Reversible Watermark Using an Accurate Predictor and Sorter Based on Payload Balancing

Sang-ug Kang, Hee Joon Hwang, and Hyoung Joong Kim

A series of reversible watermarking technologies have been proposed to increase embedding capacity and the quality of the watermarked image simultaneously. The major skills include difference expansion, histogram shifting, and optimizing embedding order. In this paper, an accurate predictor is proposed to enhance the difference expansion. An efficient sorter is also suggested to find a more desirable embedding order. The payload is differently distributed into two sub-images, split like a chessboard pattern, for better watermarked image quality. Simulation results of the accurate prediction and sorter based on the payload balancing method yield generally better performance over previous methods. The gap is wide, in particular, in low payload for natural images. The peak signal-to-noise ratio improvement is around 2 dB in low payload ranges.

Keywords: Reversible watermarking, security, image processing, histogram shifting, difference expansion.

I. Introduction

Usually digital watermarking intentionally deteriorates an original image and the degree of corruption is proportional to the amount of embedded data. In the case of a still image, one of the simplest watermarking methods is to replace the least significant bit plane of pixel values with a payload. Techniques including spread spectrum [1], quantization [2], and amplitude modulation [3] are designed to achieve various purposes such as robustness and efficiency, but they are all lossy. This lossy property of watermarking is problematic to some experts, such as graphic designers and physicians, who rely upon high image quality in their professions. It's also important that military and medical images remain intact to preserve original data. Thus, reversible watermarking can be used for authentication to retain the integrity of the image. After checking the integrity, reversible watermarking allows us to recover the original image. If the watermarked image quality is sufficiently high, it can be used without decoding so the integrity check and image recovery can be done when necessary. The proposed method helps increase convenience due to its excellent performance in low capacity.

Even though the previous achievements are remarkable and significant, the best prospect is still to minimize the difference between the original and watermarked images, while maximizing data hiding capacity. Quantum leaps have been made over the last decade by famous techniques such as difference expansion by Tian [4], histogram shifting by Ni and others [5], combining [4] and [5] by Thodi and Rodriguez [6], sorting by Kamstra and Heijmans [7], and a difference sorting method by Sachnev and others [8].

First, the difference expansion method hides the payload in a difference value between pixels in a pair. In this method, the

Manuscript received Mar. 7, 2011; revised Oct. 18, 2011; accepted Nov. 1, 2011.

This work was in part supported by the 3DLife projects of National Research Foundation, ITRC project of Ministry of Knowledge and Economy, and KEIT grant (#R1107181).

Sang-ug Kang (phone: +82 2 2131 0443, sukang@alumni.usc.edu), Hee Joon Hwang (ksteiner@korea.ac.kr), and Hyoung Joong Kim (corresponding author, khj-@korea.ac.kr) are with the Department of Information Management & Security, Korea University, Seoul, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.12.0111.0075

 © 2012 ETRI

difference value is doubled to insert one bit of the payload. Of course, more than one bit can be hidden by increasing the difference value fourfold or eightfold. However, such an excessive magnification of difference values worsens image quality significantly. Kim and others [9] introduced the concept of changeable and unchangeable pairs. Only the changeable pairs are expanded to avoid excessive distortion. Since the unchangeable pairs are obvious, marking in the location map is not necessary. This method is extended to various prediction error expansion methods that use the difference value between predicted and original values. The payload should be inserted in places with low difference values to reduce the distortion, resulting in low data embedding capacity due to the limitation of embeddable places. On the other hand, two approaches have increased the capacity. One approach is to increase the number of candidate pixels for data embedding. Tian's "pair" [4] is extended to "vector" [10] as proposed by Alattar. The capacity in Alattar's method is as many as k–1 for a k-bit vector compared to $k/2$ in Tian's method [4]. The capacity increment is due to the reuse of one pixel as the anchoring pixel for the remaining pixels in a vector. While only a horizontal vector is used in [4], Hsiao and others [11] added a vertical vector to increase prediction accuracy. The vertical vector is selected when the edge direction of surrounding blocks is vertical by calculating their variances. Kim and others [12] employed sub-sampled images, which are made by sampling the original image with sampling interval 2, 3, and so on. One sub-sampled image plays a role as a reference for prediction, and the payload is hidden in other sub-sampled images. In [4], [11]-[13], not all the pixels can be utilized to embed the payload because at least one reference pixel is necessary to reconstruct the image. The original image is divided into two sub-images with an interleaving pattern like a chessboard to overcome this constraint so every pixel can be utilized for data hiding, as used in [8], [14] and as used in this paper. Another approach is an effort to improve the prediction accuracy so that various predictors [8], [10], [13], [15] are introduced for reversible watermarking. In contrast, Chen and others [16] applied a wide range of predictors, such as six JPEG modes, a gradient-adjusted predictor, and a median edge detector, to the reversible watermarking and compared their performances. The result showed that the average predictor using the four vertically and horizontally neighboring pixels outperformed all predictors considered in [16]. This is the same predictor presented in [8], [14], [17] and is used as the basis of the accurate predictor proposed in this paper.

Second, the histogram shifting method is very useful for reducing the size of the location map. Usually the location map tells the decoder which pixel is modified or which pixel is at risk for overflow or underflow. Of course, good predictors may reduce the size of a location map, but the size reduction is insufficient. Ni and others [5] proposed a method that uses multiple pairs of peak and zero points to embed the secret messages. The peak-zero pair embedding method guarantees the minimum watermarked image quality or 48.13 dB if no multilayer embedding is used. The same histogram shifting method is also used in [13] and [14], in which the multilevel data hiding method is employed to achieve higher capacity. However, the peak-zero embedding method has limited capacity and payload embedding control. Tai and others [18] utilized multiple peak points, and Jung and others [19] also presented an adaptively adjustable embedding level by considering the HVS to overcome these problems. The generalized histogram shifting method is proposed in [8] and [17], using two or four threshold values that determine the inner and outer region in a difference histogram presented in [6]. Thus, these approaches can fundamentally remove the need of the location map by expanding only allowed pixels within the inner region and shifting others to secure rooms for expansion. Note that expansion or shifting can still cause overflow and underflow problems. This means that the generation of a location map that indicates overflow/underflow is inevitable to cope with such problems while its size is very small, and even zero, in many cases, due to advanced reversible data hiding techniques.

