

Article

High Capacity HEVC Video Hiding Algorithm Based on EMD Coded PU Partition Modes

Zhonghao Li ¹, Laijin Meng ¹, Xinghao Jiang ^{2,*} and Zhaohong Li ^{1,3,*}

¹ School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100091, China

² School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

³ Guangdong Provincial Key Laboratory of Information Security Technology, Guangzhou 510000, China

* Correspondence: xhjiang@sjtu.edu.cn (X.J.); zhgli2@bjtu.edu.cn (Z.L.)

Received: 22 June 2019; Accepted: 30 July 2019; Published: 6 August 2019



Abstract: Data hiding in videos has been a big concern as their rich redundancy can be used for embedding a lot of secret information. Further, as high efficiency video coding (HEVC) introduces many innovative technologies compared with the previous standard, H.264, it has gradually become the mainstream. Therefore, it is valuable to develop new information hiding algorithms by using novel features of HEVC. A HEVC video data hiding algorithm based on prediction unit (PU) partition modes from inter prediction process is proposed in this paper. Firstly, code units (CUs) in two sizes of 8×8 and 16×16 are selected for embedding, then the PU partition modes in these CUs are coded by a spatial coding method. After that, two specific hiding algorithms by modifying coded PU partition modes in CUs of 8×8 and 16×16 are proposed, respectively. Experimental results show that the proposed algorithm has achieved excellent performance with high visual quality, and high embedding capacity and low bitrate increase in both high- and low-resolution videos compressed with different quantization parameters (QPs). Compared with the state-of-the-art work, the proposed algorithm achieves a much higher capacity while keeping quite high visual quality with little increase of bitrate.

Keywords: video data hiding; HEVC; high capacity; PU partition modes; EMD coding

1. Introduction

Information hiding is one of the research hotspots in the field of information security. Secret transmission of important information is achieved by embedding secret information into the carrier in a special way. Since the existence of secret information has little influence on the carrier itself, the security of the secret information has been improved and it is difficult for an attacker to find it out. The main carriers for data hiding include texts, audios, images and videos. As it is limited by the bandwidth of the network, it is difficult to transmit the original data of multimedia files. Therefore, it is necessary to compress them to fit the bandwidth. The mainstream compression standard for videos nowadays is H.264, which was firstly published in 2003. Ten years later, the High Efficiency Video Coding (HEVC) standard was published. Compared with H.264, HEVC videos only require half of the space under the same visual quality conditions, and the HEVC standard is more efficient at resolutions of 1080P and above. In the future, H.264 will be replaced by HEVC because of the wide popularity of high-definition (HD) and ultra-high-definition (ultra HD) videos. When the time comes, some of the information hiding algorithms for Moving Picture Experts Group (MPEG-4) and H.264 videos would not work for HEVC videos. Therefore, it is valuable to develop new information hiding algorithms for HEVC videos with the latest technology of data compression.

Initially, the video information hiding algorithms for MPEG-4 and H.264 videos are very similar to those for images. As an important method for image and video loss compression, discrete cosine transform (DCT) maps pixels to DCT coefficients, so embedding secret information in transform domain is the first concern. Information hiding algorithms [1–8] can be used, which embed secret information by modifying quantized DCT coefficients. Besides, a large quantity of information regarding hiding algorithms in the compression domain has been proposed [9–19]. One common method is to modify motion vectors [9,19]. Modifying intra prediction modes [13–17] is also popular. In addition, there are some algorithms based on modifying dividing macroblocks [10,12] and the replacement of code words [14].

Since the basic coding principle of HEVC is similar to the existing video compression coding standards, for HEVC video data hiding, some similar modification methods in the DCT domain as for MPEG-4 and H.264 videos are adopted. For example, Chang et al. [20,21] proposed a HEVC video information hiding algorithm in the DCT transform domain, which solved the problem of distortion drifting. Feng et al. [22] proposed a reversible information hiding algorithm by modifying the intermediate frequency coefficients of 8×8 blocks. The authors in [23,24] also proposed an algorithm by modifying the DST coefficients of 4×4 luminance blocks. Later, the special features of HEVC video besides those grafting from traditional information hiding algorithms were investigated. Wang et al. [25–30] proposed a series of information hiding algorithms based on the intra prediction mode, which modified the features extracted from the intra prediction process of I-pictures in a specific strategy.

In addition to the intra prediction mode, the inter prediction mode of HEVC is also significantly innovative compared with H.264. However, only a few data hiding algorithms that embedded information in inter prediction process have been proposed [31–33]. In 2014, Tew et al. [31] had proposed a video information hiding algorithm by modifying the PU partition modes of inter prediction for the first time. Until 2018, the information hiding algorithm based on inter prediction mode was noticed again by [32,33]. Reference [32] proposed an information hiding algorithm based on motion vector space encoding. Reference [33] proposed a multilevel information hiding algorithm based on PU partition modes, which is considered to be one of the most important innovative features of HEVC. Obviously, this kind of information hiding algorithm based on the inter prediction mode [31–33] has the merit of quite high visual quality. The PSNR of modified video sequences is nearly the same as that of the clean ones compressed without information hiding, which is the notable advantage of these kinds of hiding algorithms compared to other algorithms based on transform domain or intra prediction mode. However, low embedding capacity has occurred since the modification of motion vectors [32] and PU partition modes [31,33] are performed in units of CU, and the number of CUs in P-pictures is limited. Motivated by the advantage of high visual quality and disadvantage by the low capacity of these kinds of hiding algorithms, a novel data hiding algorithm based on the space coding of PU partition modes in the inter prediction mode is proposed.

As illustrated above, the low capacity is caused by the limited number of CUs, so it is the first issue to enlarge the number of CUs that can be used for embedding data. After investigating the distribution of the number of CUs with four different sizes, the CUs of two sizes 8×8 and 16×16 are selected as they account for the dominant number of CUs. Then the PU partition modes in these selected CUs are coded by a spatial coding method called exploiting modification direction (EMD) [34] to further increase the capacity. Finally, by modifying the PU partition modes for embedding secret data, the proposed algorithm has greatly improved the capacity while keeping high visual quality with little increase of bitrate.

The rest of the paper is organized as follows. Section 2 briefly reviews the prediction units in HEVC and investigates the distribution of CUs. In Section 3, the EMD coding of the PU partition modes and the proposed data hiding algorithm are described in detail. Experimental results are shown in Sections 4 and 5 and conclude the paper.

2. EMD Encoding of PU Partition Modes

2.1. Overview of Prediction Units in HEVC

The quadtree partition structure of HEVC is one of the key technologies to improve its compression efficiency, which is shown in Figure 1. HEVC chooses to use block-based coding, in which the coding tree unit and coding unit partition can be changed according to different image content. An image to be encoded in HEVC is firstly divided into coding tree units (CTUs) of uniform size, closeness, and they are non-overlapping. According to the selected compression configuration, the maximum size of it can be 64×64 pixels. The CTU can be divided into smaller code units (CUs) through the quadtree structure. The rule is that if the CTU is to be divided, it must be divided into four CUs with the size of half CTU. If the CU needs to be divided further, the same rules must be followed. However, the number of layers is limited. The smallest CU size is 8×8 , and the largest CU size is the same as CTU, which is 64×64 .

