References

[1] A. Piva and S. Katzenbeisser, "Signal processing in the encrypted domain," *EURASIP J. Inf. Security*, vol. 2007, p. 1, 2007, Article 82790.

[2] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. 2000 ACM SIGMOD Int. Conf. Management of Data (SIGMOD'00)*, New York, 2000, vol. 29(2), pp. 439–450.

[3] B. Pinkas, "Cryptographic techniques for privacy-preserving data mining," *SIGKDD Explor. Newsl.*, vol. 4, no. 2, pp. 12–19, 2002.

[4] J. Shashank, P. Kowshik, K. Srinathan, and C. Jawahar, "Private content based image retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8.

[5] J. F. Canny, "Collaborative filtering with privacy," in *Proc. IEEE Symp. Security and Privacy*, 2002, pp. 45–57.

[6] W. Du and M. J. Atallah, "Privacy-preserving cooperative scientific computations," in *Proc. 14th IEEE Computer Security Foundations Workshop*, Nova Scotia, Canada, Jun. 11–13, 2001, pp. 273–282.

[7] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[8] N. Memon and P. Wong, "A buyer-seller watermarking protocol," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 643–649, Apr. 2001.

[9] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms," in *Foundations of Secure Computation*, R. A. De-Millo, Ed. *et al.* New York: Academic, 1978, pp. 169–179.

[10] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.

[11] A. C. Yao, "Protocols for secure computations," in *Proc. 23rd IEEE Symp. Foundations of Computer Science*, Chicago, IL, Nov. 1982, pp. 160–164.

[12] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *Proc. 19th Annu. ACM Conf. Theory of Computing*, New York, 1987, pp. 218–229.

[13] R. Cramer, I. Damgård, and J. B. Nielsen, "Multiparty computation from threshold homomorphic encryption," *Lecture Notes in Computer Science*, vol. 2045, pp. 280–299, 2001.

[14] Z. Erkin, A. Piva, S. Katzenbeisser, R. L. Lagendijk, J. Shokrollahi, G. Neven, and M. Barni, "Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing," *EURASIP J. Inf. Security*, vol. 2007, p. 20, 2007, Article 78943.

[15] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 86–97, Mar. 2009.

[16] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 1999, vol. 1592, pp. 223–238.

[17] M. Chen, O. Boric-Lubecke, and V. M. Lubecke, "$0.5-\mu$m CMOS implementation of analog heart-rate extraction with a robust peak detector," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 4, pp. 690–698, Apr. 2008.

[18] J.-R. T. Pastoriza, S. Katzenbeisser, M. Celik, and A. Lemma, "A secure multidimensional point inclusion protocol," in *ACM Multimedia and Security Workshop (MMSEC'07)*, New York, 2007, pp. 109–120.

[19] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of Paillier's probabilistic public-key system," *Public Key Cryptography*, pp. 119–136, 2001.

[20] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory of Computing (STOC'09)*, New York, 2009, pp. 169–178.

[21] D. Catalano, R. Gennaro, and N. Howgrave-Graham, "The bit security of Paillier's encryption scheme and its applications," in *Proc. Int. Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT'01)*, London, U.K., 2001, pp. 229–243.

[22] B. Schoenmakers and P. Tuyls, "Efficient binary conversion for Paillier encrypted values," in *Advances in Cryptology (EUROCRYPT 2006)*. Berlin, Heidelberg, Germany: Springer, 2006, pp. 522–537.

[23] A. V. Oppenheim and R. W. Schafer*, Digital Signal Processing*. : Prentice-Hall International Inc., 1975.

Reversible Image Watermarking Using Interpolation Technique

Lixin Luo, Zhenyong Chen, Ming Chen, Xiao Zeng, and Zhang Xiong

***Abstract*—Watermarking embeds information into a digital signal like audio, image, or video. Reversible image watermarking can restore the original image without any distortion after the hidden data is extracted. In this paper, we present a novel reversible watermarking scheme using an interpolation technique, which can embed a large amount of covert data into images with imperceptible modification. Different from previous watermarking schemes, we utilize the interpolation-error, the difference between interpolation value and corresponding pixel value, to embed bit "1" or "0" by expanding it additively or leaving it unchanged. Due to the slight modification of pixels, high image quality is preserved. Experimental results also demonstrate that the proposed scheme can provide greater payload capacity and higher image fidelity compared with other state-of-the-art schemes.**

***Index Terms*—Additive interpolation-error expansion, data hiding, interpolation-error, reversible watermarking.**

I. Introduction

Digital watermarking is a kind of data hiding technology. Its basic idea is to embed covert information into a digital signal, like digital audio, image, or video, to trace ownership or protect privacy. Among different kinds of digital watermarking schemes, reversible watermarking has become a research hotspot recently. Compared with traditional watermarking, it can restore the original cover media through the watermark extracting process; thus, reversible watermarking is very useful, especially in applications dictating high fidelity of multimedia content, such as military aerial intelligence gathering, medical records, and management of multimedia information.

