

# A Reversible Data Hiding Scheme Based on Code Division Multiplexing

Bin Ma Yun Q. Shi *Fellow, IEEE*

**Abstract**— In this paper, a novel code division multiplexing (CDM) algorithm based reversible data hiding (RDH) scheme is presented. The covert data are denoted by different orthogonal spreading sequences and embedded into the cover image. The original image can be completely recovered after the data have been extracted exactly. The Walsh Hadamard matrix is employed to generate orthogonal spreading sequences, by which the data can be overlappingly embedded without interfering each other, and multilevel data embedding can be utilized to enlarge the embedding capacity. Furthermore, most elements of different spreading sequences are mutually cancelled when they are overlappingly embedded, which maintains the image in good quality even with a high embedding payload. A location-map free method is presented in the paper to save more space for data embedding, and the overflow/underflow problem is solved by shrinking the distribution of the image histogram on both ends. This would further improve the embedding performance. Experimental results have demonstrated that the CDM-based RDH scheme can achieve best performance at the moderate-to-high embedding capacity compared with other state-of-the-art schemes.

**Index Terms**—Code division multiplexing (CDM), embedding capacity, reversible data hiding (RDH), prediction-error, location-map free.

## I. INTRODUCTION

DATA hiding is an effective method for covert communication; it aims at hiding covert data into a cover-medium such that the invader will not be aware of its existence. Generally, the data hiding introduces permanent distortion to the cover image and the original cover cannot be reconstructed. But in some sensitive applications, such as military image protection, medical image sharing, law forensics and multimedia archive management, the cover image is so crucial that no distortion is allowed. This requirement creates a new research field named reversible data hiding (RDH), in which the original cover can be completely restored after the embedded data have been extracted. In general, the major concerns of data hiding techniques are imperceptibility and

embedding capacity. Many RDH schemes have been proposed in recent years. They can be roughly classified into two categories: schemes performed in spatial domain and those in transform domain.

Since the earliest RDH scheme was presented in Barton's patent [1], many RDH schemes have been reported in the literature. In the spatial domain, three major approaches have been developed for image RDH. They are RDH based on lossless compression, on histogram modification and on difference expansion (or prediction-error expansion). The lossless compression based approach substitutes part of host image information with the to-be-embedded data and the compressed codes of the substituted part. Fridrich *et al.* [2] provided an invertible image authentication method by embedding a hash into the image. By losslessly compressing the lowest bit-plane of the image (or its feature), they could make enough space to insert the image of a hash. Celik *et al.* [3] improved Fridrich *et al.*'s approach by employing the lossless image compression algorithm CALIC (context-based, adaptive, lossless image code) for lossless image compression and achieved higher embedding capacity. Kalker and Willems [4] derived theoretical capacity bounds for image RDH and proposed the recursive code construction to achieve high data hiding performance. Zhang *et al.* [5] embedded the data by recursively utilizing the decompression and compression processes of an entropy code, and proved that the rate-distortion bound of RDH can be approached as long as perfect compression is realized. However, in order to keep imperceptibility of the marked image, the substituted parts generally locate in the least significant bits and it can only provide a limited compression ratio. Thus, the lossless compression based RDH schemes usually provide limited embedding capacity.

Introduced by Tian [6], difference expansion is a very fruitful RDH research field, which expands the difference of a pixel-pair two times, if no overflow or underflow appears, one bit can be added to the expanded difference (the maximum achievable embedding rate is 0.5 bit/pixel). The embeddable pixel-pair is identified by using a location map with one bit for each pair of pixels. Since the location map size is the same as the embedding capacity, its performance is partially decided by the compression efficiency of the location map. Alattar [7] developed an efficient algorithm to hide several bits into the expansion of the difference between adjacent pixels in a vector, by which they can still hide certain data even though the location map is not compressed. The scheme expands one cell from a pair to a triple, or a quad, to hide more bits per cell in experiments. Simulation results have shown that they can

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org

This work was supported in part by Shandong Provincial Natural Science Foundation, China (No. ZR2012F014), the National Natural Science Foundation of China (No. 41202206) and Jinan university & institutes innovation program (JN201402005)

B. Ma is with the School of Information Science, Qilu University of Technology, Jinan 250014, China (e-mail: mab@qlu.edu.cn).

Yun Q. Shi is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: shi@njit.edu).

embed larger payload than Tian's at the same image distortion (the maximum achievable embedding rate is 0.67 bit/pixel for triple and 0.75 bit/pixel for quad). Kamstra and Heijmans [8] presented an alternative method to solve the capacity control problem in Tian's method. The main idea of this technique is sorting pixel pairs according to correlation measures, which can automatically create just enough capacity to embed the desired payload and keep the distortion minimal. Kim *et al.* [9] presented a difference expansion method with a simplified location-map and achieved high embedding capacity while keeping the distortion at a low level. The location map only records the ambiguous pixels between expandable and un-expandable parts, and the size is reduced considerably. Its performance is proved to be better than the schemes proposed by Tian and its improved version provided by Kamstra and Heijmans. Thodi and Rodriguez [10] proposed a histogram shifting technique as an alternative to embedding the location map, which improves the distortion performance at low embedding capacities and mitigates the capacity control problem. They also introduced the prediction-error expansion method which exploits the correlations inherent in the neighborhood of pixels better than Tian's difference expansion scheme. Prediction-error expansion and histogram shifting are combined to form an effective method for RDH. Generally, the prediction-error histogram is Laplacian-like and distributed around zero sharply. Therefore, the better the predictor is, the smaller the prediction-error is, and the higher the embedding capacity would be achieved at low distortion. From then on, many error-prediction algorithms such as the median edge detector (MED) predictor [11], the gradient adjusted predictor (GAP) [12], simple gradient adjusted predictor (SGAP) [13] and rhombus-error predictor [14][15] are proposed aiming to provide smaller prediction-errors. Sachnev *et al.* [15] employed a rhombus-error predictor to achieve smaller prediction-errors, and a sorting technique is employed to record the prediction-errors based on the magnitude of its local variance. Using sorted prediction-errors and a reduced size location map, they embed more data into the image with less distortion. Chen *et al.* [16] investigated the selection of a "model order" for RDH through theoretical analysis and experimental examination. They have demonstrated that the full context predictor can achieve highly fitted modeling at a low order. It is capable of providing better performance than most popular prediction algorithms and need only negligible computing. Li *et al.* [17] presented an efficient RDH scheme based on the adaptive pixel selection and the prediction-error expansion techniques, 1 or 2 bits are adaptively embedded into expandable pixels according to local complexity. This doing avoids expanding pixels with large prediction-errors and decreases the pixel value modification. Thus, the embedding capacity is improved and the image quality is well retained. Han *et al.* [18] designed a reversible image authentication method that does not memorize the parameters for extracting the embedded authentication data from an image. The hidden data can be extracted without memorization of parameters used in the algorithm. Fujiyoshi *et al.* [19] introduced a parameter memorization-free lossless data hiding method accepting various sized payload. The method hides data into consecutive pixel blocks in an ascending order of their roughness, so that it does not need to remember the embedded blocks in the

decoding stage. Moreover, the method is free from fixing the payload size. Zhang [20] established an optimal rule of value modification under a payload-distortion criterion by using an iterative procedure. The estimation errors are modified according to the optimal value transfer rule; the host image is divided into some pixel subsets, and the auxiliary information of a subset is always embedded into the next subset, by which a high-performance practical RDH scheme is achieved. Coltuc [21] embedded the bits not only into the current pixel, but also into its prediction context simultaneously, which would minimize the mean square errors (MSE) and thus introduced less distortion to the cover image. Dragoi *et al.* [22] adopted a local prediction technique in difference-expansion based RDH scheme. The least square predictor is computed on a square block, and the corresponding prediction-error is expanded for data embedding. It can be applied regardless of the prediction order or the prediction context and achieve advanced performance.

Histogram modification is another popular approach to RDH, in which the prominent histogram bins are selected and the spaces are created into its adjacent bins for data embedding by shifting other histogram bins certain positions. Ni *et al.* [23][24] first proposed the RDH scheme using histogram bins of the cover image for data embedding. The peak and the minimum points of the histogram of an image are utilized for data embedding. It changes the pixel value at most by 1 during one-time data embedding, and thus keeping the marked image in good visual quality. However, this method does not work well if the cover image has a "flat" histogram. Lee *et al.* [25] utilized histogram of the difference image and modified pixel values slightly to achieve more embedding capacity than Ni *et al.*'s method. Hong [26] proposed an adaptive method to increase the number of embeddable spaces by referencing a dual binary tree. This scheme significantly improved the performance especially at a low embedding capacity. Yang *et al.* [27] developed an interleaving prediction method that enables the number of prediction-errors to be as many as the number of pixels. All predictive error values are transformed into the histogram to create higher peak values and to improve the embedding capacity. Xuan *et al.*'s [28] reversibly embedded data into image prediction-errors by using histogram-pairs with four thresholds for optimal performance. A fluctuation threshold is utilized to choose prediction-errors whose associated neighboring fluctuation is below the threshold to reduce image distortion. Li *et al.* [29] presented a general framework to construct the histogram modification-based RDH scheme, and introduced two novel and efficient RDH algorithms to further demonstrate the universality and applicability of the framework. Coatrieux *et al.* [30] designed an efficient histogram modulation method which takes care of the local specificities of the image content. They make use of a classification process to enable parts of the image embedded with the most suited reversible modulations, by which the data can be inserted in rough regions where other methods fail to do so.

Besides the aforementioned schemes, many scientists have studied RDH techniques in transform domain to enhance its security and robustness. Xuan *et al.* [31] embedded data into the coefficients of high frequency sub bands of the integer wavelet domain, and adopted compression and expansion

functions to embed data into the image to achieve high performance in terms of embedding capacity and visual quality. Yang *et al.* [32] takes advantage of the character of integer discrete cosine transform (DCT) coefficients' Laplacian-like distribution to embed data in coefficients of different frequencies. Zou *et al.* [33] presented a semi-fragile lossless digital watermarking scheme based on 5/3 integer wavelet domain. The hidden message can be retrieved even after incidental alternations or image compression. Lee *et al.* [34] divided the cover image into non-overlapping blocks and adaptively embedded covert data into the high-frequency wavelet coefficients of each block, the bit-shiftable blocks in which the data can be embedded with less distortion are found and are utilized for data hiding to minimize the distortion between the marked and the original image. An *et al.* [35] introduced a pragmatic framework of wavelet-domain statistical quantity histogram shifting and clustering (WSQH-SC) to enhance RDH robustness and reduce its computing complexity.