Third, additional distortion can only be caused by histogram shifting, not by expanding. Note that the payload is embedded only through expansion. Conversely, distortion is caused by both expansion and shifting. In the case of histogram shifting, an optimal approach is proposed by Hwang and others [17] and Xuan and others [20]. In general, probability distribution of the prediction error is Laplacian and exponentially decreases from the center of the histogram. Therefore, it is natural to think that the center portion should be expanded and that the remaining parts need to be shifted to maximize the embedding capacity and minimize distortion. That is, the region to be expanded is marked as (T_{p1}, T_{p2}), where T_{p1}=0 and T_{p1}<T_{p2}. However, Hwang and others [17] show that such an assumption is not always true. They determined nonzero T_{p1} can cause less distortion. For the negative difference values, (T_{n2}, T_{n1}) range is used. Thus, (T_n, T_p) range for an ordinary histogram shifting method implies $T_{n1} = T_{p1} = 0$.

Fourth, sorting is another effort to form a small map size and reduce distortion introduced by histogram shifting. However, the sorting method is applicable only when a pixel is independent. For example, if the prediction uses a causal window as in [7], [13], [19] and the like, it is impossible to use the sorting technique. Therefore, Sachnev and others [8] made all pixels independent of each other, using an interleaving prediction method.

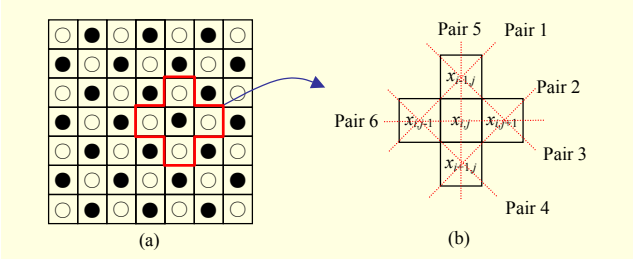

Fig. 1. (a) White and Black set and (b) pixel pairs.

Contributions of the proposed method can be summarized as follows:

1) Enhanced prediction technique using an adaptive predictor that takes two or four neighboring pixels selectively according to the level of their similarity.
2) Enhanced sorting technique that exploits a broader image area to achieve a better result.
3) Optimal payload distribution technique that can balance the amount of payload between White and Black sets as explained in Fig. 1.

II. Proposed Reversible Watermark Techniques

The detailed description and experiment results will be presented after introducing the main steps of the proposed method (that is, new prediction, sorting, and payload balancing as follows):

- Step 1. Original image is divided into two sets for two-stage embedding as illustrated in Fig. 1(a).
- Step 2. Set a proportion of payload to be embedded into the White and Black sets between 0.45 and 1.
- Step 3. The White set is selected first and encoded as explained in Steps 4 and 5 with a fixed amount of payload set in Step 2.
- Step 4. Every pixel is estimated and reordered using the proposed accurate predictor and sorter.
- Step 5. Payload is embedded into optimal histogram bin pairs according to the sorted order of pixels, checking overflow and underflow.
- Step 6. The Black set is selected and encoded as explained in Steps 7 and 8 with a fixed amount of payload set in Step 2.
- Step 7. Every pixel is estimated and reordered using an average predictor and the proposed accurate sorter.
- Step 8. Payload is embedded into optimal histogram bin pairs according to the sorted order of pixels, checking overflow and underflow.
- Step 9. Find the optimum ratio of payload for White and Black sets iteratively.
- Step 10. Payload size, embedding ratio, and location map information are embedded in the Black set.

In comparing the proposed reversible watermarking method to existing methods, the proposed method demonstrates improved performance both by appropriately controlling the payload distribution for embedding in the White and Black sets and by the combination of prediction that is more accurate and a sorting technique that exploits a broader image. Decreasing the prediction error rate consequently produces a better histogram, for which the standard deviation is smaller. This improvement allows us to find good candidates for data hiding and reduce the chance of distortion induced by both prediction error expansion and histogram shifting. Simultaneously, accurate sorting of pixels in the order of low prediction error reduces the undesirable error propagation due to histogram shifting. Enhancing the methods of prediction and sorting can give us a better result by separately balancing the respective amounts of payload embedding in the White and Black sets, because the enhanced accuracy in the White set has a negative effect on the Black set in terms of prediction and sorting. Thus, we need to rely more on the White set to achieve higher performance if the image is sufficiently natural and the payload is relatively small.

1. New Predictor and Its Performance

A cross with four opposite color pixels (that is, pixels of the White set) is made to construct a block around a pixel (for example, a pixel of the Black set) (Fig. 1(a)). The center pixel value should be predicted more accurately to reduce the prediction error. The four surrounding pixels can be referenced for prediction, since they are invariant. We have six combinations of pairs (Fig. 1(b)): pair 1 to pair 6, since there are four neighboring pixels. Each pair is denoted by p_i .

In this paper, either one of six pairs or four neighboring pixels are used for more accurate prediction. There are four diagonal pairs from pair 1 to pair 4, and pair 5 and pair 6 are vertical and horizontal, respectively. For example, pair 1 consists of two pixels: $x_{i-1,j}$ and $x_{i,j-1}$. The mean value of four neighboring pixels [7], [14], [17], which will be called the Sachnev predictor throughout this paper, usually works well. However, this is inaccurate in high frequency areas. Thus, the Sachnev predictor is used only in low frequency areas. If the difference values of all pairs are less than a threshold value t, the block is considered smooth and full of low frequency components. If there is at least one pair whose difference value exceeds t, the block is taken to be sharp and full of high frequency components. When the difference value between two pixels in a pair is larger than t, the mean value of another pair is used as a predicted value. This relationship is summarized as follows:

Table 1. Number of prediction errors, $(T_n, T_p) = (-1,1)$, using Sachnev's predictor and proposed predictor.

Predictor type	$d_{i,j}$	Lena		Baboon		Barbara		Airplane	
		White	Black	White	Black	White	Black	White	Black
Sachnev [8]	–1	16,912	15,433	5,391	5,270	13,065	12,031	25,821	24,019
	0	17,314	15,224	5,493	5,271	13,024	11,611	25,554	22,371
	1	14,767	13,301	5,323	5,190	11,195	10,027	14,301	13,826
	Total	48,993	43,958	16,207	15,731	37,284	33,669	65,767	60,216
Proposed	–1	17,252	15,562	5,589	5,323	13,795	12,423	26,548	24,407
	0	17,649	15,437	5,595	5,105	13,857	12,232	26,376	22,674
	1	15,156	13,245	5,445	5,054	12,036	10,496	14,891	14,226
	Total	50,057	44,244	16,629	15,482	39,688	35,151	67,815	61,307

