Reversible Watermarking Algorithm Using Sorting and Prediction

Vasiliy Sachnev, Hyoung Joong Kim, *Member, IEEE*, Jeho Nam *Senior Member, IEEE*, Sundaram Suresh, and Yun Qing Shi, *Fellow, IEEE*

***Abstract*— This paper presents a reversible or lossless watermarking algorithm for images without using a location map in most cases. This algorithm employs prediction errors to embed data into an image. A sorting technique is used to record the prediction errors based on magnitude of its local variance. Using sorted prediction errors and, if needed, though rarely, a reduced size location map allows us to embed more data into the image with less distortion. The performance of the proposed reversible watermarking scheme is evaluated using different images and compared with four methods: those of Kamstra and Heijmans, Thodi and Rodriguez, and Lee *et al*. The results clearly indicate that the proposed scheme can embed more data with less distortion.**

***Index Terms*— Lossless data hiding, prediction error expansion, reversible watermarking.**

I. Introduction

REVERSIBLE or lossless data hiding techniques hide data in a host signal (for example, an image) and allow extraction of the original host signal and also the embedded message. There are two important requirements for reversible data hiding techniques: the embedding capacity should be large; and distortion should be low. These two requirements conflict with each other. In general, a higher embedding capacity results in a higher degree of distortion. An improved technique embeds the same capacity with lower distortion or vice versa.

Tian's difference expansion technique [15] previously had the highest embedding capacity and the lowest distortion in image quality. His method divides the image into pairs of pixels and uses each legitimate pair for hiding one bit of information. Therefore, his embedding capacity is at best 0.5 b/pixel. However, an uncompressed location map also needs 0.5 b/pixel. Then, it is impossible to hide data reversibly into an image. Thus, reducing the size of the location map is one of the key goals in this field. Without a location map, the decoder cannot decode exactly because it does not know how and which cells were modified. The location map consists of flags, which are either 0 or 1. The purpose of the flags differs between the particular methods. The difference expansion technique is the seminal reversible data hiding scheme and was the basis for new ideas, such as Alattar's technique for triplets and quads [1], [2], and Kamstra and Heijmans' sorting method [7].

Alattar has expanded one cell from a pair to a triplet [1] or quad [2] to hide two or three bits per cell, respectively, or none at all in illegitimate cells. A cell is the unit of pixels in which the data is to be embedded. Alattar's location map covers all triplets or quads instead of pairs. Thus, the uncompressed location map size is decreased from one-half of the image resolution (for Tian's method) to one-third or one-fourth of the resolution (for triplets or quads), respectively. It is obvious that Alattar's methods have advantages over Tian's, since the forms can hide data even if the location map is not compressed whereas it is not possible in Tian's method. If the location map is compressed, Alattar's method significantly outperforms Tian's method. Later Alattar generalized his idea for a cell with n pixels in [3].

Kamstra and Heijmans [7] greatly improved Tian's difference expansion technique. They proposed a method to reduce location map size by sorting pairs according to correlation measures to facilitate compression. The main idea of this technique is its efficient utilization of the correlation between neighboring pixels in the image. Difference values of neighboring pixels in a cell (where a cell is a pair in their method) highly correlate with average values of neighboring pairs. Sorting by exploiting the correlation between neighboring pixels can strongly enhance embedding capacity of the difference expansion method. Their method achieved the best performance among the aforementioned algorithms.

The location map is the cornerstone for all the described methods [1], [2], [7], [15]. Location maps usually are huge in size and should be compressed. Even if location maps are compressed, they occupy a part of the payload. Thus, the size of the compressed location map determines the efficiency of a method. Reversible data hiding techniques with smaller, or in some cases, no location maps (like [12] and [14]), can be very desirable.

Lee *et al.* [9] used an advanced watermarking technique based on integer-to-integer wavelet transform. Their method divides the image into nonoverlapping blocks and applies a data hiding technique based on definitions of expandability

Manuscript received December 31, 2007; revised May 10, 2008. First version published April 7, 2009; current version published July 22, 2009. This work was in part supported by IT R&D program (Development of anonymity-based u-knowledge security technology, 2008-F-036-02). This paper was recommended by Associate Editor M. Barni.

V. Sachnev, H. J. Kim, and S. Suresh are with the Center of Information Security and Technology, Graduate School of Information Management and Security, Korea University, Seoul 136-701, Korea (e-mail: bassvasys@korea.as.kr; khj@korea.as.kr; sdram@korea.ac.kr).

J. Nam is with the Radio and Broadcasting Research Laboratory, Electronics and Telecommunications Research Institute, Daejon 305-700, Korea (e-mail: namjeho@etri.re.kr).

Y. Q. Shi is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: Shi@ADM.NJIT.EDU).

Digital Object Identifier 10.1109/TCSVT.2009.2020257

(which means a possibility of bit shifting operation) and changeability over high-frequency wavelet coefficients of each block. The bit-shifting approach is used for embedding data, and least significant bit (LSB) replacement approach for hiding the location map. Expanded and nonexpanded blocks are marked by different flags, 1 or 0, respectively, in the location map. It covers all blocks and its size is $(X/N) \times (Y/M)$, where N and M are the block sizes, and X and Y are image sizes. In order to achieve reversibility, the proposed method requires location map, expansion matrix P, and original LSB of coefficients from the blocks containing location map. The proposed technique outperforms existing methods such as [1], [5], [17], and [21] by exploiting high-frequency subbands and efficient data hiding technique.

Thodi and Rodriguez [13], [14] proposed a histogram shift method for embedding data in prediction errors. The location map (or flag bits) used in these schemes covers all ambiguous cells. Ambiguous cells include all cells that cannot be decoded without a location map. Ambiguity may arise due to overlapping values between expanded or shifted ones and those neither expanded nor shifted. Some values are neither expanded nor shifted according to the possibility of underflow/overflow errors. Thodi and Rodriguez [14] also proposed advanced methods based on difference expansion technique. Their methods exploit the histogram shift technique for embedding data, which is a combination of the expansion method [15] and a variation of the original histogram shifting method [12]. We describe the histogram shift method in more detail in Section II-B. Another contribution of one of their methods [14] is using the JPEG-LS prediction errors as an input signal for the embedding scheme. The prediction errors are better for embedding more data compared to the difference values [1], [2], [7], [15]. The correlation of JPEG-LS prediction errors is higher than that of the difference values between neighbors. Thus, in the prediction error histogram, the population of bins around zero is high, whereas the population of other bins drop quickly (see Fig. 2). Using the histogram shift method results in lower distortion, especially for this kind of histogram. The maximum possible capacity for their method is 1 b/pixel in the ideal situation, so this method is generally considered to be better than other methods based on pairs [7], [15] (with maximum possible capacity 0.5 b/pixel), triplets [1] (with 0.667 b/pixel), and quads [2] (with 0.75 b/pixel).