Although RDH has made considerable improvement in recent years, the embedding algorithms are growing more complicated and the computational costs are increasing sharply, in most cases, the image quality degrades rapidly as the payload increases. In this paper, a novel RDH scheme based on CDM algorithm is proposed. The to-be-embedded data are denoted by different orthogonal spreading sequences and embedded into the cover image; the original image can be completely reconstructed after the hidden data have been extracted out exactly. The *Walsh Hadamard* matrix is adopted to generate orthogonal spreading sequences, by which the data can be overlappingly embedded without interfering each other, and multilevel data embedding can be utilized to enlarge the embedding capacity. Most elements of different spreading sequences are mutually cancelled when they are overlappingly embedded, which maintains image distortion at a lower level even with a high embedding payload. Furthermore, the location map is not needed to identify the embedded locations in this scheme because some pseudo bits are embedded, and they will be dropped during the data extraction. The overflow/underflow problem is solved by shrinking the image histogram distribution from both ends. These would further enhance the embedding performance. Experimental results have demonstrated that the proposed scheme outperforms most existing RDH schemes at the moderate-to-high embedding capacity.

The rest of this paper is organized as follows. In section II, the principle of CDM-based communication system is introduced. Section III provides a CDM-Based RDH scheme. In Section IV, a location map free method is presented to increase the embedding capacity; the overflow/underflow problem is also solved practically. The implementation of the proposed scheme is provided in Section V. Section VI demonstrates the experimental results. Some discussions are given in section VII. Finally, the conclusion is drawn in section VIII.

## II. CDM-BASED COMMUNICATION SYSTEM

CDM is a kind of spectrum spreading technique widely used in communication system for secure information transmission

and channel multiplexing [36]. In a CDM-based communication system, the sender is assigned with a predetermined spreading sequence to encode the secret digital signals for transmitting. Only the receiver who knows the same sequence can correctly decode and extract the secret signals. The spreading sequence plays an important role in a CDM-based communication system. Generally, the *Walsh Hadamard* matrix is employed to derive the spreading sequences. It consists of "1" or "-1", and every two rows or columns are orthogonal to each other. Since the number of "1" and "-1" is equal in each row or column, the sequence created by the row or column of *Walsh Hadamard* matrix is zero mean and orthogonal to each other.

Suppose  $S = (g_\sigma)_{\text{bom}}$  is a row or a column from *Walsh Hadamard* matrix, it thus satisfies

$$g_\sigma \in \{-1, 1\}, \sum_{\sigma=1}^m g_\sigma = 0 \quad (1)$$

The cross-correlation of any two different sequences is

$$\langle S_r, S_t \rangle = S_r \cdot S_t^T = 0 \quad (r \neq t) \quad (2)$$

Each user is assigned a binary spreading sequence in a CDM-based communication system; bit "1" is denoted by the sequence itself, and bit "0" is represented by the negative corresponding sequence. Only the receiver, who has the same spreading sequence as the sender, can obtain the correct signals.

For example, suppose the two spreading sequences from a 4-level *Hadamard* matrix are  $S_1 = (1, -1, 1, -1)$  and  $S_2 = (1, -1, -1, 1)$ . Clearly,  $S_1$  and  $S_2$  are zero mean and orthogonal to each other. Two binary bits "1" and "0" are represented by  $S_1$  and  $-S_2$ , respectively; the two spreading sequences are added and transmitted together in a CDM-based communication system. On the sender side, the compound signal is  $S = S_1 + (-S_2) = S_1 - S_2 = (0, 0, 2, -2)$ . On the receiver side, for the receiver with the spreading sequence  $S_1$ , the decoding result is  $\frac{S \cdot S_1}{|S_1|^2} = \frac{(S_1 - S_2) \cdot S_1}{|S_1|^2} = \frac{S_1 \cdot S_1}{|S_1|^2} = 1$ , which represents bit 1; for the receiver with the spreading sequence  $S_2$ , the decoding result is  $\frac{S \cdot S_2}{|S_2|^2} = \frac{(S_1 - S_2) \cdot S_2}{|S_2|^2} = \frac{-S_2 \cdot S_2}{|S_2|^2} = -1$ , which represents bit 0. The result would be neither "1" nor "-1" if other spreading sequences are utilized for decoding. Therefore, the secret signals can be transmitted securely and the transmission capacity is enlarged. Moreover, since the noise-like orthogonal spreading sequences can be generated in different ways, which result in a highly complicated encoding and it is impossible for an intruder to guess them. The CDM-based communication system thus provides high capacity and security.

## III. CDM FOR REVERSIBLE DATA HIDING

The reversible data hiding process has great similarity with a communication system, where the digital signals are transmitted through public transmission channels to receivers who can reconstruct it completely (the covert data as the digital signals and the cover image as the public transmission channel).

Inspired by the CDM-based communication system [36], we present a CDM-based RDH scheme in this section, which satisfies high embedding capacity, high visual quality, and low

computing complexity. After data embedding is discussed, data extraction and image reconstruction are presented; and then multilevel data embedding are explained in detail.

### A. Data Embedding

Suppose  $W_o = [\omega_1, \omega_2, \dots, \omega_n] (\omega_z \in \{1, 0\}, z \in \{1, 2, \dots, n\})$  is the primitive data to be embedded, whose elements are changed to antipodal bits with the following formula

$$b_e = \begin{cases} 1 & \text{if } \omega_z = 1; \\ -1 & \text{if } \omega_z = 0; \end{cases} \quad (3)$$

The modified data is  $W_e = [b_1, b_2, \dots, b_n]$ , where  $b_e \in \{-1, 1\}$ ,  $e \in \{1, 2, \dots, n\}$ . Choose  $k$  mutually orthogonal sequences  $S_i = \{s_{i1}, s_{i2}, \dots, s_{il}\} (i \in \{1, 2, \dots, k\})$  from a specified *Walsh Hadamard* matrix. The sequence length  $l$  is an even number, and the number of “1” and “-1” in each sequence is equal, which implies that the spreading sequences are zero mean and orthogonal to each other. Let  $I$  be the original image with the size of  $N \times N$ . Select adjacent pixels from the image to form the vector  $V_j = [p_1, p_2, \dots, p_l]$ , whose length  $l$  is the same as that of  $S_i$ . The bits can be embedded as

$$\hat{V}_j = V_j + \alpha [b_1 S_1 + b_2 S_2 + \dots + b_k S_k] \quad (4)$$

where  $k$  is the number of to-be-embedded bits,  $\alpha$  is the gain factor, which is a positive integer controlling the embedding intensity. The bigger the  $\alpha$  value is, the stronger the embedding is, and the noisier the image would be. The marked image is achieved by replacing all  $V_j$  vectors with  $\hat{V}_j$ .

### B. Data Extraction and Image Recovery

Let  $\hat{I}$  be the marked image. Build  $\hat{V}_j$  with the same method as the embedding stage. The  $j$ -th embedded data bit can be extracted by calculating the cross correlation between the marked vector  $\hat{V}_j$  and the spreading sequence  $S_i$  as

$$\langle \hat{V}_j, S_i \rangle = \hat{V}_j \cdot S_i^T = V_j \cdot S_i^T + \alpha [b_1 S_1 \cdot S_i^T + b_2 S_2 \cdot S_i^T + \dots + b_k S_k \cdot S_i^T] \quad (5)$$

Since the spreading sequences are mutually orthogonal, Eq. (5) can be simplified as

$$\langle \hat{V}_j, S_i \rangle = V_j \cdot S_i^T + \alpha b_e S_i \cdot S_i^T \quad (6)$$

where the value of  $b_e$  is “1” or “-1”,  $\alpha$  is a positive integer and  $S_i \cdot S_i^T$  is always positive, the sign of expression  $\alpha b_e S_i \cdot S_i^T$  is determined by  $b_e$ . Therefore, under the condition of  $|V_j \cdot S_i^T| < |\alpha S_i \cdot S_i^T|$ , if  $b_e = 1$ , we have  $\alpha b_e S_i \cdot S_i^T > |V_j \cdot S_i^T| > 0$ , thus  $\langle \hat{V}_j, S_i \rangle = V_j \cdot S_i^T + \alpha b_e S_i \cdot S_i^T > 0$ ; On the other hand, if  $b_e = -1$ , we have  $\alpha b_e S_i \cdot S_i^T < -|V_j \cdot S_i^T| < 0$ , there must be  $\langle \hat{V}_j, S_i \rangle = V_j \cdot S_i^T + \alpha b_e S_i \cdot S_i^T < 0$ . The sign of  $\langle \hat{V}_j, S_i \rangle$  is the same as the sign of  $b_e$ . Since the value of  $b_e$  is “1” or “-1”, the embedded bit  $b_e$  can be obtained by the function of  $\text{sign}(\langle \hat{V}_j, S_i \rangle)$  in this condition.

Whenever the condition  $|V_j \cdot S_i^T| < |\alpha S_i \cdot S_i^T|$  is satisfied, the embedded bit  $b_e$  can be extracted correctly according to

$$b_e = \text{sign}(\langle \hat{V}_j, S_i \rangle) \quad (\text{if } |V_j \cdot S_i^T| < |\alpha S_i \cdot S_i^T|) \quad (7)$$

Furthermore, at the decoding stage, for the usable vectors, Eq. (6) satisfies

$$\langle \hat{V}_j, S_i \rangle = V_j \cdot S_i^T + \alpha b_e S_i \cdot S_i^T < 2|\alpha S_i \cdot S_i^T| \quad (8)$$

Then, if the Eq. (8) is satisfied at the receiver side, the embedded bit can be extracted by the function of  $\text{sign}(\langle \hat{V}_j, S_i \rangle)$ . Since  $S_i$  is a zero-mean spreading sequence, the expression  $V_j \cdot S_i^T$  is equal to sum the differences of all pixel-pairs in vector  $V_j$ . The value of  $V_j \cdot S_i^T$  would be small if the elements of  $V_j$  are similar. Considering the redundancy existing in natural image content, the neighboring pixels are quite similar especially in smooth (low-frequency) areas, the product of  $V_j \cdot S_i^T$  is usually very small in practice. Thus the bits can be embedded into the image in most cases. In addition, with the increase of  $\alpha$ , the value of  $\alpha S_i \cdot S_i^T$  grows rapidly. More pixels would be involved to form the vectors  $V_j$  that suitable for data embedding, and thus the embedding capacity is enlarged.