Since the earliest reversible watermarking scheme was invented by Barton [1] in 1997, dozens of reversible watermarking methods have been reported in the literature and classified into three categories by Feng *et al.* [2]: reversible watermarking using data compression, reversible watermarking using difference expansion (DE), and reversible watermarking using histogram operation. In these categories, the first kind has complex computation and limited capacity, whereas the latter two are better in both of two criteria.

DE, also known as a kind of integer wavelet transform, was first proposed by Tian [3]. By expanding the difference between the two neighboring pixels of pixel pairs, Tian explored the redundancy in digital images to achieve a high-capacity and low-distortion reversible watermarking. Later on, Alattar [4] extended Tian's scheme by a generalized DE method which hid several bits in the DE of vectors of adjacent pixels. Then, Kim *et al.* [5] proposed a novel scheme devoted to reduce the size of the location map. Furthermore, Lin *et al.* [6] proposed another DE scheme, where the location map was removed completely.

Manuscript received July 16, 2009; accepted October 09, 2009. First published November 06, 2009; current version published February 12, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ton Kalker.

The authors are with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: lixinluo@cse.buaa.end.cn; chzhyong@buaa.edu.cn; mingchen@cse.buaa.edu.cn; zengxiao29@gmail.com; xiongz@buaa.edu.cn).

Digital Object Identifier 10.1109/TIFS.2009.2035975

1556-6013/$26.00 © 2010 IEEE

Recently, Hu *et al.* [7] applied the DE method to prediction-errors and proposed a scheme with an improved overflow map.

Using histogram operation is another effective strategy for reversible watermarking schemes. Vleeschouwer *et al.* [8] randomly divided a group of pixels into two sets. Due to the statistical feature, histograms of the two sets were similar. By circular interpretation of the two histograms, secret messages can be embedded into one of them and can be extracted in a lossless manner. Ni *et al.*'s scheme [9] utilized a zero point and a peak point of a given image histogram to embed messages, where the amount of embedding capacity was the number of pixels with peak point. Then, Hwang *et al.* [10] extended Ni's scheme and applied location map to restore original image without the knowledge of the peak point and zero point. Lin and Hsueh [11] applied the bin exchanging approach to histogram of three-pixel block differences, which provided large pure capacity and achieved low distortion at the same time. In Tsai *et al.*'s article [12], they achieved a high embedding capacity by using a residue image indicating a difference between a basic pixel and each pixel in a nonoverlapping block. In 2009, Kim *et al.* [13] presented an efficient reversible watermarking algorithm where the difference histogram between subsampled images was modified to embed messages.

Additionally, Kalker *et al.* [14] presented theoretical analysis of reversible watermarking and gave a practical code construction, where the host could be memoryless. Later, Maas *et al.* [15] illustrated the detailed implementation of such a code construction for recursive reversible data ~~hiding~~.

In this paper, we propose a ~~reversible~~ watermarking scheme based on additive interpolation-error expansion, which features very low distortion and relatively large capacity. Different from previous watermarking schemes, we utilize an interpolation technique to generate residual values named interpolation-errors and expand them by addition to embed bits. The strategy is efficient since interpolation-errors are good at decorrelating pixels and additive expansion is free of expensive overhead information.

The rest of the paper is organized as follows. The key issues of the proposed watermarking scheme are described in Section II. Then the details of the proposed algorithm including the embedding and extracting processes are elaborated in Section III. Experimental results are given and analyzed in Section IV. Finally, conclusions are drawn in Section V.

II. Additive Interpolation-Error Expansion

In this section, first, a detailed description of additive interpolation-error expansion is given. Following that is the introduction of interpolation-error.