Kim *et al.* [8] improved the difference expansion methods [7], [15] by using a novel simplified location map approach. Goljan, Fridrich, and Du [6] used a lossless compression algorithm for reversible data hiding. Later Fridrich, Goljan, and Du [5] improved proposed method. Van der Veen *et al.* [16] proposed a companding reversible data hiding technique for audio. Leest *et al.* [10] extended companding technique for images. Celik *et al.* [4] proposed an LSB substitution technique using an efficient entropy coder. Ni *et al.* [12] used a shifting technique on the histogram of the pixels for data hiding. Ni *et al.* [11] also proposed a robust lossless data hiding technique. Yang *et al.* [20] generalized a reversible data hiding method for coefficients of integer discrete cosine transform (DCT). Another paper [21] exploited histogram expansion technique for embedding data to high-frequency wavelet coefficients. Xuan *et al.* [17]–[19] proposed several reversible data hiding techniques based on integer to integer wavelet transform. Zou *et al.* [22] proposed a semi-fragile reversible data hiding technique based on integer wavelet transform.

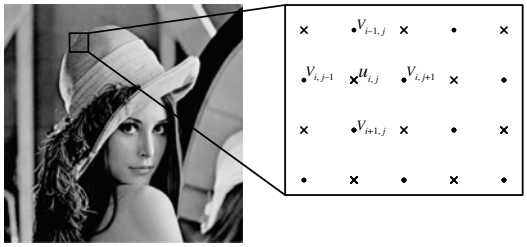

Fig. 1. Prediction pattern. The pixel value u of the Cross set can be predicted by using the four neighboring pixel values of the Dot set and expanded to hide one bit of data.

Even though the JPEG-LS prediction scheme is remarkable in terms of histogram shape, it was optimized for lossless compression. Hence, we considerably extend it in order to apply it for a data hiding scheme. An alternate prediction scheme specialized for sorting (see Section II-C) is used in this paper for achieving increased embedding capacity. Unlike the JPEG-LS predictors, the rhombus prediction scheme enables sorting and therefore achieves better performance. The combined use of the rhombus prediction scheme, sorting, histogram shift method, and, as result, small size of location map produces relatively superior results compared to existing schemes [7], [9], [14].

The organization of this paper is as follows: Section II discusses the important issues regarding the proposed method including the rhombus prediction scheme, histogram shift scheme, sorting, overflow and underflow problems, and double embedding schemes, and presents threshold values. Section III describes the encoder and decoder of the proposed method. Section IV presents the experimental results. Section V concludes the paper.

II. Proposed Algorithm

A. Rationale of Prediction Using a Rhombus Pattern

As mentioned previously, the JPEG-LS prediction scheme is not suitable for sorting of the prediction errors. Therefore, a new prediction scheme is necessary that permits sorting. For this reason we propose a rhombus pattern prediction scheme. In order to predict the pixel value of position $u_{i,j}$ in Fig. 1, four neighboring pixels (i.e., $v_{i,j-1}$, $v_{i+1,j}$, $v_{i,j+1}$, and $v_{i-1,j}$) are used. The five pixels including $u_{i,j}$ comprise a cell which is used to hide one bit of data.

All pixels of the image are divided into two sets: the "Cross" set and "Dot" set (see Fig. 1). The Cross set is used for embedding data and Dot set for computing predictors. Henceforth, this scheme will be called the Cross embedding scheme.

The encoder of the Cross embedding scheme for a single cell is as follows.

Center pixel $u_{i,j}$ of the cell can be predicted from the four neighboring pixels $v_{i,j-1}$, $v_{i+1,j}$, $v_{i,j+1}$, and $v_{i-1,j}$. The predicted value $u'_{i,j}$ is computed as follows:

$$u'_{i,j} = \left\lfloor \frac{v_{i,j-1} + v_{i+1,j} + v_{i,j+1} + v_{i-1,j}}{4} \right\rfloor. \quad (1)$$

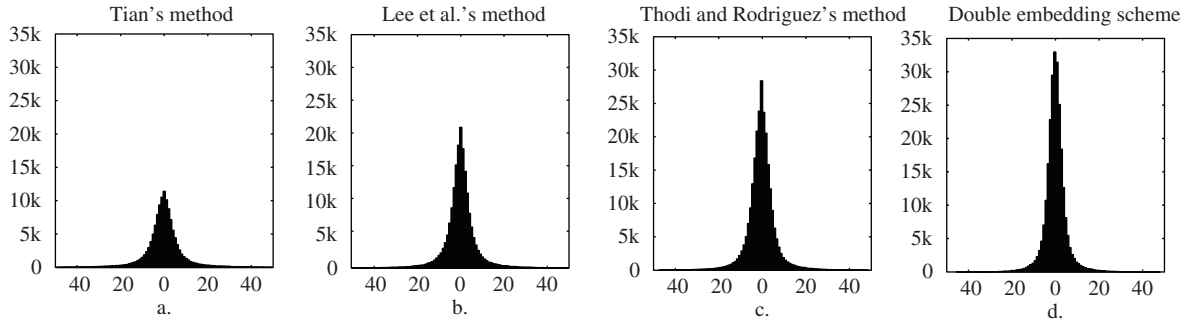

Fig. 2. (a) Histogram of differences between neighboring pixels. (b) Histogram of high-frequency wavelet coefficients. (c) Histogram of JPEG-LS prediction errors. (d) Histogram of prediction errors for Double embedding scheme and (e) for Lena image.

Based on the predicted value $u'_{i,j}$ and original value $u_{i,j}$, the prediction error $d_{i,j}$ is computed as

$$d_{i,j} = u_{i,j} - u'_{i,j}. \tag{2}$$

This prediction error can be expanded to hide information like the difference expansion algorithm proposed by Tian [15] as follows:

$$D_{i,j} = 2d_{i,j} + b \tag{3}$$

where $D_{i,j}$ is the prediction error after expansion called modified prediction error. The value b is one bit of the hidden message. Note that $D_{i,j}$ is modified according to the histogram shift method, which will be described later (see Section II-B).

After data hiding, the original pixel value $u_{i,j}$ is changed to $U_{i,j}$ as

$$U_{i,j} = D_{i,j} + u'_{i,j}. \tag{4}$$

The decoding procedure for the Cross embedding scheme for a single cell is an inverse of the encoding scheme. During data hiding, pixels from the Dot set are not modified, so the predicted values $u'_{i,j}$ are also not changed. Using the predicted value $u'_{i,j}$ and the modified pixel value $U_{i,j}$, the decoder can exactly recover the embedded bit and original pixel value.

The modified prediction error is computed as

$$D_{i,j} = U_{i,j} - u'_{i,j}. \tag{5}$$

The embedded bit value is computed as

$$b = D_{i,j} \bmod 2. \tag{6}$$

The original prediction error is computed as

$$d_{i,j} = \left\lfloor \frac{D_{i,j}}{2} \right\rfloor. \tag{7}$$

The original pixel's value is computed as

$$u_{i,j} = u'_{i,j} + d_{i,j}. \tag{8}$$

Note that the two sets (the Cross set and Dot set) are independent of each other. Independence means changes in one set do not affect the other set, and vice versa. Pixels from the Dot set are used for computing predicted values $u'_{i,j}$, whereas pixels from the Cross set $u_{i,j}$ are used for embedding data. The order of hiding data in cells is not important and can be changed. Sorting reorders cells according to the magnitudes of local variance (see Section II-C) and enables hiding data in cells with small prediction errors. Thus, sorting can significantly improve the data embedding scheme. The sorting adapted to the rhombus pattern will be presented in more detail later in Section II-C.