After the extraction of embedded data from the marked image, the original image can be restored with all  $\hat{V}_j$  replaced by  $V_j$  as

$$V_j = \hat{V}_j - \alpha [b_1 S_1 + b_2 S_2 + \dots + b_k S_k] \quad (9)$$

To well explain the idea of our proposed scheme, we choose the first  $2 \times 2$  pixels in the top-left corner of  $512 \times 512$  grayscale image *Lena* to form the vector  $V_1 = [165, 165, 166, 166]$ ; let  $S_1 = [1, -1, 1, -1]$ ,  $\alpha = 1$ , and the to-be-embedded data bit  $b_1 = 1$ . The vector after data embedding is  $\hat{V}_1 = V_1 + \alpha b_1 S_1 = [166, 164, 167, 165]$ . At the receiver side, we have  $\langle \hat{V}_1, S_1 \rangle = (166, 164, 167, 165) \cdot (1, -1, 1, -1)^T = 4$  and  $2|\alpha S_1 \cdot S_1^T| = 2|(1, -1, 1, -1)(1, -1, 1, -1)^T| = 8$ , the condition  $\langle \hat{V}_1, S_1 \rangle < 2|\alpha S_1 \cdot S_1^T|$  is satisfied. Thus, the embedded bit and the original vector can be recovered according to the equations of  $b_1 = \text{sign}(\langle \hat{V}_1, S_1 \rangle) = \text{sign}(4) = 1$  and  $V_1 = \hat{V}_1 - \alpha b_1 S_1 = [165, 165, 166, 166]$ . Similarly, in the light of Eq. (7), all the vectors in the original image satisfy  $|V_j \cdot S_i^T| < 4$  (i.e.,  $V_j \cdot S_i^T = 0, \pm 1, \pm 2, \pm 3$ ) hence are suitable for RDH in this example.

### C. Multilevel Data Embedding

RDH aims to embed covert data into an image while only slightly degrade its quality. Since the embedded data and the cover image can be restored completely from the marked image, multilevel data embedding is feasible. Furthermore, as the spreading sequences are mutually orthogonal, the cross-correlation of any two different spreading sequences is 0; the spreading sequences can be overlappingly embedded without interfering each other.

For example, regarding the example we just presented above, we can embed another bit into the marked vector  $\hat{V}_1$ . Suppose the second to-be-embedded bit is 0 (then,  $b_2 = -1$  according to Eq. (3)), set the spreading sequence  $S_2 = [1, 1, -1, -1]$  and  $\alpha = 1$ , the vector  $V_1 = [165, 165, 166, 166]$  would be changed to  $\hat{\hat{V}}_1 = \hat{V}_1 + \alpha b_2 S_2 = [165, 163, 168, 166]$  after double data embedding. At the decoding side, the embedded bits and the original vector can be restored according to the following equations (as shown in Table I):

TABLE I  
THE DETAILED DATA EXTRACTION PERIODS OF MULTILEVEL  
DATA EMBEDDING

First period of decoding	Second period of decoding
$\langle \hat{V}_1, S_2 \rangle = (165, 163, 168, 166)(1, 1, -1, -1) = -6$ ;	$\langle \hat{V}_2, S_1 \rangle = (166, 164, 167, 165)(1, -1, 1, -1)^T = 4$ ;
$2 \alpha S_2 \cdot S_2^T  = 8$ ;	$2 \alpha S_1 \cdot S_1^T  = 8$ ;
$\therefore  \hat{V}_1 \cdot S_2^T  =  \langle \hat{V}_1, S_2 \rangle  < 2 \alpha S_2 \cdot S_2^T $	$\therefore  \hat{V}_2 \cdot S_1^T  =  \langle \hat{V}_2, S_1 \rangle  < 2 \alpha S_1 \cdot S_1^T $
thus,	thus,
$b_2 = \text{sign}(\langle \hat{V}_1, S_2 \rangle) = \text{sign}(-6) = -1$ ;	$b_1 = \text{sign}(\langle \hat{V}_2, S_1 \rangle) = \text{sign}(4) = 1$ ;
$\alpha_2 = 0$ ;	$\alpha_1 = 1$ ;
$\hat{V}_1 = \hat{V}_1 - \alpha b_2 S_2 = [166, 164, 167, 165]$	$\hat{V}_2 = \hat{V}_2 - \alpha b_1 S_1 = [165, 165, 166, 166]$

In this process, only the second and the third element of the marked vector  $\hat{V}_1$  changed by 2 in value; other elements of the marked vector  $\hat{V}_1$  are the same as the original vector. The first and the forth element of two embedded spreading sequences are mutually cancelled after two times data embedding. Actually, as most elements of different spreading sequences are mutually cancelled in multilevel data embedding, it enables the proposed scheme to achieve high embedding capacity while restraining distortion to a low level.

Here, we study the capability of the elements to mutually cancel each other in the case of different spreading sequences that are overlappingly embedded. The gain factor  $\alpha$  is predefined as 1. Suppose the spreading sequence is  $S_1 = [1, -1, 1, -1]$  and the to-be-embedded bit is “1” (i.e.  $b_1 = 1$ ) at first, each pixel of vector  $V_j$  is changed by 1 in grayscale value after the first time of embedding, and we denoted the vector after first embedding as  $\hat{V}_j$ . In the second time of embedding, the bit to-be-embedded maybe “0” or “1”, and the statistical probability is 50%, respectively. We define the spreading sequence  $S_2 = [1, 1, -1, -1]$ , if the to-be-embedded bit is “0” (i.e.  $b_2 = -1$ ), then,  $\alpha b_2 S_2 = [-1, -1, 1, 1]$  would be added to the modified vector  $\hat{V}_j$ , the final change of the elements in  $V_j$  is  $\alpha b_1 S_1 + \alpha b_2 S_2 = [0, -2, 2, 0]$ . If the to-be-embedded bit is “1”

(i.e.  $b_2 = 1$ ),  $\alpha b_2 S_2 = [1, 1, -1, -1]$  will be added to the modified vector  $\hat{V}_j$ . The final change of the elements in  $V_j$  would be  $\alpha b_1 S_1 + \alpha b_2 S_2 = [2, 0, 0, -2]$ . Thus, considering the occurrence probability of the to-be-embedded bits “1” and “0”, the probable mean square error (PMSE) can be computed as  $PMSE = ((\frac{2^2 + (-2)^2}{4}) + (\frac{(-2)^2 + 2^2}{4})) / 2 = 2$ . Similarly, we can

obtain the PMSE after third time embedding is  $PMSE = 3$ , and so on. The value of PMSE equals the number of embedding levels. The detailed results are listed in Table II. Further investigations demonstrate that the value of PMSE is also the same as the embedding levels when the length of spreading sequence is 2, 8, 16 and so on. As most elements of different spreading sequence are mutually cancelled in multilevel data embedding, the image quality drops slowly when the payload increases.

Generally, the maximum modification to each pixel is 1 on its grayscale value for single embedding; the PSNR of a marked image is about 48.13dB after the first level embedding (MSE is 1) and 45.12dB after the second level embedding (MSE is 2). Since no degradation is noticeable to the human eyes if the PSNR value above 36dB, the proposed scheme has excellent imperceptibility until the MSE reaches 16 (the PSNR is 36.08dB). That is, the cover image can be overlappingly embedded for 16 times while still maintaining acceptable visual quality (if  $\alpha = 1$ ). On the other hand, as the embedded bit is denoted by a spreading sequence as discussed before, say  $S_1$  or  $S_2$ , in the proposed scheme, there are “ $l$ ” ( $l$  is the length of spreading sequence) pixels would be changed for every bit embedding. Thus, the maximum achievable embedding capacity is 0.5 bpp for one-time data embedding even if the spreading sequence length is 2 (i.e.,  $l = 2$ ). The maximum achievable embedding capacity would be  $0.5 \times n$  bpp in multilevel data embedding ( $n$  is the number of levels of data embedding). In practice, the capacity of each embedding decreases gradually as the redundancy of the marked image becomes less and less in multilevel data embedding.

TABLE II  
EXAMPLE OF PROBABLE MSE CHANGES OF MULTILEVEL DATA EMBEDDING

First embedding $S_1 = [1, -1, 1, -1]$		Difference after 1 <sup>st</sup> embedding	Second embedding $S_2 = [1, 1, -1, -1]$		Difference after 2 <sup>nd</sup> embedding	Third embedding $S_3 = [1, -1, -1, 1]$		Difference after 3 <sup>rd</sup> embedding
Probable bit	Spreading sequence		Probable bit	Spreading sequence		Probable bit	Spreading sequence	
1 ( $b_e = 1$ )	[1, -1, 1, -1]	[1, -1, 1, -1]	1 ( $b_e = 1$ )	[1, 1, -1, -1]	[2, 0, 0, -2]	1 ( $b_e = 1$ )	[1, -1, -1, 1]	[3, -1, -1, -1]
			0 ( $b_e = -1$ )	[-1, 1, 1, 1]	[0, -2, 2, 0]	0 ( $b_e = -1$ )	[-1, 1, 1, -1]	[1, 1, 1, -3]
			1 ( $b_e = 1$ )	[1, -1, -1, 1]	[0, 2, -2, 0]	1 ( $b_e = 1$ )	[1, -1, -1, 1]	[1, 1, -3, 1, 1]
			0 ( $b_e = -1$ )	[-1, 1, 1, -1]	[-1, -1, 3, -1]			
0 ( $b_e = -1$ )	[-1, 1, -1, 1]	[-1, 1, -1, 1]	1 ( $b_e = 1$ )	[1, 1, -1, -1]	[0, 2, -2, 0]	1 ( $b_e = 1$ )	[1, -1, -1, 1]	[1, 1, -3, 1]
			0 ( $b_e = -1$ )	[-1, -1, 1, 1]	[-2, 0, 0, 2]	0 ( $b_e = -1$ )	[-1, 1, 1, -1]	[-1, 3, -1, -1]
			1 ( $b_e = 1$ )	[1, -1, -1, 1]	[-1, -1, -1, 3]			
			0 ( $b_e = -1$ )	[-1, 1, 1, -1]	[-3, 1, 1, 1]			
<b>pMSE</b>		<b>1</b>			<b>2</b>			<b>3</b>

Note: In the column headers of “Difference after 1<sup>st</sup> embedding”, “Difference after 2<sup>nd</sup> embedding” and “Difference after 3<sup>rd</sup> embedding”, the difference is the change between the Original and Marked Vector.

