

Article

Two-Dimensional Histogram Shifting-Based Reversible Data Hiding for H.264/AVC Video

Yuzhang Xu and Junhui He * 

School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China; regenttsui@gmail.com

* Correspondence: hejh@scut.edu.cn

Received: 10 April 2020; Accepted: 9 May 2020; Published: 13 May 2020

Abstract: Histogram shifting (HS) has been proved to be a great success in reversible data hiding (RDH). To reduce the quality loss of marked media and the increase in file size, several two-dimensional (2D) HS schemes based on the characteristics of cover media have been proposed recently. However, our analysis shows that the embedding strategies used in these methods can be further optimized. In this paper, two new 2D HS schemes for RDH in H.264/AVC video are developed, one of which uses the DCT coefficient pairs with both values 0 and the other does not. The embedding efficiency of a DCT coefficient pair in different embedding modes is firstly calculated. Then, based on the obtained embedding efficiency along with the statistical distribution of DCT coefficient pairs, two better embedding strategies are proposed. The secret data is finally embedded into the pairs of DCT coefficients of the middle and high frequencies using our proposed strategies. The comparison experiment results demonstrate that our schemes can achieve enhanced visual quality in terms of PSNR, SSIM, and entropy in most cases, and the increase in file size is smaller.

Keywords: two-dimensional histogram shifting; reversible data hiding; H.264/AVC video

1. Introduction

As a special type of data hiding, RDH schemes imperceptibly embed secret data into cover media in a reversible manner, meaning the cover media can be losslessly recovered after data extraction. Due to the reversibility, RDH schemes are especially useful in the scenarios where any distortion may be unacceptable, such as military applications, medical imaging, and law enforcement. For example, the integrity check code of a video can be embedded into it to assure the video used for law enforcement has not been modified. So far, many RDH schemes have been proposed, which can be classified into three main categories: lossless compression [1,2], difference expansion [3–6] and histogram shifting [7–23]. HS-based RDH scheme was first proposed by Ni et al. [7], and it is improved for years afterwards using the histograms of difference image [8,9] or prediction errors [10–13], multiple histograms [14–17] and 2D HS [18–23].

Most aforementioned RDH schemes are only suitable for uncompressed images, and cannot be directly applied to compressed images and videos. However, compressed media such as JPEG images and H.264/AVC videos are more commonly used in daily life. Several RDH schemes have been proposed for JPEG images [24–29]. Huang et al. [24] proposed an HS-based RDH scheme for JPEG images by expanding the AC coefficients with values ± 1 . Moreover, a block selection strategy is used to adaptively choose DCT blocks for data embedding. An ordered embedding method to further reduce the increase in the file size of marked images was proposed in [25]. Subsequently, two different coefficient selection methods are proposed in [26,27] to further improve the embedding efficiency. Recently, He et al. [28] established the negative influence models of image visual distortion and file size change, which can be employed to optimize the selection of DCT blocks and coefficient frequencies.

Cheng et al. [29] proposed a 2D HS-based RDH scheme for JPEG images as well as a selection strategy based on the optimal frequency band of the DCT coefficient pairs.

For H.264/AVC video, Chung et al. [30] proposed embedding the motion vectors (MVs) into DCT coefficients using the HS method for the purpose of intra-frame error concealment. In [31], position of the last nonzero level of DCT block is used to embed secret data. Although the distortion caused by data hiding can be reduced, the embedding efficiency is not high. To avert the intra-frame distortion drift, the directions of intra-frame prediction are used in the RDH scheme of [32]. To reduce the quality distortion, a 2D HS-based RDH scheme was introduced by Xu et al. [33] to embed secret data into DCT coefficients of middle and high frequencies. A different 2D HS-based method is also proposed in [34] to improve the embedding efficiency. Kim et al. [35] proposed an RDH algorithm based on compensation, reducing the modification of DCT coefficients. Niu et al. [36] presented an algorithm based on the HS of MVs, and to further improve the embedding performance, they also presented a 2D HS-based method of MVs [37].

Although many video coding schemes [38–41] based on DCT or wavelet transform [42] have been proposed, H.264/AVC is the most commonly used video coding format. Thus, the RDH technique for H.264/AVC video is of great value. In this paper, in order to embed additional data into H.264/AVC videos, the embedding efficiency of a DCT coefficient pair in different embedding modes is firstly calculated. Then, based on the computed embedding efficiency along with the statistical distribution of DCT coefficient pairs, the defects in several 2D HS schemes are analyzed. In addition, two better embedding strategies are proposed. The secret data is finally embedded into the pairs of DCT coefficients of the middle and high frequencies using our proposed methods. The experimental results demonstrate the effectiveness of our embedding strategies. Compared with the related schemes, the marked videos of our schemes have better visual quality in most cases, and the increase in file size of them is smaller.

The remainder of this paper is organized as follows. Firstly, the 2D HS-based RDH technique is briefly reviewed in Section 2. Then, based on the analysis of several 2D HS schemes for compressed media, the proposed two 2D HS-based RDH schemes are described in detail in Section 3. The experimental results and analysis are then presented in Section 4. Finally, the conclusions are given in Section 5.

2. HS-based RDH Technique

The one-dimensional (1D) HS-based RDH technique was first developed in [7] for uncompressed images, whose main idea is briefly reviewed here. Firstly, the histogram of pixel values in an image is generated by

$$h(k) = \#\{x_i | x_i = k\}, \quad (1)$$

where # denotes the cardinal number of a set, $k \in [0, 255] \cap \mathbb{Z}$, and x_i is a pixel from the image. Then the bins between the peak and zero bins are shifted toward the zero bin by one unit, i.e.,

$$x_i = \begin{cases} x_i + 1 & \text{if } x_i \in [k_p + 1, k_z - 1], \\ x_i & \text{if } x_i \notin [k_p + 1, k_z - 1], \end{cases} \quad (2)$$

where k_p and k_z denote the pixel values of the peak and zero points of the histogram respectively, and without loss of generality, it is assumed that $k_p < k_z$. Finally, the data is embedded into the pixels by

$$x_i = \begin{cases} x_i + 1 & \text{if } x_i = k_p \ \& \ m_i = 1, \\ x_i & \text{if } x_i = k_p \ \& \ m_i = 0, \end{cases} \quad (3)$$

where m_i is one bit of secret data to be embedded. The 1D HS-based method is illustrated in Figure 1.

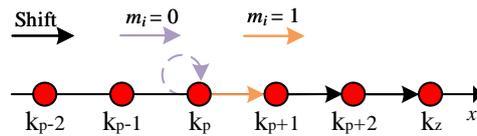


Figure 1. Illustration of 1D HS-based method.

The classic 2D HS-based RDH technique [19], which is extended from 1D HS-based method, is illustrated in Figure 2. Compared with 1D histogram, 2D histogram is generated by the statistical distribution of value pairs, so the line shown in Figure 1 is changed into a plane, as shown in Figure 2. The point (x, y) in the plane is a value pair composed of different kinds of objects (e.g., prediction errors of pixel values, transform coefficient values) used for data embedding. When the DCT coefficients are used to carry secret data, the value pair is also called a coefficient pair. Thus, the $(0,0)$ DCT coefficient pair, which will be used later in the paper, denotes a pair of DCT coefficients with both values 0. There are various ways of pairing objects. For example, two consecutive DCT coefficients in a block or two DCT coefficients from adjacent blocks with the same frequency can be paired. Each arrow in Figure 2 indicates the possible modification of the value pairs. The number of arrows ending at a certain point can be called the in-degree of the point, and the number of arrows starting with a point can be called its out-degree. Generally, the amount of the modification to a value pair will be different when the value pair is modified along different directions. For instance, when a value pair $(x, y) = (1, 0)$ is modified to $(2, 1)$ or $(2, 0)$ when $m_i = 0$ or $m_i = 1$, the corresponding amount of the modification to $(1, 0)$ is 2 or 1, respectively. For ease of discussion, the modification method of a point with a given out-degree is referred to as the embedding mode of the point, and the combination of different embedding modes is called an embedding strategy in the rest of this paper.

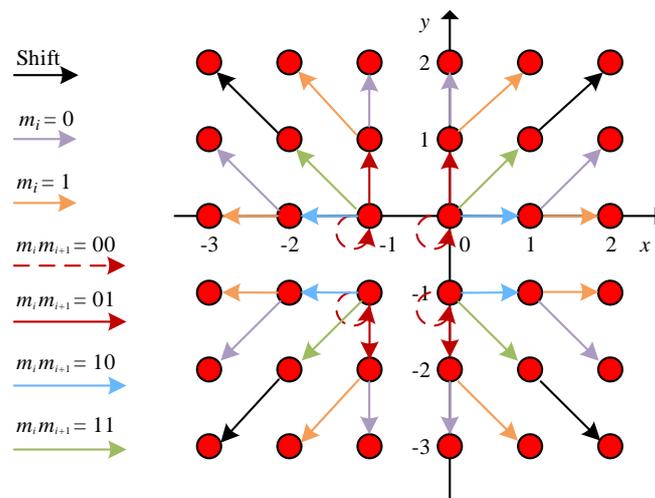


Figure 2. Classic 2D HS-based method.

Since each point can be modified with many different embedding modes, there are various embedding strategies to design a 2D HS-based scheme, which will result in different embedding efficiency. High embedding efficiency means that more data can be embedded per unit modification. For compressed media, the classic 2D HS-based method may not be efficient enough. The reason is that, unlike pixels in uncompressed images, the objects used for data embedding in the compressed domain need to be encoded. For those commonly used objects, e.g., DCT coefficients and MVs, the results of entropy coding are sensitive to their values. For example, entropy coding of DCT coefficients in H.264/AVC video is related to the coefficient values of both current block and neighboring blocks. In addition, the distribution of zero values also has a great impact on the efficiency of entropy encoding. To improve embedding efficiency, several 2D HS-based RDH schemes have recently been proposed for JPEG images [29] and H.264/AVC videos [33,34,37], which will be analyzed in the following section.

3. Proposed Schemes

In this section, we first use the embedding efficiency to analyze the embedding strategies of several 2D HS-based RDH schemes in compressed domain. Then according to the analysis results, two new 2D HS-based RDH schemes are proposed, one of which uses the (0,0) DCT coefficient pairs and the other does not.