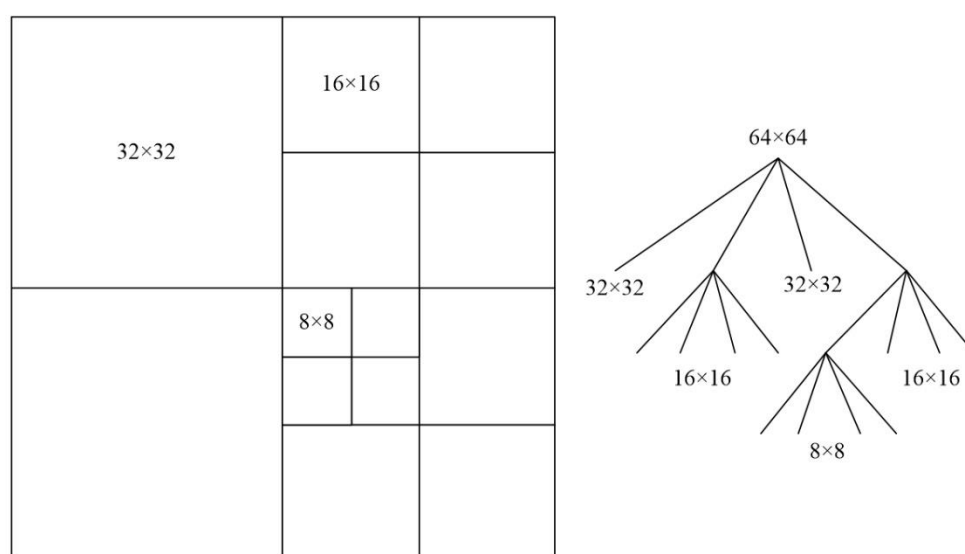


Figure 1. Example of quadtree partition structure.

When the process of CUs partition is completed, it is necessary to predict the partitioned units. The prediction unit (PU) is the basic unit for performing block prediction, and its size can be the same as the size of the CU in which it resides, or it can be partitioned further. What is more, the PU can be divided into many types, which may be symmetrical or asymmetrical.

For a CU using the intra prediction mode, there are two types of PU partition, which are the same partition size as the CU and half of the CU size as shown in Figure 2. As for the inter prediction mode, HEVC adopts eight different types of divisions for CUs, which are shown in Figure 3. For a CU of 16×16 or bigger, the PU partition mode may select any one of the seven types except the $N \times N$ partition mode. Since the prediction residual is mainly generated by the motion of objects, the residual will be smaller if the partition of the PU blocks can better fit the boundary. HEVC calculates the residuals generated by various PU partition modes and selects the most appropriate partition mode for prediction. For the CU of 8×8 , in order to reduce the complexity of the HEVC algorithm, it is only allowed to be divided into four types, which are $2N \times 2N$, $N \times 2N$, $2N \times N$, and $N \times N$.

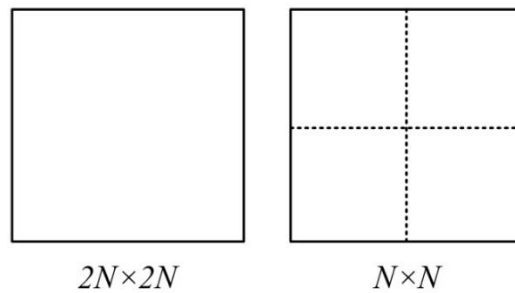


Figure 2. Prediction unit (PU) partition modes for intra-prediction.

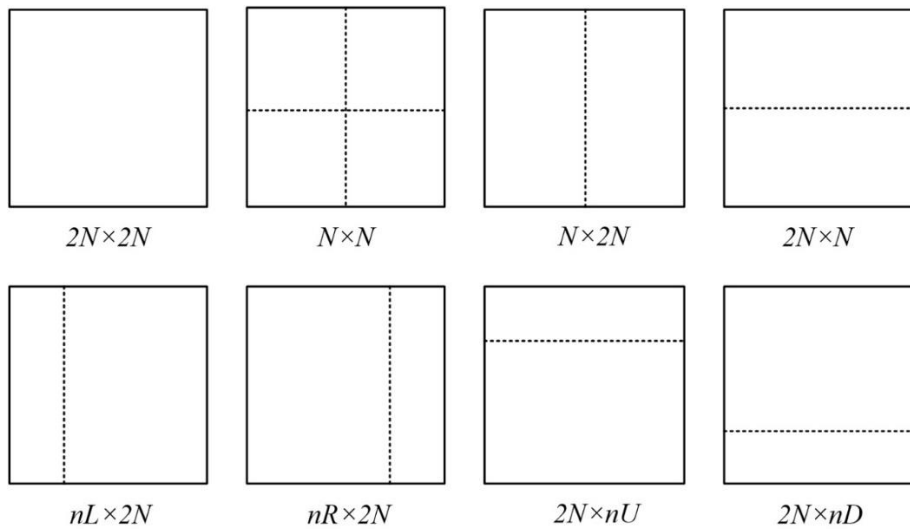


Figure 3. PU partition modes for inter-prediction.

2.2. The Distribution of CUs with Different Sizes

Eventually, for HEVC inter prediction, the partition of CUs and the selection of optimal PU partition modes are intended to reduce prediction residuals. For a stationary object in the video, the residual of the inter prediction is approximately 0, so the CU size will be large, and the PU partition mode will be simple. But for moving objects, there are often more prediction residuals in the boundary, so more precise CU and PU partition modes are required to adapt to the more accurate shape of the moving target, which results in CUs with smaller sizes. Figure 4 shows two CU partitioning examples at different resolutions of 1920×1080 and 832×480 , where the green line represents the boundary of the CTU, the white line represents the boundary of the CU, and the red line represents the PU partition boundary, which is different from the size of the CU. It can be seen from Figure 4 that although their resolutions are different, the still parts of a large piece such as the tree trunk in Figure 4a and the background in Figure 4b, the size of CUs is 64×64 . However, as for the boundary of moving objects, such as the horse in Figure 4b, the number of CUs of 8×8 is dominant. Thus, the partition size of CU and PU in a frame are closely related to the video content.

In this paper, the proposed algorithm is going to embed the secret information by modifying PU partition modes which resides in CUs. There is only one PU partition mode in one CU, so a larger embedding capacity requires more CUs, but it means more optimal PU partition modes will be modified, which will increase the prediction residual that leads to a decline of visual quality and an increase of bitrate. Therefore, it is necessary to analyze the distribution of CUs with different sizes, which can supervise the video hiding algorithm to achieve a balance of capacity, visual quality and bitrate increase by selecting appropriate CU sizes for embedding.

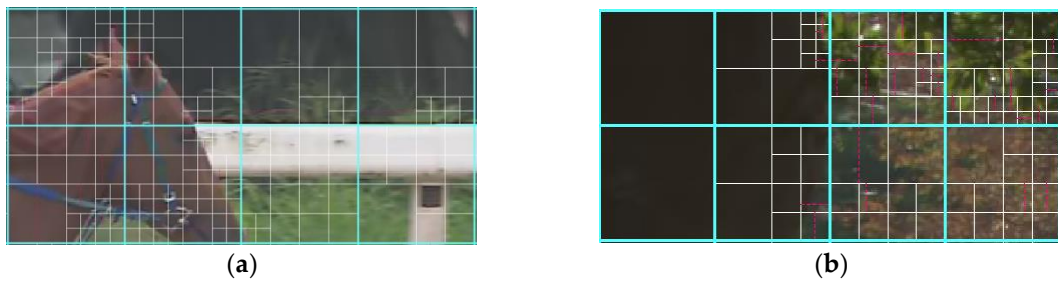


Figure 4. Code units (CU) structures and PU partition modes of the example slice. (a) is from ParkScene 1920×1080 and (b) is from Keiba 832×480 .

Therefore, we use seven different videos with different resolutions of 1920×1080 and 832×480 to get the distribution of CUs in four sizes which are 8×8 , 16×16 , 32×32 , and 64×64 as shown in Figure 5. It can be observed that at either resolution, except for sequence Kimono, the quantity of CUs of 8×8 size is the most, while the number of CUs of 64×64 size is the least. The phenomenon occurs because even the video picture is spatially redundant, a large piece of the same content is still limited, which results in the percentage of 64×64 CUs below 10%. What is more, in the HEVC partition process, a 64×64 CU can be divided into four CUs of 32×32 , and one 32×32 CU can be divided into four CUs of 16×16 , and each 16×16 CU can be divided into four CUs of 8×8 , thus accumulating down, CUs of these small partition sizes are increased by several times. Consequently, in most of different videos with different resolutions as shown in Figure 5, the number of 8×8 CUs or 16×16 CUs is the most. To be more specific, the number of 8×8 CUs and 16×16 CUs accounts for more than 80% of total quantity of CUs in most sequences. Two factors are taken into consideration. First, the dominant number of 8×8 and 16×16 CUs benefits for large embedding capacity. Second, the content of image block with smaller size of CUs is more complicated than that of larger size of CUs, so modifying the PU partition modes in smaller size CUs is supposed to cause less distortion to video visual quality. Therefore, we select CUs of 8×8 and 16×16 for embedding secret information to achieve high capacity as well as high visual quality.