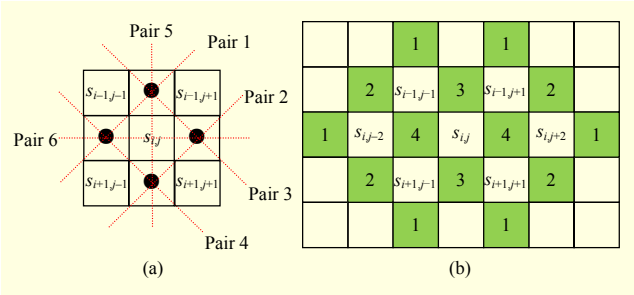

Fig. 2. (a) Pixel pairs for sorting and (b) neighborhood similarity and weighting numbers of surrounding pixels.

$$\hat{x}_{i,j} = \begin{cases} \lfloor m(p_i) \rfloor, & \text{if } |p_j| > t, \\ \left\lfloor \frac{1}{4}(x_{i,j-1} + x_{i-1,j} + x_{i,j+1} + x_{i+1,j}) \right\rfloor, & \text{otherwise,} \end{cases} \quad (1)$$

where m denotes a function of computing the mean value of a pair, and $|p_i|$ denotes the difference between pixels in the pair p_i. In (1), p_i and p_j do not share a common pixel. That is, they do not overlap. For example, if the difference value of pixels in p_1 exceeds threshold value t, then the predicted value is $\lfloor m(p_2) \rfloor$. In this case, note that p_2 is chosen since p_1 is considered the opposite of it. There are three opposite pairs: (p_1, p_2), (p_3, p_4), and (p_5, p_6) in (1).

In this paper, after thorough experiments, we choose t=20 for diagonal pairs and t=14 for vertical and horizontal pairs in (1). The threshold value for diagonal pairs is larger because the distance of pixels in horizontal pairs is relatively longer than that of pixels in diagonal pairs. This approach assumes that the actual pixel value $x_{i,j}$ is likely to be similar to its neighbor pixels in the low frequency area. This predictor is called "accurate predictor," and it adaptively chooses either an average or median value as the predicted value by considering situations of the surrounding pixels.

The performance of the Sachnev predictor is compared to the accurate predictor (see Table 1). The smaller the prediction error, the better the embedding performance. Thus, the numbers of small prediction errors such as –1, 0, and 1 are counted for the entire White set and then for the Black set, considering that pixel values are inevitably modified after randomly hiding 0 or 1 in the White set. The overall result of the accurate predictor outperforms the Sachnev predictor especially for the Barbara image that has many stripe patterns. The result is always better than the Sachnev predictor is for the White set. However, the performance for the Black set is relatively worse than for the White set, even though they are almost the same images and sometimes the accurate predictor has a weaker performance than the Sachnev predictor. With these results, we realize that the accurate predictor works better for natural images, but not for manipulated or synthesized ones. Note that the White set is a kind of synthesized one because it is modified by data hiding and histogram shifting. This is one reason to use the Sachnev predictor only for the Black set.

2. New Sorter and Its Performance

The concept of neighborhood similarity is introduced here to sort pixels more meaningfully. Neighborhood similarity of a pixel, defined in (2), roughly tells us whether a pixel is located in a low or high frequency area or the level of complexity around the pixel $x_{i,j}$. If a pixel position is in the White set, then its neighborhood similarity should be calculated using adjacent pixels in the Black set so that the reversibility is guaranteed. In addition, it is reasonable that the sorting method should be accordingly modified to reflect the prediction method by checking the difference of pixel values in pairs.

$$s_{i,j} = \begin{cases} |p_i|, & \text{if } |p_i| > t, \\ \frac{1}{4}\sum_{i=1}^{4} |p_i|, & \text{otherwise.} \end{cases} \quad (2)$$

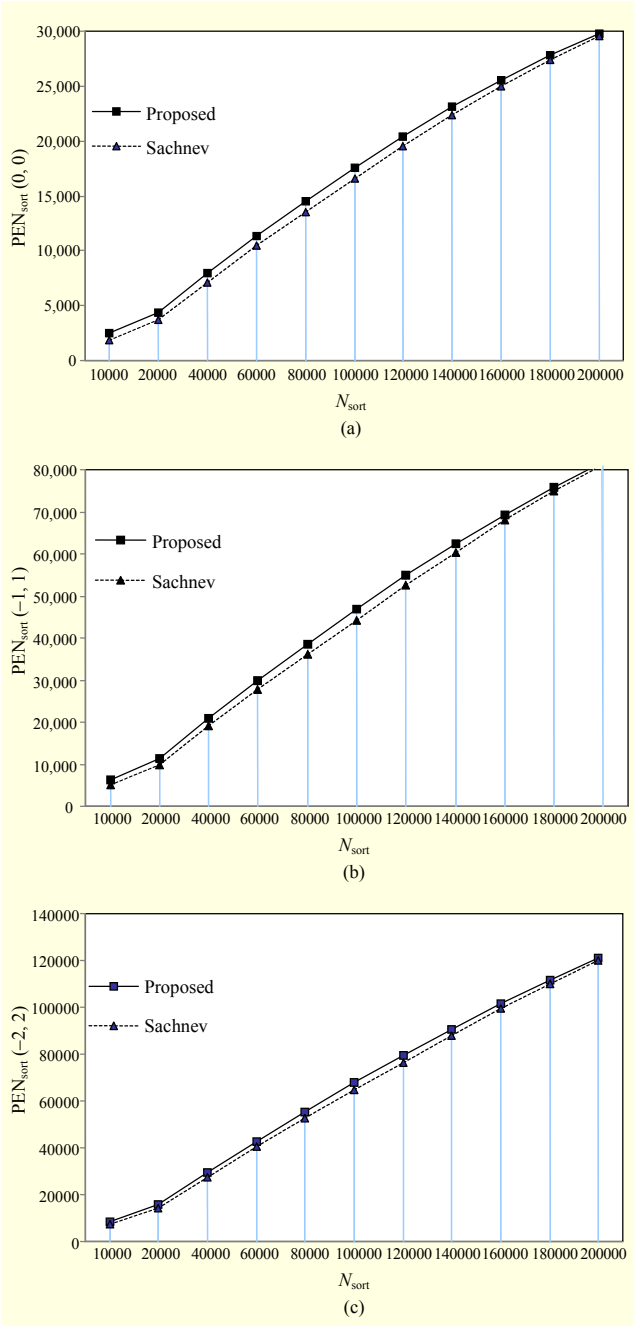

Fig. 3. Performance of proposed predictor and sorter over Sachnev method for (a) (T_n, T_p)=(0, 0), (b) (T_n, T_p)=(–1, 1), and (c) (T_n, T_p)=(–2, 2).