3.1. Analysis of 2D HS-based RDH Schemes in Compressed Domain

Although our proposed scheme is general for DCT coefficients and MVs, the modification of MVs may introduce huge prediction errors, and with the increase of frame number, the error propagation will greatly degrade the quality of the video. Therefore, only the DCT coefficients are selected for embedding. The embedding efficiency of the embedding mode i related to a coefficient pair is defined as follows.

$$E_i = \frac{B_i}{V_i} = \frac{\sum_{n=1}^N p_n b_n}{\sum_{n=1}^N p_n v_n}, \tag{4}$$

where B_i is the number of bits that can be embedded into the coefficient pair with the embedding mode i , and V_i is the corresponding amount of modification to the coefficient pair. p_n is the occurrence probability of a certain modification direction n of the embedding mode i , and N is the out-degree of the DCT coefficient pair, thus, $\sum_{n=1}^N p_n = 1$. b_n is the number of bits that can be embedded through the modification direction n , and v_n is the corresponding amount of modification to the coefficient pair. In 2D HS, the shifting across a coefficient pair will introduce excessive modification, so this kind of shifting will not be considered. On this basis, the maximum out-degree of a coefficient pair is nine, including eight neighbors and the coefficient pair itself. To embed secret data, the out-degree of a coefficient pair must larger than one. The embedding efficiency is not only related to the value of out-degree, but also to the length of secret data that can be embedded with the chosen modification directions.

Without loss of generality, it can be assumed that the secret data to be embedded is evenly distributed on 0 and 1, i.e., the probabilities of 0 and 1 in the data are both 0.5. Then, the occurrence probability of a binary string consisting of 0 and 1 is inversely proportional to the length of the string. The longer the string, the lower the probability. For example, the probability of a string of length 1 (e.g., '0') is $\frac{1}{2}$, while the probability of a string of length 2 (e.g., '10' or '11') is $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$. Therefore, to obtain more efficient embedding modes for a given out-degree, the direction that would cause large modifications should be used to embed long data string. Based on these observations, the embedding modes with the highest embedding efficiency for a given out-degree can be obtained. The results are illustrated in Figure 3, and the corresponding embedding efficiency of each embedding mode can be calculated as follows.

$$E_2 = \frac{B_2}{V_2} = \frac{\frac{1}{2} \times 1 + \frac{1}{2} \times 1}{\frac{1}{2} \times 1} = \frac{1}{0.5} = 2, \tag{5}$$

$$E_3 = \frac{B_3}{V_3} = \frac{\frac{1}{2} \times 1 + \frac{1}{2} \times 1}{\frac{1}{2} \times 2} = \frac{1.5}{0.5} = 3, \tag{6}$$

$$E_4 = \frac{B_4}{V_4} = \frac{2}{\frac{3}{4} \times 1} = \frac{2}{0.75} = 2.67, \tag{7}$$

$$E_5 = \frac{B_5}{V_5} = \frac{\frac{3}{4} \times 2 + \frac{1}{4} \times 3}{\frac{3}{4} \times 1} = \frac{2.25}{0.75} = 3, \tag{8}$$

$$E_6 = \frac{B_6}{V_6} = \frac{\frac{1}{2} \times 2 + \frac{1}{2} \times 3}{\frac{5}{8} \times 1 + \frac{1}{8} \times 2} = \frac{2.5}{0.875} = 2.857, \tag{9}$$

$$E_7 = \frac{B_7}{V_7} = \frac{\frac{1}{4} \times 2 + \frac{3}{4} \times 3}{\frac{4}{8} \times 1 + \frac{2}{8} \times 2} = \frac{2.75}{1} = 2.75, \tag{10}$$

$$E_8 = \frac{B_8}{V_8} = \frac{3}{\frac{4}{8} \times 1 + \frac{3}{8} \times 2} = \frac{3}{1.25} = 2.4, \tag{11}$$

$$E_9 = \frac{B_9}{V_9} = \frac{\frac{7}{8} \times 3 + \frac{1}{8} \times 4}{\frac{4}{8} \times 1 + \frac{3}{8} \times 2} = \frac{3.125}{1.75} = 2.6. \tag{12}$$

From the above calculation results, it can be seen that the highest embedding efficiency can be achieved with the out-degree is 3 or 5, and the embedding capacity with an out-degree of 5 is higher. Similarly, the embedding efficiency of other embedding modes with different out-degrees can be easily obtained. Accordingly, the defects in the embedding strategies of the related 2D HS-Based RDH schemes are analyzed in Sections 3.1.1 and 3.1.2.

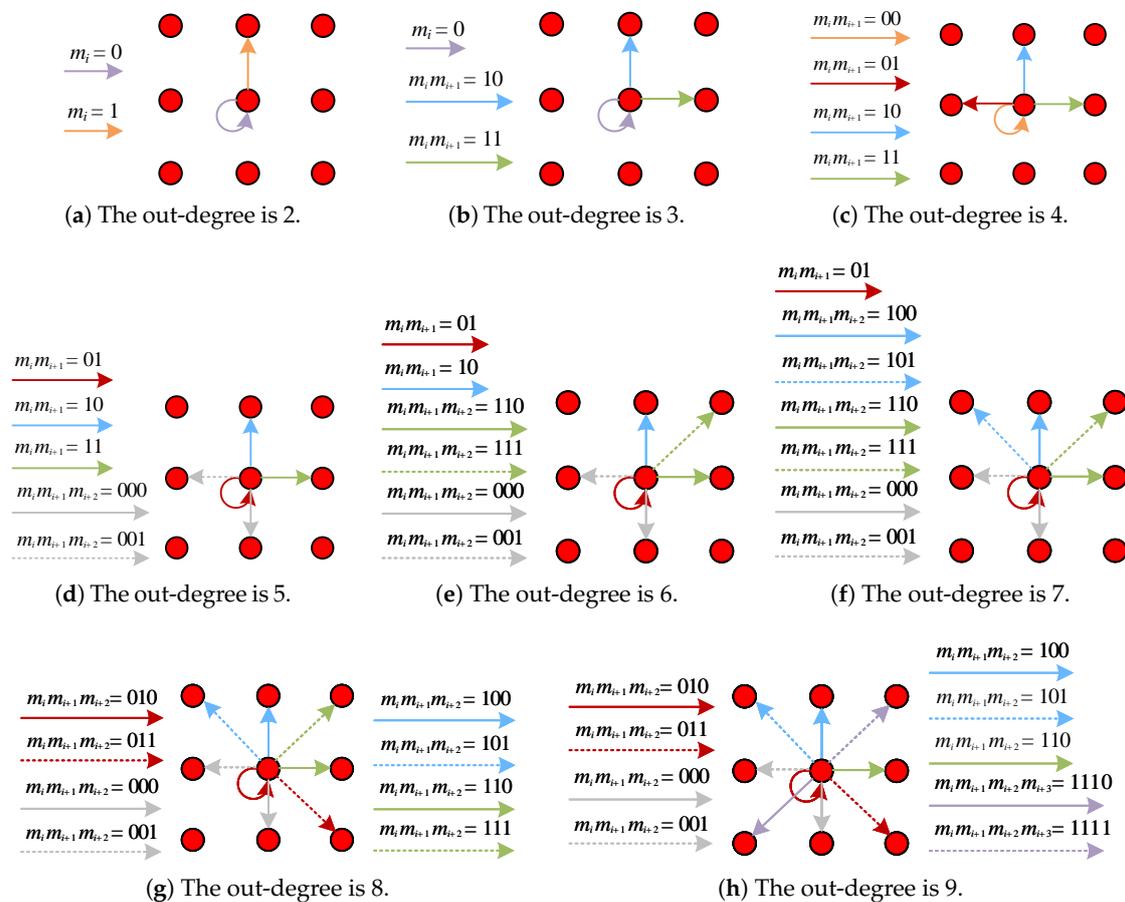


Figure 3. The best embedding modes for a given out-degree.

3.1.1. Related Schemes Using the (0,0) Coefficient Pairs

The number of coefficient pairs (0,0) are usually much larger than those of other coefficient pairs, so very high capacity can be obtained in the schemes using the (0,0) coefficient pairs. However, at the same time, when many zero coefficients are changed to non-zeros during data embedding, there will be a considerable increase in the file size of marked videos. Therefore, the schemes using the (0,0) coefficient pairs may only be suitable for the situations where high capacity is required regardless of file size.

In [34], an embedding mode with an out-degree of 3 is applied to most points on the coordinate axis; however, this embedding mode is not the most efficient embedding mode with

an out-degree of 3. More importantly, in order to make the scheme reversible with this embedding mode, both values of many pairs will be modified without data embedding, so many modifications are introduced without increasing the embedding capacity. Thus, the overall embedding efficiency may decrease. In [33], only the points in the right half plane are used, so not only are many points in the other half plane not fully used, but the best embedding mode with an out-degree of 5 cannot be used for the $(0,0)$ coefficient pairs. Thus, the use of the embedding mode with an out-degree of 4 make this method generally less efficient than the method proposed in [34] for $E_4 < E_5$.

3.1.2. Related Schemes Without Using the $(0,0)$ Coefficient Pairs

To reduce the increase in file size, the $(0,0)$ coefficient pairs should not be used. The corresponding schemes are usually suitable for the case where the increase in file size should be as small as possible, but the required embedding capacity is not large. In this case, the number of the coefficient pairs $(0,1)$, $(0,-1)$, $(-1,0)$ and $(1,0)$ is the largest, so these pairs are the best candidates for data embedding.

In [29], to reduce the modification to the zero coefficients, the best embedding mode with the out-degree being 2 is applied to the coefficient pairs $(0,1)$, $(0,-1)$, $(-1,0)$ and $(1,0)$. Although the probability of modifying the zero coefficients is 0.5 when the best embedding modes with an out-degree of 2 or 3 are used during data embedding, $E_2 < E_3$. Hence, this embedding strategy lowers the overall embedding efficiency without reducing the modifications. In [37], the best embedding modes with an out-degree of 4 are used for the coefficient pairs $(-1,0)$ and $(1,0)$, but the less efficient embedding modes with an out-degree of 4 are used for the coefficient pairs $(0,1)$ and $(0,-1)$. In addition, the embedding efficiency of the two used embedding modes is lower than that of the best embedding mode with an out-degree of 3. Moreover, the two values of many coefficient pairs need to be modified at the same time, so the videos will be greatly modified.

