A BRIEF REVIEW ON REVERSIBLE DATA HIDING: CURRENT TECHNIQUES AND FUTURE PROSPECTS

Xiaolong Li[1], *Weiming Zhang*[2], *Bo Ou*[3] *and Bin Yang*[1]

[1]Institute of Computer Science and Technology, Peking University, Beijing 100871, China
[2]School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China
[3]Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China

ABSTRACT

Reversible data hiding (RDH), in which the original cover medium can be exactly extracted from the marked content, has attracted considerable interests from the information hiding community. In this paper, current RDH techniques are first reviewed by introducing the following five types of schemes: early lossless-compression-based schemes, expansion-based schemes, recent lossless-compression-based schemes, content-adaptive schemes, integer-transform-based schemes. Then, by proposing some unsolved problems, the future prospects of RDH are also discussed.

Index Terms— Reversible data hiding, lossless compression, expansion embedding, histogram modification, adaptive embedding.

1. INTRODUCTION

In the last decade, reversible data hiding (RDH) has received much attention from the information hiding community [1–3]. RDH is a special type of data hiding and it ensures a lossless recovery of cover data. Specifically, by RDH, not only the embedded message, the cover content can also be exactly extracted from the marked data. A framework of RDH for digital images is illustrated in Fig. 1. Notice that RDH is a fragile technique and the marked data cannot undergo any degradation. In this light, a RDH scheme is usually evaluated by its capacity-distortion performance, i.e., for a given embedding capacity (EC), one expects to minimize the embedding distortion measured by PSNR in dB. The first several RDH schemes are mainly developed for image authentication [4,5]. Later on, RDH has been extensively studied and various schemes have been proposed. And, this technique has also been applied in some image/video applications such as medical image processing [6–8], multimedia archive management [9], image trans-coding [10], video error-concealment coding [11], and data coloring in the cloud [12], etc.

In this paper, as a brief review, we first introduce current RDH techniques in Section 2. Then, in Section 3, we discuss the future prospects of RDH by proposing some unsolved problems. We hope that this work will facilitate and benefit the research of RDH.

2. CURRENT RDH TECHNIQUES/SCHEMES

2.1. Early lossless-compression-based RDH schemes

Initial RDH schemes are mainly based on the lossless compression technique [13–15]. The idea behind these schemes is to release some space by losslessly compressing a feature set S of the cover image, and utilize the saved space to embed data. The embedding is implemented by replacing S with its compressed form S_C and the message, so the maximum EC is the size of $S - S_C$. The performance

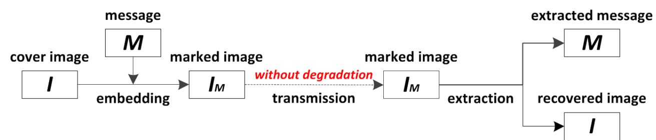

Fig. 1. Framework of RDH embedding/extraction.

of these methods is determined by the employed compression algorithm and the selected feature set. In [13], Fridrich *et al.* proposed to find the space by compressing proper bit-plane with the minimum redundancy. In their method, unless the image is noisy, the lowest bit-plane is compressed and embedded with a hash value. In [15], Celik *et al.* proposed a generalized LSB compression method to improve the compression efficiency by using unaltered portions of cover data as side-information. However, the above lossless-compression-based methods cannot yield a satisfactory performance, since the correlations among a bit-plane is too weak to provide a high EC. As EC increases, one needs to compress more bit-planes, thus the distortion increases dramatically.

2.2. Expansion-based RDH schemes

2.2.1. Histogram shifting (HS)

HS-based RDH is first proposed by Ni *et al.* [16] and this type of schemes are implemented by modifying the image histogram of a certain dimension. We start our presentation for HS by introducing a modified version of Ni *et al.*'s method which is based on modifying the pixel-intensity-histogram.

Consider here a gray-scale image I. For a given integer a, $1 \le a \le 253$, the hidden data is embedded into I in the following way to get the marked image $\widetilde{I}$:

$$\widetilde{I}_{i,j} = \begin{cases} I_{i,j} - 1, & \text{if } I_{i,j} < a \\ I_{i,j} - m, & \text{if } I_{i,j} = a \\ I_{i,j} + m, & \text{if } I_{i,j} = a+1 \\ I_{i,j} + 1, & \text{if } I_{i,j} > a+1 \end{cases} \quad (1)$$

where (i,j) is a pixel location and $m \in \{0,1\}$ is a to-be-embedded bit. In this procedure, each pixel value is modified at most by 1, and thus the PSNR of marked image versus the original one is at least 48.13 dB. Consequently, a high visual quality of marked image is guaranteed.

