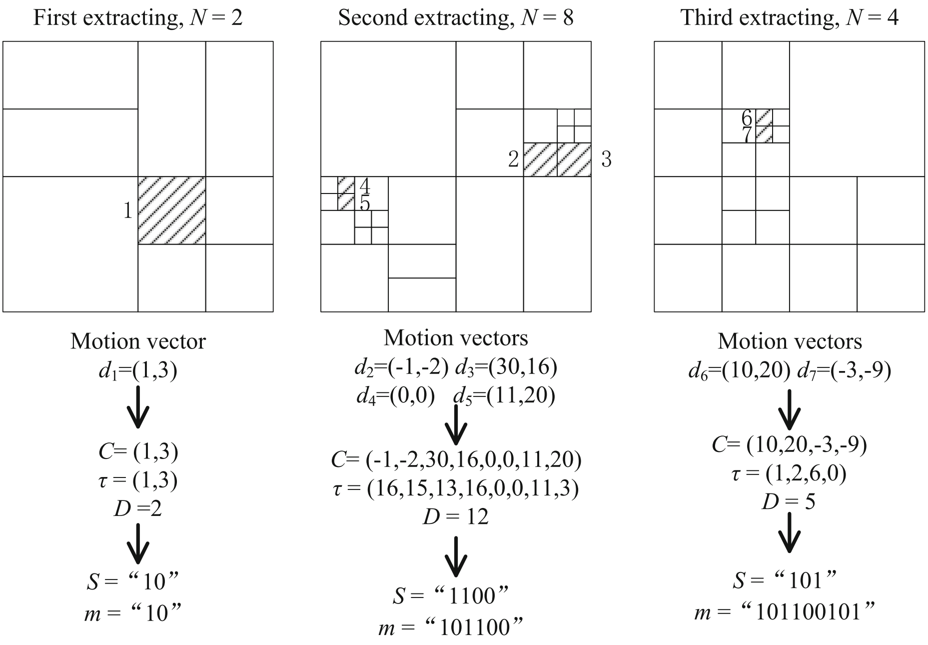


Fig. 7 An example of secret information extracting for CTU

the motion vector N-tuple according to N; thirdly, computer and extract the secret information in the current CTU. In Fig. 7, the selected motion vectors are located in the PUs with shadow mark. In the case of the second extracting, the current space dimension is $N = 8$, and the corresponding motion vector N-tuple is $C = (-1, -2, 30, 16, 0, 0, 11, 20)$. The corresponding point of C in motion vector space Γ is (16, 15, 13, 16, 0, 0, 11, 3) computed by (1). According to property 1, the point value is $f(16, 15, 13, 15, 0, 0, 11, 3) = 12$, that is, the mapping value D of the motion vector N-tuple in the current CTU is 12. The mapping value D can be converted into a binary string "1100" that is the secret information embedded in the current CTU. After extracting and connecting all embedded secret information bit in CTUs, we can obtain the complete embedded secret information "101100101".

5 Experiments and results analyses

In order to fully evaluate the performance of our propose algorithm, firstly, we will estimate the video quality and bit rate with the information hiding method for decoded video; secondly, the embedding efficiency of this algorithm will be analyzed; thirdly, we will analyze the anti-steganalysis performance of the algorithm; finally, we will give the complexity of the embedding and extracting algorithms.

5.1 Experimental setting

Since HEVC has been put forward for a short time, there is no information hiding methods based on motion vector for HEVC reported according to our research. Therefore, in order to facilitate comparison, we consider two typical embedding rules. One is substituting secret information bits for the horizontal and vertical component LSBs of motion vector selected as carrier. Two bits can be embedded for each motion vector. The other one is embedding secret information by modifying the horizontal or vertical component of motion vector. After the modification, the XOR result of the two components is equal to the secret information bit and one bit can be embedded for each motion. The two embedding rules are respectively recorded as LSB and Parity for ease of description. We exploit the method in [3] to analyze the anti-steganalysis performance of our algorithm. The method in [3] is effective for steganography based on motion vector published in recent years. Based on [3], we have achieved a steganography detector suitable for information hiding methods based on motion vector for HEVC.