The neighborhood similarity is determined by analyzing the frequency component of four neighboring pixel values if the difference of each pair exceeds a threshold or two neighboring pixel values, or vice versa, similar to the prediction case. For example, if p_1 is excluded in the prediction process, only $|p_2|$ is taken such that $s_{i,j} = |x_{i+1,j} - x_{i,j+1}|$. However, the similarity defined in (2) is insufficient to accurately estimate the level of complexity around the pixel due to the small neighborhood area for which coverage is only 3×3. Thus, the similarity of the pixel is extended to (3) to accommodate a broader area, such as 5×7, (as shown in Fig. 2(b)) so that the estimation accuracy is increased.

$$s'_{i,j} = \frac{1}{7}(s_{i,j} + s_{i,j-2} + s_{i-1,j-1} + s_{i-1,j+1} + s_{i,j+2} + s_{i+1,j-1} + s_{i+1,j+1}). \quad (3)$$

The new sorting discriminant function $s'_{i,j}$ implicitly has the form of a weighting factor on neighboring pixels as shown in Fig. 2(b) to reflect the distance from $x_{i,j}$ to adjacent pixels. For example, the right-hand side pixel of $x_{i,j}$ has a weighting factor of four, which means this pixel is used four times to calculate $s'_{i,j}$. Therefore, the numbers in Fig. 2(b) can be interpreted as weighting factors.

Figure 3 shows the improved performance using both the accurate predictor and sorter, by counting the number of small prediction errors from the first to the N-th pixel in the sorted list and comparing the accurate and Sachnev predictors. From now on, the N-th pixel will be represented as N_{sort} and the number of prediction errors in the range of (T_n, T_p) within N_{sort} is denoted by $PEN_{sort}(T_n, T_p)$. The three graphs in Fig. 3 are drawn by processing Steps 1 through 10, excluding Steps 2 and 9, which instead have a fixed embedding ratio of 0.5. In Fig. 3, there are two monotonically increasing lines and one decreasing line. The increasing ones display $PEN_{sort}(T_n, T_p)$ and the decreasing ones show the improvement ratio of the proposed method compared to Sachnev and others' method. The improvement ratios are 34.9%, 24.7%, and 15%, respectively, for the embeddable prediction error range (T_n, T_p)=(0, 0), (−1, 1), and (−2, 2), in the case of N_{sort}=10,000. For example, if the accurate predictor and sorter are used, then $PEN_{sort}(0, 0)$ is 2,499 at N_{sort}=10,000 for the Lena image, in which there are 15,395 zero prediction errors according to Table 1. This means that 16% (2,499/15,395) of zero prediction errors are concentrated within 3% (10,000/262,144) of the sorted pixel list. Conversely, the numbers are changed to 1,852 ($PEN_{sort}(0, 0)$) and 15,224 (the total number of zero prediction errors) when the Sachnev predictor and sorter are used. Thus, the improvement ratio is about 34.9% ([2,499–1,852]/1,852). This tells us that the embedding capacity is increased 34.9% and that the error propagation effect induced by histogram shifting is reduced, correspondingly. However, the improvement rate decreases as N_{sort} increases, as depicted in the diamond dotted lines in Fig. 3(a) through (c); this is natural because the sorting technique is usually effective in low payload cases. In addition, if we want to use a broader data embedding range for larger embedding capacity, the proposed sorter yields better results, but the improvement rate slightly decreases to 24.7% ([6,196–4,967]/4,967) in (T_n, T_p)=(–1, 1) and 15% ([8,254–7176]/7,176) in (T_n, T_p)=(–2, 2), in the case of N_{sort}

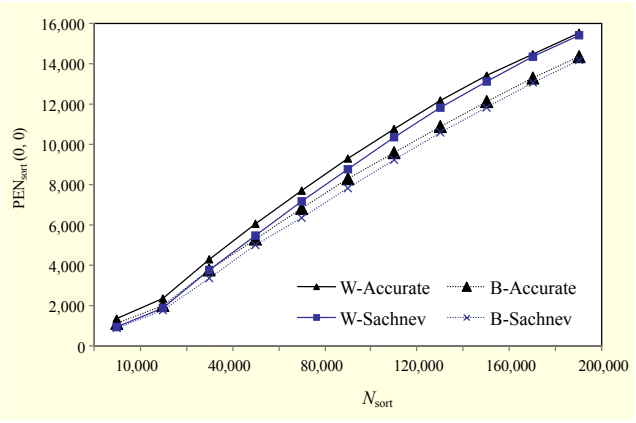

Fig. 4. Performance difference between White and Black sets for Lena image. White set performs better than Black set.

=10,000.

3. Characteristics of Image Sets and Payload Balancing

It is impossible to have a good predictor for a randomly generated image. However, good predictors are available for natural images. Thus, the predictor explained in subsection II.1 can be accurate for the White set but may not be for the Black set, because pixels in the White set are randomly altered and they are used as reference pixels for the prediction of pixels in the Black set. What should be stressed here is that the statistical characteristic of the Black set can deteriorate after data hiding occurs in the White set. It means that enhancing the prediction and sorting accuracy in the White set worsens the effect on the Black set. The accurate predictor is depicted in (1) for the White set and (4) for the Black set. The detailed explanation is following.

In Fig. 4, N_{sort} in four cases is illustrated to show that the accurate predictor and sorter scheme has a relatively big performance difference between the White and Black sets, which is only 81.5% (1,109/1,360) identical at N_{sort} =10,000. Meanwhile, there is small difference in the case of the Sachnev method, which is 94.3% (899/953) identical at N_{sort} = 10,000. The numbers are changed to 87.7% and 94.9% on the Baboon image.