3.2. Proposed 2D HS-Based RDH Schemes

Since the modifications of zero DCT coefficients have a great negative impact on the compression rate, long string of data should be preferentially embedded through the modification directions that will modify more zero coefficients. As analyzed in Section 3.1, the probability of long data string is small, so the probability of the modification to zero coefficients can be reduced. Based on this premise and the previous conclusions about the embedding efficiency in Section 3.1, two new 2D HS schemes are developed for RDH in H.264/AVC video, one of which uses the $(0,0)$ DCT coefficient pairs and the other does not. The details of these two schemes are described in the following two sections.

3.2.1. 2D HS Using the $(0,0)$ Coefficient Pairs

Let (x,y) denote a cover coefficient pair, and the corresponding marked coefficient pair is represented by (x',y') . The proposed 2D HS scheme using the $(0,0)$ coefficient pairs is illustrated in Figure 4. First, all points are divided into several disjoint sets shown below.

$$\begin{aligned}
 S_1 &= \{(0,0)\}, \\
 S_2 &= \{(0,y) \mid y > 0 \ \& \ y \neq 2\}, \quad S_3 = \{(-1,1)\}, \quad S_4 = \{(0,y) \mid y < 0 \ \& \ y \neq -2\}, \quad S_5 = \{(1,-1)\}, \\
 S_6 &= \{(x,0) \mid x > 0 \ \& \ x \neq 2\}, \quad S_7 = \{(1,1)\}, \quad S_8 = \{(x,0) \mid x < 0 \ \& \ x \neq -2\}, \quad S_9 = \{(-1,-1)\}, \\
 S_{10} &= \{(x,y) \mid x > 0 \ \& \ y > 0\} - S_7, \quad S_{11} = \{(0,2)\}, \\
 S_{12} &= \{(x,y) \mid x < 0 \ \& \ y > 0\} - S_3, \quad S_{13} = \{(-2,0)\}, \\
 S_{14} &= \{(x,y) \mid x < 0 \ \& \ y < 0\} - S_9, \quad S_{15} = \{(0,-2)\}, \\
 S_{16} &= \{(x,y) \mid x > 0 \ \& \ y < 0\} - S_5, \quad S_{17} = \{(2,0)\}.
 \end{aligned}$$

Then, the method of embedding data into the coefficient pairs belonging to different sets is described as follows.

If $(x, y) \in S_1$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y) & \text{if } m_i m_{i+1} = 01, \\ (x + 1, y) & \text{if } m_i m_{i+1} = 10, \\ (x, y - 1) & \text{if } m_i m_{i+1} = 11, \\ (x - 1, y) & \text{if } m_i m_{i+1} m_{i+2} = 000, \\ (x, y + 1) & \text{if } m_i m_{i+1} m_{i+2} = 001. \end{cases} \quad (13)$$

If $(x, y) \in S_2 \cup S_3$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y + 1) & \text{if } m_i = 0, \\ (x - 1, y) & \text{if } m_i = 1. \end{cases} \quad (14)$$

If $(x, y) \in S_4 \cup S_5$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y - 1) & \text{if } m_i = 0, \\ (x + 1, y) & \text{if } m_i = 1. \end{cases} \quad (15)$$

If $(x, y) \in S_6 \cup S_7$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x + 1, y) & \text{if } m_i = 0 \ \& \ y = 0 \ \text{or} \ m_i = 1 \ \& \ y \neq 0, \\ (x, y + 1) & \text{if } m_i = 1 \ \& \ y = 0 \ \text{or} \ m_i = 0 \ \& \ y \neq 0. \end{cases} \quad (16)$$

If $(x, y) \in S_8 \cup S_9$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x - 1, y) & \text{if } m_i = 0 \ \& \ y = 0 \ \text{or} \ m_i = 1 \ \& \ y \neq 0, \\ (x, y - 1) & \text{if } m_i = 1 \ \& \ y = 0 \ \text{or} \ m_i = 0 \ \& \ y \neq 0. \end{cases} \quad (17)$$

If $(x, y) \in S_{10} \cup S_{11} \cup S_{12} \cup S_{13} \cup S_{14} \cup S_{15} \cup S_{16} \cup S_{17}$, any secret data cannot be embedded, so the coefficient pair will be just shifted as

$$(x', y') = \begin{cases} (x, y + 1) & \text{if } (x, y) \in S_{10} \cup S_{11}, \\ (x - 1, y) & \text{if } (x, y) \in S_{12} \cup S_{13}, \\ (x, y - 1) & \text{if } (x, y) \in S_{14} \cup S_{15}, \\ (x + 1, y) & \text{if } (x, y) \in S_{16} \cup S_{17}. \end{cases} \quad (18)$$

Although our method and the method proposed in [34] use the same embedding mode for the $(0, 0)$ coefficient pairs, the embedding modes used at other points in our scheme is different from that used in [34]. To evaluate the embedding performance of different schemes, the overall embedding efficiency is defined by

$$OE = \frac{\sum_{m=1}^M r_m B_m}{\sum_{m=1}^M r_m V_m}, \quad (19)$$

where r_m is the ratio of points using the embedding mode m to the total number of points, M is the number of embedding modes included in a scheme, thus $\sum_{m=1}^M r_m = 1$. B_m is the number of bits that can be embedded with the embedding mode m , and V_m is the corresponding amount of modification to the DCT coefficient pair. Here, the point shifting without data embedding is considered a special embedding mode that can embed 0 bits. Since the embedding efficiency represents the embedding capacity per unit modification, high embedding efficiency means that under the same payload, the amount of modification will be smaller, which will have less impact on video quality and file size.

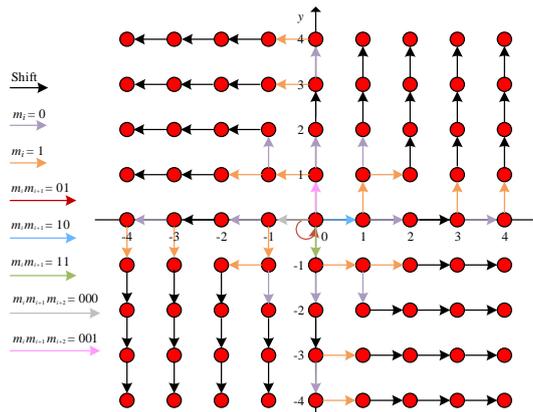


Figure 4. The proposed 2D HS using the (0,0) coefficient pairs.

A video clip called ‘bus’ from Xiph.org (<https://media.xiph.org/video/derf/>) is used to illustrate calculation of OE . The first 90 frames of this video is encoded using the H.264/AVC codec with a quantization parameter (QP) of 16, and a group of pictures (GOP) structure IPBPBPBPBPBPBPBP. The number of each DCT coefficient pair with both values in the range $[-4, 4]$ in the first GOP of video ‘bus’ is summarized in Table 1.

Table 1. The number of different DCT coefficient pairs in the video ‘bus’.

x \ y	-4	-3	-2	-1	0	1	2	3	4
-4	35	65	109	185	411	217	100	57	36
-3	55	108	196	434	974	453	219	104	56
-2	86	155	435	1272	3453	1332	495	198	99
-1	164	371	1175	5263	24,321	5516	1298	417	168
0	277	809	3274	26,301	251,897	26,346	3450	823	311
1	160	390	1314	5472	24,283	5330	1262	370	158
2	98	194	506	1357	3473	1257	494	184	86
3	50	99	238	457	1021	432	199	95	54
4	34	54	105	197	395	176	100	77	34

Let OE_o and OE_z denote the overall embedding efficiency of our method and the method presented in [34], respectively. Based on the statistical results given in Table 1, OE_o and OE_z can be estimated as below.

$$OE_o = \frac{\frac{251,897}{413,725} \times 2.25 + \frac{127,853}{413,725} \times 1 + \frac{33,975}{413,725} \times 0}{\frac{251,897}{413,725} \times 0.75 + \frac{127,853}{413,725} \times 1 + \frac{33,975}{413,725} \times 1} = \frac{694,621.25}{350,750.75} = 1.980.$$

$$OE_z = \frac{\frac{251,897}{413,725} \times 2.25 + \frac{64,392}{413,725} \times 1.5 + \frac{48,604}{413,725} \times 1 + \frac{6926}{413,725} \times 1 + \frac{41,906}{413,725} \times 0}{\frac{251,897}{413,725} \times 0.75 + \frac{64,392}{413,725} \times 1 + \frac{48,604}{413,725} \times 1.5 + \frac{6926}{413,725} \times 1 + \frac{41,906}{413,725} \times 2} = \frac{718,886.25}{416,958.75} = 1.724.$$

It can be seen from the above calculation results that our overall embedding efficiency is higher than that of the method proposed in [34]. The reason is that the embedding modes used in [34] for the points on the coordinate axis affect the shift of those points without capacity gain. In [34], both values of those points that are shifted without data embedding need to be modified, while only one value will be modified in our scheme. When the video content is more complex and the compression rate is lower, the number of shifting-only points will increase, thus, the impact of bigger modifications will be more obvious.

3.2.2. 2D HS without Using the (0,0) Coefficient Pairs

The proposed 2D HS scheme without using the (0,0) coefficient pairs is illustrated in Figure 5. First, the points except (0,0) are divided into several disjoint sets as follows.

$$\begin{aligned}
 S_1 &= \{(1,0)\}, S_2 = \{(0,1)\}, S_3 = \{(-1,0)\}, S_4 = \{(0,-1)\}, \\
 S_5 &= \{(1,1)\}, S_6 = \{(-1,1)\}, S_7 = \{(-1,-1)\}, S_8 = \{(1,-1)\}, \\
 S_9 &= \{(x,1) \mid x > 1\}, S_{10} = \{(x,1) \mid x < -1\}, S_{11} = \{(x,-1) \mid x < -1\}, S_{12} = \{(x,-1) \mid x > 1\}, \\
 S_{13} &= \{(x,0) \mid x > 1\}, S_{14} = \{(x,0) \mid x < -1\}, S_{15} = \{(x,y) \mid y > 1\}, S_{16} = \{(x,y) \mid y < -1\}.
 \end{aligned}$$

Then, the method of embedding data into the coefficient pairs belonging to different sets is described as below.