The performance of the embedding scheme depends on the shape of the histogram. In general, distribution of the prediction errors has a Laplacian distribution. The shape of the distribution is determined by the mean and variance. In general, the mean is zero, so the variance essentially determines the shape of the histogram. The smaller the variance values, the better the performance of the data hiding scheme. It is obvious that Tian's [15] and Lee *et al.*'s [9] methods show larger values than the proposed scheme [see Fig. 2(a) and (b)] in terms of variance value. For this reason the proposed scheme achieves better performance than these methods. However, Thodi and Rodriguez's scheme [14] produces similar variance values compared with the proposed scheme [see Fig. 2(c) and (d)]. Yet, due to the sorting utilized in the proposed method, the performance still exceeds their scheme.

B. Use of Histogram Shift Scheme

The histogram shift scheme of Thodi and Rodriguez [13] is utilized in this paper to enhance performance. The histogram shift method is an efficient reversible data hiding technique in terms of low distortion. Another advantage of this scheme is that it allows us to avoid overlapping problems caused by expansion. The combination of histogram shifting and expansion has been previously used [13]. In our scheme, two threshold values T_n and T_p are used, where T_n is the negative threshold value, and T_p is the positive threshold value. Expandable set E consists of the predicted errors in $[T_n; T_p]$ that can be expandable according to (3) without causing underflow or overflow errors in the spatial domain. The predicted errors not belonging to $[T_n; T_p]$ are going to be shifted to make room for the expansion. The set S is the set of predicted errors that can be shifted and is referred to as the shiftable set. However, the predicted errors in their associated domain will not be expanded or shifted due to the underflow or overflow errors in the spatial domain. The embedding capacity of the proposed scheme is $|E| - |L|$, where $|L|$ is the size of the location map (which will be discussed in Section II-D).

The histogram shift encoding algorithm modifies prediction errors d as follows:

$$D_{i,j} = \begin{cases} 2d_{i,j} + b, & \text{if } d_{i,j} \in [T_n; T_p] \\ d_{i,j} + T_p + 1, & \text{if } d_{i,j} > T_p \text{ and } T_p \geq 0 \\ d_{i,j} + T_n, & \text{if } d_{i,j} < T_n \text{ and } T_n < 0. \end{cases} \quad (9)$$

The decoder recovers original prediction errors $d_{i,j}$ and bits of the embedded data b from $D_{i,j}$ according to the following:

$$d_{i,j} = \begin{cases} \lfloor D_{i,j}/2 \rfloor, & \text{if } D_{i,j} \in [2T_n; 2T_p + 1] \\ D_{i,j} - T_p - 1, & \text{if } D_{i,j} > 2T_p + 1 \text{ and } T_p \geq 0 \\ D_{i,j} - T_n, & \text{if } D_{i,j} < 2T_n \text{ and } T_n < 0 \end{cases} \quad (10)$$

$$b = D_{i,j} \bmod 2, \quad D_{i,j} \in [2T_n; 2T_p + 1]. \quad (11)$$

The ratio between E and S sets can be controlled by changing thresholds T_n and T_p, which in turn controls the capacity. Distortion depends on the histogram shape and thresholds. The characteristics of distortion are different. The distortion due to the expandable set depends on the magnitude of each element. On the other hand, the magnitude of distortion caused by the shiftable set is uniformly T_p+1 over $d_i > T_p+1$ and $-T_n$ over $d_i < T_n$. Thus, distortion in the shiftable set highly depends on the thresholds. In addition, the cardinality of the shiftable set is also important. Summed distortion strongly depends on the cardinality.

C. Use of Sorting

Kamstra and Heijmans' [7] use of sorting introduced a significant performance advantage over previous methods. However, sorting is possible only when cells are independent. In other words, embedding data into one cell should not affect the other cells. However, Thodi and Rodriguez's method [14] produces dependent cells, where embedding data to one cell changes prediction errors of other cells.

Note that the Dot and Cross sets of the rhombus scheme are independent each other. In order to hide more data with less visual degradation, the order to hide data into the cells needs to be changed. Thus, the cells can be rearranged by sorting according to the correlation of neighboring pixels. Local variance $\mu_{i,j}$ for each cell can be computed from the neighboring pixels $v_{i,j-1}$, $v_{i+1,j}$, $v_{i,j+1}$, and $v_{i-1,j}$ (see Fig. 1) as follows:

$$\mu_{i,j} = \frac{1}{4} \sum_{k=1}^{4} (\Delta v_k - \Delta \bar{v}_k)^2 \quad (12)$$

where $\Delta v_1 = |v_{i,j-1} - v_{i-1,j}|$, $\Delta v_2 = |v_{i-1,j} - v_{i,j+1}|$, $\Delta v_3 = |v_{i,j+1} - v_{i+1,j}|$, $\Delta v_4 = |v_{i+1,j} - v_{i,j-1}|$, $\Delta \bar{v}_k = (\Delta v_1 + \Delta v_2 + \Delta v_3 + \Delta v_4)/4$. Local variances μ calculated by using (12) achieve the most appropriate sorting for improving performance of data hiding. The local variance $\mu_{i,j}$ has several features. First, this value remains unchanged after data hiding. Second, this value is proportional to the magnitude of prediction error $d_{i,j}$ of the cell. For example, a small variance indicates a small magnitude of prediction errors, and vice versa.

Assume that d_{sort} is the sorted row of all $d_{i,j}$. The histogram shift method embeds data with the thresholds T_n and T_p. Cells are sorted in ascending order of the local variance values. Cells with smaller variance values are better for data hiding. Thus, the embedding process starts from the cell with the smallest variance value in the sorted row, and moves on to the next cells until the last bit of data is embedded. The histogram shift method is applied to the cells with small variance values which are comprised of sets E and S where $P = |E|$. Of course, the payload P also depends on threshold values T_n and T_p. Needless to say, for extracting the embedded data and recovering the original image, a few bits of information about the position of the last changed cell (or embedding capacity) and threshold values are necessary.

D. Overflow and Underflow Problem

Overflow/underflow problem of the histogram shift method over prediction errors cannot be avoided. Cells in E and S sets causing overflow/underflow errors should be excluded. The condition

$$0 \leq u'_{i,j} + D_{i,j} \leq 255 \quad (13)$$

is used for locating such problematic cells, where $D_{i,j}$ is the modified error after data hiding using the histogram shift method: i.e., shifting or expanding (see Section II-B).

The problematic cells are the cells which stay unchanged since they cause overflow/underflow errors after data hiding. Some modified cells may overlap with the problematic cells after data hiding. Assume that S_p is the set of all problematic cells, and S_{op} is the set of overlapping cells with S_p after data hiding. The overlapping problem can be easily resolved using a location map. The location map tells the cells in S_p from the cells in S_{op} with different flags.