The effect of two predictors on two image sets is almost the same after about N_{sort} =80,000 for Lena and N_{sort} =40,000 for Baboon, but it gets bigger if N_{sort} becomes smaller for both images. As you can see in Fig. 5(a) and (b), the performance of the proposed (accurate) predictor and sorter in the White set compared to the performance of the Sachnev method is a 42.7% improvement at N_{sort} =10,000, but only a 18.5% improvement in the Black set. The performance gap widens as N_{sort} approaches 0. Use of the Sachnev predictor is suggested for the Black set to cope with this phenomenon. Figure 5(b) displays the result in which the performance improvement rate

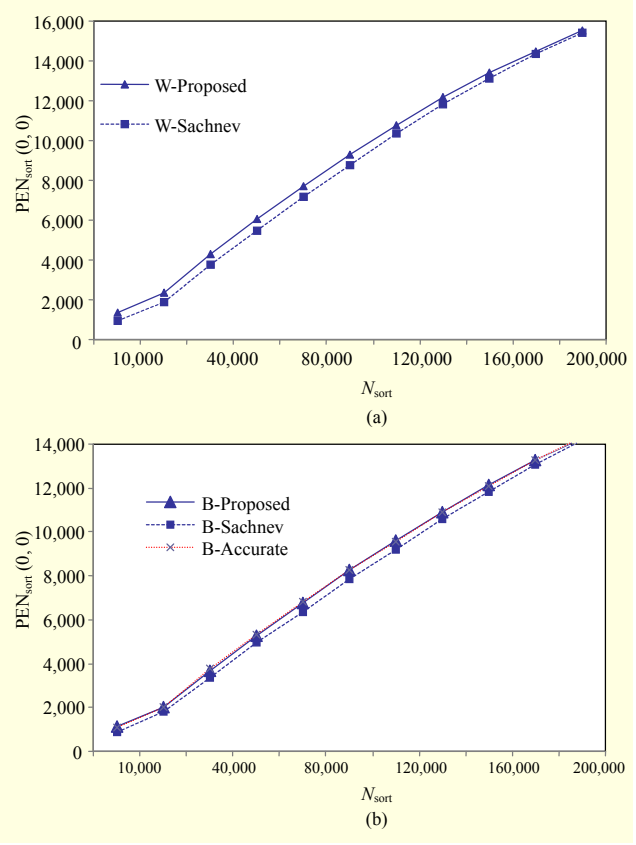

Fig. 5. Different effects of predictors on (a) White set and (b) Black set.

increases to 26.7% from 18.5% for the Black set while it remains the same for the White set. The numbers are changed to 20.6% from 17.3% for the Baboon image. Such a trend continues as N_{sort} approaches the number of image pixels or 262,144.

The accurate predictor is applied only to the White set and the Sachnev predictor in (4) is used for the Black set to reflect the result of a series of performance tests. Therefore, the proposed predictor is a combination of the accurate and Sachnev predictors.

$$\hat{x} = \left\lfloor \frac{1}{4}(x_{i,j-1} + x_{i-1,j} + x_{i,j+1} + x_{i+1,j}) \right\rfloor. \quad (4)$$

The biased performance of the predictor motivates the amount of the payload to be hidden in both image sets to be balanced to achieve a better result. It is desirable to embed more payloads in the White set and less in the Black set, especially for small capacity regions to take advantage of a bigger improvement of the proposed method on the White set. Let the ratio of the amount of the payload embedded in the White set to that in the Black set be denoted by R. For example, R=1 means every payload is hidden only in the White set and none are hidden in the Black set. Thus, the optimal R, or R_{opt}, which yields maximum watermarked image quality, should be

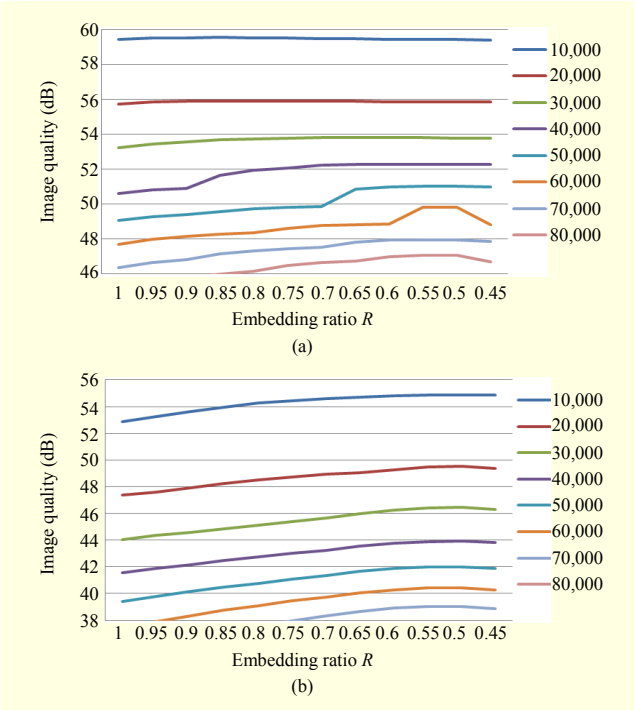

Fig. 6. Image quality changes as R changes for various payloads embedded in (a) Lena and (b) Baboon.

determined in the encoding process and sent to the decoder. Unfortunately, it is very hard to estimate R_{opt}, as shown in Fig. 6, because R_{opt} changes according to image characteristics and the amount of the payload.

Two methods are proposed to obtain R_{opt} to reduce the computational complexity. Firstly, the step search (SS) method calculates every peak signal-to-noise ratio (PSNR) from R=1 to R=0.45, stepping down R by 0.05. The search range is between R=1 and R=0.45, as found empirically. Of course, it is not guaranteed that the searched R is optimum, but this is at the expense of computational complexity reduction. Twelve PSNRs have to be calculated to get R_{opt}. Secondly, the peak search (PS) method is based on the SS method. It navigates from R=1, not from R=0.45, to take advantage of the fact that accurate prediction and sorter based on payload balancing (APS-PB) yields better results if more of the payload is hidden in the White set in the case of a small payload, usually less than 10,000 bits. We start from R=1 and then R=0.95. If the PSNR value grows, then we keep navigating by decreasing R by 0.05; otherwise, we stop. Finally, the peak value is chosen to be R_{opt}. The searched R_{opt} values are found to be the same as for the SS method for Lena and Barbara and very similar for Baboon and Airplane.