We realize our method and the other compared methods based on the HEVC reference software version 14.0, and experiment with twelve common test video sequences. They consist of 6 sequences with resolution 1920×1080: Cactus, BasketballDrive, BQTerrace, Kimono1, ParkScene and Tennis; 2 sequences with resolution 2560×1600: PeopleOnStreet and Traffic; 1 sequence with resolution 1024×768: ChinaSpeed; 3 sequences with resolution 832×480: RaceHorses, Kerba and PartyScene. These sequences are encoded with a GOP structure "IPP..." and frame rate "25", and the remaining parameters are set to the HM default configuration. We analyze the video quality, bit rate and embedding capacity based on single P frames and we encode 100 frames for each sequence while analyzing the anti-steganalysis performance.

We record the position information of motion vectors located in PU and CTU with our method while the embedding strength is 1.0. Then we embed secret information into the same PUs with the compared methods of LSB and Parity. Besides, we experiment our method while the embedding strength is 0.2 and 0.5 for comprehensive evaluation. We will use ME1.0, ME0.5, ME0.2 to respectively represent algorithm with three kinds of different embedding strength for description.

5.2 Video quality analysis

Video quality analysis includes two major aspects subjective impression and objective assessment. Subjective assessment analyzes the vision distortion of the image. As a common criterion, the peak signal-to-noise ratio (PSNR) is widely used in the objective assessment because of its effective evaluation for video image quality and is also adopted in our experiments.