The decoder can extract the embedded data and restore the original image by simply reading marked pixel values:

• If $\widetilde{I}_{i,j} < a - 1$, there is no hidden data in the pixel and its original value is $\widetilde{I}_{i,j} + 1$.
• If $\widetilde{I}_{i,j} \in \{a-1, a\}$, the pixel is used to carry data and its original value is a. The embedded bit is $m = a - \widetilde{I}_{i,j}$.

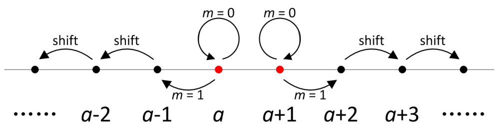

Fig. 2. Ni *et al.*'s HS-based RDH scheme.

• If $\widetilde{I}_{i,j} \in \{a+1, a+2\}$, the pixel is also used to carry data and its original value is $a+1$. The embedded bit is $m = \widetilde{I}_{i,j} - (a+1)$.
• If $\widetilde{I}_{i,j} > a+2$, similar to the first case, there is no hidden data and the original value is $\widetilde{I}_{i,j} - 1$.

Fig. 2 shows an illustration of this method, in which black values are shifted and red ones are expanded to embed data. This simple method can well illustrate the general mechanism of HS-based RDH schemes: for data embedding, certain pixel values are shifted to create vacant spaces whereas some others are expanded to carry hidden data by filling those vacant spaces.

Later on, this method is improved by Lee *et al.* [17] by using the histogram of difference image. Lee *et al.*'s method outperforms Ni *et al.*'s by improving both EC and visual quality. Compared with the ordinary image pixel-intensity-histogram, the difference-histogram is better for RDH since it is regular in shape and has a much higher peak point. As a result, the spatial correlation of natural images is exploited in Lee *et al.*'s method and thus an improved performance is obtained. After that, Ni *et al.*'s method is extensively investigated and many subsequent works are proposed. Specifically, in a recent work [18], by extending Ni *et al.* and Lee *et al.*'s schemes, a general construction for designing HS-based RDH is proposed by Li *et al.*, and this construction includes many previous RDH schemes as special cases.

We now briefly introduce Li *et al.*'s general construction. Let S and T be a partition of $\mathbb{Z}^n$: $S \cup T = \mathbb{Z}^n$ and $S \cap T = \emptyset$, where n is the image block size. Suppose that three functions $g : T \to \mathbb{Z}^n$, $f_0 : S \to \mathbb{Z}^n$ and $f_1 : S \to \mathbb{Z}^n$ satisfy the following two conditions:
-$C1$: The functions g, f_0 and f_1 are injective.
-$C2$: The sets $g(T)$, $f_0(S)$ and $f_1(S)$ are disjointed with each other.
Here, g is called "shifting function" and will be used to shift pixel values, f_0 and f_1 are called "embedding functions" and will be used to embed data. More specifically, after dividing the cover image into non-overlapping blocks of size n, each block with value $\mathbf{x} \in T$ will be shifted to $g(\mathbf{x})$, and the block with value $\mathbf{x} \in S$ will be expanded to either $f_0(\mathbf{x})$ or $f_1(\mathbf{x})$ to carry one bit. The shifting and embedding functions will give a HS-based RDH algorithm where the reversibility is guaranteed by the conditions $C1$ and $C2$.

Furthermore, to avoid underflow/overflow (i.e., for a gray-scale image, the shifted and expanded values should be restricted in the range of $[0, 255]$), the above defined sets T and S need be further processed. Let $A_n = \{\mathbf{x} = (x_1, ..., x_n) \in \mathbb{Z}^n : 0 \le x_i \le 255\}$ be the set of all pixel-value-arrays of length n of gray-scale image. We define $T_s = A_n \cap g^{-1}(A_n)$, $S_e = A_n \cap f_0^{-1}(A_n) \cap f_1^{-1}(A_n)$, $T_{u,o} = A_n \cap T - T_s$ and $S_{u,o} = A_n \cap S - S_e$. Here, the sub-indices "*s*", "*e*" and "*u*, *o*" mean "shift", "embed" and "underflow/overflow", respectively. Obviously, the four sets T_s , S_e, $T_{u,o}$ and $S_{u,o}$ are disjointed with each other and constitute a partition of A_n. Moreover, the sets $g(T_s)$, $f_0(S_e)$ and $f_1(S_e)$ are contained in A_n and the condition $C2$ ensures that they are also disjointed.

By these definitions, each block with value $\mathbf{x} \in T_s$ will be shifted, each block with value $\mathbf{x} \in S_e$ will be expanded to carry one bit, and the block with value $\mathbf{x} \in T_{u,o} \cup S_{u,o}$ will remain unchanged since it cannot be shifted or expanded due to undeflow/overflow. The data embedding procedure based on this general framework is illustrated in Fig. 3. By this construction, one only needs to design the embedding and shifting functions to derive a RDH scheme.