A. Additive Interpolation-Error Expansion

Essentially, the data embedding approach of the proposed reversible watermarking scheme, namely additive interpolation-error expansion, is a kind of DE. But it is different from most DE approaches [3]–[7] in two important aspects:

1) It uses interpolation-~~error, instead~~ of interpixel difference or prediction-error, to embed data.
2) It expands difference~~, which~~ is interpolation-error here, by addition instead of bit-shifting.

First, interpolation values of pixels are calculated using interpolation technique, which works by guessing a pixel value from its surrounding pixels. Then interpolation-errors are obtained via

$$e = x - x' \tag{1}$$

where x' are the interpolation values of pixels x. Let LM and RM denote the corresponding values of the two highest points of interpolation-errors histogram and be formulated as

$$\begin{cases} LM = \arg\max\limits_{e\in E} \mathrm{hist}(e) \\ RM = \arg\max\limits_{e\in E-\{LM\}} \mathrm{hist}(e) \end{cases} \tag{2}$$

where $\mathrm{hist}(e)$ is the number of occurrence when the interpolation-error is equal to e and E denotes the set of interpolation-errors. Without loss of generality, assume $LM < RM$. Then, we divide the interpolation-errors into two parts:

1) Left interpolation-errors (LE): interpolation-error e satisfies $e \le LM$.
2) Right interpolation-errors (RE): interpolation-error e satisfies $e \ge RM$.

The additive interpolation-error expansion is formulated as

$$e' = \begin{cases} e + \mathrm{sign}(e) \times b, & e = LM \text{ or } RM \\ e + \mathrm{sign}(e) \times 1, & e \in (LN, LM) \cup (RM, RN) \\ e, & \text{otherwise} \end{cases} \tag{3}$$

where e' is the expanded interpolation-error, b is the bit to be embedded, and $\mathrm{sign}(.)$ is a sign function defined as

$$\mathrm{sign}(e) = \begin{cases} 1, & e \in RE \\ -1, & e \in LE. \end{cases} \tag{4}$$

In (3), the parameters LN and RN are defined as

$$\begin{cases} LN = \arg\min\limits_{e\in LE} \mathrm{hist}(e) \\ RN = \arg\min\limits_{e\in RE} \mathrm{hist}(e). \end{cases} \tag{5}$$

Usually, LM is a very small integer and in most cases 0, while LN is a smaller integer that with no interpolation-error satisfying $e = LN$. Similarly, in most cases, RM is equal to 1 and RN is a larger integer with no interpolation-error satisfying $e = RN$. After expansion of interpolation-errors, the watermarked pixels x'' become

$$x'' = x' + e'. \tag{6}$$

During the extracting process, with the same interpolation algorithm, we can obtain the same interpolation values x' and the corresponding interpolation-errors via

$$e' = x'' - x'. \tag{7}$$

Note that (7) is the deformation of (6). Once the same LM, LN, RM, and RN are known, embedded data can be extracted through

$$b = \begin{cases} 0, & e' = LM \text{ or } RM \\ 1, & e' = LM - 1 \text{ or } RM + 1. \end{cases} \tag{8}$$

Then the inverse function of additive interpolation-error expansion is applied to recover the original interpolation-errors

$$e = \begin{cases} e' - \mathrm{sign}(e') \times b, & e' \in [LM-1, LM] \cup [RM, RM+1] \\ e' - \mathrm{sign}(e') \times 1, & e' \in [LN, LM-1) \cup (RM+1, RN] \\ e', & \text{otherwise}. \end{cases} \tag{9}$$

Finally, we can restore the original pixels through

$$x = x' + e. \tag{10}$$

Compared with previous DE [3]–[7], the additive interpolation-error expansion is advantageous in three aspects: first, the distortion of additive expansion is smaller since each pixel is altered at most by 1.

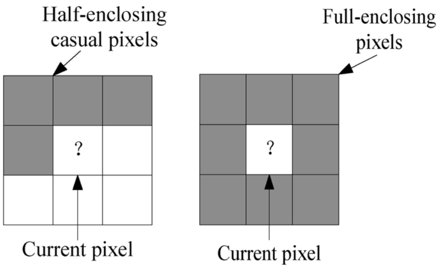


Fig. 1. Interpretations of half-enclosing casual pixels and full-enclosing pixels.

Second, no location map is needed to tell between expanded interpolation-errors and nonexpanded ones since they are distinguishable with LM, LN, RM, and RN. Last, interpolation-errors are more expandable than interpixel differences or prediction-errors, which will be explained in the following section.