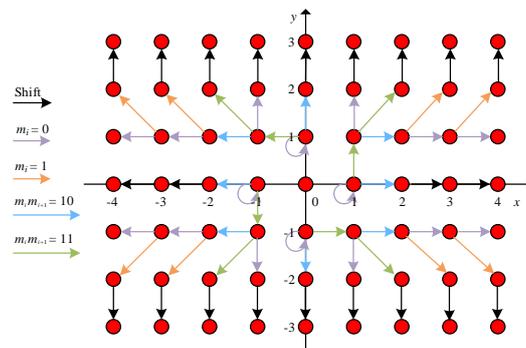


Figure 5. The proposed 2D HS without using the (0,0) coefficient pairs.

If $(x, y) \in S_1$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y) & \text{if } m_i = 0, \\ (x + 1, y) & \text{if } m_i m_{i+1} = 10, \\ (x, y + 1) & \text{if } m_i m_{i+1} = 11. \end{cases} \tag{20}$$

If $(x, y) \in S_2$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y) & \text{if } m_i = 0, \\ (x, y + 1) & \text{if } m_i m_{i+1} = 10, \\ (x - 1, y) & \text{if } m_i m_{i+1} = 11. \end{cases} \tag{21}$$

If $(x, y) \in S_3$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y) & \text{if } m_i = 0, \\ (x - 1, y) & \text{if } m_i m_{i+1} = 10, \\ (x, y - 1) & \text{if } m_i m_{i+1} = 11. \end{cases} \tag{22}$$

If $(x, y) \in S_4$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y) & \text{if } m_i = 0, \\ (x, y - 1) & \text{if } m_i m_{i+1} = 10, \\ (x + 1, y) & \text{if } m_i m_{i+1} = 11. \end{cases} \tag{23}$$

If $(x, y) \in S_5$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y + 1) & \text{if } m_i = 0, \\ (x + 1, y) & \text{if } m_i m_{i+1} = 10, \\ (x + 1, y + 1) & \text{if } m_i m_{i+1} = 11. \end{cases} \tag{24}$$

If $(x, y) \in S_6$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y + 1) & \text{if } m_i = 0, \\ (x - 1, y) & \text{if } m_i m_{i+1} = 10, \\ (x - 1, y + 1) & \text{if } m_i m_{i+1} = 11. \end{cases} \quad (25)$$

If $(x, y) \in S_7$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y - 1) & \text{if } m_i = 0, \\ (x - 1, y) & \text{if } m_i m_{i+1} = 10, \\ (x - 1, y - 1) & \text{if } m_i m_{i+1} = 11. \end{cases} \quad (26)$$

If $(x, y) \in S_8$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x, y - 1) & \text{if } m_i = 0, \\ (x + 1, y) & \text{if } m_i m_{i+1} = 10, \\ (x + 1, y - 1) & \text{if } m_i m_{i+1} = 11. \end{cases} \quad (27)$$

If $(x, y) \in S_9$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x + 1, y) & \text{if } m_i = 0, \\ (x + 1, y + 1) & \text{if } m_i = 1. \end{cases} \quad (28)$$

If $(x, y) \in S_{10}$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x - 1, y) & \text{if } m_i = 0, \\ (x - 1, y + 1) & \text{if } m_i = 1. \end{cases} \quad (29)$$

If $(x, y) \in S_{11}$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x - 1, y) & \text{if } m_i = 0, \\ (x - 1, y - 1) & \text{if } m_i = 1. \end{cases} \quad (30)$$

If $(x, y) \in S_{12}$, the marked coefficient pair will be

$$(x', y') = \begin{cases} (x + 1, y) & \text{if } m_i = 0, \\ (x + 1, y - 1) & \text{if } m_i = 1. \end{cases} \quad (31)$$

If $(x, y) \in S_{13} \cup S_{14} \cup S_{15} \cup S_{16}$, any secret data cannot be embedded, and the coefficient pair will be shifted as

$$(x', y') = \begin{cases} (x + 1, y) & \text{if } (x, y) \in S_{13}, \\ (x - 1, y) & \text{if } (x, y) \in S_{14}, \\ (x, y + 1) & \text{if } (x, y) \in S_{15}, \\ (x, y - 1) & \text{if } (x, y) \in S_{16}. \end{cases} \quad (32)$$

Because the method described in [37] uses more points on the coordinate axis for data embedding, while the method presented in [29] does not use any points on the coordinate axis except $(0, 1), (0, -1), (-1, 0)$ and $(1, 0)$ for data embedding. It can be easily inferred that the embedding capacity of our scheme illustrated in Figure 5 will be lower than that of [37], and higher than that of [29]. Our scheme achieves a good balance between embedding capacity and modification to video,

so its performance is better than the methods of [29,37] for most payloads, which will be demonstrated in the massive experiments.

3.2.3. Data Extraction and Video Recovery

In the proposed scheme, the data extraction and video recovery can be completed by the inverse operation of embedding. From Figures 4 and 5, it can be observed that the in-degree of each point is one. Therefore, each coefficient pair in the marked video denoted by the point (x', y') can be uniquely restored to the original coefficient pair in the cover video denoted by the point (x, y) by following the opposite direction of the arrow ending at (x', y') , and at the same time, the embedded data can be obtained according to the rules of shifting (x, y) to (x', y') .

4. Experimental Results

The proposed schemes are implemented based on the reference software JM 19.0 (<http://iphome.hhi.de/suehring/tml/>) for H.264/AVC. Six typical sequences with the resolution of 352×288 from Xiph.org video dataset are used in our experiments. These videos contain different motion and content, allowing for a wide range of payloads. The first 90 frames of each video are encoded with main profile, and the GOP structure is IPBPBPBPBPBPBP, which means that there are six GOPs in total.

To compare the performance of different 2D HS schemes fairly, the method proposed in [29] is modified to make it suitable for H.264/AVC video, and the objects used for data embedding in [37] is changed from motion vectors to DCT coefficients. Moreover, to reduce the impact of data embedding on video quality, only P frames and B frames are used for data embedding. In addition, the DCT coefficient pair is composed of two sequential coefficients in a zig-zag scanning order. There are 16 coefficients in a 4×4 block of H.264/AVC video, but only the 7th to 16th coefficients are selected, because modifying more low-frequency coefficients may cause larger video distortion.

In the following sections, Ours⁺ is used to denote our proposed scheme using the $(0, 0)$ coefficient pairs, and Ours⁻ denotes our proposed scheme without using the $(0, 0)$ coefficient pairs. To present the comparison results more clearly, we also use gray cells to rank the results. There are three types of gray cells. The darker the cell, the higher the ranking of the result. In addition, the best results are displayed with underlined numbers.

4.1. Embedding Capacity

Although the primary goal of our schemes is to reduce the loss of video quality and the increase in file size, the embedding capacity should not decrease too much. In this section, the embedding capacity of different schemes is evaluated. The results of the schemes using the $(0, 0)$ coefficient pairs are shown in Table 2. It can be learned that the capacity of [33] is lowest, which is significantly lower than that of our scheme and [34]. Furthermore, the capacity of our scheme is very close to that of [34], and the difference is generally around 1%, which is basically negligible.

Table 2. The embedding capacity of the schemes using the $(0, 0)$ coefficient pairs, where the percentage figures in parentheses indicate the relative difference between other schemes and Ours⁺.

Video	Embedding Capacity (bits)					
	QP = 16			QP = 28		
	Ours ⁺	[33]	[34]	Ours ⁺	[33]	[34]
foreman	<u>3,415,024</u>	2,991,010 (−12.4%)	3,411,252 (−0.1%)	442,374	387,443 (−12.4%)	<u>445,023</u> (+0.6%)
container	2,197,152	1,914,967 (−12.8%)	<u>2,206,489</u> (+0.4%)	180,453	156,344 (−13.4%)	<u>182,238</u> (+1.0%)
bus	<u>3,885,660</u>	3,216,720 (−17.2%)	3,874,676 (−0.3%)	1,442,234	1,267,463 (−12.1%)	<u>1,453,286</u> (+0.8%)
crew	<u>3,888,517</u>	3,488,241 (−10.3%)	3,886,072 (−0.1%)	1,048,725	928,943 (−11.4%)	<u>1,051,886</u> (+0.3%)
hall_monitor	4,213,795	3,762,970 (−10.7%)	<u>4,219,961</u> (+0.1%)	250,843	217,846 (−13.2%)	<u>253,323</u> (+1.0%)
mobile	2,814,823	2,134,882 (−24.2%)	<u>2,823,286</u> (+0.3%)	1,597,461	1,357,855 (−15.0%)	<u>1,614,352</u> (+1.1%)

The results of the schemes without using the (0,0) coefficient pairs are shown in Table 3. As can be seen from Table 3, although the embedding capacity of our proposed scheme is lower than that of [37], it is still higher than that of [29]. The experimental results are consistent with the previous analysis presented in Section 3.2.

Table 3. The embedding capacity of the schemes without using the (0,0) coefficient pairs, where the percentage figures in parentheses indicate the relative difference between other schemes and Ours⁻.

Video	Embedding Capacity (bits)					
	QP = 16			QP = 28		
	Ours ⁻	[37]	[29]	Ours ⁻	[37]	[29]
foreman	674,237	864,838 (+28.3%)	480,248 (-28.8%)	16,892	22,252 (+31.7%)	11,761 (-30.4%)
container	388,377	499,770 (+28.7%)	273,428 (-29.6%)	11,747	15,398 (+31.1%)	8118 (-30.9%)
bus	1,253,046	1519,264 (+21.2%)	879,967 (-29.8%)	156,593	205,092 (+31.0%)	110,696 (-29.3%)
crew	616,595	795,013 (+28.9%)	440,713 (-28.5%)	24,478	32,174 (+31.4%)	16,845 (-31.2%)
hall_monitor	701,549	920,756 (+31.2%)	493,344 (-29.7%)	14,704	19,408 (+32.0%)	10,171 (-30.8%)
mobile	1,371,842	1,616,303 (+17.8%)	989,803 (-27.8%)	311,538	404,755 (+29.9%)	220,584 (-29.2%)

4.2. Video Quality

To obtain a reasonably comprehensive evaluation of the impact of data embedding on the quality of H.264/AVC video, the video sequences are encoded with two QPs of 16 and 28, and five different payloads are selected according to the embedding capacity of each video. The 10th frame of the six cover videos with a QP of 16 and the corresponding marked frames generated by our schemes are shown in Figure 6, where the payload is the maximum value we use for each video in our experiments. It can be seen that the visual distortions in the marked frames are almost unnoticeable. Hence, the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [43] and entropy are used to further demonstrate the visual quality of marked video.