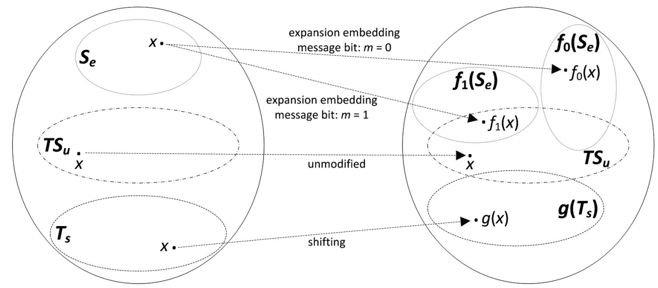

Fig. 3. RDH using shifting and embedding functions. After data embedding, the sets $g(T_s)$, $f_0(S_e)$ and $f_1(S_e)$ are disjointed with each other, but the two sets $TS_u \triangleq T_{u,o} \cup S_{u,o}$ and $g(T_s) \cup f_0(S_e) \cup f_1(S_e)$ may be overlapped, and a location map will be used to record the locations of pixel blocks whose values belong to TS_u.

2.2.2. *Difference expansion (DE)*

We now present Tian's DE scheme [19]. For a pixel pair (x_0, x_1), define their integer average and difference as $l = \lfloor (x_0 + x_1)/2 \rfloor$ and $h = x_1 - x_0$. In order to embed one bit $m \in \{0, 1\}$, the difference h is expanded to $h^* = 2h + m$, and the marked pixel pair (y_0, y_1) is determined as $y_0 = l - \lfloor h^*/2 \rfloor$ and $y_1 = l + \lfloor (h^* + 1)/2 \rfloor$. By a simple reduction, one can get

$$\begin{cases} y_0 = 2x_0 - \lceil (x_0 + x_1)/2 \rceil \\ y_1 = 2x_1 - \lceil (x_0 + x_1)/2 \rceil + m \end{cases}. \tag{2}$$

In this form, the decoder can determine the embedded bit m as the LSB of $y_1 - y_0$, and recover the original pixel pair (x_0, x_1) as $x_0 = l' - \lfloor h'/2 \rfloor$ and $x_1 = l' + \lceil h'/2 \rceil$, where $l' = \lfloor (y_0 + y_1)/2 \rfloor$ and $h' = \lfloor (y_1 - y_0)/2 \rfloor$. Compared with the lossless-compression-based schemes [14, 15] and HS-based scheme [16], the DE-based method performs much better by providing a higher EC while keeping the distortion low.

DE is a fundamental technique of RDH, and this technique has also been widely investigated and developed, mainly in the aspects of integer-to-integer transformation [20–24], location map reduction [25–27], and prediction-error expansion (PEE) [28–39]. Among these extensions, the PEE-based one has attracted considerable attention since this approach has the potential to well exploit the spatial redundancy in natural images. Unlike in DE where only the correlation of two adjacent pixels is considered, the local correlation of larger neighborhood is exploited in PEE, and thus a better performance can be expected. In this light, we think PEE is an important expansion-based RDH technique, and we will introduce PEE in details in the next subsection. Moreover, we remark that DE is essentially a special case of PEE since in DE, a pixel is in fact predicted by its adjacent neighbor [40, 41].

2.2.3. *Prediction-error expansion (PEE)*

By summarizing previous typical PEE-based schemes such as [28–31], we first introduce the PEE embedding procedure. The procedure contains three steps:
Step 1: Predict image pixels to obtain a prediction-error sequence. First, under a specific scan sequencing, the cover pixels are collected into a 1D sequence as $(x_1, ..., x_N)$. Then, a predictor is used to determine the prediction of x_i denoted as $\widehat{x}_i$. Next, the prediction-error is computed by $e_i = x_i - \widehat{x}_i$ (suppose here for simplicity that $\widehat{x}_i$ is an integer). Finally, the prediction-error sequence $(e_1, ..., e_N)$ is derived.
Step 2: Generate the prediction-error histogram (PEH) by counting the frequencies of prediction-errors. Usually, the PEH obeys a

Laplacian-like distribution centered at 0 or close to 0.

Step 3: Embed data by modifying the PEH through expansion and shifting. Specifically, for each e_i, it is expanded or shifted as

$$e_i' = \begin{cases} 2e_i + m, & \text{if } e_i \in [-T, T) \\ e_i + T, & \text{if } e_i \in [T, +\infty) \\ e_i - T, & \text{if } e_i \in (-\infty, -T) \end{cases} \tag{3}$$

where T is a capacity-dependent integer-valued parameter, and $m \in \{0, 1\}$ is a to-be-embedded bit. Here, the bins in $[-T, T)$ are expanded to embed data, and those in $(-\infty, -T) \cup [T, +\infty)$ are shifted outwards to create vacancies. Finally, each pixel value x_i is modified to $x_i' = \widehat{x}_i + e_i'$ to obtain the marked image.