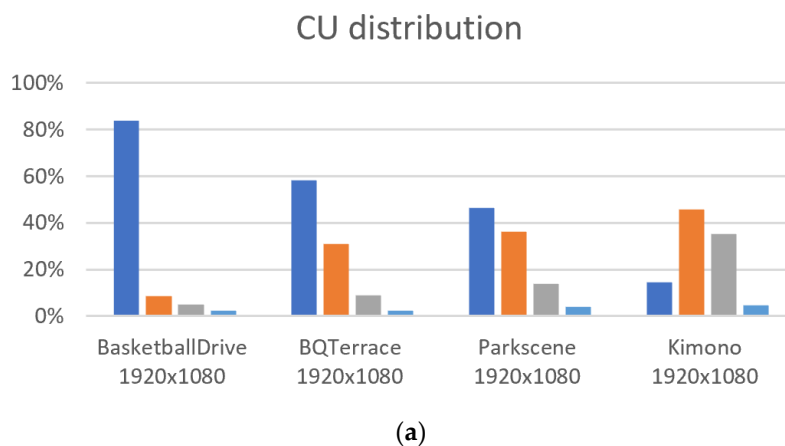
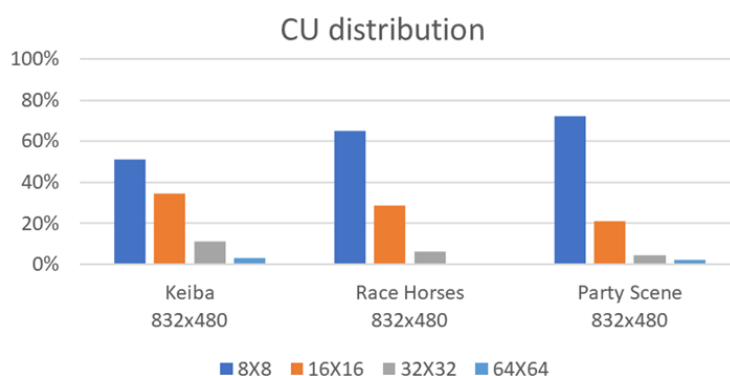


Figure 5. Cont.



(b)

Figure 5. The distribution of CUs with different sizes. (a) CU distribution under high resolution; (b) CU distribution under low resolution.

3. The Proposed Hiding Algorithm

Based on determination of the CU sizes in Section 2.2, the PU partition modes of 8×8 and 16×16 CUs are firstly mapped to be integers and then coded by a spatial coding method called EMD. Then a video data hiding algorithm based on the spatial coding of PU partition modes is proposed. The EMD coding of PU partition modes and the procedure of modifying them will be illustrated in the following sections.

3.1. EMD Coding of PU Partition Mode

Firstly, we assume that a CTU has M CUs, and divide all the CUs into a series of CU-groups, each of which containing S ($S \leq M$) CUs. Then the assigned PU partition mode of each CU is mapped to be an integer according to Tables 1 and 2. It can be seen that, in a CU size of 16×16 , 32×32 and 64×64 , the eight possible PU partition modes are mapped into integers ranging from 0 to 7, and in a CU size of 8×8 , the four possible PU partition modes are mapped into integers ranging from 0 to 3. After that, for each CU-group, a S -tuple of integers can be obtained and denoted as (p_1, p_2, \dots, p_S) . For example, as shown in Figure 6, this CTU has been divided into one 32×32 CU and twelve 16×16 CUs, and if S is set to be 3 and only 16×16 CUs are adopted to embed the secret message, then we can get four 3-tuple groups of CUs (count it from left to right and up to down) in this CTU. According to the mapping rule in Table 1, four corresponding S -tuple arrays $(0, 0, 0)$, $(0, 0, 0)$, $(0, 0, 5)$ and $(2, 0, 0)$ can be obtained.

Table 1. Mapping of PU partition modes in 16×16 , 32×32 , and 64×64 CUs.

Modes	$2N \times 2N$	$N \times N$	$N \times 2N$	$2N \times N$	$nL \times 2N$	$nR \times 2N$	$2N \times nU$	$2N \times nD$
Integers	0	1	2	3	4	5	6	7

Table 2. Mapping of PU partition modes in 8×8 CUs.

Modes	8×8	4×4	4×8	8×4
Integers	0	1	2	3

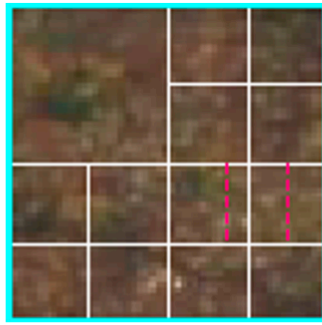


Figure 6. Magnified coding tree units (CTU) in the bottom-right corner from Figure 4a.

Based on the S -tuple of integers (p_1, p_2, \dots, p_S) , another S -tuple of base $(2S + 1)$ array (g_1, g_2, \dots, g_S) can be found by Equation (1).

$$g_i = p_i \bmod (2S + 1) \tag{1}$$

Then, an S -dimensional space called PU partition modes space can be built up, as g_i is the corresponding i -th dimensional coordinate of the space. Obviously, arbitrary S -tuple (g_1, g_2, \dots, g_S) can be mapped as a point in the S -dimensional space, and each point's value $f(g_1, g_2, \dots, g_S)$ in space lattice can be calculated by Equation (2),

$$f(g_1, g_2, \dots, g_S) = \left[\sum_{i=1}^S (g_i \cdot i) \right] \bmod (2S + 1) \tag{2}$$

where $[]$ means rounding operation. For example, if $S = 2$, the range of the space coordinate will be $\{0, 1, 2, 3, 4\}$ according to Equation (1). Secondly, according to Equation (2), we can obtain the points' values in the 2-dimensional lattice as shown in Figure 7.

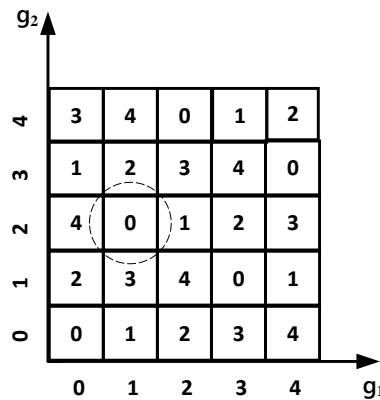


Figure 7. The points' values in the 2-dimensional lattice.

From Figure 7, we can see that any point's value and its $2S$ neighbors, which are integers within $[0, 2S]$, are mutually different, and the point's value can be assigned to be any $(2S + 1)$ base number by modifying at most one element in the S -tuple of (g_1, g_2, \dots, g_S) . For example, if $(g_1, g_2) = (1, 2)$, then $f(g_1, g_2) = 0$, as shown in the circle of Figure 7. When the data to be embedded is 3, we can make the modified point value 3 by only modifying g_2 from 2 to 1. Consequently, in a S -tuple group of CUs, only by modifying at most one PU partition mode in one CU, $\log_2(2S + 1)$ secret bits can be embedded. What is more, by EMD space coding of PU partition modes, the embedding capacity can be increased a lot. However, how to decide the proper parameter S for the selected CUs of 8×8 and 16×16 is an important issue, and it will be discussed in next section.