According to the rules of the histogram shift method, each cell is checked using a technique which will be referred to as two-pass testing, where each pass is an attempt to embed a test bit. Rather than using a random bit value, the test bit should be the hardest and extreme for embedding, i.e., "1" for positive d values and "0" for negatives (note that the output signal has maximum distortion under these cases). Each pass of the testing process should satisfy condition (13). Three possible cases may arise as a result of the encoder testing (ET), as detailed below.

- ET(a) If the current cell is modifiable twice, the corrective location map is not necessary.
- ET(b) If the current cell is modifiable once owing to overflow/underflow errors during the second pass, this cell is marked as "0" in the corrective location map. The set of cells that results in this case are denoted as S_{op}. A cells in this category overlaps with the cells in set S_p after modification.
- ET(c) If the current cell is not modifiable even once, the cell cannot be used in the embedding phase. This cell is marked as "1" in the location map. The set of such cells is denoted as S_p.

Consider an example for testing possible overflow/underflow errors (see Figs. 3 and 4). Let the

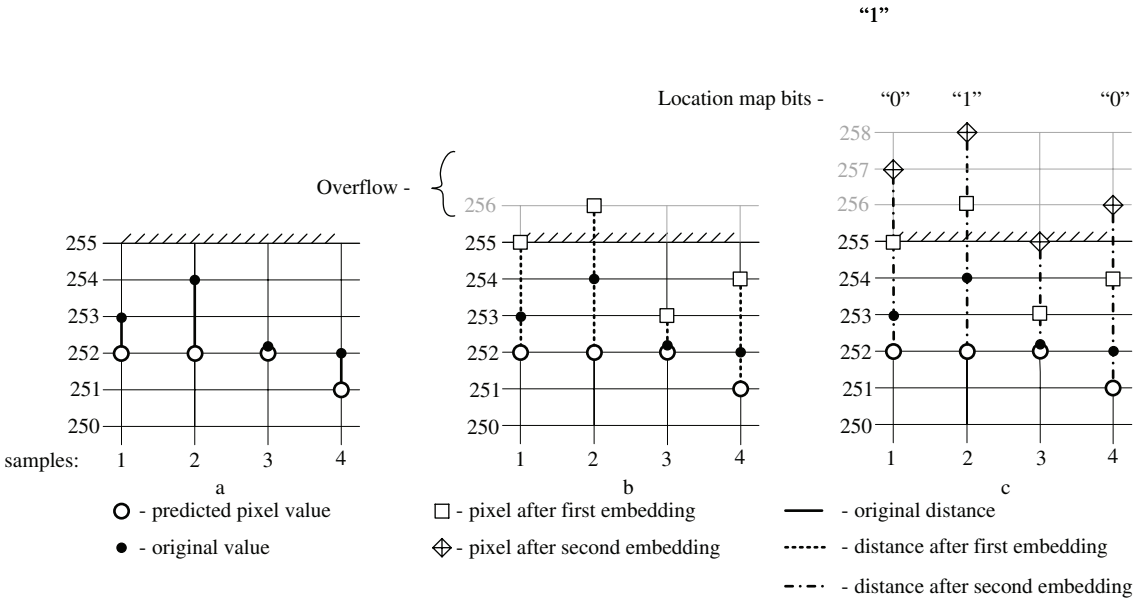

Fig. 3. Overflow testing: (a) original data, (b) first time embedding, and (c) second time embedding.

threshold value be 1 (i.e., $T = 1$). There are four samples: sample 1 with $u_{n+1,m} = 253$, $u'_{n+1,m} = 252$, and $d_{n+1,m} = 1$; sample 2 with $u_{n-1,m} = 254$, $u'_{n-1,m} = 252$, and $d_{n-1,m} = 2$; sample 3 with $u_{n,m+1} = 252$, $u'_{n,m+1} = 252$, and $d_{n,m+1} = 0$; and sample 4 with $u_{n,m-1} = 252$, $u'_{n,m-1} = 251$, and $d_{n,m-1} = 1$, where u' and d are computed by using (1) and (2) over four neighboring pixels v. Note that the test bit for all samples is 1. Modified values over 255 are considered as overflow errors.

Note that the sample 3 is modifiable twice. Its value even after modifying twice is on the border [see Fig. 3(c)]. In other words, this sample is not problematic, has no overlap with problematic cells, and corresponds to the Case ET(a). Thus, the location map is not needed for this sample. On the other hand, the samples 1 and 4 cause overflow errors in the second pass, and hence these samples are modifiable only once. These samples are marked as "0" in the location map since they belong to the Case ET(b). Sample 2 causes overflow error in the first step because it is not modifiable at all. This sample belongs to the Case ET(c) and is marked as "1" in the location map. Thus, the corrective location map is marked as "010" with three bits in this example.

The decoder testing (DT) procedure is as follows.

DT(a) If the cell to be decoded is modifiable once, the corrective location map bit is not needed, because the cell was modifiable twice at the encoding phase and thus was not marked. Refer to the Case ET(a).

DT(b) If the cell to be decoded is not modifiable at all, the location map must be consulted, because the cell was either modifiable once or not modifiable at all at the encoding phase (i.e., $S_{op} \bigcup S_p$). If the corrective location map bit is "0," the current pixel belongs to the Case ET(b). The decoder has to extract original value of the cell. However, if the location map bit is "1," the cell is disregarded as it was not modifiable in the encoder. Refer to the Case ET(c).

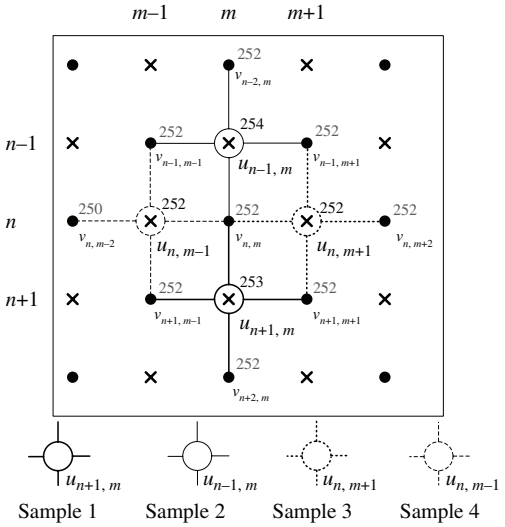

Fig. 4. Example of samples for overflow testing (associated with Fig. 3).

As an exception, certain expandable cells (as defined in Section II-B) belonging to the S_{op} set should be excluded from the expandable set E. Embedding certain bit values (0 for positive d and 1 for negative) to these cells can change them from Case ET(b) to Case ET(a). This will not present any problem on the encoder side. However, during the decoding process, the cell would not be classified as requiring location map consultation. As a result of this misclassification, the location map would be skewed. Consider a simple example of this situation. Assume that sample 4 in the previous example (see Fig. 4) will be used for embedding bit "0." The pixel from sample 4 would be expanded to 253. This cell belongs to the Case ET(b) and will be marked in the location map. During the decoder testing procedure this cell would be classified as Case DT(a) [instead of the correct Case DT(b)] and the location map would not be consulted. The next time the location map

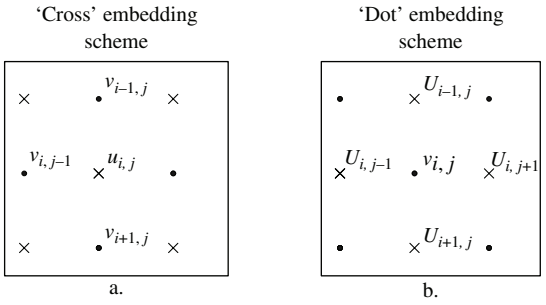

Fig. 5. Relationship among pixels in the Cross set or the Dot set.