According to (3), one can see that the maximum modification to each pixel value is limited by the capacity-parameter T, and the marked image quality can be well controlled by taking a proper T.

In PEE extraction procedure, the original prediction-error e_i is recovered from the marked prediction-error e_i' as

$$e_i = \begin{cases} \lfloor e_i'/2 \rfloor, & \text{if } e_i' \in [-2T, 2T) \\ e_i' - T, & \text{if } e_i' \in [2T, +\infty) \\ e_i' + T, & \text{if } e_i' \in (-\infty, -2T) \end{cases} \tag{4}$$

and the embedded bits are extracted as the LSBs of those prediction-errors $e_i' \in [-2T, 2T)$. Finally, the cover image is restored using the recovered prediction-errors. Notice that, to guarantee the reversibility, the key point is that the prediction values used in extraction should be the same as that in embedding.

In summary, PEE can embed a large payload by exploiting PEH, and control the embedding distortion by simultaneously utilizing expansion embedding and shifting. PEE is currently a research hotspot and the most powerful technique of RDH. And, most recently proposed RDH works are based on this technique by incorporating some strategies such as better prediction algorithm utilization [29, 32, 33, 42–44], double-layered embedding [32, 33, 45], embedding-position-selection [32, 37, 46], context-modification [35], optimal-bins-selection [34, 38, 47, 48], and 2D PEH utilization [49, 50], etc.

2.3. Recent lossless-compression-based RDH schemes

One basic problem for RDH is what is the upper bound of the payload for a given cover sequence and a distortion constraint. For i.i.d. cover sequence, this problem has been solved by Kalker and Willems [51], who formulated the RDH as a special rate-distortion problem, and obtained the rate-distortion function, i.e., the upper bound of the embedding rate under a given distortion constraint Δ, as follows:

$$\rho_{rev}(\Delta) = maximize\{H(Y)\} - H(X) \tag{5}$$

where X and Y denote the random variables of cover and marked signal respectively. The maximum entropy is over all transition probability matrices $P_{Y|X}(y|x)$ satisfying the distortion constraint $\sum_{x,y} P_X(x) P_{Y|X}(y|x) D(x, y) \leq \Delta$, where the metric $D(x, y)$ is usually defined as the square error distortion, i.e., $(x - y)^2$.

Recently, some lossless-compression-based coding methods are independently proposed in [52–55], which can approach the rate-distortion bound (5). All these methods can be viewed as improved versions of the recursive code construction in [51]. We take the recursive-histogram-modification (RHM) method [55] as an example to briefly introduce such lossless-compression-based schemes. First, RHM should solve the problem (5) to estimate the optimal probability transition matrix $P_{Y|X}(y|x)$ and $P_{X|Y}(x|y)$. Second, RHM divides the cover sequence into disjoint blocks and embeds the message by recursively modifying the histogram of each block with the compression and decompression algorithms of an entropy coder according to the optimal probability transition matrices.

As RHM behaves exactly the same within each block, we take a single block here to concisely illustrate the data embedding. Assume the block sequence $\mathbf{x} = (x_1, ..., x_K)$ is a K-tuple composed of K samples drawn with probability distribution P_X. According to the probability transition matrix $P_{Y|X}(y|x)$, we decompress S bits of message $\mathbf{m} = (m_1, ..., m_S)$ into a marked sequence $\mathbf{y} = (y_1, ..., y_K)$ and replace $\mathbf{x}$ with $\mathbf{y}$. Note that the message is usually encrypted before embedded, so we assume $\mathbf{m}$ is a binary random sequence. Afterwards, because $\mathbf{x}$ is similar to $\mathbf{y}$, we generate a compressed version of $\mathbf{x}$ according to the probability transition matrix $P_{X|Y}(x|y)$, denoted by $O(\mathbf{x})$. In order to restore the original $\mathbf{x}$ at the receiver side, $O(\mathbf{x})$ is embedded into the next block. The data extraction and cover restoration are executed in a backward manner.

For a given cover sequence and a desired payload, the capacity-approaching codes can minimize the embedding distortion. Therefore, with such codes, designers of RDH only need pay their attention to generating the cover signal X with small entropy.