B. Interpolation-Error

Different from the recent conventional schemes, we exploit interpolation-error, the difference between pixel value and its interpolation value, to embed data. There are several reasons that lift interpolation-error to be a better alternative to interpixel difference or prediction-error.

The reasons for using interpolation-error instead of interpixel difference in the proposed scheme can be summarized as two points. The first one is that interpolation-error requires no blocking which can significantly reduce the amount of differences and in turn lessen the potential embedding capacity. For instance, in Tian's method, the number of differences is only a half of the number of total pixels. And for a pixel vector or a pixel block of X pixels, as presented in [4], the number of differences is $(X-1)/X$ of the number of total pixels, which means that $1/X$ pixels are spent to find differences. The cost is considerably expensive because X cannot be large. But in our scheme, we almost get all pixels as candidates to embed messages except some negligible marginal pixels, which promises a larger embedding capacity. The second one is that interpolation-error exploits the correlation between pixels more significantly. Fundamentally, the feasibility of reversible image watermarking is due to high interpixel redundancy or interpixel correlation existing in practical images. To maximize the capacity of image watermarking, we should exploit the correlation of pixels to the greatest extent. The proposed scheme utilizes the full-enclosing pixels to interpolate the target pixel, so the interpolation-error tends to be smaller, which means that we can obtain a higher capacity.

Compared with prediction-error, our method also has several advantages. First, referring to Fig. 1, we select the full-enclosing pixels that can be before and after the current pixel, not the half-enclosing casual pixels that must be before the current pixel to estimate the target pixel, which exploits the correlation between pixels more significantly. In addition, because of the diversity of images, it is difficult to find an appropriate predictor in the prediction-error schemes, and the complexity of prediction depends on the characteristics of the image, but our method only samples pixels from the original image and utilizes a simple interpolation algorithm, which is another important reason for preferring interpolation-error to prediction-error.

Next, we offer a feasible image interpolation algorithm to obtain interpolation values and interpolation-errors. Image interpolation is the process of producing a high-resolution image from its low-resolution counterpart. It has applications in medical imaging, remote sensing, and digital photographs. For better understanding, we adopt a concrete interpolation algorithm, which is Zhang *et al.*'s [16] simplified method, to explain the interpolation-error. However, the proposed reversible watermarking scheme does not rely on the specific interpolation algorithm.

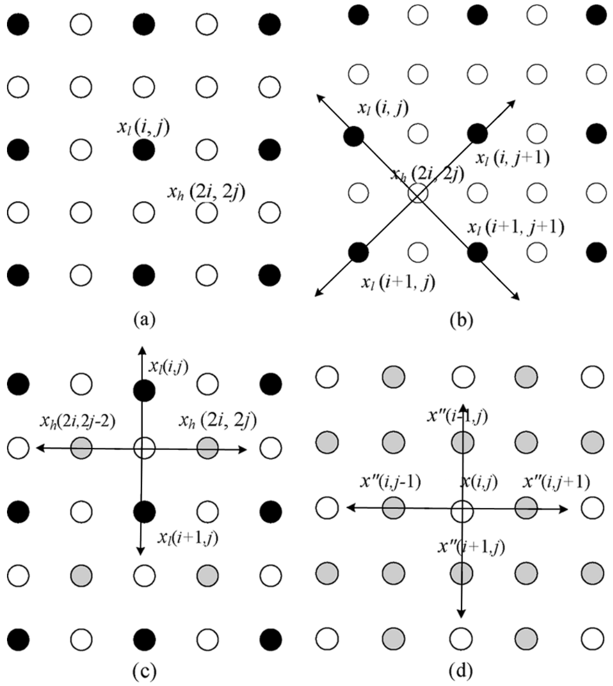


Fig. 2. (a) Formation of a low-resolution image x_l from the high-resolution image x_h; (b), (c) interpolation of residual samples of high-resolution; (d) the interpolation of the sample pixels.

Assume a low-resolution image x_l is directly down-sampled from an associated high-resolution x_h through $x_l(i,j) = x_h(2i-1, 2j-1)$, $1 \le i \le N$, $1 \le j \le M$. Referring to Fig. 2(a), the black dots represent the pixels of x_l and the white dots represent the missing pixels of x_h. The interpolation aims to estimate the missing pixels in high-resolution x_h, whose size is $2N \times 2M$, from the pixels in low-resolution x_l, whose size is $N \times M$.