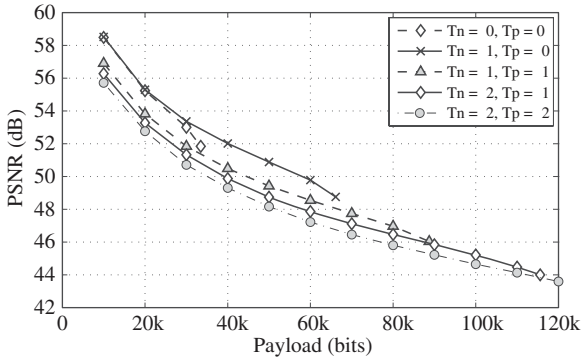

Fig. 6. Effect of threshold values on PSNR given a certain payload for the Lena image.

is consulted, the current position will be off by one and hence the value would be incorrect. This would result in a cascade of incorrect decoding from that point onward.

In order to resolve such ambiguity at the decoder, during the encoding process all expandable cells classified as the Case ET(b) should have bit "1" embedded for all positive d values and bit "0" for all negative d. All expanded cells from S_{op} are denoted as set S_{op_E}. All shiftable cells with Case ET(b) are shifted according to the histogram shift method rules (see Section II-B).

It is worth mentioning that the location map is usually not necessary in our method and even when it is necessary the size of this map is negligible.

E. Double Embedding Scheme

The double embedding scheme involves consecutive usage of the Cross embedding scheme and the Dot embedding scheme, and results in nearly double the embedding capacity. The maximum capacity can be increased to 1 b/pixel (from 0.5 b/pixel when using only Cross embedding). The Cross and Dot embedding schemes are similar in nature.

The Cross set embedding scheme computes predicted values using the Dot set and embeds data using the Cross set [see Fig. 5(a)]. Note that the pixel $u_{i,j}$ belonging to the Cross set has four neighboring pixels $v_{i-1,j}$, $v_{i+1,j}$, $v_{i,j+1}$, and $v_{i,j-1}$, which belong to the Dot set. The predicted value $u'_{i,j}$ using four neighboring pixels is computed by (1) and the prediction error $d_{i,j}$ is computed to hide data. After hiding data using the Cross embedding scheme, pixels from the Cross set $u_{i,j}$ are modified to $U_{i,j}$. Similarly, the Dot embedding scheme uses the central pixel $v_{i,j}$ belonging to the Dot set and four modified neighboring pixels $U_{i-1,j}$, $U_{i+1,j}$, $U_{i,j+1}$, and $U_{i,j-1}$ belonging to the Cross set (see Fig. 5(b)]. Furthermore, payload P should be divided into two sets with similar sizes P_{Cross} and P_{Dot} for the Cross and Dot embedding schemes, respectively.

It is clear that the capacity of the single embedding scheme is less than that of the double embedding scheme. However, contrary to expectations, the distortion after data hiding using the double embedding scheme is less than the distortion resulting from a single embedding. The reason for such an oddity is that, when using single embedding, cells with larger errors have to be modified in order to hide required payload. In the double embedding, two sets of sorted prediction errors with smaller magnitudes are used. The required payload for each set is approximately half of the single embedding scheme, which allows us to utilize the cells with smaller magnitudes of errors.

Note that the distortion resulting from the double embedding scheme is always better than previous methods (such as [7], [9], and [14]).

F. Appropriate Threshold Values

As mentioned before, there are many possible combinations of threshold values. However, in order to achieve the best possible PSNR, it is necessary to find and use appropriate threshold values based on the required payload. Such appropriate threshold values can be found by iteratively adjusting (or finetuning) the threshold values in order to produce improved PSNR for our required payload. It is also possible to achieve appropriate thresholds by using histograms. However, the details of this method are outside the scope of this paper.

As an example, consider a case where we need to hide 60 000 bits. Fig. 6 shows that pairs of thresholds $\left[T_n; T_p\right]$ of $[-1, 0], [-1, 1], [-2, 1]$, and $[-2, 2]$ are sufficient to hide 60 000 bits. Among them the most appropriate pair is $[-1, 0]$ because it achieves 49.58 dB of image quality.

III. Encoder and Decoder

This section describes the encoding and decoding processes in detail and uses ideas all described above. Fig. 7 presents a simple block diagram representing the embedding and decoding processes. Before data embedding, all pixels should be divided into two sets: either Cross set or Dot set (see arrows with "Cross" and "Dot," respectively, in the left-hand side of the encoder in Fig. 7). Payloads for the Cross and Dot embedding schemes are P_{Cross} and P_{Dot}, respectively.

For recovering data, threshold values $T_{n\text{Cross}}$ and $T_{p\text{Cross}}$, and actual payload size $|P_{\text{Dot}}|$ (for Dot embedding scheme) or $|P_{\text{Cross}}|$ (for Cross embedding scheme) should be sent to the decoder. The LSB values of the first 34 prediction errors from d_{sort} are replaced with threshold values $T_{n\text{Cross}}$ (7 bits) and $T_{p\text{Cross}}$ (7 bits), payload size $|P_{\text{Dot}}|$ (20 bits), or $|P_{\text{Cross}}|$ (20 bits). Original 34 LSB values should be collected to a set of collected LSB values S_{LSB}, and included to the payload. These

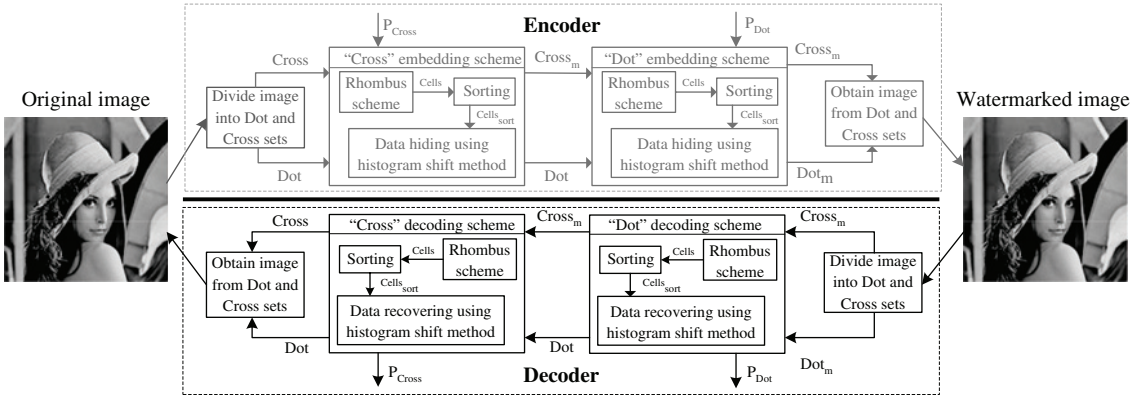

Fig. 7. Framework of the double encoding and decoding scheme.