#### IV. TWO IMPORTANT ISSUES

In the process of data embedding, some assistant techniques are needed to guarantee that the embedded data can be extracted exactly and the original image can be restored completely. Usually, they contain the following two items:

- 1) The method to identify those vectors containing embedded bits or not.
- 2) The method to solve the problem of overflow/underflow.

In this section, we present a location-map-free method to record the usable vectors, and develop a “flag array” to solve the problem of overflow/underflow.

##### A. Location Map Free Method

In most existing data hiding schemes, the location map is employed to locate the data embedding positions. It is generally embedded into the image as auxiliary information together with the payload, which reduces the net embedding capacity and decrease the efficiency of extraction at receiver side. Furthermore, since it is not easy to estimate the size of a location map, the iteration has to be employed to embed data into the cover image. It also reduces the efficiency of the scheme. Thus, a RDH scheme free from the location map is highly desirable and helpful [18][19].

According to Eq. (7), if the condition  $|v_j \cdot s_i^T| < |\alpha s_i \cdot s_i^T|$  is satisfied at encoding stage, then, at decoding stage, we have  $|\hat{v}_j \cdot s_i^T| < 2|\alpha s_i \cdot s_i^T|$ , and thus the embedded bits can be correctly extracted and the original image can be restored completely. On the other hand, the vector  $v_j$  is unusable for data embedding if  $|v_j \cdot s_i^T| \geq |\alpha s_i \cdot s_i^T|$ . We handle this case as follows.

- If  $v_j \cdot s_i^T \geq |\alpha s_i \cdot s_i^T|$ , we let  $b_e = 1$  and embed one pseudo bit “1” to the vector  $v_j$ . Thus

$$\langle \hat{v}_j, s_i \rangle = (v_j + ab_e s_i) \cdot s_i^T = v_j \cdot s_i^T + ab_e s_i \cdot s_i^T \geq 2|\alpha s_i \cdot s_i^T| \quad (10)$$

- If  $v_j \cdot s_i^T \leq -|\alpha s_i \cdot s_i^T|$ , we let  $b_e = -1$  and embed one pseudo bit “-1” to the vector  $v_j$ . Thus

$$\langle \hat{v}_j, s_i \rangle = (v_j + ab_e s_i) \cdot s_i^T = v_j \cdot s_i^T + ab_e s_i \cdot s_i^T \leq -2|\alpha s_i \cdot s_i^T| \quad (11)$$

Therefore, it is simple to determine whether a vector has been embedded with a bit at the decoding stage. If  $\langle \hat{v}_j, s_i \rangle < 2|\alpha s_i \cdot s_i^T|$ , there must be one bit has been embedded in the vector. We restore the original vector with the bit and save it properly. On the other hand, if  $\langle \hat{v}_j, s_i \rangle \geq 2|\alpha s_i \cdot s_i^T|$ , there must be one pseudo bit “1” embedded in the vector  $v_j$ . Alternatively, if  $\langle \hat{v}_j, s_i \rangle \leq -2|\alpha s_i \cdot s_i^T|$ , there must be one pseudo bit “-1” embedded in the vector  $v_j$ . We only restore the original vector with the pseudo bit and then discard it.

We provide an example to illustrate this method. Considering eight pixels as shown in Table III, every pixel-pair constitutes one vector  $v_j$ . Suppose the payload bit stream is  $w_o = (1, 0)$ , which is changed to  $w_e = (1, -1)$  according to Eq. (3); the spreading sequence  $s_1 = (1, -1)$  is utilized for data embedding on each vector and the gain factor is predefined as  $\alpha = 1$ . For the first vector  $v_1$ , we have

$v_1 \cdot s_1^T = (98, 96) \cdot (1, -1)^T = 2$  and  $|\alpha s_1 \cdot s_1^T| = |(1, -1) \cdot (1, -1)^T| = 2$ , thus,  $v_1 \cdot s_1^T = |\alpha s_1 \cdot s_1^T|$ ; so, the first pair is unsuitable for data embedding. We let  $b_1 = 1$ , and then the spreading sequence  $(1, -1)$  is added to the first vector. The marked vector is changed to  $(99, 95)$ . At the decoding stage, the product of  $\langle \hat{v}_1, s_1 \rangle = (99, 95) \cdot (1, -1)^T = 4$  and the value of  $2|\alpha s_1 \cdot s_1^T| = 2|(1, -1) \cdot (1, -1)^T| = 4$ , thus,  $\langle \hat{v}_1, s_1 \rangle = 2 \cdot |\alpha s_1 \cdot s_1^T|$ . According to Eq. (10), one pseudo bit “1” must have been embedded in the first vector. We just remove the spreading sequence from the first embedded vector and then discard the pseudo bit “1”. For the second and third vectors, since  $|v_j \cdot s_i^T| < |\alpha s_i \cdot s_i^T|$  ( $j = 2, 3$ ), these two vectors are suitable for data embedding, and the corresponding spreading sequences  $(ab_e s_i)$  ( $i = 2, 3$ ) are added onto the vectors according to Eq. (4). At the decoding side, the embedded bits are achieved by the function of sign of  $\text{sign}(\langle \hat{v}_j, s_i \rangle)$  ( $j = 2, 3$ ), and the original vectors are restored through removing the corresponding spreading sequences. As for the fourth vector, one pseudo bit of -1 is embedded to the vector because of  $v_4 \cdot s_1^T < -|\alpha s_1 \cdot s_1^T|$ , the original vector is restored by removing a negative spreading sequence  $(-s_1)$  from the embedded vector and then throw the pseudo bit “-1” away. The detailed data embedding, extraction and pixel recovery process are listed in Table III.

TABLE III  
EXAMPLE OF PIXEL VALUE CHANGES OF CDM-BASED DATA EMBEDDING AND DECODING

Original vector	(98 96)	(100 100)	(101 100)	(102 105)
$\langle v_j, s_i \rangle$	2	0	1	-3
Embedded pseudo bit	1	<del>0</del>	<del>1</del>	-1
Embedded secret bit	<del>1</del>	1	-1	<del>-1</del>
Embedded sequence	(1,-1)	(1,-1)	(-1,1)	(-1,1)
marked vector	(99 95)	(101 99)	(100 101)	(101 106)
$\langle \hat{v}_j, s_i \rangle$	4	2	-1	-5
Extracted bit( $b_e$ )	1	1	-1	-1
Extracted Sequence	(1,-1)	(1,-1)	(-1,1)	(-1,1)
Recoverd vector	(98 96)	(100 100)	(101 100)	(102 105)
Secret bit( $w_e$ )	discard	1	0	discard

By embedding some pseudo bits, the location map is not needed at the embedding stage. Comparing with most classical works which use location-map to record the embedding positions as overhead, the method of embedding pseudo bits is simple and efficient.

Actually, the CDM-based RDH scheme can be viewed as a special histogram modification algorithm

$$\begin{cases} \hat{v}_j = v_j - \alpha s_i & v_j \cdot \alpha s_i^T \leq -T \\ \hat{v}_j = v_j + ab_e \cdot s_i & -T < v_j \cdot \alpha s_i^T < T \\ \hat{v}_j = v_j + \alpha s_i & v_j \cdot \alpha s_i^T \geq T \end{cases} \quad (12)$$

where  $T$  is the absolute value of  $\alpha \langle s_i, s_i \rangle$ . All the vectors satisfying  $-T < v_j \cdot \alpha s_i^T < T$  can be utilized for data embedding, which provides larger embedding capacity than traditional histogram modification-based works. Its superiority lies in that it involves more histogram bins for data embedding.

##### B. Prevention of Overflow and Underflow

In a RDH system, the overflow/underflow may occur after data embedding, i.e., the values of some pixels in a marked

image may exceed the upper or/and the lower bound, e.g., 255 or 0 for an eight-bit grayscale image. Since the conventional overflow/underflow map needs to be embedded as overhead, this doing would occupy much embedding space. Here, we use the “flag array” to record the problem vectors. The equation  $0 \leq V_j + \alpha[b_1S_1 + b_2S_2 + \dots + b_kS_k] \leq 255$  is utilized to determine the occurrence of overflow/underflow. One can see that the maximum modification on pixel value is  $\alpha k$  after data embedding in a CDM-based scheme (note that  $\alpha$  is the gain factor and  $k$  is the embedding levels). In order to prevent overflow/underflow, we shrink the histogram distribution from both ends. The range of pixel values of an eight-bit level grayscale image is changed towards the center from  $[0, 255]$  to  $[\alpha k, 255 - \alpha k]$ , and the pixel whose value lies in the overlapping areas are recorded (i.e., pixel values in  $[\alpha k, 2\alpha k - 1]$  and  $[256 - 2\alpha k, 255 - \alpha k]$ ) in a “flag array” to identify it is an original or a modified one. The modified pixels are recoded as “1” and the original pixels are recoded as “0”. Thus, the “flag array” is a binary array and every element corresponding to an overlapping-pixel, its size is the same as the number of overlapping-pixels.

As far as image *Lena* is concerned, its pixels' grayscale values distribute from 39 to 247 (as shown in Fig.2 (a)). Particularly, at the right-hand side of the histogram, there are 6 pixels with the grayscale values occurring in the range of 236 to 255; and these 6 pixels assume the following 6 different non-zero values, they are 236, 238, 241, 244, 245 and 247, respectively. Thus, if the shrunken threshold is 10, the pixels with grayscale value occurring in the range of  $[0, 9]$  would be changed to  $[10, 19]$  by means of adding 10 to each of these pixel values; and the pixels with their grayscale values occurring in the range of  $[246, 255]$  would be reduced to  $[236, 245]$  by subtracting 10 from each pixel value. As the minimal pixel grayscale value is 39, no pixels would be overlapped at the left-hand side of the histogram; while, at the right-hand side, the pixel with the value of 247 will be changed to 237 (as shown in Table IV). Therefore, we mark the 6 pixels with a binary sequence (i.e., the “flag array”) according to their order of appearance in the image (such as from top-left to bottom-right, we have the “flag array” as ‘010000’). In the data extracting process, if the pixel's value occurs in the range of  $[236, 245]$  and its corresponding bit in the “flag array” is 0, we know that it is an original one, otherwise, it is a modified one and we just change it back by adding 10 into the value. Thus, the modified pixel with the value of 237 will be recovered to 247. The original image is then recovered completely.

TABLE IV

THE CHANGES OF PIXELS DISTRIBUTION IN THE HISTOGRAM SHRUNKEN PROCESS OF IMAGE LENA

Pixel values	236	237	238	241	244	245	247
Original distribution	1	0	1	1	1	1	1
Distribution after shrinking	1	1	1	1	1	1	0
Recovered distribution	1	0	1	1	1	1	1

Note: As many pixel values between 236-255 (such as 239,240,etc) do not exist in image *Lena*, we do not show them in this table.