3.2. Information Hiding Algorithm

Equal importance should be given to deciding the number S for EMD coding of PU partition modes, because there exist eight possible PU partition modes in 16×16 CUs, but only four in 8×8 CUs. To be more specific, we assume that the proposed algorithm sets a larger S' rather than a smaller S . As a result, a S' -tuple group of CUs has more CUs, which means that by modifying the PU partition mode of one CU, $\log_2(2S' + 1)$ secret bits can be embedded instead of $\log_2(2S + 1)$ bits, so that higher capacity efficiency can be achieved. Apart from this, what should be clear is that the CU structure of the CTU will not change before and after information hiding. So, with the fixed quantity of CUs in a video sequence, the larger S' , the fewer S' -tuple groups of CUs. Because only one optimal PU partition mode of one CU will be modified in a group, less visual quality distortion will be caused with less PU partition modes being modified.

However, the larger S' not only has advantages as mentioned above, but also leads to capacity reduction and limitation. As for the capacity reduction, it can be explained by Equations (3) and (4), where Q represents the total quantity of CUs in a video sequence. Equation (3) represents the total capacity of the information hiding algorithm, Equation (4) is derived from Equation (3), which indicates that with S increasing, the total capacity will decrease.

$$\text{Capacity} = \frac{Q \times \log_2(2S + 1)}{S} \quad (3)$$

$$\frac{d\text{Capacity}}{dS} = Q \times \frac{\frac{S}{(2S+1) \times \ln 2} - \log_2(2S + 1)}{S^2} < 0, S >= 1 \quad (4)$$

What is more, in the process of EMD coding, it is required that the value range of p_i has to be larger than $2S + 1$ so that the base $(2S + 1)$ integer g_i can be realized and every point $f(g_1, g_2, \dots, g_S)$ of the S -dimensional space can be reached. As the PU partition mode is mapped to be p_i , so the number of possible PU partition modes in a CU has to be larger than $2S + 1$. Besides, the distribution of 8×8 and 16×16 CUs is also taken into consideration. As shown in Figure 5, at either resolution, the quantity of 8×8 CUs is the most and quantity of 16×16 is the second most. Consequently, to balance all the effects of S discussed above, S is set to be 3 for 16×16 CUs, so that higher hiding efficiency and visual quality can be obtained, and capacity will not be low because of the second most quantity of 16×16 CUs. For 8×8 CUs, S is set to be 1 in case the visual quality decreases too much and the capacity can be assured with the largest ratio of 8×8 CUs and small S .

As illustrated above, different EMD coding parameters and mapping are determined for PU partition modes in 8×8 CUs and 16×16 CUs. Then in general, before data-embedding, the binary secret message should be conveniently converted into base $(2S + 1)$ number. The binary secret message can be divided into many pieces that each of them contains L bits, and the value of each secret piece can be represented by K digits in a base $(2S + 1)$ notational system. The principle can be proved from Equation (5).

$$2^L \leq (2S + 1)^K \quad (5)$$

Based on Equation (5), L can be determined by Equation (6).

$$L = \lfloor K \cdot \log_2(2S + 1) \rfloor \quad (6)$$

where $\lfloor \cdot \rfloor$ means rounding down. For instance, if the binary message to be embedded is (00000 10001 11111), they can be converted into base 7 data as (00 23 45), where $L = 5$ and $K = 2$.

After that, a sequence of secret data of base $(2S + 1)$ can be achieved. Each time when a secret number is to be embedded, the PU partition mode of one CU in the corresponding S -tuple group of CUs will be modified. The rule of modifying the PU partition mode is to make the mapping value $f(g_1, g_2, \dots, g_S)$ of the modified S -tuple group of CUs equal to the secret data. Based on this rule, two

specific information hiding algorithms for CU size of 8×8 and 16×16 will be illustrated in Algorithms 1 and 2, respectively.

Algorithm 1. The embedding algorithm for one S-tuple group of 16×16 CUs with $S = 3$

Input: The 3-tuple 16×16 CU group array $P = (p_1, p_2, p_3)$, where p_i represents the mapping integer of PU partition mode in the i th 16×16 CU decided by HEVC according to Table 1.

Hidden base 7 data d ;

Output: the modified 3-tuple 16×16 CU group array $P' = (p'_1, p'_2, p'_3)$;

1. $P' = P$
 2. $f = p'_1 + 2 \times p'_2 + 3 \times p'_3$
 3. **if** $(f \% 7) + 1 == d$ **then**
 4. $p'_1 = p'_1 + 1$
 5. **else if** $(f \% 7) + 2 == d$ **then**
 6. $p'_2 = p'_2 + 1$
 7. **else if** $(f \% 7) + 3 == d$ **then**
 8. $p'_3 = p'_3 + 1$
 9. **else if** $(f \% 7) - 1 == d$ **then**
 10. $p'_1 = p'_1 - 1$
 11. **else if** $(f \% 7) - 2 == d$ **then**
 12. $p'_2 = p'_2 - 1$
 13. **else if** $(f \% 7) - 3 == d$ **then**
 14. $p'_3 = p'_3 - 1$
 15. **end if**
-

Algorithm 2. The embedding algorithm for one S-tuple group of 8×8 CUs with $S = 1$

Input: The 1-tuple 8×8 CU group array $P = (p_1)$, where p_1 represents the mapping integer of the PU partition mode in this 8×8 CU decided by HEVC according to Table 2.

Hidden base 3 data d ;

Output: the modified 1-tuple 8×8 CU group array $P' = (p'_1)$;

1. $P' = P$
 2. $f = p'_1$
 3. **if** $(f \% 3) + 1 == d$ **then**
 4. $p'_1 = p'_1 + 1$
 5. **else if** $(f \% 3) - 1 == d$ **then**
 6. $p'_1 = p'_1 - 1$
 7. **end if**
-

Based on Algorithms 1 and 2, the specific process of embedding information is given below:

Step 1: For a CTU, the CU division depth and the selected PU partition modes are recorded during the first-round calculation where the HEVC encoder is calculating the optimal CU structure by default.

Step 2: If the optimal CU structure for the CTU does not contain any 8×8 or 16×16 CUs, the default HEVC encoding process will be applied, and the next CTU will be dealt with following Step 1. Otherwise, go to Step 3.

Step 3: The PU partition modes of CUs in size of 8×8 or 16×16 will be modified following Algorithms 1 and 2 respectively.

Step 4: The default HEVC coding process will be applied for this CTU. What is different from the first-round calculation is that some CUs have different PU partition modes from those that the CUs are assigned in the first-round calculation, because they are modified to embed data, and the leftover CUs have the PU partition modes that are the same as those they got in the first-round calculation.

While extracting the embedded data, for 16×16 CUs, firstly the 3-tuple group of CUs are got by decoding the encoded video. Then the 3-tuple array (d_1, d_2, d_3) are mapped according to Table 1, After

that, $f(g_1, g_2, g_3)$ can be calculated by Equations (1) and (2). Finally, the embedded secret data can be directly extracted as it equals to the point value $f(g_1, g_2, g_3)$. When it comes to 8×8 CUs, the secret data extraction procedure is the same as that of 16×16 CUs illustrated above, except for the parameter S that is 1 for 8×8 CUs instead of 3 for 16×16 CUs.

4. Experimental Results

4.1. Configuration

In this subsection, the experimental configuration will be given. The video codec platform is HM 16.15. The proposed method has been applied to nine common video sequences including five sequences with a resolution of 1920×1080 : Tennis, ParkScene, Kimono, BQTerrace and BasketballDrive; three sequences with resolution 832×480 : Keiba, RaceHorse, PartyScene; one sequence with resolution 1024×768 : ChinaSpeed. The encoding settings are as follows: group of pictures (GOP) structure is "IPPP" with GOP size 4, and the frame rate is 25 fps. For each video, they are encoded for three times with different quantization parameters (QPs), which are 26, 32 and 38.

4.2. Performance Evaluation

4.2.1. Subjective Visual Quality

It is using human eyes to recognize the distortion of the video that achieves the subject assessment. This assessment is necessary because until now, the subjective evaluation can still not fully reflect the subjective impression on the human brain. Hence, to illustrate that the proposed hiding algorithm cannot be detected subjectively by human eyes, the subjective visual quality of the video sequences with embedded information will be given in this part.