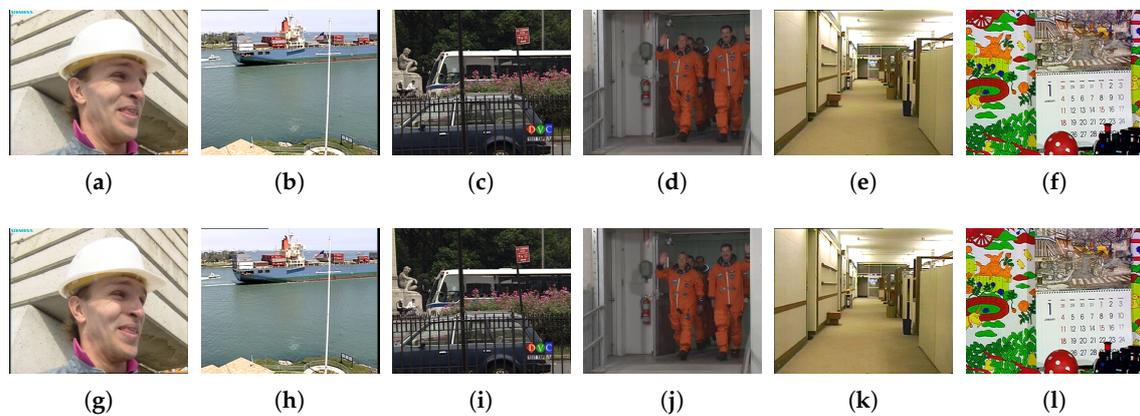


Figure 6. The 10th frame of original videos (a) foreman, (b) container, (c) bus, (d) crew, (e) hall_monitor, and (f) mobile, and the corresponding 10th frame of marked videos, where (g) marked foreman, (h) marked container, and (i) marked bus are generated by our scheme using the (0,0) coefficient pairs; (j) marked crew, (k) marked hall_monitor, and (l) marked mobile are generated by our scheme without using the (0,0) coefficient pairs, respectively.

The results of the schemes using the (0,0) coefficient pairs are shown in Table 4, and the corresponding percentages for each ranking are shown in Table 5. From Tables 4 and 5, it can be seen that when QP is 16, as far as PSNR is concerned, the quality of the marked video generated by our scheme is the best in 66.7% of the cases, while in the remaining cases, our results are all ranked in the middle, which are superior to [34] and inferior to [33]; in terms of SSIM, our scheme achieves the best video quality in 93.3% of the cases. When QP is 28, 70.0% of the PSNR values of our scheme are the

highest, and for SSIM, our scheme obtains the best results in about 73.3% of the cases, both results are higher than the comparison methods. The main reason for the small difference between the PSNR and SSIM values of our results and the comparison methods is that the proportion of the (0, 0) coefficient pairs are very high, and thus large part of data will be embedded into these coefficient pairs. However, our embedding mode of the (0, 0) coefficient pairs is the same as that of [34], and the difference in embedding efficiency between our scheme and [33] is not very large. To sum up, our scheme achieves better video quality in terms of PSNR or SSIM in most cases, as demonstrated by the average results shown in Table 4 and the results in Table 5.

Table 4. PSNR and SSIM for different schemes using the (0, 0) coefficient pairs, where payload is the number of bits embedded into each GOP.

Video	QP = 16							QP = 28						
	Payload (bits)	PSNR			SSIM			Payload (bits)	PSNR			SSIM		
		Ours ⁺	[33]	[34]	Ours ⁺	[33]	[34]		Ours ⁺	[33]	[34]	Ours ⁺	[33]	[34]
foreman	40,000	<u>43.76</u>	43.62	43.76	<u>0.985</u>	0.984	<u>0.985</u>	8000	<u>36.45</u>	36.44	36.38	<u>0.940</u>	0.939	<u>0.940</u>
	75,000	43.32	<u>43.40</u>	43.06	<u>0.983</u>	<u>0.983</u>	0.982	16,000	35.44	<u>35.53</u>	35.41	0.933	0.933	<u>0.934</u>
	110,000	<u>42.56</u>	42.34	42.18	<u>0.980</u>	<u>0.978</u>	<u>0.980</u>	24,000	<u>35.09</u>	34.87	35.07	<u>0.929</u>	0.927	<u>0.929</u>
	145,000	41.80	<u>41.95</u>	41.43	<u>0.977</u>	<u>0.977</u>	0.976	32,000	34.59	34.39	<u>34.88</u>	0.924	0.922	<u>0.925</u>
	180,000	<u>41.94</u>	41.56	41.27	<u>0.977</u>	0.975	0.976	40,000	<u>34.27</u>	34.10	34.15	<u>0.921</u>	0.919	0.920
container	40,000	<u>43.44</u>	43.20	43.31	<u>0.979</u>	<u>0.979</u>	<u>0.979</u>	4000	<u>36.32</u>	36.28	36.30	<u>0.925</u>	<u>0.925</u>	<u>0.925</u>
	75,000	<u>42.33</u>	42.12	42.15	<u>0.976</u>	0.973	<u>0.976</u>	8500	<u>35.92</u>	35.85	35.90	<u>0.922</u>	0.920	<u>0.922</u>
	110,000	<u>41.59</u>	41.32	41.36	<u>0.970</u>	0.968	0.969	13,000	<u>35.61</u>	35.50	35.58	<u>0.919</u>	0.917	<u>0.919</u>
	145,000	<u>40.94</u>	40.76	40.72	<u>0.965</u>	0.964	0.964	17,500	<u>35.35</u>	35.26	35.33	<u>0.915</u>	0.914	<u>0.915</u>
	180,000	<u>40.62</u>	40.44	40.35	<u>0.963</u>	0.960	0.962	22,000	<u>35.20</u>	35.14	35.15	<u>0.913</u>	0.912	<u>0.913</u>
bus	40,000	<u>42.95</u>	42.92	42.76	<u>0.992</u>	0.991	0.991	20,000	<u>34.02</u>	33.92	33.90	<u>0.951</u>	0.950	0.950
	80,000	42.43	<u>42.75</u>	41.97	<u>0.990</u>	<u>0.990</u>	0.990	50,000	<u>32.82</u>	<u>32.82</u>	32.68	<u>0.942</u>	<u>0.942</u>	0.941
	120,000	41.42	<u>41.57</u>	41.26	<u>0.988</u>	<u>0.988</u>	<u>0.988</u>	80,000	<u>32.08</u>	31.83	31.80	<u>0.936</u>	<u>0.933</u>	0.935
	160,000	40.90	<u>41.21</u>	40.19	<u>0.986</u>	<u>0.986</u>	0.985	110,000	<u>31.31</u>	31.16	31.09	<u>0.928</u>	0.926	<u>0.928</u>
	200,000	40.68	<u>40.78</u>	40.42	<u>0.986</u>	0.985	0.985	140,000	<u>30.92</u>	30.73	30.61	<u>0.924</u>	0.921	0.923
crew	40,000	<u>40.99</u>	<u>40.99</u>	40.79	0.978	0.978	<u>0.979</u>	20,000	<u>34.49</u>	34.17	34.37	0.917	0.915	<u>0.920</u>
	80,000	<u>40.75</u>	<u>40.75</u>	39.85	0.976	<u>0.977</u>	0.973	45,000	32.64	<u>32.82</u>	32.72	0.891	0.890	<u>0.893</u>
	120,000	<u>39.76</u>	38.94	38.60	<u>0.971</u>	0.967	0.969	70,000	31.94	31.67	<u>32.30</u>	0.873	0.869	<u>0.876</u>
	160,000	<u>39.07</u>	38.59	38.14	<u>0.966</u>	0.965	0.962	95,000	<u>31.70</u>	30.77	31.35	<u>0.860</u>	0.846	0.857
	200,000	<u>39.34</u>	38.29	37.94	<u>0.965</u>	0.962	0.962	120,000	30.47	<u>30.76</u>	30.63	0.838	0.838	<u>0.841</u>
hall_monitor	40,000	41.79	<u>41.94</u>	41.20	<u>0.981</u>	0.980	<u>0.981</u>	2400	<u>37.44</u>	37.21	37.42	<u>0.954</u>	0.953	<u>0.954</u>
	80,000	<u>41.47</u>	41.36	41.33	<u>0.978</u>	0.977	<u>0.978</u>	4800	36.85	<u>36.96</u>	36.61	<u>0.952</u>	<u>0.952</u>	<u>0.952</u>
	120,000	<u>41.28</u>	39.88	40.73	<u>0.976</u>	0.973	0.975	7200	36.32	<u>36.65</u>	36.53	<u>0.951</u>	<u>0.951</u>	<u>0.951</u>
	160,000	<u>40.22</u>	39.95	39.65	<u>0.971</u>	0.970	0.969	9600	35.42	<u>36.28</u>	35.89	0.949	<u>0.951</u>	0.950
	200,000	<u>40.44</u>	39.89	39.80	<u>0.972</u>	0.969	0.970	12,000	<u>36.34</u>	36.08	36.03	<u>0.950</u>	<u>0.950</u>	<u>0.950</u>
mobile	40,000	43.72	<u>44.12</u>	43.06	<u>0.994</u>	<u>0.994</u>	<u>0.994</u>	32,000	33.52	<u>33.53</u>	33.35	0.970	0.971	0.970
	75,000	<u>43.25</u>	42.94	42.74	<u>0.994</u>	0.993	0.993	64,000	32.64	32.55	32.45	<u>0.966</u>	<u>0.966</u>	<u>0.966</u>
	110,000	42.17	<u>42.47</u>	41.37	<u>0.992</u>	<u>0.992</u>	0.991	96,000	<u>32.03</u>	31.89	31.82	<u>0.963</u>	0.962	0.962
	145,000	<u>41.89</u>	41.85	41.16	<u>0.992</u>	0.991	0.991	128,000	<u>31.53</u>	31.41	31.28	<u>0.960</u>	0.959	0.959
	180,000	41.21	<u>41.36</u>	40.36	<u>0.990</u>	<u>0.990</u>	<u>0.990</u>	160,000	<u>31.14</u>	31.05	30.89	<u>0.957</u>	0.956	0.956
average	—	<u>41.60</u>	41.44	41.10	<u>0.980</u>	0.979	0.979	—	<u>33.99</u>	33.92	33.93	<u>0.929</u>	0.928	<u>0.929</u>

Table 5. The percentages of PSNR and SSIM values of different schemes using the (0, 0) coefficient pairs in each ranking.