Generally, only one “flag array” is needed for the recovery of the original image no matter how many times data embedding would be executed, which reduces the overhead size

significantly. Moreover, the “flags array” is usually very short and no “flag array” is needed for most images. Table V is the number of flag bits at different shrunken thresholds; it shows that nearly all images do not need “flag array” for one-time data embedding. In practice, even if the shrunken threshold is 10, the size of a “flag array” is still very small. The embedding of a “flag array” has little impact on embedding capacity.

TABLE V

SIZE OF “FLAG ARRAY” AT DIFFERENT SHRUNKEN THRESHOLD

Threshold \ Image	1	2	3	4	5	6	7	8	9	10
Lena	0	0	0	0	1	3	3	4	5	6
Airplane	0	0	0	0	0	0	0	0	7	33
Barbara	0	0	0	0	0	0	0	0	0	17
Boat	0	0	0	0	0	0	0	0	0	0
Baboon	36	36	44	46	54	61	72	91	139	176
Peppers	2	2	6	9	13	25	44	83	196	392

Note: the threshold is the value that the pixel has been changed towards the center from the both ends of the histogram.

## V. PROPOSED SCHEME

### A. Formation of Vector $v_j$

According to Eq. (7), the vector  $v_j$  is suitable for data embedding in the case of  $|v_j \cdot s_i^T| < |\alpha s_i \cdot s_i^T|$ . As the elements “1” and “-1” are equally distributed in spreading sequence  $s_i$ , the product of  $\langle v_j, s_i \rangle$  equals the sum of the differences of all element-pairs in vector  $v_j$ . Hence, the product of  $\langle v_j, s_i \rangle$  would be minimized if the elements of vector  $v_j$  are similar. The embedding capacity of the proposed scheme depends on the elements similarity in every vector  $v_j$ . Therefore, it is highly desirable that the vector elements are similar and closely correlated.

Fortunately, according to the redundancy existing in natural images, the values of pixels in a small local area are quite similar. Then, a vector  $v_j$  with similar elements can be obtained if it is formed by neighboring pixels; and more vectors suitable for data embedding can be obtained if the cover image has plenty of smooth areas.

Furthermore, inspired by reference paper, we propose a new approach that using rhombus-error-prediction method to enhance the similarity of the vector elements [14][15]. Fig. 1 reveals the main idea of the rhombus-error-prediction method. Denote the central pixel value as  $u_{x,y}$ , where  $x$  and  $y$  are subscripts indices of row and column. The four neighboring pixels ( $u_{x,y-1}, u_{x+1,y}, u_{x,y+1}, u_{x-1,y}$ ) are utilized to predict the value of  $u_{x,y}$  as

$$u'_{x,y} = \left[ \frac{(u_{x,y-1} + u_{x+1,y} + u_{x,y+1} + u_{x-1,y})}{4} \right] \quad (13)$$

The prediction-error  $d_{x,y}$  is computed as  $d_{x,y} = u_{x,y} - u'_{x,y}$ . The data can be embedded by modifying of  $d_{x,y}$  to  $D_{x,y}$ , and then change the original pixel value  $u_{x,y}$  to  $U_{x,y} = D_{x,y} + u'_{x,y}$ . Since the prediction-errors are usually very small and distribute around zero with a high peak, the vector  $v_j$  with more similar elements can be obtained, and the embedding capacity is then increased.

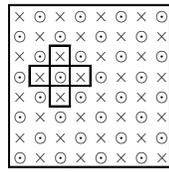


Fig. 1. Diagram of rhombus-error predictor. The “Dot” pixel value is predicted by using the four “Cross” neighboring pixel values.

Fig.1 indicates that all image pixels are divided into two sets (the “Cross” set and the “Dot” set) in the light of the rhombus-error-prediction method. The four neighbors of a pixel must belong to the set different from the set occupied by

the central pixel. In the first embedding stage, the “Cross” set containing pixels with even  $(x+y)$  are predicted by the “Dot” set containing pixels with odd  $(x+y)$ . Pixels in the “Dot” set are not modified and the predicted value  $u'_{x,y}$  is stable. Then, the decoder can get the same  $U_{x,y}$  and  $u'_{x,y}$  as the encoder, by which the embedded bit and the original pixel can be exactly recovered. Moreover, since the two sets are independent of each other, the “Dot” set can be predicted with the “cross” set for data embedding in the second stage. The order of two data embedding stages can be exchanged.

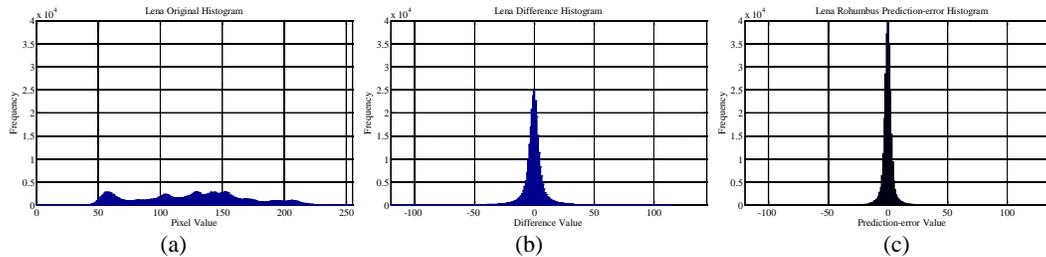


Fig. 2. (a) Histogram of image pixels; (b) Histogram of the differences of adjacent pixel-pairs; (c) Histogram of the prediction-errors of image Lena.

Fig. 2 shows the histogram of the pixels, the histogram of the differences of adjacent pixel-pairs, and the histogram of the rhombus-prediction-errors of image *Lena*. It is obvious that the histogram distribution of rhombus-prediction-errors is steeper than the others. The reason is that the rhombus-error-prediction method can exploit image redundancy better, so that the prediction-errors distribute more closely around zero; and thus more elements can be involved for data embedding while introducing less image distortion. The scheme works on the rhombus-prediction-error matrix would achieve better performance practically.

### B. Data Embedding and Extraction

This section describes the data embedding and extraction processes of the proposed scheme step by step (as shown in Fig.3).

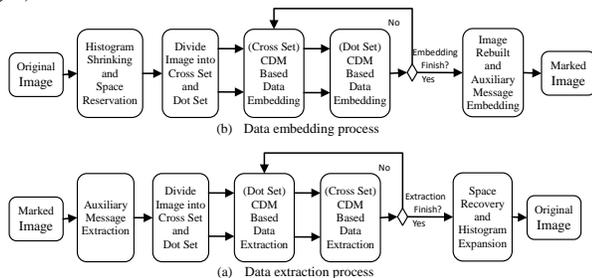


Fig. 3. Block diagrams of CDM-based (a) data embedding; (b) data extraction

#### Step 1: Image modification

- 1) Reserve specific area as “LSB replaceable” block and keep it non-modified. The position of “LSB replaceable” block is fixed or specified by a secret key shared with decoder. Generally, the marginal region of the cover image is taken as the “LSB replaceable” block.
- 2) According to the payload and image content, predetermine the multi-embedding levels, gain factor  $\alpha$  and spreading sequence. In general, the capacity of one time data embedding is about 0.2-0.4 bpp for natural image of size  $512 \times 512$ .

- 3) Except for the reserved “LSB replaceable” block, shrink the histogram distribution of the cover image according to the embedding levels and gain factor  $\alpha$ , using “flag array” to identify the overlapped pixels.

#### Step 2: Data embedding

- 1) According to the rhombus-error-prediction method, divide the cover image into “Cross” set and “Dot” set. Use the “Dot” set pixels to predict the “Cross” set pixel values and get the prediction-error matrix.
- 2) Choose elements of the matrix (formed by prediction-errors) to form the vector  $V_j$  from top-left to bottom-right.

Considering the  $j$ -th vector  $V_j$ , if  $|V_j \cdot S_i^T| < |\alpha S_i \cdot S_i^T|$  is satisfied, embed one bit according to Eq. (4), otherwise, embed a pseudo bit to guarantee the reversibility of the scheme.

- 3) The output after the “Cross” set embedding is the marked “Cross” set and the unchanged “Dot” set. Then, the data embedding in the “Dot” set is accomplished with prediction-errors generated by the modified pixels from the “Cross” set. The data embedding process can be overlappingly executed until all of the data have been embedded. Then we denote  $N^*$  as the index of the last vector.

#### Step 3: Auxiliary information embedding

- 1) Record the LSB of the “LSB-replaceable” block to obtain a binary sequence  $S_R$ . Embed  $S_R$  consecutively (i.e. the vectors with index of  $N^* + 1$  to  $N_{end}$ ) with the same method as described in Step 2. Here  $N_{end}$  is the last marked vector.
- 2) Finally, replace the LSB of the “LSB-replaceable” block with auxiliary data including the spreading sequence (16 bits), gain factor  $\alpha$  (8 bits), embedding levels (8 bits), embedding index  $N^*$  ( $\lceil \log_2 N^* \rceil$ ) and  $N_{end}$  ( $\lceil \log_2 N_{end} \rceil$ ), “flag array,” and the size of the payload. Thus, the marked image is achieved

It should be mentioned that, in order to reduce the image distortion, the gain factor is usually taken as the smallest

positive integer “1”. The data are overlappingly embedded and most elements of different spreading sequences are mutually cancelled in the process of multilevel data embedding. It enables the proposed scheme to achieve higher embedding capacity while restraining distortion to a low level.

The data extraction and image restoration is the inverse of the embedding process. The corresponding steps are summarized as follows:

Step 1: Auxiliary information extraction

- 1) Read LSB of pixels in “LSB-replaceable” block to obtain the spreading sequence, gain factor, multi-embedding levels, payload size, embedding index and the “flag array”.

Step 2: Data extraction

- 1) In the reverse order of the data embedding process, divide the marked image into “cross” set and “dot” set. Use “dot” (or “cross”) set pixel values to predict the “cross” (or “dot”) set pixel values, and achieve the corresponding prediction-error matrix.
- 2) Scan the object matrix and build the vector  $\hat{v}_j$  the same as in the embedding process, calculate the cross correlation between  $\hat{v}_j$  and the spreading sequence  $s_i$  to obtain the embedded bit according to Eq. (7).
- 3) Substrate the embedded spreading sequence from vector  $\hat{v}_j$ , and keep the useful bits safely. If it is a pseudo bit, just discard it and move the process ahead.
- 4) According to the multi-embedding levels, extract the embedded data and substrate the spreading sequence repeatedly.