First, we analyze the subjective impression. Figure 8 illustrates the reconstructed images of video sequences Cactus and BasketballDrive before and after steganography with different methods. The rows marked with "LSB" and "Parity" respectively denote the decoded images embedding secret information with the rules of LSB and Parity. Similarly, the rows marked with "ME1.0", "ME0.5" and "ME0.2" respectively denote the decoded images using our method with the embedding strength 1.0, 0.5 and 0.2. In this figure, we can see that our method with three different kinds of embedding strength does not cause to significant deterioration for the decoded images and there is no visual jagged, mosaics and dislocation compared with the decoded images without embedding. Besides, there is no difference through subjective impression between our method and the compared methods.

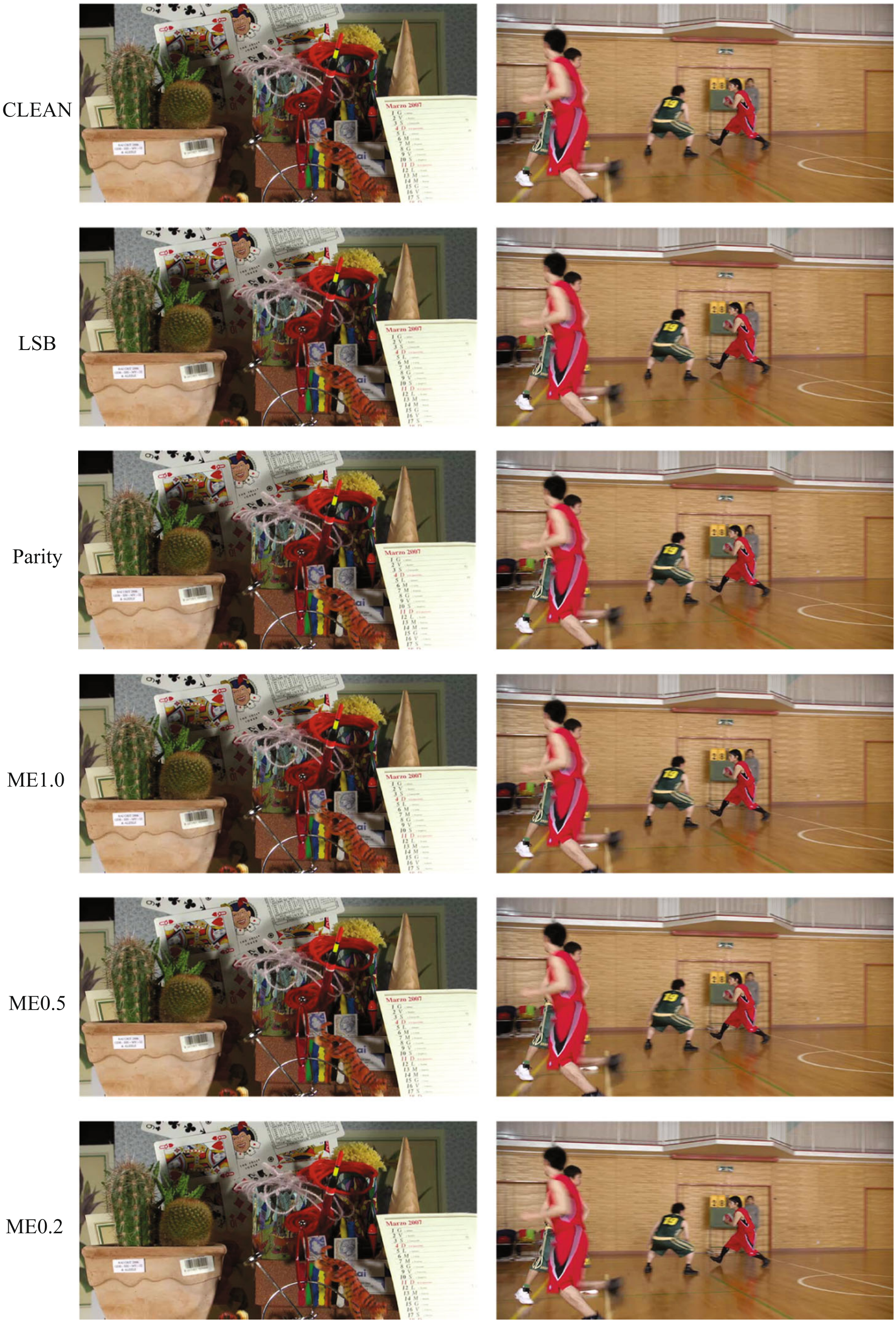
Then we will exploit PSNR to evaluate the objective comparison of video quality. Figure 9 presents the PSNR values in the average luminance component Y and two chrominance channels U and V of twelve HEVC test sequences' decoding images without embedding and with different embedding. LSB, Parity and ME1.0 have the same quantity of embedding bits. In this figure, we can see that, in most cases, the PSNR with the embedding strength 1.0 is smaller than the other two embedding strength using our method, that is, the video distortion is larger with the embedding strength 1.0. With the embedding strength decrease, the PSNR value rises. This is because the decrement of embedding strength means the number of CTU selected as carriers becomes fewer, which brings to fewer changes to original video area. Hence, the impact on the video quality will be reduced accordingly. Obviously, the effect of our method with the embedding strength 1.0 on luminance component is better than LSB and Parity. On the two chrominance components, our method also remains certain advantages. The results illustrate that our method brings to less distortion than the other two methods of the video quality while embedding the same amount of secret bits.