It can be seen from Figure 8 that no matter the high-resolution sequence ParkScene or the low-resolution sequence Keiba, the reconstructed slice by the proposed method does not have mosaics, dislocation or visual distortions compared to the original slice. Therefore, it is hard for human eyes to distinguish the embedded videos.

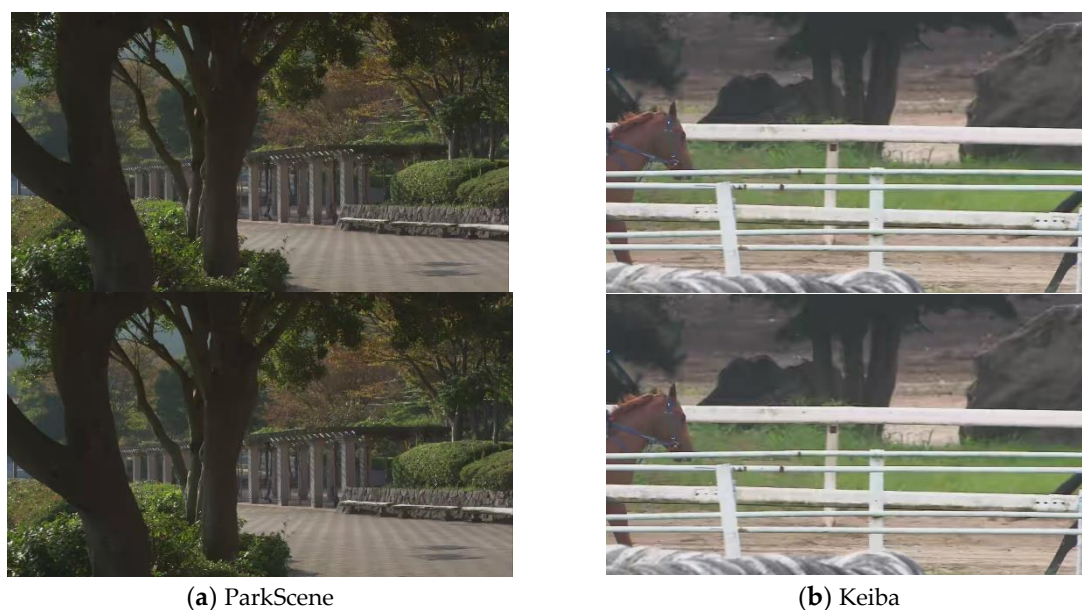


Figure 8. Subjective performance evaluation of the second P-slices from: (a) ParkScene 1920×1080 ; (b) Keiba 832×480 . The clean slices without embedded information are laid on the first line, and slices with hidden information are placed in the second line.

4.2.2. Objective Performance

To evaluate the objective performance of the proposed data hiding algorithm, three representative parameters of the embedded videos are adopted. The first one is peak signal-to-noise ratio (PSNR). PSNR is where the signal in this case is the original slice, and the noise is the error introduced by compression and data hiding. It is the common method to estimate the visual quality of the video sequences.

The next significant evaluation method is bit rate increasing (BRI). As the proposed data hiding is achieved by modifying the optimal PU partition modes, which will result in more prediction redundancy, and further lead to the increasing of bitrate. In many cases, videos' bit stream needs to be transmitted or stored or broadcast through the internet, so less video bitrate is better for internet transmission rate. For this consideration, it is necessary to measure the bitrate increase produced by data hiding. BRI is defined as Equation (7), where BR' is the bitrate of video with embedded information and BR means the bitrate of the video without modification in HEVC encoding process.

$$BRI = \frac{BR' - BR}{BR} \quad (7)$$

Thirdly, the embedding capacity is assessed because it is the direct performance of a data hiding algorithm to show how much information the proposed algorithm can embed into the videos. Embedding capacity is defined as the number of bits that have been embedded into the video sequence per frame. Because of the effectiveness and significance of the three parameters mentioned above, the experimental results with multi QPs of the proposed method are given in Table 3.

In Table 3, PSNR(clean) is the PSNR of the coded video sequences without information hiding, and PSNR(hiding) means the PSNR of the sequences with information embedded. The result shows that the PSNR remains high even after data hiding as the average PSNR(hiding) is 36.0322 dB and the PSNR(hiding) of most sequences is above 36.0000 dB. What is more, compared to PSNR(clean), the PSNR(hiding) is approximately the same as PSNR(clean) and the difference between them is mostly within 0.1 dB. The reason behind the little PSNR decrease can be explained as follows.

It is during the coding process that the PU modification was applied, so the motion vector, DCT coefficients and redundancy will be calculated again based on the modified PU partition modes with CU structure of CTU unchanged. What is more, PU partition modes of 32×32 CUs, 64×64 CUs are not modified, as well as two third of 16×16 CUs because in a 3-tuple group of CUs, only the PU partition mode of one CU will be modified according to the proposed S -dimensional space coding method. As a result, the error caused by the proposed algorithm will be eliminated to a very low level. Therefore, the visual quality experiences nearly no distortion through the proposed information hiding algorithm.

Additionally, PSNR varies in a wide range for the same video sequence with multi QPs, and for all sequences, the PSNR follows the same trend in that it decreases with the QP going up. The phenomenon occurs because as the quantization configuration QP increases, more details of the video to be encoded will be lost during quantization. However, no matter what the QP or video content is, it can be seen from the data that the PSNR(hiding) is always close to the PSNR(clean). Therefore, the video sequences with data hidden by the proposed algorithm still keep high visual quality.

As shown in Table 3, the BRI value of the proposed algorithm is limited to a small range that the maximum BRI is only 0.0462 occurring in Racehorse with QP = 26. Because BRI is a ratio value of bitrate difference between the compressed clean video and the compressed hiding video, and both of them are compressed with the same QP, the BRI value only indicates bitrate influence brought by the modified PU partition modes and does not has a positive correlation with QP like PSNR. What is more, it appears to be always small and close to each other that the average BRI of all sequences is 0.0094, which shows the small bitrate influence of the proposed algorithm. Like the cause of PSNR performance, the calculation of prediction redundancy is applied based on the PU partition modes that just those in 8×8 CUs and a third of 16×16 CUs have been modified. Consequently, the proposed method introduces a small BRI rate.

Table 3. The peak signal-to-noise ratio (PSNR), bit rate increasing (BRI) and capacity of the proposed algorithm with different quantization parameters (QPs).

Sequence	QP	PSNR(Clean)	PSNR(Hiding)	BRI	Capacity
BasketballDrive 1920 × 1080	26	39.9836	39.9576	0.0209	2252
	32	38.1135	38.0798	0.0226	926
	38	35.9492	35.9018	0.0063	350
BQTerrace 1920 × 1080	26	36.9362	36.9268	0.0148	7578
	32	34.6276	34.6540	0.0059	977
	38	32.1492	32.1668	0.0025	141
Kimono 1920 × 1080	26	41.0887	41.1022	0.0112	1702
	32	38.3209	38.3373	0.0140	790
	38	35.5277	35.5761	0.0150	313
ParkScene 1920 × 1080	26	38.9657	38.9486	0.0162	4783
	32	36.1111	36.1024	0.0108	1396
	38	33.4029	33.4146	0.0056	280
Tennis 1920 × 1080	26	40.5594	40.5679	0.0155	3226
	32	38.0411	38.0633	0.0110	1254
	38	35.5259	35.5689	0.0022	419
ChinaSpeed 1024 × 768	26	41.2659	41.1669	0.0410	2656
	32	36.9406	36.8492	0.0272	1427
	38	33.1107	33.0431	0.0143	651
Keiba 832 × 480	26	38.7105	38.8770	−0.0191	1148
	32	35.5753	35.7236	−0.0165	601
	38	32.6737	32.8271	−0.0330	199
Racehorses 832 × 480	26	37.0262	36.9092	0.0462	2558
	32	33.6613	33.5330	0.0392	1282
	38	30.5866	30.5369	0.0399	547
PartyScene 832 × 480	26	36.5585	36.5190	−0.0368	2032
	32	32.6259	32.5839	−0.0153	1070
	38	28.9680	28.9322	−0.0069	243

For some sequences like Keiba and PartyScene, even though a specific frame introduces a little more redundancy after modification, this modified frame has a stronger correlation with the following frame, and in inter-prediction mode, stronger correlation results in less redundancy of several continuous frames in a second. Because BRI represents the bit number increase per second, and the frame rate is 25 frames per second, if the redundancy of 25 frames becomes smaller after modification, the BRI will be negative.