Step 3: Original image recovery

- 1) Replace LSB of the “LSB-replaceable” block by the extracted sequence bits from  $N^i + 1$  to  $N_{end}$ .
- 2) Use the “flag array” to identify the overlapped pixels and reconstruct the original image. Finally, the embedded data and the original image are restored completely.

## VI. EXPERIMENTAL RESULTS

Four  $512 \times 512$  grayscale images *Lena*, *Airplane*, *Baboon* and *Barbara* from USC\_SIPi database were firstly employed in our experiments. All of these images have been widely used in RDH research community. The four selected images have different features, the image *Airplane* has more low-frequency (smooth areas) components, *Baboon* contains more high-frequency components (noisy areas), and *Lena* and *Barbara* are rich in moderate-frequency components. Besides, the image *Barbara* has regions containing many strips, which is unfavorable for error prediction and data embedding. Thus the performance of the proposed scheme can be evaluated comprehensively with these four images.

We designed a prediction-error based RDH scheme to evaluate the performance of our proposed method. The to-be-embedded bits were denoted by different spreading sequences and embedded into the rhombus-prediction-error matrix. To facilitate comparison, all experiments were performed by embedding and decoding a random binary sequence generated by the Matlab function *randint()*. The capacity-distortion behavior has been utilized to evaluate the performance of the proposed scheme. The net embedding

capacity was calculated in bit per pixel (bpp) and the distortion was measured by peak signal to noise ratio (PSNR).

### A. Results of Different Gain Factor $\alpha$

In the proposed scheme, the gain factor  $\alpha$  is employed to control the embedding intensity. More pixels would be involved for embedding if  $\alpha$  is big and thus the embedding capacity is enlarged. According to Eq. (7), if the length of spreading sequence is 2 and the gain factor  $\alpha = 1$ , all of the pixel-pairs with differences less than 2 would be employed for data embedding. Similarly, when the gain factor  $\alpha$  is 2, all of the pixel-pairs with differences less than 4 would be involved, and so on.

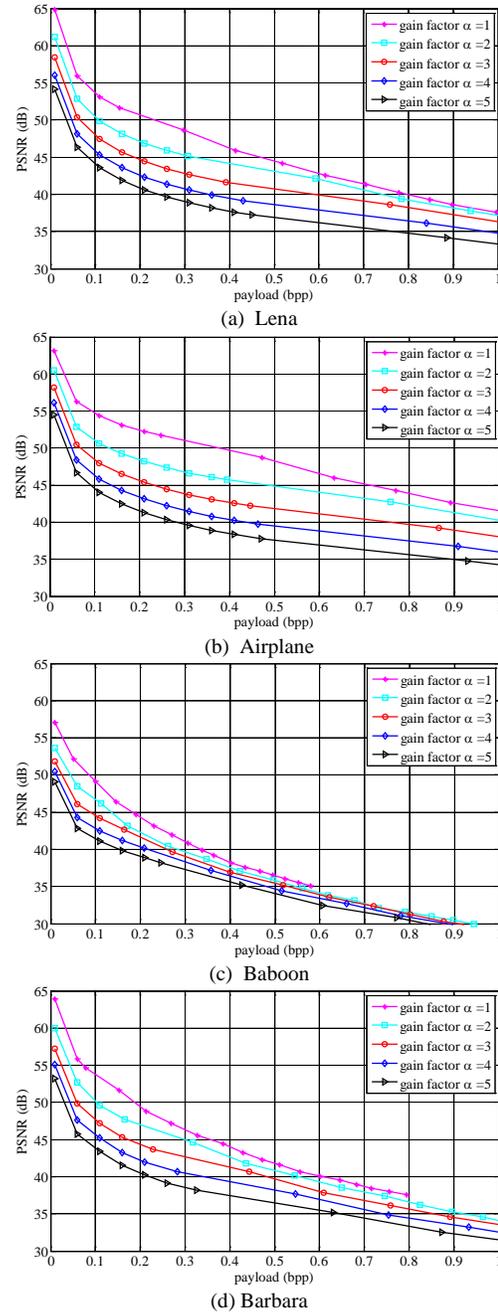


Fig.4. Comparisons of PSNR versus payload with different gain factor  $\alpha$ .

Fig. 4 demonstrates that the performance drops clearly as the value of  $\alpha$  increases. As to the image *Lena*, its PSNR value is 48.63dB at 0.27bpp and 44.82dB at 0.5bpp when the gain factor  $\alpha=1$ , whereas, it dropped to 46.11dB at 0.27bpp and 42.78dB at 0.5bpp when the gain factor  $\alpha=2$ . Since the MSE of the marked image is the square of the gain factor  $\alpha$ , the PSNR decreases rapidly with the increases of  $\alpha$ . The larger the gain factor is, the noisier the image would be. It can be calculated that the PSNR value of a marked image is 48.12dB after one-time data embedding when  $\alpha=1$ , whereas the PSNR becomes 36.08dB after one-time data embedding when  $\alpha=4$  (MSE=16). On the other hand, a bigger  $\alpha$  can enhance the robustness of a marked image. The data can be exactly extracted even after the marked image suffers from some unintentional attacks during the transmission.

Moreover, Fig. 4 also indicates that the PSNR value drops slowly at the moderate-to-high embedding capacity. As to the image *Lena*, its PSNR value is 54.16dB at 0.1bpp, 44.82dB at 0.5bpp, and still retains at 38.12dB at 1.0bpp. The proposed scheme keeps image in good quality even at high embedding capacity. Since two pixels would be changed after one bit has been embedded into the image in the proposed scheme, the PSNR of the marked image drops rapidly in the process of first time data embedding; whereas, as the payload increases, the data are overlappingly embedded into the image, most elements of different spreading sequences are mutually cancelled in the process of multilevel data embedding, the image quality thus degrades slowly.

### B. Results of Different Spreading Sequence Length

The length of the spreading sequence also impacts the performance of the proposed scheme. The maximum achievable capacity of one-time data embedding is 0.5bpp when the sequence length is 2; but the maximum capacity of single embedding is only 0.25bpp when the sequence length is 4. The short spreading sequence modifies less pixels of the image (or elements of the prediction-error matrix) than the long one for data embedding (if the spreading sequence is 2, 2 pixels/elements would be modified for one bit embedding, whereas 4 pixels/elements would be modified if the spreading sequence is 4), which would cause less distortion at the same embedding capacity. In addition, according to the local consistency of a natural image, the short spreading sequence can exploit more vectors  $v_j$  suitable for data embedding than a long sequence. To make good use of image redundancy, we form the vector  $v_j$  in the following three ways to evaluate its impact on data hiding performance. As shown in Fig. 5, the adjacent of 2, 4, and 16 pixels/elements are involved to form vector  $v_j$ .

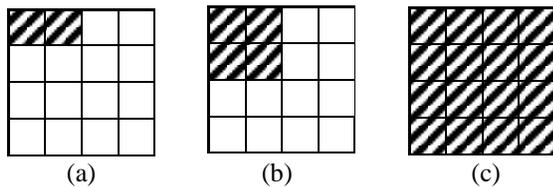


Fig. 5. Formation of vector with adjacent pixels/elements (a) 2, (b) 4, (c) 16

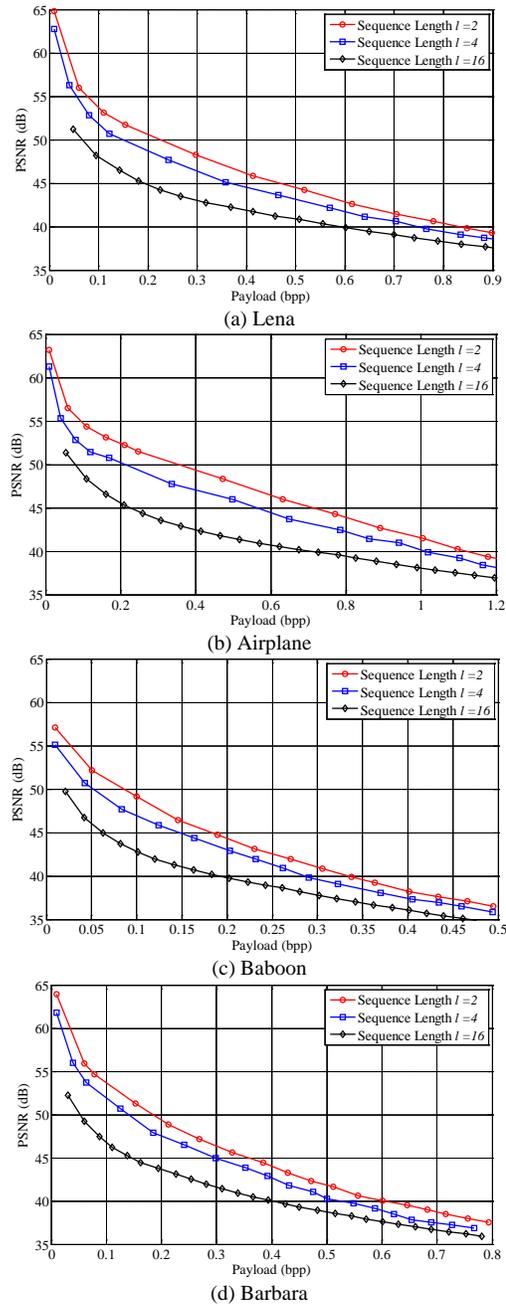


Fig. 6. Comparisons of embedding capacity versus image distortion with different length of spreading sequence.

Fig. 6 contains the results of embedding with different lengths of spreading sequences. The superiority of embedding with a short spreading sequence is obvious. The PSNR differences are 2.1, 2.4, 1.8 and 2.7dB, respectively, at 0.2 bpp between 2 and 4 elements spreading sequence for the images *Lena*, *Airplane*, *Baboon* and *Barbara*.