34 prediction errors should be excluded from d_{sort}. The number of bits to be collected depends on the application requirements.

The Cross embedding scheme is designed as follows.

1) Find all cells according to the Cross embedding scheme.
2) For each cell compute:
   a) predicted value u' using (1);
   b) prediction errors d using (2);
   c) local variance μ using (12).
3) Sort cells according to the local variances μ, and produce the sorted row of prediction errors d_{sort} (see Section II-C). Skip the first 34 elements in d_{sort}. Collect the original 34 LSB values of the prediction errors to set S_{LSB} and include S_{LSB} as a part of the payload.
4) Find appropriate threshold values $T_{n\text{Cross}}$ (negative threshold) and $T_{p\text{Cross}}$ (positive threshold) according to the payload P_{Cross} using sorted row d_{sort}.
5) Classify cells into the expandable set E and shiftable set S, and create the location map L. For each cell starting from the 35th sorted cell.
   5.1) Check for the possibility of overflow/underflow [either the Case ET(a), ET(b), or ET(c)] using Section II-D. If a current cell belongs to the Case ET(b) or Case ET(c), mark it in the location map L. If the current cell belongs to Case ET(c), go to Step 5.3.
   5.2) Check for expandability using Section II-B. Place a current cell in expandable set E or shiftable set S (exclusively). If the current cell belongs to the expandable set E and to the Case ET(b), move it from the expandable set E to the special set S_{op_E} (see Section II-D).
   5.3) If the current cell is the last cell and the condition $|P_{\text{Cross}}|+|L| = |E|$ is not met, increase magnitude of threshold $|T_{n\text{Cross}}|$ or $T_{p\text{Cross}}$ and repeat Step 5 again. If the condition is met, move on to Step 6.
6) Embed the payload P_{Cross} and location map L to expandable set E using histogram shift encoding algorithm (see Section II-B). Make a set of modified prediction errors D. Modify cells from set S_{op_E} according to guidelines in Section II-D.
7) Compute the modified pixels U using (4) and obtain the modified Cross set.
8) Modify the first 34 LSB values of the prediction errors excluded in Step 3 by binary representation of the threshold values $T_{n\text{Cross}}$ and $T_{p\text{Cross}}$ and the payload size $|P_{\text{Cross}}|$.

The output of the Cross embedding scheme is unchanged pixels v from the Dot set and the modified pixels U from the Cross set (see the arrows with companion words "Dot" and "Cross$_m$," respectively, in the middle of the encoder part in Fig. 7).

The Dot embedding scheme is designed according to Section II-E. The Dot embedding scheme uses the modified pixels from the Cross set for computing predicted values and original pixels from the Dot set (see Fig. 7) for embedding data. The output result of the Dot embedding scheme is the modified Dot and Cross sets (see arrows with accompanying words "Dot$_m$" and "Cross$_m$" in the right-hand side of the encoder part in Fig. 7), which is the watermarked image. Dot$_m$ is a modified Dot set, and Cross$_m$ a modified Cross set.

Double decoding scheme is the inverse of the double encoding scheme (see the decoder part in Fig. 7). Computations for the Dot and Cross decoding schemes are similar. Thus, to simplify explanation of the double decoding scheme, we only describe the Cross decoding scheme.

The Cross decoding scheme is designed as follows.

1) See Step 1 and Step 2 of the Cross embedding scheme.
2) Sort cells according to the local variances μ, and obtain the set of sorted prediction errors d_{sort} (see Section II-C). Read the 34 LSB values from d_{sort} and recover the threshold values $T_{n\text{Cross}}$ and $T_{p\text{Cross}}$ and payload size $|P_{\text{Cross}}|$. Skip the first 34 sorted cells.
3) Classify cells into expandable set E and shiftable set S according to the appropriate threshold values $T_{n\text{Cross}}$ and $T_{p\text{Cross}}$ and the payload size $|P_{\text{Cross}}|$.
4) Find problematic cells using the decoding testing for overflow/underflow and remove them from the sets E and S (see Section II-D).
5) Using decoder of histogram shift method, recover original prediction errors d, location map L, and payload P_{Cross}. Remove the set S_{LSB} from the payload P_{Cross}.
6) Recover original prediction errors d from the problematic cells using the decoder of the histogram shift method according to the location map L.