Embedding capacity represents the performance of the data hiding algorithm directly. In Table 3, the value of capacity is the number of bits that have been embedded per frame, so it is easy to assess the embedding ability regardless of (GOP) size and the number of frames in test video sequences. The average capacity for all sequences with multi QPs is 1511 bits, which presents a high embedding capacity. It shows a tendency that more bits can be embedded into high resolution sequences than that in low resolution sequences, as well as more bits in a video with low QP than that with high QP. This is because high resolution and a small quantization parameter QP lead to the CU structure that contains more CUs in small sizes, which means more PU partition modes can be modified for embedding more secret information, thus higher embedding capacity can be achieved by video sequences with high resolution or small QP.

As illustrated above, PSNR, BRI and Capacity represent the performance of the proposed method in three aspects. But each of these three parameters has a restriction on the other ones. Hence, a comprehensive data hiding algorithm is supposed to balance the performance of these three aspects. For instance, if the PU partition modes of bigger CUs in size of 32×32 or 64×64 are adopted to be

modified to hide information, the capacity will increase with no doubt, but because of not choosing the optimal division mode, the PSNR will decrease as a result of visual quality distortion, and BRI will go up as the redundancy is introduced by modifying more and bigger CUs. Therefore, in order to balance the three assessments, the proposed algorithm adopted a proper modification category, and has achieved outstanding performance. It can be seen from Table 3 that by only modifying CUs in small sizes in the proposed algorithm, the PSNR and BRI has been strained to an excellent level, while the capacity is insured to be high because of the large quantities of 8×8 CUs and 16×16 CUs.

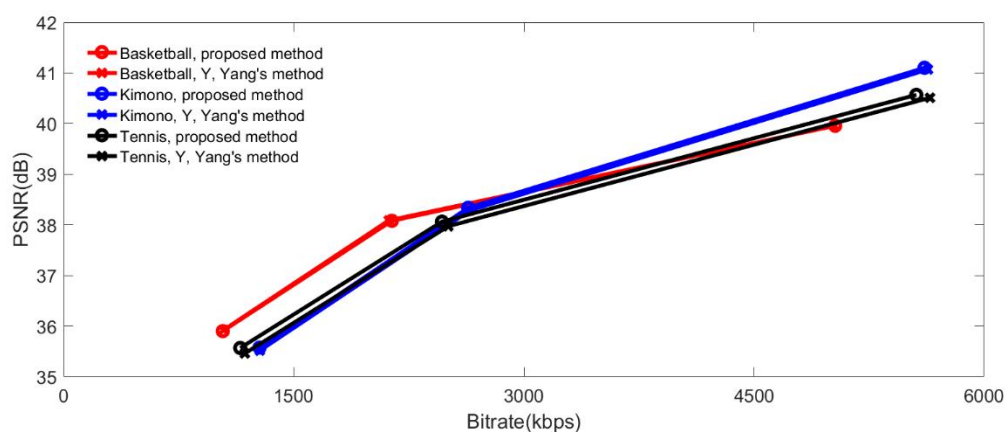
The video sequence Kimono can be taken as a special example to represent the correlation of CU distribution with algorithm performance. As shown in Figure 5, the quantity of 16×16 CUs of Kimono is the most and the percentage of 32×32 CUs is 38%, which is different from the most sequences. The data in Table 3 show that the capacity of Kimono with the CU distribution shown in Figure 5 has the smallest capacity 1702 bits among 1920×1080 video sequences, which is the result of the smaller proportion of CUs in size of 8×8 and 16×16 compared to that of the others. However, the capacity stays high with more than 1500 bits with excellent PSNR and BRI performance.

The computation time per frame of sequence RaceHorses 832×480 , ChinaSpeed 1024×768 and ParkScene 1920×1080 is 72 s, 97 s and 276 s, respectively with QP = 38 and that of sequence ChinaSpeed 1024×768 is 97 s, 120 s and 157 s when QP is set 26, 32 and 38 correspondingly. The computation time per frame exhibits an increasing trend with higher resolution or smaller QP because more details of a sequence take more time to encode, but whatever the QP or the resolution is, the computation time reaches an acceptable level.

4.2.3. Comparative Analysis

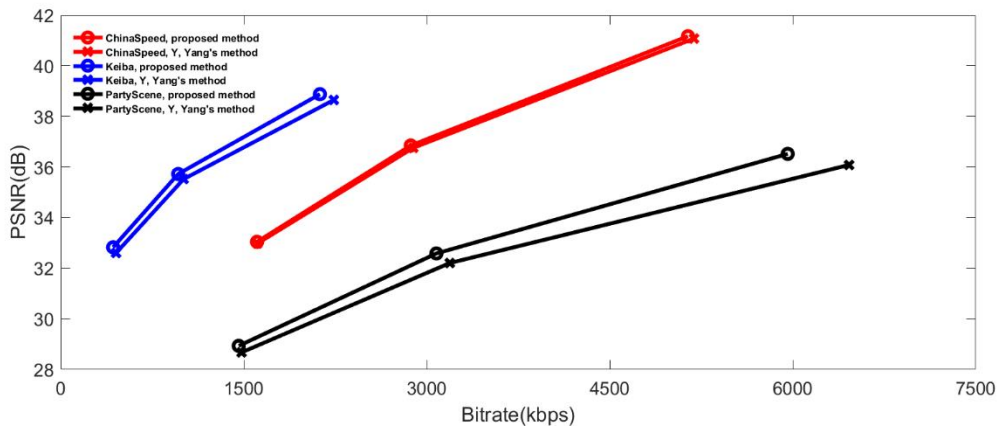
In this section, the comparison of the proposed algorithm with [33] and [32] will be given, because these two algorithms have the similar principle that the inter-frame prediction feature is adopted to embed secret message. Meanwhile, both papers were published in 2018, which represent the latest and most outstanding level of data hiding in HEVC videos.

Even though the performance on the same test sequences was given by [33], the bitrate used to compress the video is different, which brings about difficulty in comparison. Therefore, the plots of bitrates vs distortion (RD) and bitrates vs capacity (RC) are used to compare the performance between the proposed algorithm and that of [33]. The RC and RD plots on sequences with multi resolutions are shown in Figures 9 and 10.



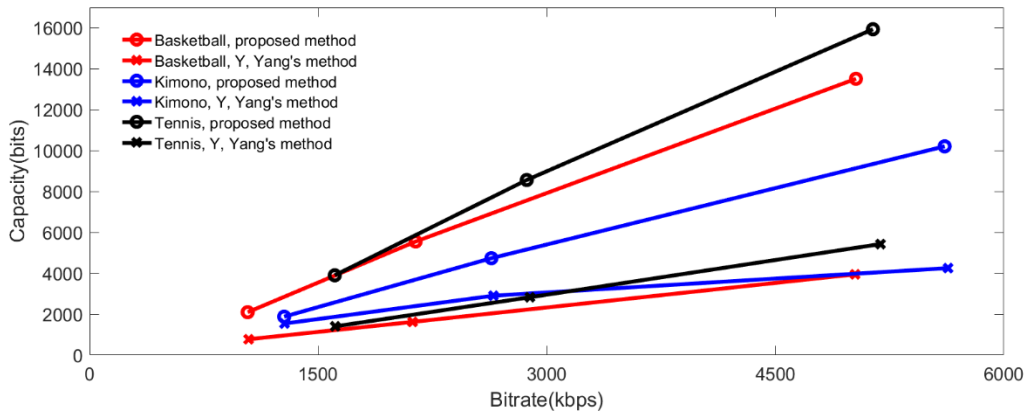
(a) RD with Y. Yang's method under high resolution.