2.4. Content-adaptive RDH schemes

The first content-adaptive RDH scheme is the one proposed by Kamstra and Heijmans [25]. In [25], Tian's DE method is improved by sorting pixel pairs according to the local variance before embedding data. Notice that in DE, the integer-average of two pixels in a pair is invariant and only the difference value is altered, and thus these embedding-invariants can be utilized by both encoder and decoder to compute the local variance to sort pixel pairs. Clearly, if the local variance of a pixel pair is small, the pair is located in a flat image region and it is more probably to be expandable with small difference value. Therefore, by sorting, the location map can be compressed remarkably. The experiments reported in [25] indicated significant improvement over the original DE-based method. Moreover, in some recent works [32, 37, 43, 45, 46, 49, 50, 56–58], the authors illustrated that combining sorting (or, pixel-selection, embedding-location-selection) with other reversible techniques such as PEE, can dramatically improve the embedding performance.

The key issue of content-adaptive RDH is to assign each pixel a complexity measurement to determine whether the pixel is located in a flat image region or not. Then, by considering only the pixels located in flat regions while ignoring the noisy ones, an accurate prediction can be made and a more sharply distributed PEH can be generated. With this adaptively generated PEH, the performance of RDH can then be enhanced through expansion embedding and shifting. In our opinion, the content-adaptive embedding is an important strategy of RDH and a better utilization of this strategy is very helpful for enhancing the embedding performance.

2.5. Integer-transform-based RDH schemes

The integer transform (IT) can be employed to design RDH. According to (2), one can see that DE is actually a kind of IT, and Tian's DE-based method can be viewed as the first IT-based RDH. Later on, Alattar [20] proposed an IT by generalizing DE. The method takes several adjacent pixels as a vector and embeds data into each selected embeddable vector. Unlike DE that embeds data one bit by one bit, it can, for example, simultaneously embed three bits into a pixel quad. In [59], Coltuc and Chassery proposed a method based on the so-called reversible contrast mapping which is an IT of integer pair. Particularly, the method does not need additional lossless data compression, and thus it is efficient in terms of computational complexity. Coltuc and Chassery's method is improved by Chen *et* *al.* [60] by extending the reversible contrast mapping to integer array of arbitrary size. In [21], Wang *et al.* generalized the DE also by using a new IT. They show that the embedding rule of DE can be reformulated as a transform of integer pair and give a novel algorithm by extending the transform. Recently, from another IT viewpoint for DE, Peng *et al.* [22] proposed a novel RDH scheme. In general, the IT-based methods group several pixels into a unit and embed data unit by unit. It advantages in reducing the impact of location map to the embedding performance. However, for the IT-based schemes, it usually uses a less efficient prediction in which the average value of a block is used to predict each pixel within the block. On the other hand, unlike PEE, the maximum modification cannot be controlled by IT. Based on these observations, we argue that the IT-based RDH is only efficient for the case of high EC.

3. FUTURE PROSPECTS

1) High dimensional histogram utilization: The reversibility of RDH mainly relies on the lossless compressibility of natural image. Therefore, an insightful understanding of complex image structure and a better redundant information exploitation may further enhance the performance of RDH. Most PEE-based schemes are based on modifying a 1D histogram. These methods generally contain two steps: (Step 1, histogram generation) First, each local image region consisting of several pixels is projected to a 1D space to get a scalar sequence. Then, a 1D histogram is generated by counting the frequency of the resulting sequence. (Step 2, histogram modification) Finally, embed data into the cover image by modifying the histogram, e.g., via expansion and shifting. In the first step, the complex local image correlation is simplified to a 1D statistic. By this simplification, the image redundancy cannot be fully exploited and it only contributes to the second step since a 1D histogram is easy to deal with. Based on this observation, to better exploit the image redundancy, some new schemes based on 2D-histogram-modification are proposed [49, 50]. For example, in [50], every two adjacent prediction-errors (e_{2i-1}, e_{2i}) are counted to generate a 2D histogram, and then, a new embedding strategy named pairwise PEE is proposed by modifying the resulting histogram (see Fig. 3 of [50]). However, these 2D-histogram-modification-based schemes are rather heuristic and lack a systematic investigation. Then, how to construct a more suitable 2D histogram for RDH, and how to better modify the 2D histogram to optimize the embedding performance, are meaningful problems for the feature research. On the other hand, although difficult, one may also consider higher dimensional histogram rather than 2D.