4. Proposed Reversible Watermarking Framework

Firstly, an image is separated into two sets, one marked with white circles and one with black circles, as shown in Fig. 1(a). Next, consider a pixel with value $x_{i,j}$ in an eight-bit grayscale image at position (i, j), where i represents the i-th row and j represents the j-th column. The pixel value range is $0 \le x_{i,j} \le L$, where L is 255. A pixel intensity $x_{i,j}$ is predicted using two or four neighboring pixels using (1) for the White set and the Sachnev predictor depicted in (4) for the Black set. The difference value between $x_{i,j}$ and its predicted value $\hat{x}_{i,j}$ is represented by $d_{i,j} = x_{i,j} - \hat{x}_{i,j}$. One bit of the payload b is inserted in $d_{i,j}$ by doubling it and adding b, then the revised difference $D_{i,j}$ is equal to $2d_{i,j}+b$, where $b \in (0,1)$. The revised pixel value of $x_{i,j}$ after data embedding is described by (5).

$$X_{i,j} = D_{i,j} + \hat{x}_{i,j} = d_{i,j} + b + x_{i,j}. \tag{5}$$

Note that (5) should satisfy $0 \le X_{i,j} \le L$, which results in (6).

$$\begin{cases} x_{i,j} \le L - d_{i,j} - 1, & d_{i,j} \ge 0, \\ x_{i,j} \ge -d_{i,j}, & d_{i,j} < 0. \end{cases} \tag{6}$$

According to (5), it is desirable to hide data in locations with small prediction errors, so that the difference between $X_{i,j}$ and $x_{i,j}$, or image distortion, is as small as possible. These locations are determined by setting four parameters, two negative prediction error thresholds, T_{n2}, T_{n1}, and two positive ones, T_{p1}, T_{p2}. Let E_c denote a set of locations selected for data embedding without considering the overflow and underflow problem. That is, $E_c = (i,j)|_{T_{n2} \le d_{i,j} \le T_{n1}}^{T_{p1} \le d_{i,j} \le T_{p2}}$. Before data embedding, all the pixel locations in the White set are reordered so that pixels with low $d_{i,j}$ are first, according to the specific measurements based on image attributes with (2) and (3). Additionally, we need to keep in mind that the overflow or underflow problem may be caused by the data embedding and shifting processes. Since thresholds are defined, (6) should be modified to (7) to embrace them as follows:

$$|N| \le x_{i,j} \le L - |P| - 1, \tag{7}$$

where T_{n2}–T_{n1}=N, T_{p2}–T_{p1}=P. All the pixels in a host image can be categorized into two sets, a set of embeddable pixels $E_c = (i,j)|_{T_{n2} \le d_{i,j} \le T_{n1}, T_{p1} \le d_{i,j} \le T_{p2}}^{|N| \le x_{i,j} \le L-|P|-1}$ and a complementary set $\overline{E} = (i,j)|_{(i,j) \notin E}$, which is not embeddable. When $d_{i,j}$ is out of the threshold range, the image worsens due to modified pixel values by P+1 in the case of a positive difference or N–1 in the case of a negative difference, as shown in (8).

$$D_{i,j} = \begin{cases} d_{i,j} + N - 1, & d_{i,j} < T_{n2}, \\ 2(d_{i,j} - T_{n1}) + T_{n1} - 1 + b, & T_{n2} \le d_{i,j} \le T_{n1}, \\ d_{i,j}, & T_{n1} \le d_{i,j} \le T_{p1}, \\ 2(d_{i,j} - T_{p1}) + T_{p1} + b, & T_{p1} \le d_{i,j} \le T_{p2}, \\ d + P + 1, & T_{p2} < d_{i,j}. \end{cases} \tag{8}$$

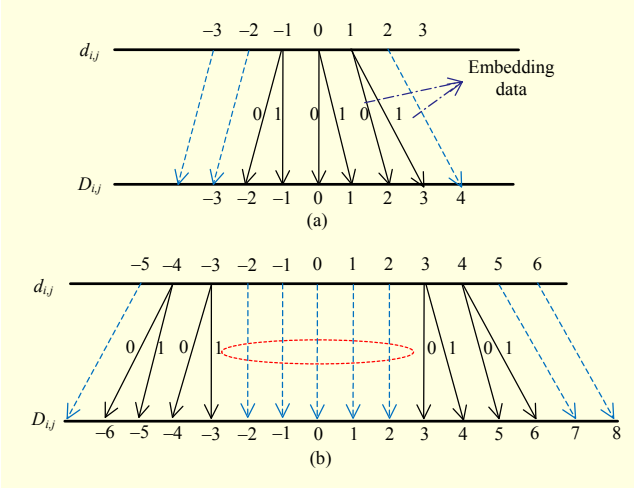

Fig. 7. (a) Normal histogram shifting and (b) generalized Hwang and others' histogram shifting method.

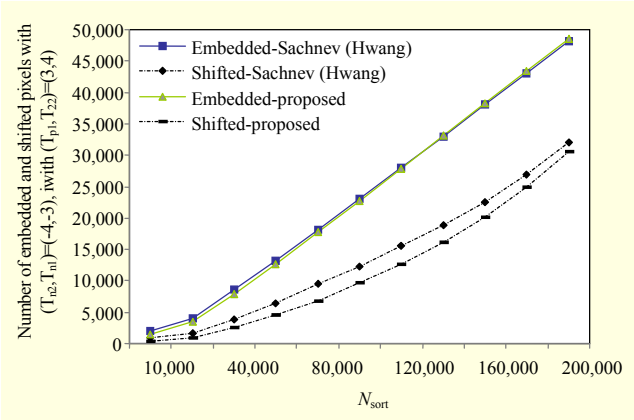

Fig. 8. Ratio of number of embedded and shifted pixels effect of predictor and sorter.

The reversible watermarking algorithm should completely recover the host image in the receiver, and the embedded payload bit can be extracted from (9).

$$b = D_{ij} \bmod 2. \tag{9}$$

The original prediction error is computed as $d_{i,j} = \lfloor D_{i,j}/2 \rfloor$. Then, the original pixel value is recovered with (5).

Now we are ready to embed the payload into an image. Considering the payload embedding strategy, Sachnev and others [8] utilized pixels with relatively small prediction errors, that is, zero prediction error first, –1 and 1 second, –2 and 2 third, and so on, regardless of image set type. In [17], the payload can be hidden in any histogram bins starting from any bin. This is called the optimum histogram pair shifting (OHPS) method, so the error propagation effect, illustrated in Fig. 7, is minimized. The proposed APS-PB method is also valid for both reversible watermarking frameworks. OHPS is adopted in this paper. In addition, it is not allowed to embed the payload alternatively in the White and Black sets, because it is not causal. With this strategy, the payload is embedded in every available pixel position within (T_{n2}, T_{n1}) and (T_{p1}, T_{p2}) in the White set initially and then the remaining amount of the payload is embedded in the available pixel positions in the Black set, until the entire payload is hidden, including the location map.

For instance, if the histogram bins (T_{n2}, T_{n1}) = (–4, –3) and (T_{p1}, T_{p2}) = (3, 4) are selected during the encoding process, then pixel values corresponding to bin –2, –1, 0, 1, and 2 do not have to change their values and only bins outside –4 and 4 are shifted, creating distortion after watermarking.