Figure 9. Cont.

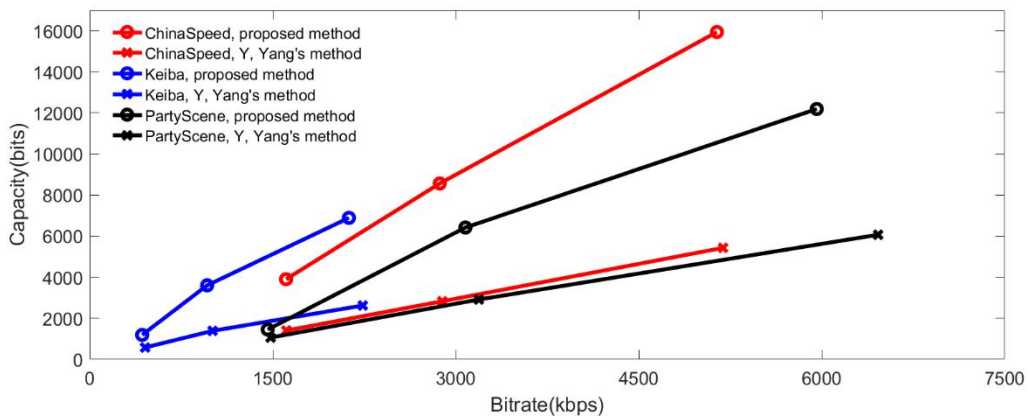


(b) RD with Y. Yang's method under low resolution.

Figure 9. Comparison of distortion (RD) with Y. Yang's paper ([33]): (a) RD of BasketballDrive 1920 × 1080, Kimono 1920 × 1080 and Tennis 1920 × 1080; (b) RD of ChinaSpeed 1024 × 768, Keiba 832 × 480 and PartyScene 832 × 480.



(a) RC with Y. Yang's method under high resolution.



(b) RC with Y. Yang's method under low resolution.

Figure 10. Comparison of capacity (RC) with [33]: (a) RC of BasketballDrive 1920 × 1080, Kimono 1920 × 1080 and Tennis 1920 × 1080; (b) RC of ChinaSpeed 1024 × 768, Keiba 832 × 480 and PartyScene 832 × 480.

Figure 9 shows the PSNR comparison of the [33] algorithm and the proposed algorithm with different bitrates. The reference [33] algorithm has four levels, and the PSNR of each one is almost the same while the capacity is quite different, so Level 4 is chosen to be compared with the proposed algorithm because it has the maximum capacity. It can be seen from Figure 9 that PSNR of the proposed algorithm is higher than that of [33], whatever the bitrate or the resolution is. This is caused by the algorithm that only the 8×8 CUs and one third of 16×16 CUs have been modified instead of all sizes of CUs in the [33] algorithm. Therefore, with more CUs divided following the optimal PU partition mode, visual quality distortion introduced by the proposed method is smaller than in [33].

In order to perform a fair comparison, the RC plot of Level 4 of [33] is shown in Figure 10 as Level 4 has the strongest embedding strength. As shown in the RC plots, the capacity of the proposed algorithm is much more than that of the [33] algorithm under the same resolution or bitrate. For the [33] algorithm, considering the sharp drop of the visual quality, the PU partition mode $2N \times 2N$ is not adopted to embed information. But the proposed algorithm only modifies one CU in a S -tuple group of CUs while guaranteeing the quantity of bits to be embedded. Therefore, even though the $2N \times 2N$ PU partition modes are used as secret message carriers in the proposed algorithm, the visual quality still stays at a high level. Because of the large quantities of $2N \times 2N$ PU partition modes, the capacity of the proposed algorithm can be much greater than that of [33].

As for the comparison with [32], because experimental data of [32] are under only one QP, the PSNR and capacity comparison illustration has been carried out in Table 4. It demonstrates that on most video sequences, the proposed method outperforms PSNR better than [32]. Apart from this, when it comes to the capacity, the proposed algorithm has achieved dozens of times the capacity of that of [32], as well as exceeding in PSNR. What is more, even though the capacity is much larger than [32], the BRI of the proposed algorithm is still limited to the low level, similar to [32], as shown in Figure 11.

Table 4. The PSNR and capacity with QP = 26.

Sequence	PSNR(Hiding)	PSNR (Ref. [32])	Capacity	Capacity (Ref. [32])
BasketballDrive 1920 × 1080	39.9576	40.1833	2252	82
BQTerrace 1920 × 1080	36.9268	37.6133	7578	26
Kimono 1920 × 1080	41.1022	39.7700	1702	80
ParkScene 1920 × 1080	38.9486	37.8867	4783	62
Tennis 1920 × 1080	40.5679	40.0667	3226	108
ChinaSpeed 1024 × 768	41.1669	38.4333	2656	84
Keiba 832 × 480	38.8770	38.9333	1148	26
Racehorses 832 × 480	36.9092	35.5000	2558	118
PartyScene 832 × 480	36.5190	34.0667	2032	78

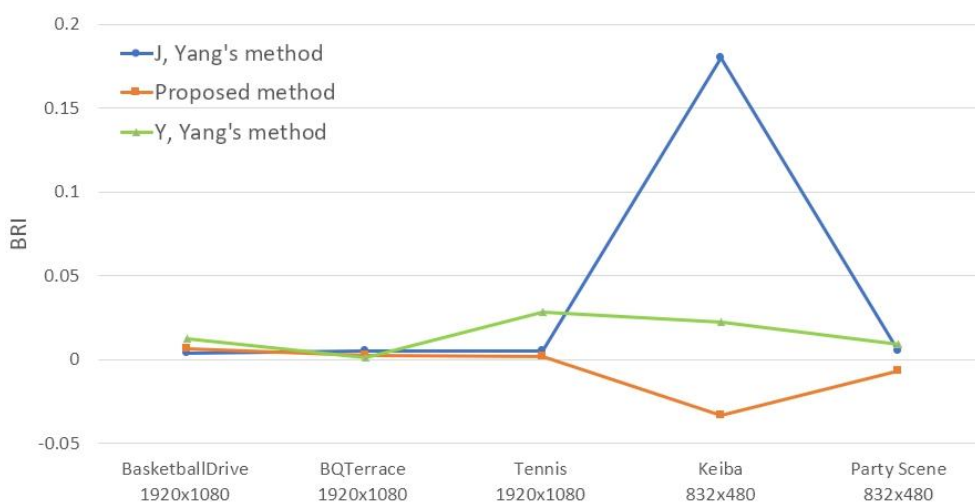


Figure 11. BRI comparison with Yang and Li ([32]) and Yang et al. ([33]).

5. Conclusions

In this paper, we have proposed a novel data hiding algorithm for HEVC videos. After comprehensively analyzing the distribution of CUs, the PU partition modes in the selected CUs are firstly coded by EMD method and then modified for embedding secret data according to hiding Algorithms 1 and 2, respectively. Experimental results show that the video sequences with data embedded achieve almost the same excellent visual quality as that of sequences without data hiding, and the increase of bitrate caused by data hiding is limited to an acceptably low level. The biggest contribution of the proposed algorithm is to overcome the drawback of low capacity that happens in the kind of hiding algorithms based on the inter prediction mode and the capacity of the proposed algorithm has been increased dozens of times. In future, we will study other HEVC coding modules and look for more potential features for data hiding to further improve the embedding capacity.

Author Contributions: Z.L.; Data curation, Z.L.; Formal analysis, Z.L. and L.M.; Investigation, Z.L.; Methodology, Z.L. and X.J.; Software, Z.L.; Supervision, X.J. and Z.L.; Visualization, L.M. and Z.L.; Writing—original draft, Z.L. and L.M.; Writing—review and editing, Z.L. and Z.L.