5.3 Video bitrate increment analysis

Due to the modification on PU's motion vector in the process of information hiding, the bitrate of compressed stream usually increases. We utilize the bitrate increment (BRI) to measure the bitrate variation of the video sequences after embedding. The BRI is defined as:

$$BRI = \frac{(BR' - BR)}{BR'} \tag{10}$$

where BR' is the video bitrate without embedding, and BR is the video bitrate after embedding. The BRI of the twelve HEVC test sequences with different information hiding methods is shown in Fig. 10. LSB, Parity and ME1.0 have the same quantity of embedding bits. We can see that the method with embedding strength 0.2 has the lowest bitrate increment, which is closest to zero. With the embedding strength increasing, the bitrate of video



(a) Cactus (b) BasketballDrive

Fig. 8 Decoding video images before and after steganography with different methods

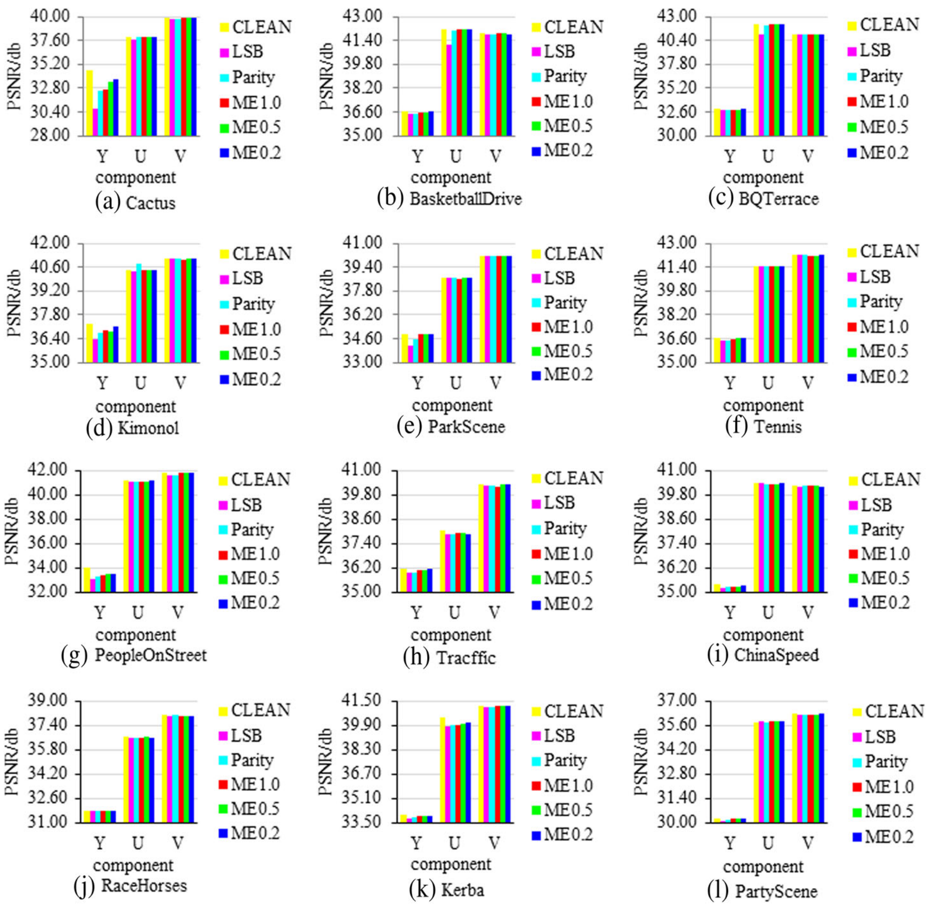


Fig. 9 The PSNR of decoded video sequences with different steganography methods

stream also increases. Obviously, the method with embedding strength 1.0 is better than LSB and Parity on the whole with less bitrate increment.

5.4 Embedding efficiency analysis

We define the embedding efficiency E as the average quantity of secret information embedded when modifying one motion vector component. Suppose that embed m bits to one frame in video in which n motion vector components are modified, then the embedding efficiency is $E = m/n$. In Table 1, we list the embedding efficiencies on twelve HEVC videos with three embedding strength and two compared information hiding algorithms.Sequence, Steganography method, Embedding bits, Motion vector components and Average embedding quantity are abbreviated as Seq, SM, EBs, MVCs and AEQ respectively. The columns “EBs” and “MVCs” refer to the average number per frame.

From the data in Table 1, our method has fewer embedding bits in weaker embedding strength when hiding information; but, the embedding efficiency of our methods with three

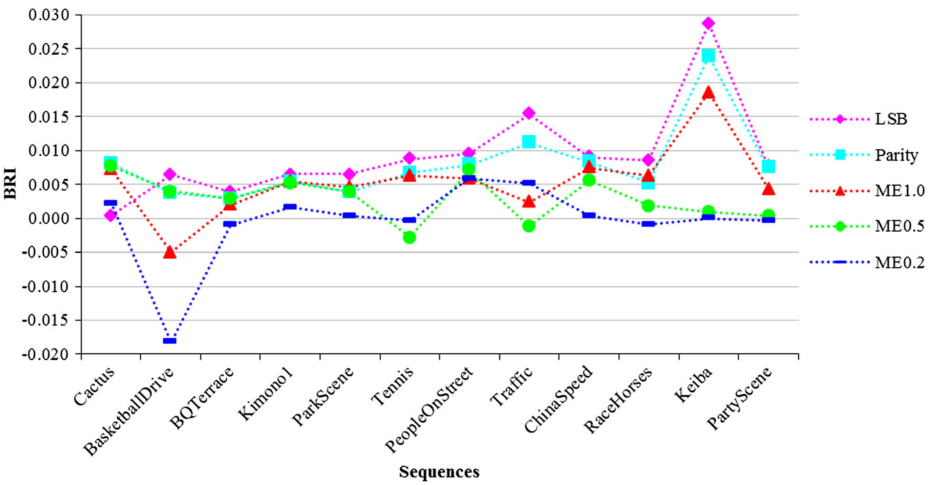


Fig. 10 The BRI of decoded video sequences with different steganography methods

embedding strength are better than LSB and Parity. Although the embedding capacity of our methods is lower than LSB under the same number of carriers, the embedding capacity of our methods is high than LSB under the same number of motion vector components modified. This shows that, the number of motion vector components modified of our methods is less while embedding the same of secret information, which further illustrates our methods have better performance in the aspects of video quality and code rate with the same quantity of embedding bits.