2) Multi-histograms modification: In content-adaptive PEE, a complexity measurement n_i is computed for each pixel x_i according to its context. Then, only the pixels satisfying $n_i < NL$ are selected to generate PEH and to embed data, while those with $n_i \geq NL$ are ignored, where NL is a given threshold. Another viewpoint for this procedure is that, by taking h_k as the PEH generated form the pixels with $n_i = k$ (we assume that n_i is an positive integer for simplicity), one can see that only the histograms h_k with $0 \leq k < NL$ are utilized for data embedding. And, the EC provided by h_k is $\sum_{e=-T}^{T-1} h_k(e)$ if $0 \leq k < NL$ and 0 if $k \geq NL$, where T is the capacity-parameter (see the context of (3)). In this multi-histograms-modification viewpoint, a natural problem is that, for a given EC denoted as C, how to minimize the total distortion $\sum_{k=0}^{NL_{\max}} D_k$ subject to the condition $\sum_{k=0}^{NL_{\max}} C_k = C$, where $NL_{\max}$ is the maximum of the complexity measurement, and D_k is the distortion introduced by embedding the payload of size C_k into h_k. Notice that, the content-adaptive PEE provides a solution to this problem, however, its optimality is questioned, and some primary experimental results suggest that a better solution exists. Then, it is meaningful to investigate the multi-histograms-modification-based optimization problem. Surely, other constructions for multi-histograms are also valuable for consideration.

3) Capacity partition in multi-layered embedding: The double-layered embedding technique proposed by Sachnev *et al.* [32] is a promising issue of RDH. In [32], the image is first divided into two parts to get "cross set" and "dot set" (see Fig. 1 of [32]). Then, the cross set is used for embedding a half of the payload and the dot set for computing predictors. After that, the modified cross set is used for predicting the dot pixels, and the dot set will be embedded with the rest half of the payload. With double-layered embedding, an accurate prediction is made using full-enclosing pixels, i.e., a pixel is predicted by the average of its four nearest neighbors. We argue that, a major problem of double-layered embedding is the "equally capacity partition", since in the 2nd embedding layer, the prediction is less efficient than the 1st layer, and the resulting PEH is somewhat flatter than that of the 1st layer. We think that embedding a relatively larger payload into the 1st layer will enhance the embedding performance, and this observation is validated by our experiments. However, how to get a better payload partition for double-layered embedding to maximize the embedding performance, is still questioned and worthy studying. Moreover, the "equally capacity partition" problem also exists in other multi-layered embedding schemes such as [18, 45] and should be fixed in the future.

4) Color image RDH: Unlike the case of gray-scale images, RDH for color images is a rarely studied topic [61]. However, color images are more popular than gray-scale ones in reality. So, RDH for color images should be emphasized in the future work. We think that the key issue for color image RDH is how to utilize the correlations of different channels. One possible way to this problem is to take the prediction-errors of three channels into a triple to generate a 3D PEH, and then reversibly embed data by modifying this 3D PEH. Moreover, a better prediction based on the cross-channel correlation will also improve the embedding performance.

4. REFERENCES

[1] Y. Q. Shi, Z. Ni, D. Zou, C. Liang, and G. Xuan. Lossless data hiding: fundamentals, algorithms and applications. In *Proc. IEEE ISCAS*, volume 2, pages 33–36, 2004.

[2] Y. Q. Shi. Reversible data hiding. In *Proc. IWDW*, volume 3304 of *Springer LNCS*, pages 1–12, 2004.

[3] R. Caldelli, F. Filippini, and R. Becarelli. Reversible watermarking techniques: An overview and a classification. *EURASIP J. Inf. Secur.*, 2010. Article ID 134546.

[4] J. M. Barton. Method and apparatus for embedding authentication information within digital data, 1997. US Patent 5,646,997.

[5] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel. Lossless recovery of an original image containing embedded data, 2001. US Patent 6,278,791.

[6] G. Coatrieux, C. L. Guillou, J. M. Cauvin, and C. Roux. Reversible watermarking for knowledge digest embedding and reliability control in medical images. *IEEE Trans. Inf. Technol. Biomed.*, 13(2):158–165, Mar. 2009.

[7] M. Fontani, A. D. Rosa, R. Caldelli, F. Filippini, A. Piva, and M. Consalvo. Reversible watermarking for image integrity verification in hierarchical PACS. In *Proc. ACM MM&SEC*, pages 161–168, 2010.

[8] F. Battisti, M. Carli, and A. Neri. Secure annotation for medical images based on reversible watermarking in the integer Fibonacci-Haar transform domain. In *Proc. SPIE EI*, volume 7870 of *SPIE*, 2011.

[9] S. Lee, C. D. Yoo, and T. Kalker. Reversible image watermarking based on integer-to-integer wavelet transform. *IEEE Trans. Inf. Forens. Security*, 2(3):321–330, Sep. 2007.

[10] R. Li, O. C. Au, C. K. M. Yuk, S. Yip, and T. Chan. Enhanced image trans-coding using reversible data hiding. In *Proc. IEEE ISCAS*, pages 1273–1276, 2007.