In Fig. 8, we know the APS-PB method outperforms in terms of the ratio of the number of embedded pixels to the number of shifted pixels. At N_{sort}= 10,000, the ratio is 2.4 (that is, 1,982/842) when the prediction and sorting techniques in [8] are used and 3.8 (1,384/362) when the APS-PB method is used, as shown in Fig. 8. The result tells us that the APS-PB can embed more data and fewer pixels are shifted. The ratio is important, because we can move to a more spacious histogram bin or allocate another bin if the data embedding space is insufficient. Likewise, the lines indicating the ratios meet as the amount of embedded payload increases.

III. Experiment Results

The proposed APS-PB method is implemented on the basis of the OHPS technique [17] and compared to Sachnev and others' method [8] and Hwang and others' method [17] on four 512×512 eight-bit grayscale test images: Lena, Baboon, Barbara, and Airplane, as shown in Fig. 9. The APS-PB method can also be implemented based on Sachnev and others' method. Figure 9 compares performance in the low bit-per-pixel (bpp) range, from 1,000 bits (that is, 0.0038 bpp) to 10,000 bits (that is, 0.038 bpp) to show the APS-PB effect more clearly. The enhancement of image quality reaches 71.7 dB, which is 2.1 dB larger than Sachnev and others' method, as in Fig. 9, and 72.3 dB, which is 1.3 dB larger than Hwang and others' method, as in Fig. 9(a) at 0.0038 bpp for the Lena image. The quality improvement is around 1 dB for the other images, as in Fig. 9 (b) through (d).

The proposed method shows consistently better results throughout all bpp ranges compared to the results achieved using the existing methods, including [8], [11], [13], [17], [19], [21]. Since the sorting technique is less effective if the bpp approaches 1, it is reasonable to limit the comparison range to around 0.6 to 0.7 bpp. The proposed method exceeds Luo and others' method [13] on three test images (Fig. 10). Our method is better than Jung and others' method [19] when

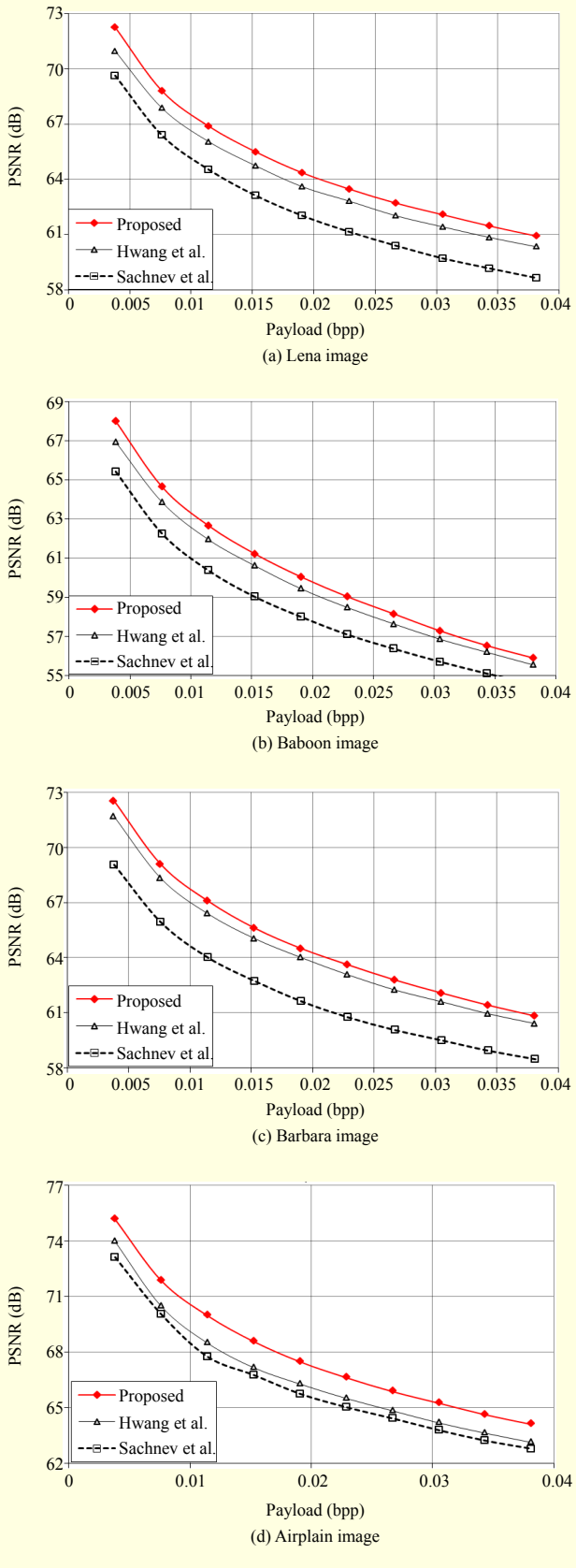

Fig. 9. Performance comparison of proposed method.

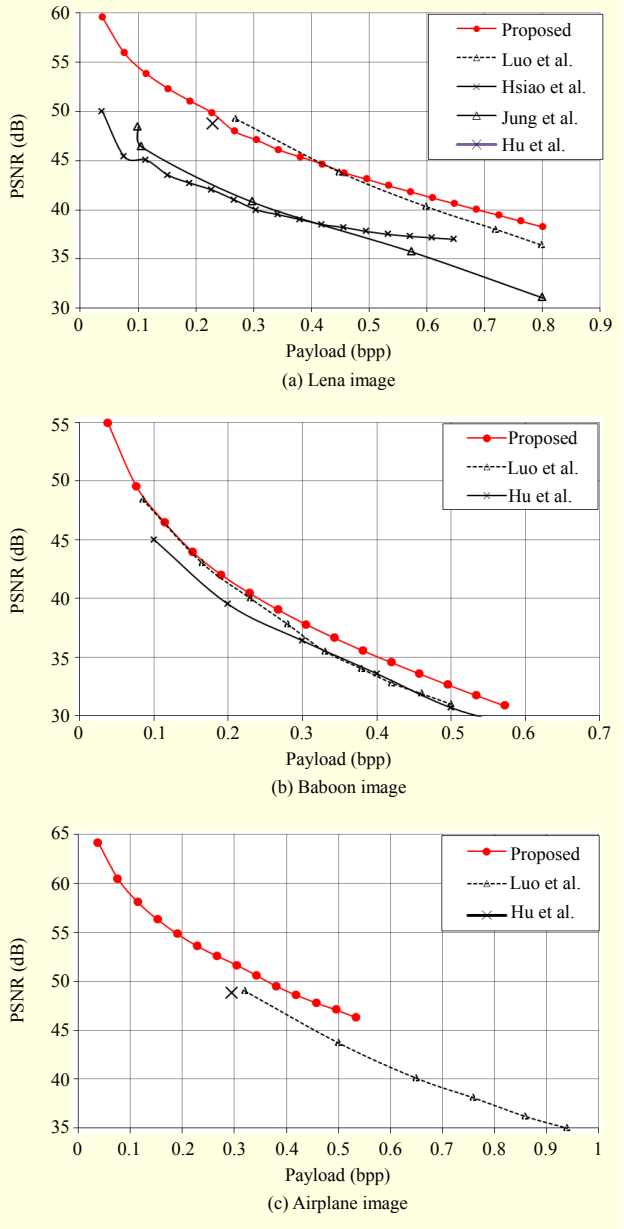

Fig. 10. Experiment results in large bpp range on (a) Lena, (b) Baboon, and (c) Airplane.

the bit rate is large.