Ranking	QP = 16						QP = 28					
	PSNR (%)			SSIM (%)			PSNR (%)			SSIM (%)		
	Ours ⁺	[33]	[34]									
High	66.7	40.0	3.3	93.3	33.3	36.7	70.0	26.7	6.7	73.3	26.7	63.3
Middle	33.3	43.3	13.3	6.7	26.7	36.7	16.7	33.3	50.0	23.3	20.0	33.3
Low	0.0	16.7	83.3	0.0	40.0	26.7	13.3	40.0	43.3	3.3	53.3	3.3

The experimental results of the schemes without using the (0,0) coefficient pairs are shown in Table 6, and the corresponding percentages for each ranking are shown in Table 7. When QP is 16, it can be observed from Tables 6 and 7 that 73.3% of the PSNR values and 96.7% of the SSIM values of our scheme are the highest. When QP is 28, due to the significant reduction in payload, the SSIM values of our scheme and the related schemes are almost the same in about 60% of the cases. However, for PSNR, 83.3% of the results of our schemes are the best, which is apparently superior to the related schemes. Moreover, the average results given in Table 6 also show that our scheme achieves the best video quality in most cases. Compared with the schemes using the (0,0) coefficient pairs, the improvement of our scheme without using the (0,0) coefficient pairs is more obvious.

Table 6. PSNR and SSIM for different schemes without using the (0,0) coefficient pairs, where payload is the number of bits embedded into each GOP.

Video	QP = 16							QP = 28						
	Payload (bits)	PSNR			SSIM			Payload (bits)	PSNR			SSIM		
		Ours	[37]	[29]	Ours	[37]	[29]		Ours	[37]	[29]	Ours	[37]	[29]
foreman	10,000	<u>45.18</u>	44.99	45.13	0.988	0.988	0.989	260	<u>37.36</u>	37.33	37.34	0.945	0.945	0.945
	20,000	<u>44.83</u>	44.46	44.58	0.988	0.987	0.988	520	<u>37.29</u>	37.23	37.28	0.945	0.945	0.945
	30,000	<u>44.28</u>	44.02	44.16	0.987	0.986	0.987	780	<u>37.26</u>	37.21	37.25	0.945	0.945	0.945
	40,000	<u>44.07</u>	43.60	43.91	0.987	0.985	0.986	1040	<u>37.21</u>	37.16	37.20	0.945	0.945	0.945
	50,000	<u>43.85</u>	43.34	43.62	0.986	0.985	0.986	1300	<u>37.20</u>	37.14	37.18	0.945	0.945	0.945
container	7500	<u>44.96</u>	44.78	44.81	0.986	0.986	0.986	220	<u>36.75</u>	36.74	36.75	0.927	0.927	0.927
	15,000	<u>44.45</u>	44.14	44.20	0.985	0.984	0.984	440	<u>36.72</u>	36.70	36.71	0.927	0.927	0.927
	22,500	<u>44.05</u>	43.59	43.79	0.984	0.982	0.982	660	<u>36.70</u>	36.66	36.68	0.927	0.926	0.927
	30,000	<u>43.78</u>	43.22	43.55	0.983	0.981	0.982	880	<u>36.67</u>	36.62	36.66	0.926	0.926	0.926
	37,500	<u>43.54</u>	42.92	43.39	0.982	0.979	0.981	1100	<u>36.66</u>	36.59	36.64	0.926	0.926	0.926
bus	20,000	43.79	42.85	<u>43.87</u>	0.994	0.993	0.994	3000	<u>35.23</u>	35.02	35.18	0.958	0.958	0.958
	40,000	42.85	42.65	<u>42.96</u>	0.993	0.992	0.992	5500	<u>35.03</u>	34.85	34.88	0.957	0.957	0.957
	60,000	<u>42.46</u>	41.35	42.25	0.992	0.991	0.991	8000	<u>34.80</u>	34.41	34.61	0.957	0.956	0.956
	80,000	<u>41.79</u>	41.22	41.70	0.991	0.990	0.991	10,500	<u>34.53</u>	34.21	34.45	0.956	0.955	0.956
	100,000	<u>41.44</u>	40.68	41.32	0.990	0.989	0.990	13,000	<u>34.44</u>	33.91	34.35	0.956	0.954	0.955
crew	10,000	<u>43.50</u>	42.63	43.19	0.987	0.986	0.987	320	<u>37.73</u>	37.58	37.48	0.947	0.947	0.947
	20,000	<u>42.90</u>	41.98	42.55	0.986	0.984	0.986	640	37.29	37.22	<u>37.39</u>	0.946	0.946	0.946
	30,000	<u>42.45</u>	41.50	41.95	0.985	0.983	0.984	960	<u>37.46</u>	37.36	37.31	0.946	0.946	0.946
	40,000	41.78	41.02	<u>41.83</u>	0.983	0.981	0.982	1280	37.01	<u>37.31</u>	37.05	0.945	0.945	0.945
	50,000	41.02	40.74	<u>41.48</u>	0.982	0.980	0.982	1600	37.10	37.00	<u>37.14</u>	0.946	0.945	0.945
hall_monitor	14,000	44.28	43.77	<u>44.30</u>	0.987	0.986	0.987	69	<u>37.99</u>	<u>37.99</u>	37.95	0.955	0.955	0.955
	28,000	<u>43.46</u>	42.96	43.17	0.986	0.985	0.985	138	37.91	<u>37.97</u>	37.77	0.954	0.955	0.954
	42,000	<u>43.12</u>	42.26	42.82	0.985	0.983	0.984	207	<u>37.91</u>	37.57	37.83	0.954	0.954	0.954
	56,000	<u>42.62</u>	42.04	42.60	0.984	0.982	0.983	276	<u>37.87</u>	37.79	37.85	0.954	0.954	0.954
	70,000	<u>42.50</u>	41.89	42.20	0.983	0.981	0.982	345	37.68	<u>37.83</u>	<u>37.84</u>	0.954	0.954	0.954
mobile	24,000	44.30	43.67	<u>44.37</u>	0.996	0.995	0.996	6600	<u>34.64</u>	34.48	34.60	0.976	0.975	0.975
	48,000	43.13	42.82	<u>43.27</u>	0.995	0.995	0.995	13,200	<u>34.35</u>	34.12	34.21	0.975	0.974	0.974
	72,000	42.50	42.08	<u>42.55</u>	0.995	0.994	0.995	19,800	<u>34.08</u>	33.73	33.95	0.974	0.973	0.974
	96,000	<u>42.07</u>	41.30	42.02	0.994	0.993	0.994	26,400	<u>33.87</u>	33.46	33.76	0.974	0.972	0.973
	120,000	<u>41.64</u>	41.00	41.63	0.994	0.993	0.994	33,000	<u>33.72</u>	33.23	33.63	0.973	0.971	0.973
average	—	<u>43.22</u>	42.65	43.11	0.988	0.987	0.987	—	<u>36.35</u>	36.21	36.30	0.951	0.950	0.950

Table 7. The percentages of PSNR and SSIM values of different schemes without using the (0,0) coefficient pairs in each ranking.

Ranking	QP = 16						QP = 28					
	PSNR (%)			SSIM (%)			PSNR (%)			SSIM (%)		
	Ours ⁻	[37]	[29]	Ours ⁻	[37]	[29]	Ours ⁻	[37]	[29]	Ours ⁻	[37]	[29]
High	73.3	0.0	26.7	96.7	6.7	56.7	83.3	10.0	13.3	96.7	66.7	76.7
Middle	26.7	0.0	73.3	3.3	20.0	43.3	10.0	10.0	73.3	3.3	13.3	23.3
Low	0.0	100.0	0.0	0.0	73.3	0.0	6.7	80.0	13.3	0.0	20.0	0.0

Furthermore, the changes in Shannon entropy of marked videos compared with the original videos are calculated to evaluate the modifications to the video. The average results of each cover video and the marked videos generated by different schemes are shown in Table 8. It can be seen from Table 8 that for the schemes using the (0,0) coefficient pairs, whether the QP is 16 or 28, the results of our method are basically closer to the entropy of cover videos than the related schemes. The same observations can be made for the schemes without using the (0,0) coefficient pairs. The closer the entropy of the marked video is to the original video, generally means the smaller modification to the video. Thus, the quality of the marked videos generated by our scheme will be better, which was already demonstrated in Tables 4 and 6.

Table 8. Shannon entropy of each cover video and changes in Shannon entropy of the marked video generated by different schemes.

Video	Shannon Entropy/Shannon Entropy Change													
	Origin	QP = 16						QP = 28						
		(0,0)s Are Used			(0,0)s Aren't Used			Origin	(0,0)s Are Used			(0,0)s Aren't Used		
		Ours ⁺	[33]	[34]	Ours ⁻	[37]	[29]		Ours ⁺	[33]	[34]	Ours ⁻	[37]	[29]
foreman	7.402	0.013	0.014	0.014	0.004	0.006	0.005	7.381	0.006	0.009	0.005	0.000	0.000	0.000
container	6.871	0.013	0.014	0.013	0.004	0.005	0.005	6.828	0.005	0.005	0.005	0.000	0.000	0.000
bus	7.316	0.010	0.012	0.014	0.007	0.011	0.008	7.286	0.030	0.032	0.030	0.004	0.006	0.005
crew	7.148	0.020	0.020	0.020	0.008	0.009	0.010	7.122	0.064	0.067	0.066	0.000	0.002	0.000
hall_monitor	7.269	0.006	0.014	0.009	0.003	0.005	0.005	7.244	0.001	0.003	0.002	0.000	0.000	0.000
mobile	7.568	0.004	0.005	0.004	0.000	0.001	0.000	7.603	0.000	0.000	0.000	-0.001	-0.001	-0.002

4.3. File Size

Generally, the file size of marked videos will increase. However, as the H.264/AVC video aims to provide good video quality at a low bit rate, so it is desirable that the RDH schemes for H.264/AVC video will not cause a significant increase in the file size of marked videos.