The embedding performance drops remarkably when the spreading sequence length is 16 (constitute by  $4 \times 4$  pixels/elements). On one hand, 16 pixels/elements would be modified for one bit embedding. On the other hand, the number of  $4 \times 4$  group pixels/elements suitable for data embedding is less, thus the PSNR-BPP curve is far below the curves at 2 or 4 elements spreading sequence. Furthermore, with payload increasing, multilevel embedding can be employed to expand embedding capacity, and most elements of different spreading

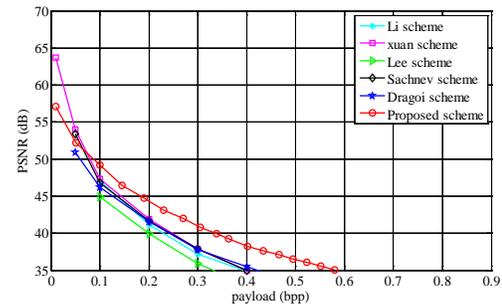
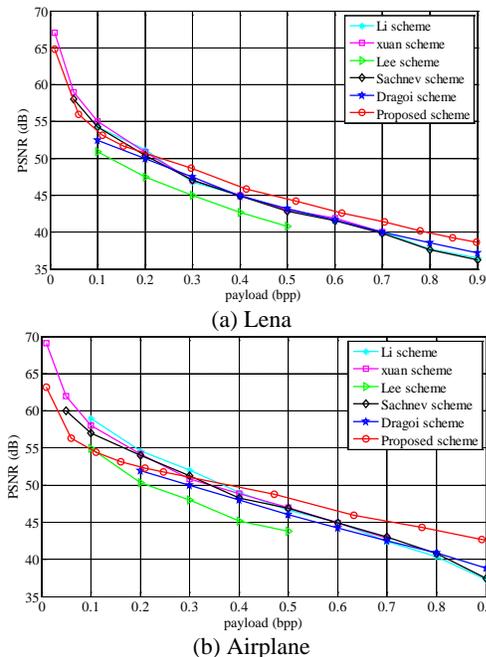
sequences would be mutually cancelled. The difference of PSNR curves resulted from different lengths of spreading sequences is reduced gradually.

### C. Comparison with Other State-of-the-art Schemes

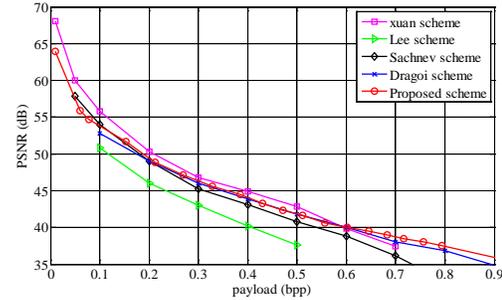
#### 1) Comparisons on six popular used test images

Here, six images from USC\_SIPi are employed to evaluate the performance of our proposed scheme by comparing with some state-of-the-art RDH works. Fig. 7 contains the performance comparisons between the proposed scheme and some state-of-the-art RDH schemes which are proposed by Li *et al.* [17], Sachnev *et al.* [15], Xuan *et al.* [28], Lee *et al.* [25] and Dragoi and Coltue [22]. These five schemes utilize different data embedding strategies, i.e., difference expansion in integer-transform domain, prediction-error expansion and sorting technique, histogram shifting and optimal embedding method, prediction-error expansion and adaptive embedding, local prediction and difference expansion method, respectively. Each of these skills has achieved excellent performance in RDH. Here, to facilitate comparison, we vary the embedding capacity from 0 to 0.9 bpp with a step of 0.1bpp.

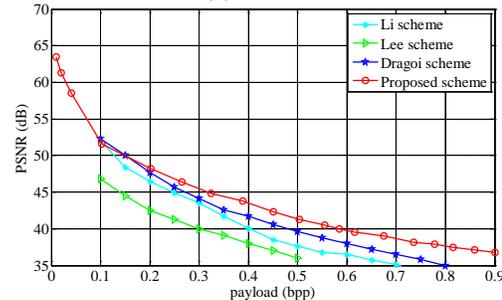
According to Fig. 7, the performance of our proposed scheme is less than others at low payload. However, it catches up and surpasses other schemes as the payload increases. The gain in PSNR exceeds other state-of-the-art schemes when the embedding bit rate equals to or greater than 0.24, 0.37, 0.08 and 0.58bpp on image *Lena*, *Airplane*, *Baboon* and *Barbara*, respectively. Moreover, for images *Boat* and *Peppers*, it also outperforms other advanced schemes when the embedding bit rate larger than 0.16 and 0.12 bpp respectively. It is noted that these two test images are not as popular as the other four test images used in RDH. Only a very few schemes have reported their performance on them. Experimental results have shown that the performance of our proposed scheme outperforms other state-of-the-art schemes at the moderate-to-large embedding capacity significantly.



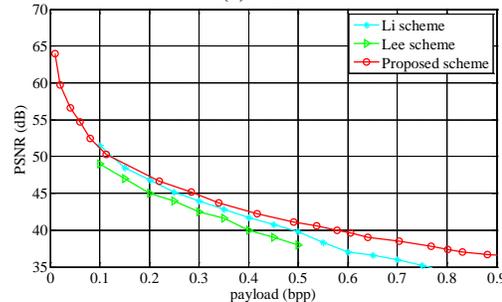
(c) Baboon



(d) Barbara



(e) Boat



(f) Peppers

Fig. 7. Comparison of PSNR versus payload between our proposed method and other advanced schemes- Li *et al.* [17], Xuan *et al.* [28], Lee *et al.* [25], Sachnev *et al.* [15] and dragoi *et al.* [22].

The reason that its performance is not the best at the small data embedding rates is as follows. As the payload is small, at least two pixels need to be modified for the proposed method to embed one bit, the performance of the proposed scheme less than those referenced works. But, as the payload increases, multilevel data embedding is employed to enlarge the embedding capacity. Most elements of different spreading sequences are mutually cancelled in the process of multilevel data embedding, and the corresponding distortion-capacity curve drops more slowly than those classical schemes. Therefore, the proposed scheme achieves higher PSNR value than others at the moderate-to-high embedding capacity.

As to image *Barbara*, the improvement is not obvious. It is because the image has regions containing many strips, which

are unfavorable for the exploitation of redundancy. The proposed scheme achieves better performance compared to other advanced schemes after more levels of data embedding have been executed.

### 2) Comparisons on JPEG2000 test image

To further evaluate the performance of our proposed scheme, we also employed one of the JPEG2000 test image *Woman* in our experiments. The size of the image has been reduced from  $1920 \times 1536$  to  $960 \times 768$  to facilitate the comparison. It is observed that most of the above mentioned six widely used images have the two ends of their histogram empty (except for images *Baboon* and *Peppers* having one-side of their histogram non-zero), whereas, the both ends of histogram of the image *Woman* have peaks. The test of our proposed scheme on such an image can further verify its performance. The performance comparison in terms of PSNR versus payload with other prior-arts (Kim *et al.*'s difference expansion method and Xuan *et al.*'s histogram modification method) is shown in Fig. 9. As Xuan *et al.*'s method makes the use of four thresholds to search for the optimal performance in RDH, it achieves better performance as the payload is no more than 0.55 bpp. However, as the payload increases, in our proposed scheme, most elements of different spreading sequences are mutually cancelled in the process of multilevel data embedding. Hence, our proposed scheme achieves higher PSNR values than Xuan *et al.*'s method when the embedding capacity is larger than 0.55 bpp.



Fig. 8. Image Woman used in paper

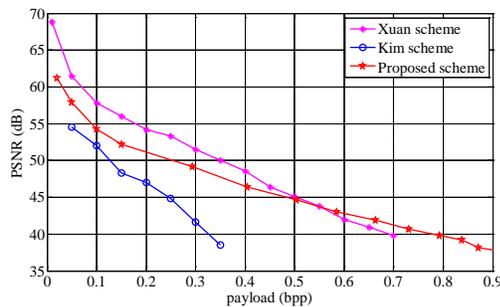


Fig. 9. Comparison of PSNR versus payload between our proposed method and other advanced schemes - Xuan *et al.*[28], Kim *et al.*[9] over image *Woman*.

## VII. DISCUSSIONS

### A. Structural Simulation (SSIM) of the Marked Images.

While the PSNR measures the difference of an image with its original version, SSIM is a perception-based model that considers image perceptual degradation in structural information. As the SSIM can capture the actual image quality closely to visual effects in human eyes, in this paper, we firstly introduce this index to evaluate the quality of marked images. Experimental results have demonstrated that the SSIM and PSNR have similar behaviors in terms of the performance evaluation for a RDH scheme. As shown in Fig. 10, the SSIM

drops clearly with the increases of the embedding capacity, especially for image with plenty of high-frequency component. The SSIM of image *Baboon* is 0.88 when the bpp of RDH embedding is 0.6; while the value of image *Airplane* is 0.97 at the same embedding capacity. As less vectors generated from noisy areas are suitable for data embedding than those from smooth areas, multilevel data embedding is needed for a noisy image to hide as much data as an image rich in smooth areas. The SSIM of a noisy image drops faster than those images having massive smooth areas as data have been embedded. The trends of curves obtained by using SSIM are similar to those obtained by using PSNR. Nevertheless, The SSIM provides us a new perception-based way to test the performance of RDH scheme.

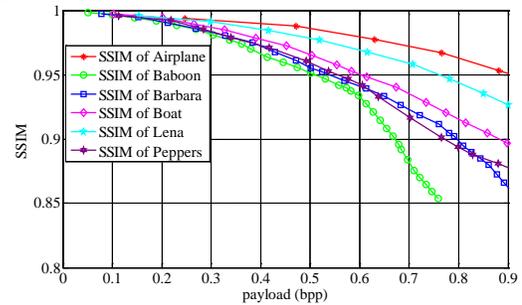


Fig. 10. Comparison of SSIM versus payload over different images.

### B. A Parameter Memorization-free Method

In a RDH system, a parameter memorization-free method is highly desired in order to enlarge the embedding capacity. Motivated by the work of Fujiyoshi *et al.* [19], we arranged the blocks include the usable vectors in ascending order, so that the vectors, whose elements having small variance, are firstly selected for data embedding, and thus the distortion of marked image is reduced especially at low embedding capacity.

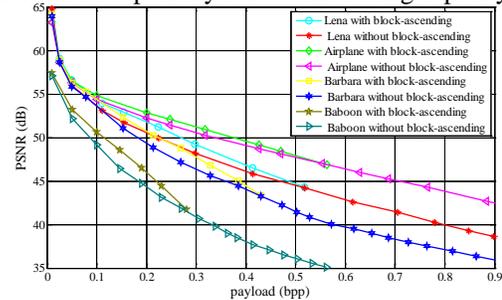


Fig. 11. Performance comparison of our proposed scheme with or without parameter memorization-free method.