5.5 Anti-steganalysis performance analysis

In this section, we will assess the performance of our information hiding method on anti-steganalysis, which is an important factor when evaluating an information hiding method. An excellent information hiding algorithm should be hard to be detected in steganalysis. We use the steganography detection method in [3] to evaluate the performance on anti-steganalysis of our method with three different embedding strength and two compared algorithms.

For each video sequence, we first compress it using HEVC, and then decode it to get the sample data with no hidden information, marked as positive samples. Then, we encode the video sequences again using our method with three different embedding strength and two compared algorithms to hide information. After decoding these video sequences, we get the samples data, marked as negative samples. At last, we use the steganalysis method in [3] to extract feature vectors and construct detector to implement steganalysis. The result is listed in Table 2. According to it, the accuracy of all the methods is about 50% which equals to guess. This is because the PUs selected from the satisfied PUs are smallest, of which the change caused by modified is very small. Therefore, our method has an effective anti-steganalysis performance.

5.6 Algorithm complexity analysis

In this section, we analyze the embedding/extracting complexity in the encoding/decoding process. According to Algorithm 1 and Algorithm 2, the secret information embedding/extracting process requires little extra storage, so the space complexity can be ignored.

Table 1 Average embedding quantity of decoded video sequences with differernt steganography methods

Seq	SM	EBs	MVCs	AEQ	Seq	SM	EBs	MVCs	AEQ
	LSB	194	99	1.96		LSB	1202	591	2.03
	Parity	97	52	1.87		Parity	614	309	1.99
Cactus	ME1.0	170	73	2.33	People-On-Street	ME1.0	1074	494	2.17
	ME0.5	112	52	2.15		ME0.5	562	259	2.17
	ME0.2	48	21	2.29		ME0.2	246	117	2.10
	LSB	108	56	1.93		LSB	156	84	1.86
	Parity	54	31	1.74		Parity	85	46	1.85
Basket-ball-Drive	ME1.0	82	40	2.05	Traffic	ME1.0	104	47	2.21
	ME0.5	64	31	2.06		ME0.5	40	17	2.35
	ME0.2	46	21	2.19		ME0.2	20	8	2.50
	LSB	34	16	2.13		LSB	98	47	2.09
	Parity	17	7	2.43		Parity	49	27	1.81
BQ-Terrace	ME1.0	26	12	2.17	China-Speed	ME1.0	84	37	2.27
	ME0.5	18	8	2.25		ME0.5	48	20	2.40
	ME0.2	6	2	3.00		ME0.2	22	10	2.20
	LSB	110	56	1.96		LSB	148	74	2.00
	Parity	55	27	2.04		Parity	72	34	2.12
Kimo-no1	ME1.0	80	32	2.50	Race-Horses	ME1.0	118	58	2.03
	ME0.5	58	27	2.15		ME0.5	64	29	2.21
	ME0.2	26	11	2.36		ME0.2	14	5	2.80
	LSB	70	38	1.84		LSB	42	20	2.10
	Parity	35	17	2.06		Parity	23	11	2.09
Park-Scene	ME1.0	62	28	2.21	Keiba	ME1.0	26	12	2.17
	ME0.5	40	19	2.11		ME0.5	12	4	3.00
	ME0.2	14	5	2.80		ME0.2	5	2	2.5
	LSB	116	54	2.15		LSB	88	36	2.44
	Parity	58	26	2.23		Parity	44	25	1.76
Tennis	ME1.0	108	50	2.16	Party-Scene	ME1.0	78	34	2.29
	ME0.5	52	25	2.08		ME0.5	48	21	2.29
	ME0.2	34	15	2.27		ME0.2	14	5	2.80

For each embedding, it is actually a process that modifies the value of a motion vector component to another value by adding 1 or subtracting 1, which is executed N times at most. The time complexity of (6) is O(N), so the time complexity of the embedding process is O(N^2) and that of the extracting process is O(N). We calculate the statistical distribution of the number of motion vectors in each CTU, and find that the number of the motion vectors in 99.96% CTUs is less than 30, so the embedding/extracting process has very low time complexity. Table 3 shows the average time cost results per frame of the encoding process with different methods. The experiments are carried out on a PC with Intel Xeon E5-2620 v3@2.40 GHz CPU and 16G RAM running under Windows 7 and Microsoft Visual Studio 2013. We also measure the time cost of the decoding process per frame which takes only several seconds and the extra time cost caused by the extracting is close to 0. It can be seen