Funding: Part of the work was supported by the National Natural Science Foundation of China (No. 61702034). Part of the work was supported by the Opening Project of Guangdong Province Key Laboratory of Information Security Technology (Grant No. 2017B030314131). Part of this work is funded by National Natural Science Foundation of China (Grant No.61572320 & 61572321).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yuan, K.G.; Zhang, R.; Xin, X.U.; Niu, X.X.; Yang, Y.X. Design of steganography algorithm based on MPEG-4 coding scheme. *J. Commun.* **2009**, *30*, 48–53.
2. Xiao, Z.; Chen, Z.; Zhang, X. Applications. Issues and solution on distortion drift in reversible video data hiding. *Multimed Tools Appl.* **2011**, *52*, 465–484.
3. Esen, E.; Alatan, A.A. Robust Video Data Hiding Using Forbidden Zone Data Hiding and Selective Embedding. *IEEE Trans. Circuits Syst.* **2011**, *21*, 1130–1138. [[CrossRef](#)]
4. Liu, Y.; Li, Z.; Ma, X.J. Reversible Data Hiding Scheme Based On H.264/AVC without Distortion Drift. *J. Softw.* **2012**, *7*, 1059–1065. [[CrossRef](#)]
5. Liu, Y.; Ju, L.; Hu, M.; Ma, X.; Zhao, H. A robust reversible data hiding scheme for H.264 without distortion drift. *Neurocomputing* **2015**, *151*, 1053–1062. [[CrossRef](#)]
6. Liu, Y.; Chen, L.; Hu, M.; Jia, Z.; Jia, S.; Zhao, H. A reversible data hiding method for H.264 with Shamir's (t, n)-threshold secret sharing. *Neurocomputing* **2016**, *188*, 63–70. [[CrossRef](#)]

7. Li, H. Research on Large Capacity H.264/AVC Video Information Hiding Algorithm in QDCT Domain. Master's Thesis, Southwest Jiaotong University, Chengdu, Sichuan, China, 19 May 2017.
8. Duan, N. Research on Robust Video Information Hiding Algorithm for Reducing H.264/AVC Distortion Drift. Master's Thesis, Xi'an University of Technology, Xian, Shaanxi, China, May 2018.
9. Fang, D.Y.; Chang, L.W. Data hiding for digital video with phase of motion vector. In Proceedings of the IEEE International Symposium on Circuits & Systems, Island of Kos, Greece, 21–24 May 2006; pp. 1422–1425.
10. Kapotas, S.K.; Varsaki, E.E.; Skodras, A.N. Data Hiding in H. 264 Encoded Video Sequences. In Proceedings of the IEEE Workshop on Multimedia Signal Processing, Chania, Greece, 1–3 October 2007; pp. 373–376.
11. Shanableh, T. Data Hiding in MPEG Video Files Using Multivariate Regression and Flexible Macroblock Ordering. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 455–464. [[CrossRef](#)]
12. Jia, C. Research of Video Information Hiding Based on H.264. Master's Thesis, University of Electronic Science and Technology of China, Chengdu, Sichuan, China, 28 June 2013.
13. Yang, G.; Li, J.; He, Y.; Kang, Z. Communications. An information hiding algorithm based on intra-prediction modes and matrix coding for H.264/AVC video stream. *AEU Int. J. Electron. Commun.* **2011**, *65*, 331–337. [[CrossRef](#)]
14. Xu, D.; Wang, J. Prediction mode modulated data-hiding algorithm for H.264/AVC. *J. Real Time Image Process.* **2012**, *7*, 205–214. [[CrossRef](#)]
15. Zhou, G.; Wu, M. Information hiding algorithm based on H.264 intra prediction mode. *Telecommun. Inf.* **2014**, *5*, 44–47.
16. Wei, F. Study on Information Hiding Based on Intra Prediction for Video. Master's Thesis, Southwest Jiaotong University, Chengdu, Sichuan, China, 8 April 2015.
17. Zhang, Y.; Zhang, M.; Niu, K.; Yang, X. Novel video steganographic algorithm based on modifying intra-prediction mode. *J. Zhengzhou Univ.* **2018**, *50*, 23–28.
18. Xu, D.; Wang, R.; Shi, Y. Data Hiding in Encrypted H.264/AVC Video Streams by Codeword Substitution. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 596–606. [[CrossRef](#)]
19. Su, Y.; Zhang, X.; Zhang, C.; Jing, Z.J. Steganography Algorithm Based on Motion Vectors of H.264. *J. Tianjin Univ.* **2014**, *47*, 67–73.
20. Chang, P.C.; Chung, K.L.; Chen, J.J.; Lin, C.H.; Lin, T.J. An error propagation free data hiding algorithm in HEVC intra-coded frames. In Proceedings of the Signal & Information Processing Association Summit & Conference, Kaohsiung, Taiwan, China, 29 October–1 November 2013.
21. Chang, P.C.; Chung, K.L.; Chen, J.J.; Lin, C.H.; Lin, T.J. A DCT/DST-based error propagation-free data hiding algorithm for HEVC intra-coded frames. *J. Vis. Commun. Image Represent.* **2013**, *25*, 239–253. [[CrossRef](#)]
22. Gui, F.; Xue, H. A Reversible Data Hiding Scheme for HEVC. In Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design, Hangzhou, China, 9–10 December 2017.
23. Liu, Y.; Liu, S.; Zhao, H.; Si, L. A new data hiding method for H.265/HEVC video streams without intra-frame distortion drift. *Multimed. Tools Appl.* **2018**, *78*, 6459–6486. [[CrossRef](#)]
24. Liu, S.; Liu, Y.; Feng, C.; Zhao, H. A Reversible Data Hiding Method Based on HEVC Without Distortion Drift. In Proceedings of the Intelligent Computing Methodologies: 13th International Conference, Liverpool, UK, 7–10 August 2017; pp. 613–624.
25. Wang, J.; Wang, R.; Li, W.; Xu, D.; Yan, D. An information hiding algorithm for HEVC based on intra prediction. *J. Optoelectron. Laser* **2014**, *25*, 1578–1585.
26. Wang, J.; Wang, R.; Li, W.; Xu, D.; Huang, M. A High-capacity Information Hiding Algorithm for HEVC Based on Intra Prediction Mode Intra-prediction in HEVC. *J. Softw.* **2014**, *10*, 8933–8943.
27. Wang, J.; Wang, R.; Li, W.; Xu, D.; Xu, J. Video information hiding in intra prediction and block codes for HEVC. *J. Optoelectron. Laser* **2015**, *26*, 942–950.
28. Wang, J. Study on Information Hiding Algorithm Based on Intra Prediction Modes for HEVC Video. Master's Thesis, Ningbo University, Ningbo, Zhejiang, China, 15 June 2015.
29. Xu, J.; Wang, R.; Huang, M. An information hiding algorithm for hevc based on intra-prediction modes and Hamming. *J. Comput. Inf. Syst.* **2015**, *11*, 5587–5598.
30. Qi, S.; Wang, R.; Pei, A.; Wang, B. An Information Hiding Algorithm for HEVC Based on Differences of Intra Prediction Modes. In Proceedings of the Cloud Computing and Security—2nd International Conference, Nanjing, China, 29–31 July 2016.

31. Tew, Y.; Wong, K.S. Information hiding in HEVC standard using adaptive coding block size decision. In Proceedings of the IEEE International Conference on Image Processing, Paris, France, 27–30 October 2014.
32. Yang, J.; Li, S. An efficient information hiding method based on motion vector space encoding for HEVC. *Multimed. Tools Appl.* **2018**, *77*, 11979–12001. [[CrossRef](#)]
33. Yang, Y.; Li, Z.; Xie, W.; Zhang, Z. High capacity and multilevel information hiding algorithm based on pu partition modes for HEVC videos. *Multimed. Tools Appl.* **2019**, *78*, 8423–8446. [[CrossRef](#)]
34. Zhang, X.; Wang, S. Efficient Steganographic Embedding by Exploiting Modification Direction. *IEEE Commun. Lett.* **2006**, *10*, 781–783. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).