[11] K. L. Chung, Y. H. Huang, P. C. Chang, and H. Y. M. Liao. Reversible data hiding-based approach for intra-frame error concealment in H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.*, 20(11):1643–1647, Nov. 2010.

[12] K. Hwang and D. Li. Trusted cloud computing with secure resources and data coloring. *IEEE Internet Comput.*, 14(5):14–22, Sep.-Oct. 2010.

[13] J. Fridrich, M. Goljan, and R. Du. Invertible authentication. In *Proc. SPIE EI*, volume 4314 of *SPIE*, pages 197–208, 2001.

[14] J. Fridrich, M. Goljan, and R. Du. Lossless data embedding - New paradigm in digital watermarking. *EURASIP J. Adv. Signal Process.*, 2002(2):185–196, Feb. 2002.

[15] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber. Lossless generalized-LSB data embedding. *IEEE Trans. Image Process.*, 14(2):253–266, Feb. 2005.

[16] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.*, 16(3):354–362, Mar. 2006.

[17] S. K. Lee, Y. H. Suh, and Y. S. Ho. Reversible image authentication based on watermarking. In *Proc. IEEE ICME*, pages 1321–1324, 2006.

[18] X. Li, B. Li, B. Yang, and T. Zeng. General framework to histogram-shifting-based reversible data hiding. *IEEE Trans. Image Process.*, 22(6):2181–2191, Jun. 2013.

[19] J. Tian. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.*, 13(8):890–896, Aug. 2003.

[20] A. M. Alattar. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. Image Process.*, 13(8):1147–1156, Aug. 2004.

[21] X. Wang, X. Li, B. Yang, and Z. Guo. Efficient generalized integer transform for reversible watermarking. *IEEE Signal Process. Lett.*, 17(6):567–570, Jun. 2010.

[22] F. Peng, X. Li, and B. Yang. Adaptive reversible data hiding scheme based on integer transform. *Signal Process.*, 92(1):54–62, Jan. 2012.

[23] D. Coltuc. Low distortion transform for reversible watermarking. *IEEE Trans. Image Process.*, 21(1):412–417, Jan. 2012.

[24] X. Gui, X. Li, and B. Yang. A novel integer transform for efficient reversible watermarking. In *Proc. ICPR*, pages 947–950, 2012.

[25] L. Kamstra and H. J. A. M. Heijmans. Reversible data embedding into images using wavelet techniques and sorting. *IEEE Trans. Image Process.*, 14(12):2082–2090, Dec. 2005.

[26] S. Weng, Y. Zhao, J. S. Pan, and R. Ni. Reversible watermarking based on invariability and adjustment on pixel pairs. *IEEE Signal Process. Lett.*, 15:721–724, 2008.

[27] 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. Inf. Forens. Security*, 4(3):456–465, Sep. 2008.

[28] D. M. Thodi and J. J. Rodriguez. Expansion embedding techniques for reversible watermarking. *IEEE Trans. Image Process.*, 16(3):721–730, Mar. 2007.

[29] M. Fallahpour. Reversible image data hiding based on gradient adjusted prediction. *IEICE Electron. Express*, 5(20):870–876, Oct. 2008.

[30] Y. Hu, H. K. Lee, and J. Li. DE-based reversible data hiding with improved overflow location map. *IEEE Trans. Circuits Syst. Video Technol.*, 19(2):250–260, Feb. 2009.

[31] W. Hong, T. S. Chen, and C. W. Shiu. Reversible data hiding for high quality images using modification of prediction errors. *J. Syst. Softw.*, 82(11):1833–1842, Nov. 2009.

[32] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi. Reversible watermarking algorithm using sorting and prediction. *IEEE Trans. Circuits Syst. Video Technol.*, 19(7):989–999, Jul. 2009.

[33] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong. Reversible image watermarking using interpolation technique. *IEEE Trans. Inf. Forens. Security*, 5(1):187–193, Mar. 2010.

[34] C. Wang, X. Li, and B. Yang. Efficient reversible image watermarking by using dynamical prediction-error expansion. In *Proc. IEEE ICIP*, pages 3673–3676, 2010.

[35] D. Coltuc. Improved embedding for prediction-based reversible watermarking. *IEEE Trans. Inf. Forens. Security*, 6(3):873–882, Sep. 2011.

[36] X. Gao, L. An, Y. Yuan, D. Tao, and X. Li. Lossless data embedding using generalized statistical quantity histogram. *IEEE Trans. Circuits Syst. Video Technol.*, 21(8):1061–1070, Aug. 2011.

[37] X. Li, B. Yang, and T. Zeng. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Trans. Image Process.*, 20(12):3524–3533, Dec. 2011.

[38] H. T. Wu and J. Huang. Reversible image watermarking on prediction errors by efficient histogram modification. *Signal Process.*, 92(12):3000–3009, Dec. 2012.