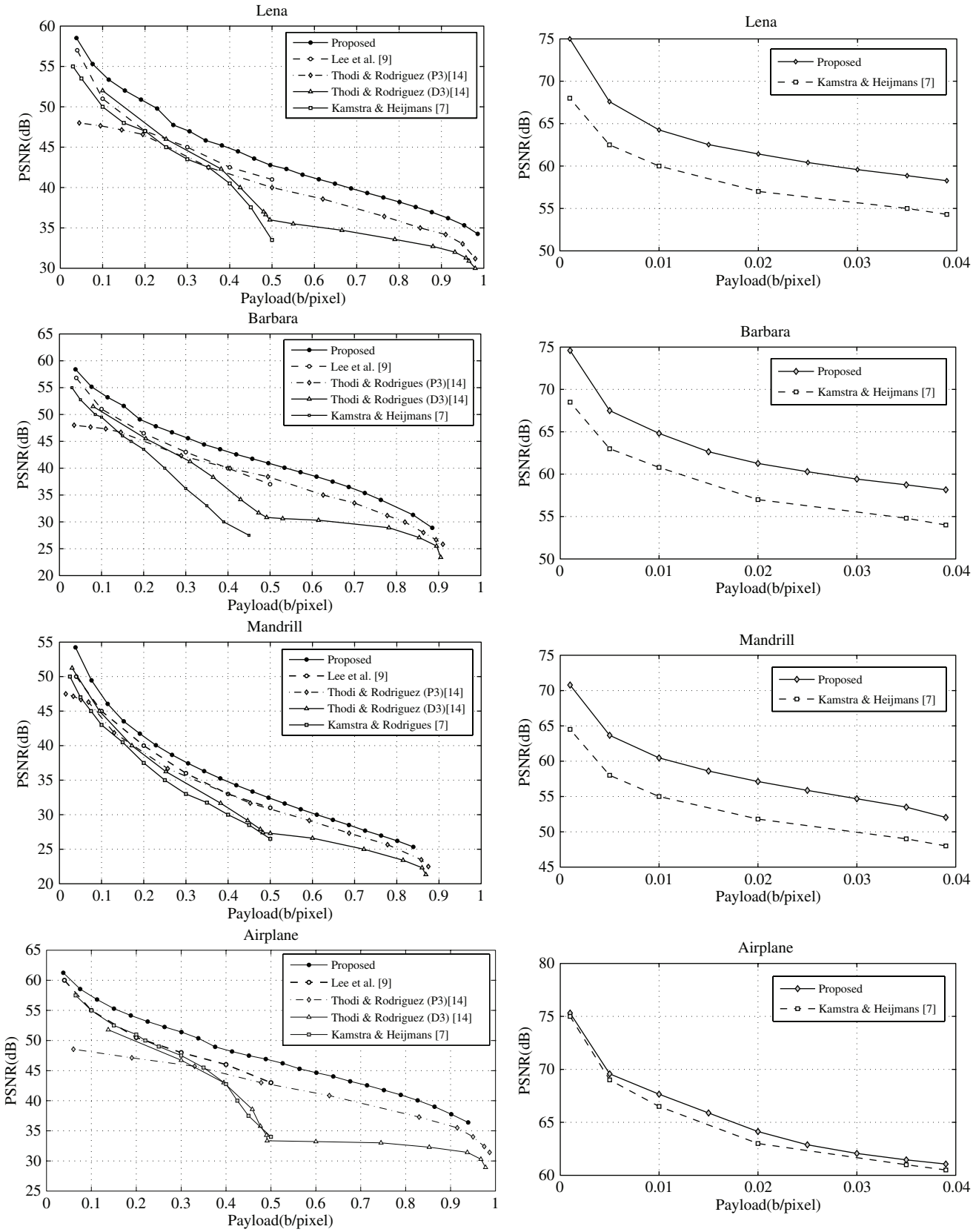

Fig. 8. Distortion vs. capacity graphs of the tested methods for each of the four images. Right column of graphs shows results for the small capacities (0–0.04 b/pixel).

TABLE I
IMPROVEMENTS OVER EXISTING METHODS FOR LENA IMAGE IN DB

Payload (b/pixel)	Improvements over Kamstra and Heijmans [7]	Improvements over Thodi and Rodriguez [14] based on		Improvements over Lee *et al.* [9]
		Difference expansion	Prediction errors	
0.040	3.97 dB	—	10.05 dB	1.51 dB
0.100	3.36 dB	2.05 dB	6.25 dB	3.02 dB
0.300	3.46 dB	2.21 dB	3.22 dB	1.96 dB
0.450	6.04 dB	3.98 dB	2.06 dB	1.85 dB
0.650	—	5.67 dB	2.47 dB	—
0.800	—	2.63 dB	2.19 dB	—

TABLE II
LOCATION MAP SIZE VS. PAYLOAD FOR TESTED IMAGES

Payload (bits/b/pixel)	Lena			Barbara			Mandrill			Airplane		
	Thresholds		Location map (bits)	Thresholds		Location map (bits)	Thresholds		Location map (bits)	Thresholds		Location map (bits)
	T_n	T_p		T_n	T_p		T_n	T_p		T_n	T_p	
100 k/0.38	−2	1	0	−2	2	0	−6	5	31	−1	1	0
150 k/0.57	−3	3	0	−6	5	2	−11	10	278	−2	2	0
200 k/0.76	−5	5	0	−9	8	115	−20	19	1758	−4	3	0
220 k/0.83	−6	6	2	−15	15	1253	−29	29	6782	−5	5	0
230 k/0.87	−8	7	6	−25	25	5602	—	—	—	−7	6	0
250 k/0.95	−12	12	9	—	—	—	—	—	—	−13	13	13

7) Compute the original pixels u using (8).
8) Replace the first 34 LSB values of the sorted prediction errors (these prediction errors were skipped with first 34 sorted cells in the step 2) with the original LSB from the set S_{LSB}.

The double decoding process is shown in Fig. 7.

IV. EXPERIMENTS

The proposed scheme is compared with the four seminal methods of Kamstra and Heijmans [7], Thodi and Rodriguez [13], [14] and Lee *et al*. [9] using typical 512 × 512 grayscale images (i.e., Lena, Mandrill, Barbara, and Airplane). Results are shown in Fig. 8. For all images, the proposed algorithm is superior to other methods.

Kamstra and Heijmans' method [7] shows good performance when the payload is small. Comparison for small payloads (see right column of Fig. 8) also shows that the proposed scheme is better. Improvements are more significant for payloads close to 0.5 b/pixel. Note that the maximum possible capacity for Kamstra and Heijmans' method is 0.5 b/pixel, while the performance in terms of PSNR is significantly decreased for payloads closed to 0.5 b/pixel. Compared to Kamstra and Heijmans' method which exploits sorting, our method is better and because it uses sorting, histogram shift, and double embedding scheme altogether, and also exploits prediction errors with lower variance.

Thodi and Rodriguez [14] proposed five versions of the reversible data hiding methods. Two methods achieve the best improvements among them. Thus, we compare our proposed method with these two methods. The first method is a combination of histogram shifting and difference expansion (which is called method D3). The second is a combination of histogram shifting and use of prediction errors (method P3). Thodi and Rodriguez's method based on use of prediction error has the same limitation of maximum possible capacity with the proposed scheme: 1 b/pixel. Our method has better performance over their method because of the use of sorting, especially for small capacities (see Table I). Their method based on difference expansion has the maximum possible capacity 0.5 b/pixel for single embedding. Thus, their method has the same limitation as Kamstra and Heijmans' method. Compared to this method, our method is better because of the exploitation of sorting and histogram shifting together.

In addition, the proposed method has significant improvements over the method of Lee *et al.* [9] (see Fig. 8). Their method exploits different bit-shifting operations for each block of which coefficients can be shifted without causing overflow/underflow and significant degradation after modification. This approach significantly increases capacity and at the same time requires additional side information. Finally, their method achieves desirable improvements, although the side information occupies part of the payload. On the other hand, our method achieves better improvements because of exploiting prediction errors with lower variance and using sorting and histogram shifting method, which provides small location map.

Table I shows the improvements of our method over the previous methods for different payloads when embedding to the Lena image. Barbara, Mandrill, and Airplane images have similar results to the Lena image for the same reasons.

Table II tabulates location map size and thresholds for different payloads. Results show that our method produces small location map in size and can also dispense with it for Lena and Airplane images when the payload is less than 0.76 and 0.87 b/pixel, respectively. Even if the location map is indispensable, it size is almost negligible compared to payload. The biggest location map (6782 bits) is used for Mandrill image when the payload is 220 000 bits.

V. Conclusion

The proposed reversible watermarking algorithm is a combination of efficient well-known existing techniques and new techniques which enables performance significantly. Using a new rhombus prediction scheme enables the efficient exploitation of sorting. A set of sorted prediction errors can be efficiently used for low distortion data hiding. The histogram shift method exploited over the sorted prediction errors produces excellent ratio between capacity and distortion. Furthermore, the histogram shift method in the proposed scheme significantly decreases location map size and may indeed eliminate the need for it sometimes. Thus, capacity can be significantly increased. The double embedding scheme allows using each pixel for data hiding in the ideal case. Thus, the maximum possible capacity reaches 1 b/pixel. Experimental results show that the proposed method has better results compared to the methods of Thodi and Rodriguez [14], Kamstra and Heijmans [7], and Lee *et al.* [9].

Acknowledgment

The authors would like to thank A. H. Khayat for proofreading.

References

[1] A. M. Alattar, "Reversible watermark using difference expansion of triplets," in *Proc. Int. Conf. Image Process.*, vol. 1. Barcelona, Spain, 2003, pp. 501–504.

[2] A. M. Alattar, "Reversible watermark using difference expansion of 611 quads," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 3. Toulouse, France, 2004, pp. 377–380.

[3] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1147–1156, Aug. 2004.