Table 2 Accuracy of information hiding detection of compressed video with different steganography methods

Sequence	LSB	Parity	ME1.0	ME0.5	ME0.2
Cactus	51.00%	51.25%	50.00%	50.50%	49.75%
BasketballDrive	50.61%	48.98%	46.39%	51.03%	47.14%
BQTerrace	52.00%	51.00%	49.75%	50.25%	49.00%
Kimono1	51.00%	51.75%	50.75%	49.75%	49.75%
ParkScene	51.25%	51.75%	50.75%	50.50%	49.75%
Tennis	50.00%	51.00%	49.75%	49.75%	49.25%
PeopleOnStreet	51.37%	51.10%	49.85%	49.75%	49.65%
Traffic	52.00%	51.25%	50.75%	51.75%	49.25%
ChinaSpeed	50.93%	51.00%	49.65%	48.75%	49.15%
RaceHorses	51.37%	51.00%	49.75%	50.26%	48.17%
Keiba	51.00%	50.3%	51.75%	50.75%	49.25%
PartyScene	50.36%	51.00%	51.45%	50.50%	48.35%
Average	51.07%	50.96%	50.05%	50.30%	49.04%

that the additional time of the methods is less than 2 seconds in Table 3, which illustrates that all the embedding methods have very low time complexity.

From the above results and analyses, it can see that our method has better performance on additional video quality distortion, video bitrate increment and embedding efficiency. The anti-steganalysis is effective and the complexity of the proposed method is very low. In total, we provide a high efficient information hiding method for the new video coding standard HEVC.

Table 3 Average time cost (s) of the encoding process with different steganography methods

Sequence	CLEAB	LSB	Parity	ME1.0	ME0.5	ME0.2
Cactus	160.477	161.056	160.526	162.294	161.585	160.995
BasketballDrive	207.821	208.656	208.326	209.460	206.818	207.703
BQTerrace	135.534	136.482	135.144	137.306	137.147	136.005
Kimono1	212.200	213.676	213.755	214.188	213.412	212.710
ParkScene	143.755	144.555	144.247	144.012	144.168	143.638
Tennis	249.657	250.526	250.460	251.358	250.838	249.494
PeopleOnStreet	558.061	560.735	558.722	560.848	559.681	558.506
Traffic	284.380	286.717	285.307	286.346	286.238	285.615
ChinaSpeed	82.147	83.055	82.994	84.055	82.656	82.357
RaceHorses	56.955	57.704	57.461	58.346	57.917	57.355
Keiba	41.884	42.589	42.332	42.761	42.461	42.562
PartyScene	52.136	53.482	52.705	53.686	53.364	52.610

6 Conclusion

A new information hiding method is proposed for the latest video coding standard HEVC in this paper. This method implements efficient information hiding through introducing motion vector space. For the N motion vector components of CTU selected as carrier, a $2N + 1$-ary number can be embedded by modifying at most one component of them. Furthermore, embedding strength, secret key and PU selection rule which are configurable are introduced to further improve the concealment of information hiding algorithm. Experiments shows that our method has good performance in multiple aspects and provides a solid foundation for some applications such as HEVC video content protection and covert communication based on HEVC video stream. Further to improve the embedding capacity of our method is our research focus in the following study.

Acknowledgements This work is supported partly by National Natural Science Foundation of China under grant U1636113, and partly by the Scientific Research Foundation Project of Haikou Laboratory, Institute of Acoustics, Chinese Academy of Sciences.