The experimental results of the schemes using the (0,0) coefficient pairs are shown in Figures 7 and 8. It can be seen from Figure 7 that when QP is 16, the increase in file size caused by our scheme is apparently smaller than that of [33] for all six videos. Although the increase in the file size of marked video generated by [34] is close to ours, it is still slightly higher, and as the payload increases, the difference become more noticeable. When QP is 28, Figure 8 shows that the file size increase of our scheme is also apparently lower than that of [33]. However, the difference between our scheme and [34] is very small. The reason for the above results is that the method proposed in [33] uses only half of the plane, resulting in more blocks of H.264/AVC video modified with the same payload. Although both values of the points that will be shifted by [34] need to be modified at the same time during data embedding, we found that these points are seldom used in our experiments because the (0,0) coefficient pairs carry most of the payload, so the file size increase of [34] is close to that of ours.

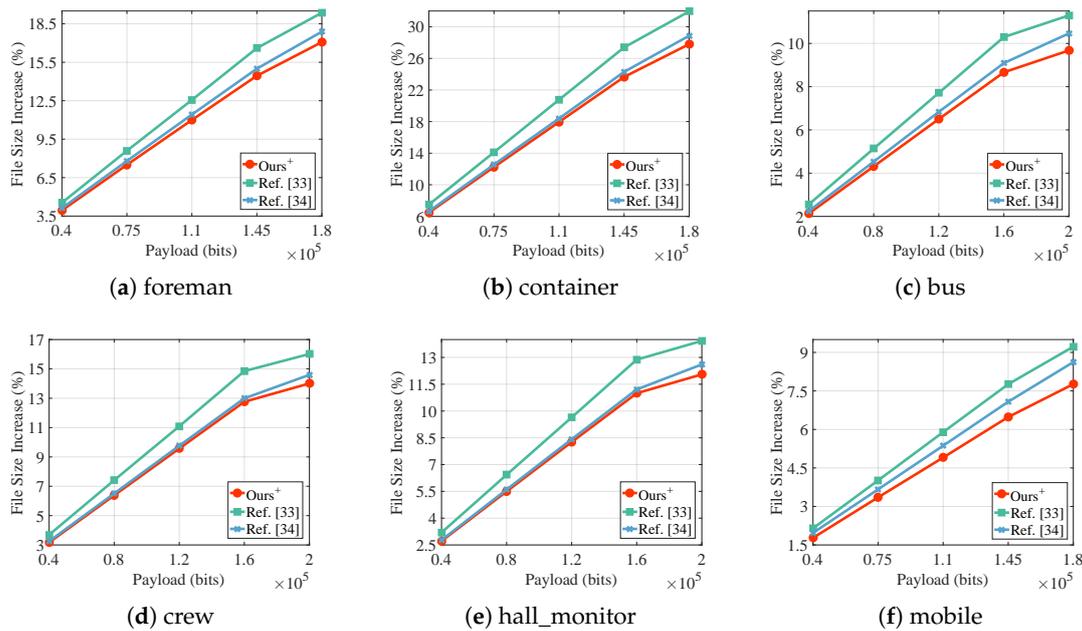


Figure 7. The increase in the file size of marked video generated by different schemes using the (0,0) coefficient pairs when QP is 16.

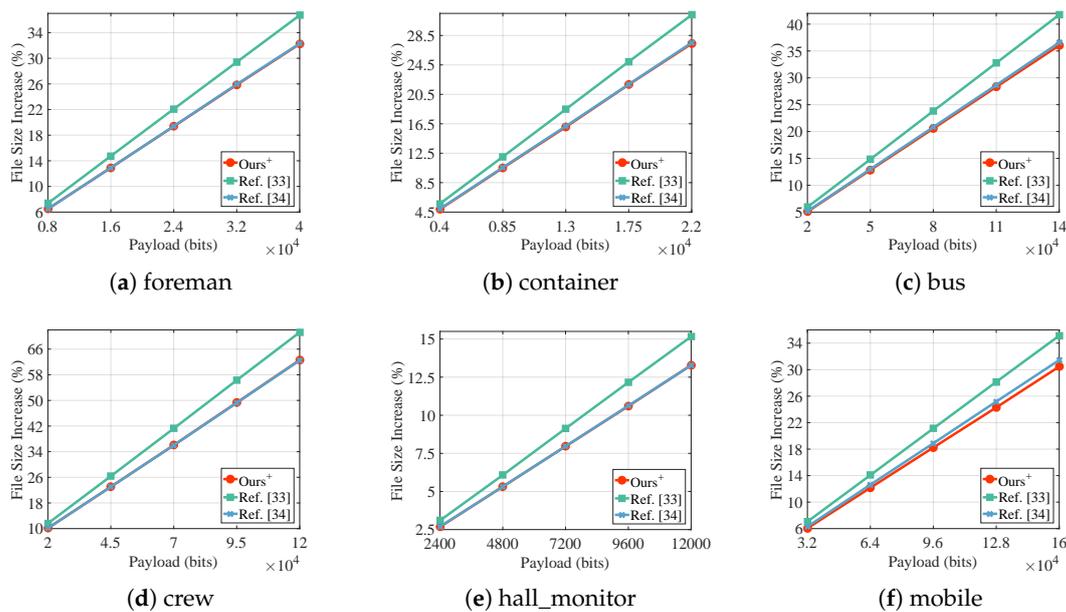


Figure 8. The increase in the file size of marked video generated by different schemes using the (0,0) coefficient pairs when QP is 28.

The experimental results of the schemes without using the (0,0) coefficient pairs are shown in Figures 9 and 10. It can be seen from Figures 9 and 10 that whether the QP is 16 or 28, the increase in file size of the marked video generated by our proposed scheme is basically smaller than that of [29,37] for all six videos. Moreover, as the payload increases, the differences between the file size increase of our scheme and those of the related schemes become more obvious. The reason is that the scheme proposed in [37] not only uses more points with lower embedding efficiency, but also needs to modify both values of many points without embedding any data. Moreover, due to the low embedding efficiency in [29], there are more coefficient pairs will be modified under the same payload, resulting

in more apparent increase in file size. The influence of these factors will be more obvious with the increase of the payload, which will lead to a growing impact on the file size.

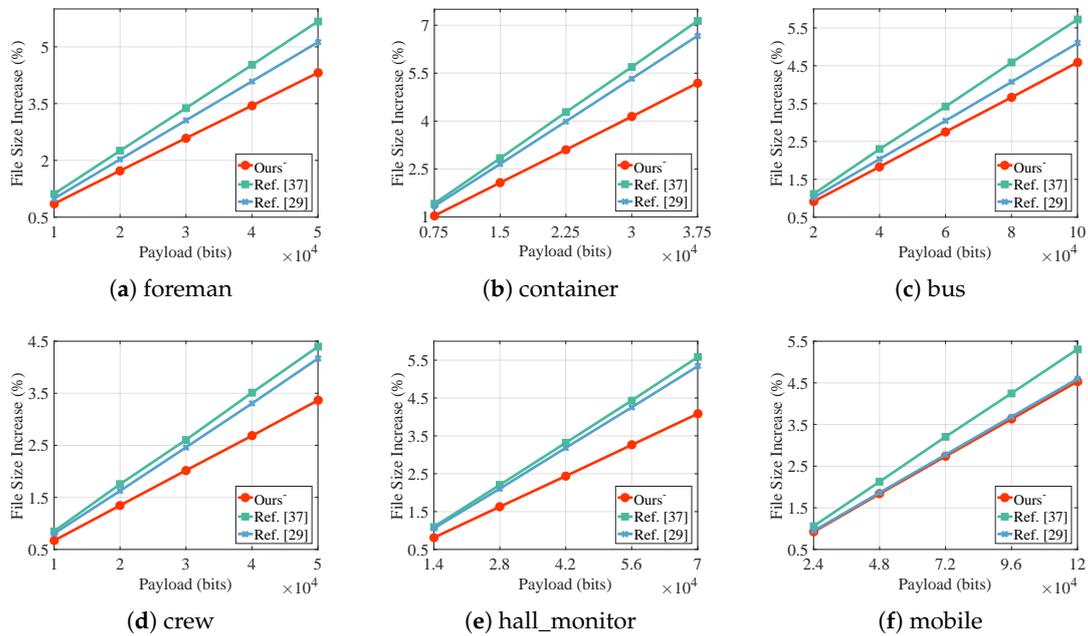


Figure 9. The increase in the file size of marked video generated by different schemes without using the (0,0) coefficient pairs when QP is 16.

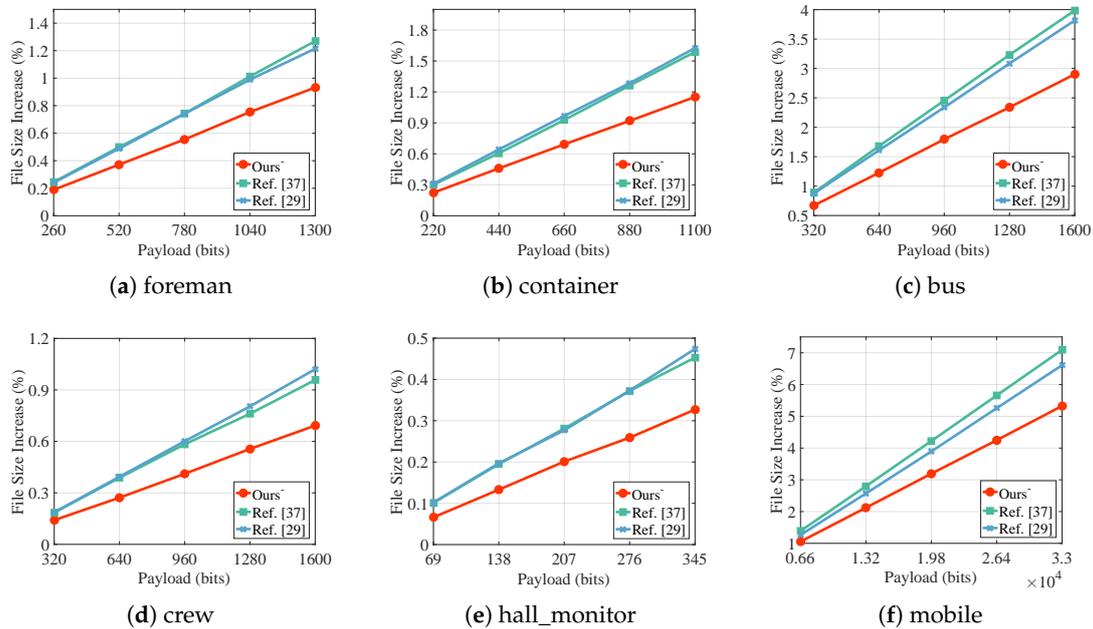


Figure 10. The increase in the file size of marked video generated by different schemes without using the (0,0) coefficient pairs when QP is 28.