As shown in Fig.11, the embedding performance of the proposed scheme has been improved at the low payload by using the block-ascending method. However, with the increases of payload, the advantage reduces gradually. When the embedding capacity largely increases, the improvement by with the block-ascending method is marginal. This is because the method of block-ascending record the magnitude of local variance, and a vector will be firstly used for data embedding if it has a small variance. Thus, the scheme performs well at low embedding capacity. As the payload increase, the smooth vectors are insufficient and the noisy ones would to be utilized for data embedding, vectors with large prediction-errors will be involved, the image quality thus drops rapidly. When the payload is largely increases, all of the usable vectors would be

involved for data embedding; the improvement derives from the using of block-ascending method for embedding position selection is negligible. Then, the performance of data embedding with or without block-ascending method is similar. In short, as far as our proposed scheme is concerned, embedding with a parameter memorization-free lossless data hiding method can improve the performance at low embedding capacity apparently.

### C. Security and Computing Complexity

As the data are embedded through spreading sequences at encoding side, only one who has the same spreading sequence can extract them exactly at decoding side. The security of the scheme not only relies on the method itself but also on the spreading sequence. The spreading sequence can be utilized as the key of the data hiding system, which provides more security than traditional RDH schemes. A short spreading sequence can be employed to enlarge the embedding capacity, whereas a long spreading sequence can be applied to increase security. The choice of the sequence length can be balanced between the embedding capacity and the system security. Furthermore, since the modification of image pixels usually is very small, the quality of the marked image degrades slightly, and it is hard to detect whether the image has been embedded or not. The key spreading sequence cannot be deduced from image analysis. It will be difficult for one to illegally extract the embedded data and recover the original image.

Additionally, we briefly compare the computational complexity of the proposed method with the methods based on difference expansion and histogram modification. Since all of these three methods can work on the prediction-error matrix derived from the cover image, we only compare the embedding and decoding complexity. The command operations, such as multiplication, addition and shift, are employed to compare the complexity of the three methods. In the embedding process, the difference expansion method includes one multiplication operation and one addition operation for one bit embedding, the histogram shifting method needs one addition operation, and the proposed method needs two addition operations to embed one bit when the length of spreading sequence is 2. However, the difference expansion method generally utilizes a location map to locate the embedded pixels. The compression of the location map is generally high computational cost in practice. The histogram shifting method needs to create an image histogram to choose the prominent bins and then shift all non-selected bins one position for data embedding, which is also quite complex. As to the proposed scheme, only a simple pre-comparison is needed to determine the embedding of one useful or pseudo bit. The decoding process is the reverse of data embedding and possesses the same complexity. Hence, the proposed scheme is less complex compared with other RDH schemes.

## VIII. CONCLUSION

A novel CDM-based RDH algorithm has been proposed in this paper. The covert data are denoted by different spreading sequences and added to the prediction-error matrix of cover image. At the decoding stage, the original image can be recovered completely and the embedded data can be extracted

exactly. In the proposed scheme, different orthogonal spreading sequences are overlappingly embedded into the cover image to enlarge embedding capacity and most elements of different spreading sequences are mutually cancelled in the process of multilevel data embedding, by which high embedding capacity is achieved while restraining distortion to a low level. This is the first attempt to design a RDH scheme based on CDM algorithm, and the proposed scheme achieves higher performance compared with most state-of-the-art works at the moderate-to-high embedding capacity. However, this algorithm is still fragile as any malicious invasion may cause errors in the decoding of the data hiding image. This issue should be investigated in the future.

## REFERENCES

- [1] J. M. Barton, "Method and apparatus for embedding authentication information within digital data," *US Patent 5 646 997*, 1997.
- [2] J. Fridrich, M. Goljan, R. Du, "Invertible authentication," *Photonics West 2001-Electronic Imaging. International Society for Optics and Photonics*, pp. 197-208, 2001.
- [3] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *Image Processing, IEEE Transactions on*, vol. 14, no. 2, pp. 253-266, 2005.
- [4] T. Kalker, FMJ Willems, "Capacity bounds and constructions for reversible data-hiding," *Electronic Imaging 2003. International Society for Optics and Photonics*, pp.604-611, 2003.
- [5] W. Zhang, X. Hu, X. Li, and N. Yu, "Recursive Histogram Modification: Establishing Equivalency Between Reversible Data Hiding and Loss less Data Compression," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2775-2785, Jul, 2013.
- [6] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 13, no. 8, pp. 890-896, 2003.
- [7] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions On Image Processing: A Publication Of The IEEE Signal Processing Society*, vol. 13, no. 8, pp. 1147-1156, 2004.
- [8] L. Kamstra, and H. J. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *Image Processing, IEEE Transactions on*, vol. 14, no. 12, pp. 2082-2090, 2005.
- [9] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H.-G. Choo, "A novel difference expansion transform for reversible data embedding," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 3, pp. 456-465, 2008.
- [10] D. M. Thodi, and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *Image Processing, IEEE Transactions on*, vol. 16, no. 3, pp. 721-730, 2007.
- [11] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *Image Processing, IEEE Transactions on*, vol. 9, no. 8, pp. 1309-1324, 2000.
- [12] M. Fallahpour, "Reversible image data hiding based on gradient adjusted prediction," *IEICE Electronics Express*, vol. 5, no. 20, pp. 870-876, 2008.
- [13] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 3, pp. 873-882, 2011.
- [14] M. Fujiyoshi, S. Sato, H. L. Jin, and H. Kiya, "A location-map free reversible data hiding method using block-based single parameter," *Proc. IEEE ICIP*, pp.257-260, 2007.
- [15] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 7, pp. 989-999, 2009.
- [16] M. Chen, Z. Chen, X. Zeng, and Z. Xiong, "Model order selection in reversible image watermarking," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 3, pp. 592-604, 2010.
- [17] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *Image Processing, IEEE Transactions on*, vol. 20, no. 12, pp. 3524-3533, 2011.

- [18] S. Han, M. Fujiyoshi, and H. Kiya, "A reversible image authentication method without memorization of hiding parameters," *IEICE Trans. Fundamentals*, vol. E92-A, pp. 2572-2579, Oct. 2009.
- [19] M. Fujiyoshi, T. Tsuneyoshi, and H. Kiya, "A parameter memorization-free lossless data hiding method with flexible payload size," *IEICE Electronics Express*, vol. 7, no. 23, pp. 1702-1708, Dec. 2010.
- [20] X. Zhang, "Reversible data hiding with optimal value transfer," *Multimedia, IEEE Transactions on*, vol. 15, no. 2, pp. 316-325, 2013.
- [21] D. Coltuc, "Low Distortion Transform for Reversible Watermarking," *Ieee Transactions on Image Processing*, vol. 21, no. 1, pp. 412-417, Jan, 2012.
- [22] I. C. Dragoi and D. Coltuc, "Local-Prediction-Based Difference Expansion Reversible Watermarking," *Ieee Transactions on Image Processing*, vol. 23, no. 4, pp. 1779-1790, 2014.
- [23] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 3, pp. 354-362, 2006.
- [24] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin, "Robust lossless image data hiding," *IEEE Trans. on Circuits & Systems for Video Technology*, vol. 18, no. 4, pp. 497-509, 2008.
- [25] S. K. Lee, Y. H. Suh and Y. S. Ho, "Reversible Image Authentication Based on Watermarking," *Multimedia and Expo, 2006 IEEE International Conference on*. pp. 1321-1324, 2006.
- [26] W. Hong, "Adaptive reversible data hiding method based on error energy control and histogram shifting," *Optics Communications*, vol. 285, no. 2, pp. 101-108, Jan 15, 2012.
- [27] C. H. Yang and M. H. Tsai, "Improving histogram-based reversible data hiding by interleaving predictions," *IET Image Processing*, vol. 4, no. 4, pp. 223-234, 2010.
- [28] G. Xuan, X. Tong, J. Teng, X. Zhang, and Y. Q. Shi, "Optimal histogram-pair and prediction-error based image reversible data hiding," *Digital Forensics and Watermarking*, pp. 368-383, 2013.
- [29] X. Li, B. Li, B. Yang and T. Zeng, "General Framework to Histogram-Shifting-Based Reversible Data Hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2181-2191, 2013.
- [30] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, "Reversible Watermarking Based on Invariant Image Classification and Dynamic Histogram Shifting," *IEEE Trans. on Information Forensics and Security*, vol. 8, no. 1, pp. 111-120, Jan, 2013.
- [31] G. Xuan, Q. Yao, C. Yang, J. Gao, P. Chai, Y. Q. Shi, and Z. Ni, "Lossless data hiding using histogram shifting method based on integer wavelets," *Digital Watermarking*, pp. 323-332, 2006.
- [32] B. Yang, M. Schmucker and W. Funk, "Integer DCT-based reversible watermarking for images using companding technique," *Electronic Imaging 2004. International Society for Optics and Photonics*, pp. 405-415, 2004.
- [33] D. Zou, Y. Q. Shi, Z. Ni, and W. Su, "A semi-fragile lossless digital watermarking scheme based on integer wavelet transform," *Circuits and Systems for Video Technology, IEEE Transactions on*. vol. 16, no. 10, pp. 1294-1300, 2006.
- [34] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *Information Forensics and Security, IEEE Transactions on*. vol. 2, no. 3, pp. 321-330, 2007.
- [35] L. An, X. Gao, X. Li, D. Tao, C. Deng, and J. Li, "Robust Reversible Watermarking via Clustering and Enhanced Pixel-Wise Masking," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3598-3611, 2012.
- [36] A. J. Viterbi, "CDMA: principles of spread spectrum communication," Addison Wesley Longman Publishing Co., Inc., pp. 112-124, 1995.



**Bin Ma** received the M.S. and ph. D degree in 2005 and 2008, respectively, from the Shandong University, Jinan, China. He was with the New Jersey Institute of Technology at Newark as a visiting scholar from November 2013 to January 2015. Currently, he is an associate professor with the School of Information Science, Qilu University of Technology.

His research interests include reversible data hiding, multimedia security and image processing.



**Yun Q. Shi** (M'90-SM'93-F'05) obtained the B.S. and M.S. degrees from Shanghai Jiao Tong University, Shanghai, China, and the M.S. and Ph.D. degrees from University of Pittsburgh, PA, USA. He has been Professor of Electrical and Computer Engineering at New Jersey Institute of Technology, Newark, since 1987. His research interests include digital

multimedia data hiding, steganalysis, forensics and information assurance, visual signal processing and communications, theory of multidimensional systems, and signal processing. He is an author/coauthor of more than 300 research papers, a book, and four book chapters. He holds 30 U.S. patents. He was the Technical Program Chair of IEEE ICME07 and MMSP05, a Co-Technical Chair of IWDW from 2006 to 2015.