References

1. Ali MA, Edirisinghe EA (2011) Multi-layer watermarking of H.264/AVC video using Differential Expansion on IPCM blocks. In: Proceedings of the 2011 IEEE international conference on consumer electronics, pp 53–54
2. Aly H (2011) Data hiding in motion vectors of compressed video based on their associated prediction error. IEEE Trans Inf Forensic Secur 6(1):14–18
3. Cao Y, Zhao X, Feng D (2012) Video steganalysis exploiting motion vector reversion-based features. IEEE Signal Process Lett 19(1):35–38
4. Chang PC, Chung KL, Chen JJ, Lin CH, Lin TJ (2014) A dct/dst-based error propagation-free data hiding algorithm for hevc intra-coded frames. J Vis Commun Image Represent 25(2):239–253
5. Dai YJ, Zhang LH, Yang YX (2003) A new method of MPEG video watermarking technology. In: International conference on communication technology proceedings, pp 1845–1847
6. Dutta T, Gupta HP (2016) A robust watermarking framework for High Efficiency Video Coding (HEVC)–Encoded video with blind extraction process. J Vis Commun Image Represent 38:29–44
7. Fallahpour M, Shirmohammadi S, Ghanbari M (2015) A high capacity data hiding algorithm for H. 264/AVC video. Secur Commun Netw 8(16):2947–2955
8. Fang DY, Chang LW (2006) Data hiding for digital video with phase of motion vector. In: Proceedings of the international symposium on circuit and systems, pp 1422–1425
9. Feng G, Wu GZ (2011) Motion vector and mode selection based fragile video watermarking algorithm. Proceedings of the 2011 IEEE international conference on anti-counterfeiting, security and identification, pp 73–76
10. Filler T, Judas J, Fridrich J (2011) Minimizing additive distortion in steganography using syndrome-trelliscodes. IEEE Trans Inf Forensic Secur 6(3):920–935
11. Guo Y, Pan F (2010) Information hiding for H.264 in video stream switching application. In: Proceedings of the 2010 IEEE international conference on information theory and information security, pp 419–421
12. Hu Y, Zhang CT, Su YY (2007) Information hiding based on intra prediction modes for H.264/AVC. In: IEEE international conference on multimedia and expo, pp 1231–1234
13. Jiang B, Yang G, Chen W (2015) A cabac based hevcc video steganography algorithm without bitrate increase. J Comput Inf Syst 11(6):2121–2130
14. Jordan F, Kutter M, Ebrahimi T, Jordan F (1997) Proposal of a watermarking technique for hiding/retrieving data in compressed and decompressed video. Technical Report M2281, ISO/IEC document, JTCI/SC29/WG11
15. Kapotas SK, Skodras AN (2009) Real time data hiding by exploiting the IPCM macroblocks in H.264/AVC streams. Real-Time Image Process 4(1):33–41
16. Kapotas SK, Skodras AN (2007) Data hiding in H.264 encoded video sequences. In: Proceedings of the IEEE 9th workshop on multimedia signal, pp 373–376