5. Conclusions

In this paper, two new 2D HS-based RDH schemes for H.264/AVC video are presented, one of which uses the (0,0) coefficient pairs and the other does not. Based on the statistical distributions of DCT coefficient pairs, both schemes employ a better embedding strategy consisting of the embedding modes with high embedding efficiency. Moreover, to further reduce the embedding

distortion, secret data is only embedded into the DCT coefficients with middle and high frequencies. The experimental results demonstrated that our proposed schemes can achieve better visual quality and smaller increase in the file size of marked video compared with the related schemes.

Author Contributions: Conceptualization, J.H. and Y.X.; methodology, Y.X.; software, Y.X.; validation, Y.X. and J.H.; formal analysis, J.H.; investigation, J.H.; resources, J.H.; data curation, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, J.H.; visualization, Y.X.; supervision, J.H.; project administration, J.H.; funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Guangdong Natural Science Foundation under grant number 2019A1515011231, Guangdong Province Key Area R&D Program of China under grant number 2019B010137004.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Goljan, M.; Fridrich, J.J.; Du, R. Distortion-Free Data Embedding for Images. In Proceedings of the Information Hiding (IH'01), Pittsburgh, PA, USA, 25–27 April 2001; Volume 2137, pp. 27–41.
2. Celik, M.U.; Sharma, G.; Tekalp, A.M.; Saber, E. Lossless Generalized-LSB Data Embedding. *IEEE Trans. Image Process.* **2005**, *14*, 253–266. [[CrossRef](#)] [[PubMed](#)]
3. Tian, J. Reversible Data Embedding Using a Difference Expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. [[CrossRef](#)]
4. Hu, Y.; Lee, H.K.; Chen, K.; Li, J. Difference Expansion Based Reversible Data Hiding Using Two Embedding Directions. *IEEE Trans. Multimed.* **2008**, *10*, 1500–1512. [[CrossRef](#)]
5. Liu, M.; Seah, H.S.; Zhu, C.; Lin, W.; Tian, F. Reducing Location Map in Prediction-based Difference Expansion for Reversible Image Data Embedding. *Signal Process.* **2012**, *92*, 819–828. [[CrossRef](#)]
6. Caciula, I.; Coanda, H.G.; Coltuc, D. Multiple Moduli Prediction Error Expansion Reversible Data Hiding. *Signal Process. Image Commun.* **2019**, *71*, 120–127. [[CrossRef](#)]
7. Ni, Z.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible Data Hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–362.
8. Lin, C.C.; Tai, W.L.; Chang, C.C. Multilevel Reversible Data Hiding Based on Histogram Modification of Difference Images. *Pattern Recognit.* **2008**, *41*, 3582–3591. [[CrossRef](#)]
9. Tai, W.L.; Yeh, C.M.; Chang, C.C. Reversible Data Hiding Based on Histogram Modification of Pixel Differences. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 906–910.
10. Hong, W.; Chen, T.S.; Shiu, C.W. Reversible Data Hiding for High Quality Images Using Modification of Prediction Errors. *J. Syst. Softw.* **2009**, *82*, 1833–1842. [[CrossRef](#)]
11. Kim, S.; Qu, X.; Sachnev, V.; Kim, H.J. Skewed Histogram Shifting for Reversible Data Hiding Using a Pair of Extreme Predictions. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 3236–3246. [[CrossRef](#)]
12. Jia, Y.; Yin, Z.; Zhang, X.; Luo, Y. Reversible Data Hiding Based on Reducing Invalid Shifting of Pixels in Histogram Shifting. *Signal Process.* **2019**, *163*, 238–246. [[CrossRef](#)]
13. Jung, S.M.; On, B.W. An Advanced Reversible Data Hiding Algorithm Using Local Similarity, Curved Surface Characteristics, and Edge Characteristics in Images. *Appl. Sci.* **2020**, *10*, 836. [[CrossRef](#)]
14. Li, X.; Zhang, W.; Gui, X.; Yang, B. Efficient Reversible Data Hiding Based on Multiple Histograms Modification. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2016–2027.
15. Wang, J.; Ni, J.; Zhang, X.; Shi, Y.Q. Rate and Distortion Optimization for Reversible Data Hiding Using Multiple Histogram Shifting. *IEEE Trans. Cybern.* **2017**, *47*, 315–326. [[CrossRef](#)] [[PubMed](#)]
16. Ou, B.; Zhao, Y. High Capacity Reversible Data Hiding Based on Multiple Histograms Modification. *IEEE Trans. Circuits Syst. Video Technol.* **2019**. [[CrossRef](#)]
17. Wang, J.; Chen, X.; Ni, J.; Mao, N.; Shi, Y. Multiple Histograms Based Reversible Data Hiding: Framework and Realization. *IEEE Trans. Circuits Syst. Video Technol.* **2019**. [[CrossRef](#)]
18. Li, X.; Zhang, W.; Gui, X.; Yang, B. A Novel Reversible Data Hiding Scheme Based on Two-Dimensional Difference-Histogram Modification. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1091–1100.
19. Ou, B.; Li, X.; Zhao, Y.; Ni, R.; Shi, Y. Pairwise Prediction-Error Expansion for Efficient Reversible Data Hiding. *IEEE Trans. Image Process.* **2013**, *22*, 5010–5021. [[CrossRef](#)]

20. Ou, B.; Li, X.; Zhang, W.; Zhao, Y. Improving Pairwise PEE Via Hybrid-dimensional Histogram Generation and Adaptive Mapping Selection. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 2176–2190. [[CrossRef](#)]
21. Xiao, M.; Li, X.; Wang, Y.; Zhao, Y.; Ni, R. Reversible Data Hiding Based on Pairwise Embedding and Optimal Expansion Path. *Signal Process.* **2019**, *158*, 210–218. [[CrossRef](#)]
22. Wu, H.; Mai, W.; Meng, S.; Cheung, Y.; Tang, S. Reversible Data Hiding With Image Contrast Enhancement Based on Two-Dimensional Histogram Modification. *IEEE Access* **2019**, *7*, 83332–83342. [[CrossRef](#)]
23. Qin, J.; Huang, F. Reversible Data Hiding Based on Multiple Two-Dimensional Histograms Modification. *IEEE Signal Process. Lett.* **2019**, *26*, 843–847. [[CrossRef](#)]
24. Huang, F.; Qu, X.; Kim, H.J.; Huang, J. Reversible Data Hiding in JPEG Images. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1610–1621. [[CrossRef](#)]
25. Qian, Z.; Dai, S.; Chen, B. Reversible Data Hiding in JPEG Images Using Ordered Embedding. *KSII Trans. Internet Inf. Syst.* **2017**, *11*, 945–958.
26. Wedaj, F.T.; Kim, S.; Kim, H.J.; Huang, F. Improved Reversible Data Hiding in JPEG Images Based on New Coefficient Selection Strategy. *EURASIP J. Image Video Process.* **2017**, *2017*, 63. [[CrossRef](#)]
27. Hou, D.; Wang, H.; Zhang, W.; Yu, N. Reversible Data Hiding in JPEG Image Based on DCT Frequency and Block Selection. *Signal Process.* **2018**, *148*, 41–47. [[CrossRef](#)]
28. He, J.; Chen, J.; Tang, S. Reversible Data Hiding in JPEG Images Based on Negative Influence Models. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2121–2133. [[CrossRef](#)]
29. Cheng, S.; Huang, F. Reversible Data Hiding in JPEG Images Based on Two-Dimensional Histogram Modification. In Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS'18), Haikou, China, 8–10 June 2018; pp. 392–403.
30. Chung, K.; Huang, Y.; Chang, P.; Liao, H.M. Reversible Data Hiding-Based Approach for Intra-Frame Error Concealment in H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 1643–1647. [[CrossRef](#)]
31. Fallahpour, M.; Shirmohammadi, S.; Ghanbari, M. A High Capacity Data Hiding Algorithm for H.264/AVC Video. *Secur. Commun. Netw.* **2015**, *8*, 2947–2955. [[CrossRef](#)]
32. 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](#)]
33. Xu, D.; Wang, R. Two-dimensional Reversible Data Hiding-based Approach for Intra-Frame Error Concealment in H.264/AVC. *Signal Process. Image Commun.* **2016**, *47*, 369–379. [[CrossRef](#)]
34. Zhao, J.; Li, Z.T.; Feng, B. A Novel Two-dimensional Histogram Modification for Reversible Data Embedding into Stereo H.264 Video. *Multimed. Tools Appl.* **2016**, *75*, 5959–5980. [[CrossRef](#)]
35. Kim, H.; Kang, S.U. Genuine Reversible Data Hiding Technology Using Compensation for H.264 Bitstreams. *Multimed. Tools Appl.* **2018**, *77*, 8043–8060. [[CrossRef](#)]
36. Niu, K.; Yang, X.; Zhang, Y. A Novel Video Reversible Data Hiding Algorithm Using Motion Vector for H.264/AVC. *Tsinghua Sci. Technol.* **2017**, *22*, 489–498. [[CrossRef](#)]
37. Li, D.; Zhang, Y.; Li, X.; Niu, K.; Yang, X.; Sun, Y. Two-dimensional Histogram Modification Based Reversible Data Hiding Using Motion Vector for H.264. *Multimed. Tools Appl.* **2019**, *78*, 8167–8181. [[CrossRef](#)]
38. Sikora, T. The MPEG-4 Video Standard Verification Model. *IEEE Trans. Circuits Syst. Video Technol.* **1997**, *7*, 19–31. [[CrossRef](#)]
39. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A. Overview of the H.264/AVC Video Coding Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 560–576. [[CrossRef](#)]
40. Sullivan, G.J.; Ohm, J.; Han, W.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [[CrossRef](#)]
41. Ferroukhi, M.; Ouahabi, A.; Attari, M.; Habchi, Y.; Taleb-Ahmed, A. Medical Video Coding Based on 2nd-Generation Wavelets: Performance Evaluation. *Electronics* **2019**, *8*, 88. [[CrossRef](#)]
42. Ouahabi, A. *Signal and Image Multiresolution Analysis*; Wiley-ISTE: London, UK, 2012.
43. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)]