[4] M. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," in *Proc. Int. Conf. Image Process.*, Rochester, NY, 2002, pp. 157–160.

[5] J. Fridrich, M. Golijan, and R. Du, "Lossless data embedding for all image formats," in *Proc. SPIE, Security Watermarking Multimedia Contents,* San Jose, CA, 2002.

[6] M. Goljan, J. Fridrich, and R. Du, "Distortion-free data embedding for images," in *Proc. Inform. Hiding Workshop*, Pittsburgh, PA, 2001, pp. 27–41.

[7] L. H. J. Kamstra and A. M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2082–2090, Dec. 2005.

[8] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. G. Choo, "A novel difference expansion transform for reversible data embedding," *IEEE Trans. Inform. Forensic Security*, vol. 3, no. 3, pp. 456–465, Sep. 2008.

[9] S. Lee, C. D. Yoo, and T. Kalker "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Trans. Inform. Forensics Security*, vol. 2, no. 3, pp. 321–330, Sep. 2007.

[10] A. Leest, M. van der Veen, and F. Bruekers, "Reversible watermarking for images," in *Proc. SPIE, Security, Steganography, Watermarking Multimedia Contents,* San Jose, CA, Jan. 2004.

[11] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin, "Robust lossless image data hiding," in *Proc. IEEE Int. Conf. Multimedia Expo*, Taipei, Taiwan, 2004, pp. 2199–2202.

[12] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[13] D. M. Thodi and J. J. Rodriguez, "Reversible watermarking by prediction-error expansion," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation*, Lake Tahoe, CA, 2004, pp. 21–25.

[14] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.

[15] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[16] M. van der Veen, F. Bruekers, A. Leest, and S. Cavin, "High capacity reversible watermarking for audio," in *Proc. SPIE, Security, Steganography, Watermarking Multimedia Contents*, vol. 5020. San Jose, CA, 2003, pp. 1–11.

[17] G. Xuan, J. Chen, J. Zhu, Y. Q. Shi, Z. Ni, and W. Su, "Lossless data hiding based on integer wavelet transform," in *Proc. IEEE Workshop Multimedia Signal Process.*, St. Thomas, VI, 2002, pp. 312–315.

[18] G. Xuan, Y. Q. Shi, Z. C. Ni, J. Chen, C. Yang, Y. Zhen, and J. Zheng, "High capacity lossless data hiding based on integer wavelet transform," in *Proc. IEEE Int. Conf. Circuits Syst.*, Vancouver, BC, 2004, pp. 29–32.

[19] G. Xuan, C. Yang, Y. Zhen, Y. Q. Shi, and Z. Ni "Reversible data hiding based on wavelet spread spectrum," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Siena, Italy, 2004, pp. 211–213.

[20] B. Yang, M. Schmucker, W. Funk, C. Busch, and S. Sun, "Integer DCT-based reversible watermarking for images using companding technique," in *Proc. SPIE Security, Steganography, Watermarking Multimedia Contents*, vol. 5306. San Jose, CA, 2004, pp. 405–415.

[21] B. Yang, M. Schmucker, C. Busch, X. Niu, and S. Sun, "Approaching optimal value expansion for reversible watermarking," in *Proc. 7th ACM Multimedia Security Workshop*, New York, 2005, pp. 95–102.

[22] D. Zou, Y. Q. Shi, Z. Ni, and W. Su, "A semi-fragile lossless digital watermarking scheme based on integer wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 10, pp. 1294–1300, Oct. 2006.

Vasiliy Sachnev received the B.S. and M.S. degrees in electrical engineering from the Komsomolsk-na-Amure State Technical University, Russia, in 2002 and 2004, respectively. He is currently pursuing the Ph.D. degree with the Multimedia Security Laboratory, the Center of Information Security and Technology, Graduate School of Information Management and Security, Korea University, Seoul.

In 2007, he joined the Multimedia Security Laboratory. His research interests include multimedia security, digital watermarking, steganography, and image processing.

Hyoung Joong Kim (A'09) received the B.S., M.S., and Ph.D. degrees from Seoul National University, Seoul, in 1978, 1986, and 1989, respectively.

He joined the faculty of the Department of Control and Instrumentation Engineering, Kangwon National University, Korea, in 1989. Since 2006, he has been a Professor at the Center of Information Security and Technology, Graduate School of Information Management and Security, Korea University, Seoul. His research interests include parallel and distributed computing, multimedia computing, and multimedia security. He has contributed to MPEG standardization for digital item adaptation, file format, symbolic music representation, and multimedia application format, with more than ten contributions and the same number of patents. In addition, he has filed many patents and published more than 30 reviewed papers in international journals including IEEE and ACM, and two peer-reviewed book chapters. He was the prime investigator of the national projects during 1997–2005 developing interactive and personalized digital television. He has served as Guest Editor of the IEEE Transactions on Circuits and Systems for Video Technology and Eurasip Journal of Advances in Signal Processing, and has been Technical Program Chair of many international conferences including International Workshop on Digital Watermarking. He is a Vice Editor-in-Chief of the LNCS Transactions on Data Hiding and Multimedia Security, Associate Editor of well-known international journals, and Editor of many Lecture Notes in Computer Science series

Prof. Kim is a member of ACM and several Korean academic societies.

Jeho Nam (M'00–SM'07) received the B.S. degree in electrical and control engineering from Hongik University, Seoul, in 1992, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Minnesota at Twin Cities, Minneapolis, in 1996 and 2000, respectively.

He was a Senior Member of Engineering Staff with the Radio and Broadcasting Research Laboratory, Electronics and Telecommunications Research Institute, Daejon, where he is currently the Leader of Convergence Contents Protection Research Team. He is also an Adjunct Professor in the field of mobile communication and digital broadcasting engineering at the Korea University of Science and Technology, Daejeon. He has participated in standardization activities, including MPEG-21 and the Digital Media Project. His research interests are in the area of signal processing for digital video technologies and multimedia applications, in particular content-based audio/video analysis, context-aware media adaptation, and multimedia security and protection.

Dr. Nam is a Member of the ACM.

Sundaram Suresh received the B.E. degree in electrical and electronics engineering from Bharathiyar University, India, in 1999, and the M.E. and Ph.D. degrees in aerospace engineering from the Indian Institute of Science, Bangalore, India, in 2001 and 2005, respectively.

From 2005 to 2007, he was with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, as a Postdoctoral Research Fellow. Since, 2008, he has been with INRIA Sophia Antipolis, France, as an ERCIM Postdoctoral Research Fellow. His research interest includes adaptive flight control, unmanned aerial vehicles, neural networks, parallel computing and computer vision.

Yun Qing 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.

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, steganaysis, forensics and information assurance, visual signal processing and communications, theory of multidimensional systems, and signal processing. He is an author/coauthor of more than 200 research papers, a book, and four book chapters. He holds two U.S. patents and has additional 20 U.S. patents pending. He is the founding Editor-in-Chief of LNCS TRANSACTIONS ON DATA HIDING AND MULTIMEDIA SECURITY (Springer). He was Technical Program Chair of ICME07, Co-Technical Chair of IWDW06, 07, MMSP05, Co-General Chair of MMSP02.