The key issue of interpolation is how to infer and utilize the correlation between the missing pixels and the neighboring pixels. With the interpolation algorithm under discussion, we partition the neighboring pixels of each missing pixel into two directional subsets that are orthogonal to each other. For each subset, a directional interpolation is made, and then we fuse the two interpolated values with an optimal pair of weights to estimate x_h. We reconstruct the high-resolution x_h in two steps. First, those missing pixels $x_h(2i,2j)$ at the center locations surrounded by four low-resolution pixels are interpolated. Second, the other missing pixels $x_h(2i-1,2j)$ and $x_h(2i,2j-1)$ are interpolated with the help of the already recovered pixels $x_h(2i,2j)$.

Then, we discuss the details of the first step. Referring to Fig. 2(b), we can interpolate the missing high-resolution pixel $x_h(2i,2j)$ along two orthogonal directions: 45° diagonal and 135° diagonal. Denoted by $x'_{45}(2i,2j)$ and $x'_{135}(2i,2j)$, the two directional interpolation results are computed as

$$\begin{cases} x'_{45} = \left(x_l(i,j+1) + x_l(i+1,j)\right)/2 \\ x'_{135} = \left(x_l(i,j) + x_l(i+1,j+1)\right)/2. \end{cases} \tag{11}$$

Here, we take e_{45} and e_{135} to represent the interpolation errors in the corresponding direction. Let

$$\begin{cases} e_{45}(2i,2j) = x'_{45}(2i,2j) - x_h(2i,2j) \\ e_{135}(2i,2j) = x'_{135}(2i,2j) - x_h(2i,2j). \end{cases} \tag{12}$$

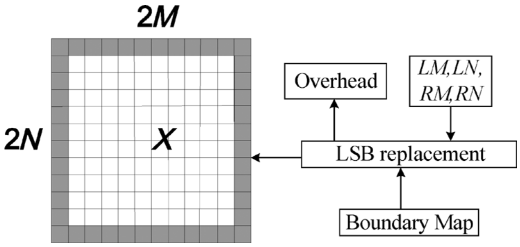


Fig. 3. LSB replacement of the overhead information. The gray part represents the marginal area of cover-image.

Instead of computing the linear minimum mean square-error estimation (LMMSE) estimate of x_h, we select an optimal pair of weights to make x'_h a good estimate of x_h. The strategy of weighted average leads to significant reduction in complexity. Let

$$\begin{cases} x'_h = w_{45} \bullet x'_{45} + w_{135} \bullet x'_{135} \\ w_{45} + w_{135} = 1. \end{cases} \tag{13}$$

The weights w_{45} and w_{135} are determined to minimize the mean square-error of x_h

$$\{w_{45}, w_{135}\} = \arg \min_{w_{45}+w_{135}=1} E\left[\left(x'_h - x_h\right)^2\right]. \tag{14}$$

According to abundant experiments, the correlation coefficient between e_{45} and e_{135}, which influences the correlation between x'_{45} and x'_{135} from (12), hardly changes the PSNR value and visual quality of the interpolation image. So we can show the weights are

$$w_{45} = \frac{\sigma(e_{135})}{\sigma(e_{45}) + \sigma(e_{135})}, \quad w_{135} = 1 - w_{45} \tag{15}$$

where $\sigma(e_{45}), \sigma(e_{135})$ are the variance estimations of e_{45} and e_{135}, respectively. They are related to the mean value of $x_h(2i, 2j)$ that we will discuss. From (15), we can see intuitively how the weighting method works. For instance, for an edge in or near the 45° diagonal direction, $\sigma(e_{135})$ is higher than $\sigma(e_{45})$ so that w_{135} will be less than w_{45}; consequently, x'_{135} has less influence on x'_h than x'_{45}, and vice versa.