IV. Concluding Remarks

Many reversible watermarking methods have been devised to embed as many secret messages as possible. These methods modify an original image minimally. This is the common goal of reversible watermarking. Thus, several methods have been proposed to improve prediction and sorting toward achieving this goal. This paper proposed an adaptive prediction technique utilizing image characteristics and a sorting technique utilizing

both image characteristics and the broader area of neighbor pixels to attain the same goal. These two improved techniques were proven to be more effective in combination with payload balancing, mainly due to the enhanced accuracy of prediction and sorting. The APS-PB technique worked especially well in low payload cases. It was applied to the reversible watermarking frameworks suggested in [8] and [17], and those were much improved. However, even though the predictor and sorter were significantly improved, the PSNR for high capacity was not significantly improved, because sorting was ineffective in a high bpp range. Even so, the proposed method was proven superior across a wide range.

References

[1] J.R. Hernández and F. Pérez-González, "Statistical Analysis of Watermarking Schemes for Copyright Protection of Images," *Proc. IEEE*, vol. 87, no. 7, 1999, pp. 1142-1166.

[2] Z.M. Lu and S.H. Sun, "Digital Image Watermarking Technique Based on Vector Quantization," *Electron. Lett.*, vol. 36, no. 4, Feb. 2000, pp. 303-305.

[3] J.R. Hernandez et al., "Performance Analysis of a 2-D-Multipulse Amplitude Modulation Scheme for Data Hiding and Watermarking of Still Images," *IEEE J. Sel. Areas Commun.,* vol. 16, no. 4, May 1998, pp. 510-524.

[4] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, 2003, pp. 890-896.

[5] Z. Ni et al., "Reversible Data Hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, 2006, pp. 354-362.

[6] D.M. Thodi and J.J. Rodriguez, "Expansion Embedding Techniques for Reversible Watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, 2007, pp. 721-730.

[7] L.H. Kamstra and A. Heijmans, "Reversible Data Embedding into Images Using Wavelet Techniques and Sorting," *IEEE Trans. Image Process.*, vol. 14, no. 12, 2005, pp. 2080-2090.

[8] V. Sachnev et al., "Reversible Watermark Algorithm Using Sorting and Prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, 2009, pp. 989-999.

[9] H.J. Kim et al., "A Novel Difference Expansion Transform for Reversible Data Embedding," *IEEE Trans. Inf. Forensics Security,* vol. 3, no. 3, 2008, pp. 456-465.

[10] A.M. Alattar, "Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, 2004, pp. 1147-1156.

[11] J. Hsiao, K. Chan, and J. Chang, "Block-Based Reversible Data Embedding," *Signal Process.,* vol. 89, no. 4, 2009, pp. 556-569.

[12] K.S. Kim et al., "Reversible Data Hiding Exploiting Spatial Correlation Between Sub-sampled Images," *Pattern Recognition,* vol. 42, no. 11, 2009, pp. 3083-3096.

[13] L. Luo et al., "Reversible Image Watermarking Using Interpolation Technique," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, Mar. 2010, pp. 187-193.

[14] C. Yang and M. Tsai, "Improving Histogram-Based Reversible Data Hiding by Interleaving Predictions," *IET Image Process.*, vol. 4, no. 4, 2010, pp. 223-234.

[15] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS," *IEEE Trans. Image Process.*, vol. 9, no. 8, 2000, pp. 1309-1324.

[16] M. Chen et al., "Model Order Selection in Reversible Image Watermarking," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 3, June 2010, pp. 592-604.

[17] H. Hwang, H. Kim, and V. Sachnev, "Reversible Watermarking Method Using Optimum Histogram Pair Shift Based on Prediction and Sorting," *KSII Trans. Internet Inf. Syst.*, vol. 3, no. 2, 2009, pp. 134-151.

[18] W. Tai, C. Yeh, and C. Chang, "Reversible Data Hiding Based on Histogram Modification of Pixel Differences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 6, June 2009, pp. 906-910.

[19] S. Jung, L.T. Ha, and S. Ko, "A New Histogram Modification Based Reversible Data Hiding Algorithm Considering the Human Visual System," *IEEE Signal Process. Lett.*, vol. 18, no. 2, Feb. 2011, pp. 95-98.

[20] G. Xuan et al., "Optimum Histogram Pair Based Image Lossless Data Embedding," *LNCS*, vol. 5041, 2008, pp. 264-278.

[21] Y. Hu, H. Lee, and J. Li, "DE-Based Reversible Data Hiding with Improved Overflow Location Map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, Feb. 2009, pp. 250-260.

Sang-ug Kang received his BS in electronic engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1993, and his MS and PhD from the University of Southern California, California, USA, and Korea University, Seoul, Rep. of Korea, in 1996 and 2011, respectively. He worked for IBM Korea as a system engineer and for Samsung Electronics Company as a research engineer at the Central R&D Center. In 2002, he joined the National Information Society Agency (NIA) in Korea and he is currently responsible for its Convergence Service Department as a director. His research interests include multimedia security, multimedia processing, multimedia-based services and video quality measurement, and telecommunication service quality measurement.

Hee Joon Hwang received his BS from the Electric and Electronic Engineering Department and earned his MS in the Graduate School of Information Management and Security of Korea University, Seoul, Rep. of Korea, in 2008. He joined the Multimedia Security Lab at the Center of Information Security and Technology (CIST), Korea University, in 2007, where he is currently pursuing a PhD. His research interests include multimedia security, reversible and robust watermarking, and steganography.

Hyoung Joong Kim received his BS, MS, and PhD from Seoul National University, Seoul, Rep. of Korea, in 1978, 1986, and 1989, respectively. He is currently a professor at Korea University, Seoul, Rep. of Korea. His research interests include data hiding.