[39] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux. Reversible watermarking based on invariant image classification and dynamical error histogram shifting. *IEEE Trans. Inf. Forens. Security*, 8(1):111–120, Jan. 2013.

[40] W. L. Tai, C. M. Yeh, and C. C. Chang. Reversible data hiding based on histogram modification of pixel differences. *IEEE Trans. Circuits Syst. Video Technol.*, 19(6):906–910, Jun. 2009.

[41] Y. C. Li, C. M. Yeh, and C. C. Chang. Data hiding based on the similarity between neighboring pixels with reversibility. *Digit. Signal Process.*, 20(4):1116–1128, Jul. 2010.

[42] W. J. Yang, K. L. Chung, H. Y. M. Liao, and W. K. Yu. Efficient reversible data hiding algorithm based on gradient-based edge direction prediction. *J. Syst. Softw.*, 86(2):567–580, Feb. 2013.

[43] C. Qin, C. C. Chang, Y. H. Huang, and L. T. Liao. An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism. *IEEE Trans. Circuits Syst. Video Technol.*, 23(7):1109–1118, Jul. 2013.

[44] B. Ou, X. Li, Y. Zhao, and R. Ni. Reversible data hiding based on PDE predictor. *J. Syst. Softw.*, 86(10):2700–2709, Oct. 2013.

[45] W. Hong. An efficient prediction-and-shifting embedding technique for high quality reversible data hiding. *EURASIP J. Adv. Signal Process.*, 2010. Article ID 104835.

[46] W. Hong. Adaptive reversible data hiding method based on error energy control and histogram shifting. *Opt. Commun.*, 285(2):101–108, Jan. 2012.

[47] H. J. Hwang, H. J. Kim, V. Sachnev, and S. H. Joo. Reversible watermarking method using optimal histogram pair shifting based on prediction and sorting. *TIIS*, 4(4):655–670, Aug. 2010.

[48] J. Wang and J. Ni. A GA optimization approach to HS based multiple reversible data hiding. In *Proc. IEEE WIFS*, pages 203–208, 2013.

[49] X. Li, W. Zhang, X. Gui, and B. Yang. A novel reversible data hiding scheme based on two-dimensional difference-histogram modification. *IEEE Trans. Inf. Forens. Security*, 8(7):1091–1100, Jul. 2013.

[50] B. Ou, X. Li, Y. Zhao, R. Ni, and Y. Q. Shi. Pairwise prediction-error expansion for efficient reversible data hiding. *IEEE Trans. Image Process.*, 22(12):5010–5021, Dec. 2013.

[51] T. Kalker and F. M. J. Willems. Capacity bounds and constructions for reversible data-hiding. In *Proc. DSP*, volume 1, pages 71–76, 2002.

[52] W. Zhang, B. Chen, and N. Yu. Improving various reversible data hiding schemes via optimal codes for binary covers. *IEEE Trans. Image Process.*, 21(6):2991–3003, Jun. 2012.

[53] X. Zhang. Reversible data hiding with optimal value transfer. *IEEE Trans. multimedia*, 15(2):316–325, Feb. 2013.

[54] S. J. Lin and W. H. Chung. The scalar scheme for reversible information-embedding in gray-scale signals: capacity evaluation and code constructions. *IEEE Trans. Inf. Forens. Security*, 7(4):1155–1167, Aug. 2011.

[55] W. Zhang, X. Hu, X. Li, and N. Yu. Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression. *IEEE Trans. Image Process.*, 22(7):2775–2785, Jul. 2013.

[56] W. Hong and T. S. Chen. A local variance-controlled reversible data hiding method using prediction and histogram-shifting. *J. Syst. Softw.*, 83(12):2653–2663, Dec. 2010.

[57] S. Jung, L. Ha, and S. Ko. A new histogram modification based reversible data hiding algorithm considering the human visual system. *IEEE Signal Process. Lett.*, 18(2):95–98, 2011.

[58] X. Li, J. Li, B. Li, and B. Yang. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Process.*, 93(1):198–205, Jan. 2013.

[59] D. Coltuc and J. M. Chassery. Very fast watermarking by reversible contrast mapping. *IEEE Signal Process. Lett.*, 14(4):255–258, Apr. 2007.

[60] X. Chen, X. Li, B. Yang, and Y. Tang. Reversible image watermarking based on a generalized integer transform. In *Proc. IEEE ICASSP*, pages 2382–2385, 2010.

[61] J. Li, X. Li, and B. Yang. Reversible data hiding scheme for color image based on prediction-error expansion and cross-channel correlation. *Signal Process.*, 93(9):2748–2758, Sep. 2013.