Referring to Fig. 2(b), the mean value of $x_h(2i, 2j)$, denoted by u, is estimated by the available low-resolution pixels around $x_h(2i, 2j)$. To balance the complexity of computation and the consistency of pixels, we compute u as

$$u = \frac{(x_l(i,j+1)+x_l(i+1,j))}{4} + \frac{(x_l(i,j)+x_l(i+1,j+1))}{4}. \tag{16}$$

Then we return to the computation of $\sigma(e_{45})$ and $\sigma(e_{135})$, which are the variance estimations of interpolation errors in the corresponding directions. They are computed as

$$\begin{cases} \sigma(e_{45}) = \frac{1}{3} \sum_{k=1}^{3} \left(S_{45}(k) - u\right)^2 \\ \sigma(e_{135}) = \frac{1}{3} \sum_{k=1}^{3} \left(S_{135}(k) - u\right)^2 \end{cases} \tag{17}$$

where the two sets are

$$\begin{cases} S_{45} = \{x_l(i, j+1), x'_{45}, x_l(i+1, j)\} \\ S_{135} = \{x_l(i, j), x'_{135}, x_l(i+1, j+1)\}. \end{cases} \tag{18}$$

We can use (11)–(18) to obtain the estimations of the missing high-resolution pixels $x_h(2i, 2j)$ and finish the first step.

After the missing high-resolution pixels $x_h(2i, 2j)$ are estimated, the residual pixels $x_h(2i-1, 2j)$ and $x_h(2i, 2j-1)$ can be estimated similarly. Referring to Fig. 2(c), the black dots represent the low-resolution pixels, the gray dots represent the estimated pixels in the first step, and the white dots represent the pixels that are to be interpolated. As illustrated in Fig. 2(c), the remainders are computed in a similar way as described in the first step, except the two directions are modified to 0° and 90°. Finally, the whole high-resolution is reconstructed through the above process.

Here, we can obtain the interpolation-errors just through calculating the difference between interpolation values and original values. However, as illustrated in Fig. 2(a), a quarter of pixels, which are exploited to reconstruct the image, cannot be utilized to embed the data. Consequently, the capacity of the algorithm is constrained, which is a problem we must deal with. To clarify this problem, we give some definitions that will be used.

1) Sample pixels: the pixels in the original image which are sampled to form the low-resolution image.
2) Nonsample pixels: the pixels in the original image except sample pixels.

In (3), each pixel is modified at most by 1 through additive interpolation-error expansion, thus high image quality is preserved and the watermarked nonsample pixels can be utilized to interpolate the sample pixels. Referring to Fig. 2(d), the gray dots represent the watermarked nonsample pixels and the white dots represent the sample pixels. As illustrated in Fig. 2(d), the sample pixels are interpolated along two orthogonal directions: 0° and 90° and the corresponding interpolation-errors are easily obtained by the interpolation algorithm.

III. Algorithm of the Proposed Scheme

This section presents the implementation details of the proposed reversible watermarking scheme.

A. *Overhead Information*

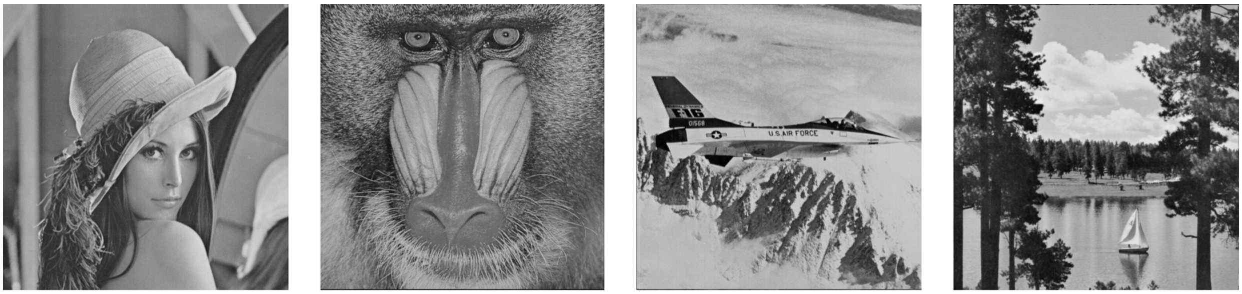
After secret messages are embedded, some overhead information is needed to extract the covert information and restore the original image. Generally, the overhead information contains the following:

1) the information to identify those pixels containing embedded bits;
2) the information to solve the overflow/underflow problem.

In our proposed scheme, we use four keys, namely LM, LN, RM, and RN in (2) and (5), to identify the pixels containing embedded bits, and exploit a boundary map, to record information on solving the overflow/underflow problem.