17. Kapotas SK, Skodras AN (2008) A new data hiding scheme for scene changing detection in H.264 encoding video sequence. In: Proceedings of the 2008 In: IEEE international conference on multimedia and expo, pp 277–280
18. Li Y, Chen H, Zhao Y (2010) A new method of data hiding based on H.264 encoded video sequences. In: Proceedings of the 2010 IEEE 10th international conference on signal processing, pp 1833–1836
19. Liao K, Ye D, Lian S, Guo Z, Wang J (2009) Lightweight Information Hiding in H.264/AVC Video Stream. International conference on multimedia information networking and security, pp 578–582
20. Liu CH, Chen TC (2008) Data hiding in inter and intra prediction modes of H.264/AVC. In: Proceedings of the 2008 IEEE international symposium on circuits and systems, pp 3025–3028
21. Mansouri A, Aznaveh AM, Torkamani-Azar F, Kurugollu F (2010) A low complexity video watermarking in H.264 compressed domain. IEEE Trans Inf Forensic Secur 5(4):649–659
22. Mehmood N, Mushtaq M (2012) Blind watermarking scheme for H.264/AVC based on intra 4x4 prediction modes. Fut Inform TechnoL Appl Servicer Lect Notes Electr Eng 179:1–7
23. Mobasseri BG, Raikar YN (2007) Authentication of H.264 streams by direct watermarking of CAVLC blocks. Proc SPIE - Int Soc Opt Eng 6505:65051W-1-5
24. Sullivan GJ, Ohm JR, Han WJ, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circ Syst Video Technol 22(12):1649–1668
25. Sun YX, Zhan RJ, Yang Y, Niu H, Wang XA (2017) An information hiding method based on context-based adaptive variable length coding. Int J Innov Comput Appl 8(1):50–57
26. Swaraja K, Latha YM, Reddy VSK, Paramkusam AV (2011) Video watermarking based on motion vectors of H.264. In: Proceedings of the 2011 annual IEEE india conference, pp 1–4
27. Tian L, Zheng N, Xue J, Xu T (2008) A CAVLC-based blind watermarking method for H.264/AVC compressed video. In: IEEE Asia-Pacific services computing conference, pp 1295–1299
28. Tew YQ, Wong K (2014) Information hiding in HEVC standard using adaptive coding block size decision. In: IEEE international conference on image processing, pp 5502–5506
29. Wang P, Zhang ZD, Li L (2008) A video watermarking scheme based on motion vectors and mode selection. In: Proceedings of the 2008 international conference on computer science and software engineering, pp 233–237
30. Wang JJ, Wang RD, Li W, Xu DW, Huang ML (2014) A Large-capacity information hiding method for HEVC video. Atlantis Press, pp 934–937
31. Wang JJ, Wang RD, Li W, Xu DW, Huang ML (2014) An information hiding algorithm for HEVC based on intra prediction mode and block code. Sensors Transducers 117(8):230–237
32. Wang JJ, Wang RD, Xu DW, Li W (2015) An information hiding algorithm for HEVC based on angle differences of intra prediction mode. J Softw 10(2):213–221
33. Wu GY, Yang XY, Niu K (2011) One real-time video information hiding method based on the CAVLC encoding. In: Proceedings of the 8th China conference on information and communications security, pp 45–50
34. Xu DW, Wang RD, Wang JC (2012) Prediction mode modulated data-hiding algorithm for H.264/AVC. J Real-Time Image Process 7(4):205–214
35. Xu DW, Wang RD, Wang JC (2010) Low complexity video watermarking algorithm by exploiting CAVLC in H.264/AVC. In: IEEE international conference on wireless communications, networking and information security, pp 411–415
36. Xu CY, Ping XJ, Zhang T (2006) Steganography in compressed video stream. In: Proceedings of the 1st international conference innovative computer information control, pp 269–272
37. Xu J, Wang RD, Huang ML, Wang JJ, Xu DW (2015) An information hiding algorithm for HEVC based on intra-prediction modes and hamming+1. J Comput Inf Syst 11(15):5587–5598
38. Yang GB, Li JJ, He YL, Kang ZW (2011) 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
39. Yamadera S, Wada N, Hangai S (2010) A study on light information hiding method compatible with conventional H.264/AVC decoder. In: Proceedings of the 2010 2nd European workshop visual information processing, pp 210–213
40. Yao YZ, Zhang WM, Yu NH, Zhao XF (2015) Defining embedding distortion for motion vector-based video steganography. Multimed Tools Appl 74(24):11163–11186
41. Zhang H, Cao Y, Zhao XF (2016) Motion vector-based video steganography with preserved local optimality. Multimed Tools Appl 75(21):13503–13519
42. Zhang J, Maitre H, Li JG, Zhang L (2001) Embedding watermark in MPEG video sequence. In: Proceeding of the 2003 international conference on communication technology, pp 1845–1847
43. Zhang X, Au OC, Pang C, Dai W (2013) Additional sign bit hiding of transform coefficients in HEVC. In: IEEE international conference on multimedia and expo workshops, pp 1–4

44. Zhu HL, Wang RD, Xu DW (2010) Information hiding algorithm for H264 based on the motion estimation of quarter-pixel. In: Proceedings of the 2nd international conference future computer communication, pp 423–427
45. Zhu HL, Wang RD, Xu DW, Zhou XX (2010) Information hiding algorithm for H.264 based on the prediction difference of intra_4×4. In: Proceedings of the 2010 3rd international congress on image and signal processing, pp 487–490
46. Zou D, Bloom JA (2010) H.264 stream replacement watermarking with CABAC encoding. In: Proceedings of the 2010 IEEE international conference on mutimedia and expo, pp 117–21

Jie Yang is a Ph.D. candidate in the Haikou Laboratory, Institute of Acoustics, Chinese Academy of Science, Haikou, China. He received his BS degree from Beijing Normal University in 2013, majoring in electronic science and technology. His research interests include multimedia signal processing and data hiding.

Songbin Li received the Ph.D. degree from the Institute of Acoustics, Chinese Academy of Science, in 2010. And then, he did his postdoctoral research at the Department of Electronic Engineering, Tsinghua University. From 2012, he is an Associate Professor in the Haikou Laboratory, Institute of Acoustics, Chinese Academy of Science, Haikou, China. His research interests focus on multimedia signal processing and information forensics.