Since overflow/underflow happens when pixels are changed from 255 to 256 or from 0 to -1 (boundary pixels), we apply additive interpolation-error expansion only to pixels valued from 1 to 254. However, ambiguities arise when nonboundary pixels are changed from 1 to 0 or from 254 to 255 (pseudoboundary pixels) during the embedding process. When a boundary pixel is encountered during the extracting process, it is originally either a boundary pixel or a pseudoboundary pixel. Therefore, to find the original boundary pixels, we only need to tell whether boundary pixels in the watermarked image are genuine or pseudo. A boundary map is the right judge to distinguish between genuine and pseudo. It is a binary array with its every element corresponding to a boundary pixel in the watermarked image, 0 for genuine and 1 for pseudo.

It is verified through abundant experiments that the boundary map is very short and needs no compression. Take the eight most popular test images for example (*Lena, Baboon, Plane, Sailboat, Peppers, Barbara, Boat, Tiffany*), the largest size is merely 16 for single-layer embedding. So the marginal area of the cover-image, illustrated in Fig. 3, is enough to accommodate it using least significant bit (LSB) replacement. The LSB replacement is not a problem only if we record the original LSB bit-plane of the marginal area at the head of the payload as overhead and place it back once it is extracted during the extracting process. Referring to Fig. 3, the LM, LN, RM, and RN in (2) and (5) can be embedded in a similar way.



(a) (b) (c) (d)

Fig. 4. Watermarked versions of test images. (a) Lena (40.69 dB with 0.59 bpp); (b) Baboon (42.92 dB with 0.16 bpp); (c) Plane (40.31 dB with 0.65 bpp); (d) Sailboat (43.17 dB with 0.26 bpp).

B. Embedding Process

The proposed scheme is mainly composed of two parts for the embedding of the watermark: interpolation and embedding. In the interpolation process, we estimate the interpolated values with the above-mentioned algorithm and calculate the interpolation-errors in the raster scan order. In the embedding process, we apply additive expansion to interpolation-errors and embed the watermark information. The detailed description of the embedding process is given as follows.

1) Record some original LSB bits of the marginal area as overhead and add "0" to the beginning of boundary map B as a label. Then, assemble overhead and watermark information to form payload W.
2) Using (1), calculate interpolation-errors e of the nonsample pixels as discussed in Section II-B.
3) Work out the frequency of every interpolation-error and find out LM, LN, RM, and RN. Next, scan the cover-image from the beginning and start to undertake the embedding operation.
4) If $x \in \{0, 255\}$, put a "0" into the boundary map B and move to the next one. Else, expand e through additive expansion and work out the watermarked pixel x''. If $x'' \in \{0, 255\}$, put a "1" into the boundary map B.
5) For convenience, let C_1 denote the condition when W is not completely embedded, and C_2 denote the condition when the current pixel is not the end of nonsample pixels. If C_1 and C_2 are both satisfied, go to Step 4). If C_1 is satisfied but C_2 is not satisfied, record the length of the boundary map B (denoted by L) and replace the header of B with "1", Then, calculate the interpolation-errors of the sample pixels and go to Step 3).
6) Embed B, L, LMs, LNs, RMs, and RNs into marginal area of the cover-image using LSB replacement.

For intuition and pellucid, we only consider the single-layer embedding in the above-mentioned steps. But we have also implemented multilayer embedding and undertaken relevant experiments, which will be discussed in Section IV. In addition, we can append an end-token to watermark data to indicate where the data hiding ends in the image, and set a label to determine whether to embed watermark data.

C. Extracting Process

The corresponding extracting process is described as follows.

1) Obtain LMs, LNs, RMs, RNs, L, and the boundary map B from the LSB of marginal area of the watermarked image. Next, scan the watermarked image and undertake the following steps.
2) Extract the first bit of the boundary map B, if it is equal to 0, go to Step 5).
3) Using (1), work out the expanded interpolation-errors e' of the watermarked sample pixels x_2 as discussed in Section II-B.
4) If $x_2 \in [1, 254]$, recover the interpolation-error e through inverse additive expansion and put the extracted bit into W_2. Else, $x_2 \in \{0, 255\}$, remove the Lth bit b from B, here, if b is equal to 0, move to the next one, else process like $x_2 \in [1, 254]$. Do this step until the latter part of payload is extracted.
5) Using (1), calculate the expanded interpolation-errors e' of the watermarked nonsample pixels x_1.
6) If $x_1 \in [1, 254]$, recover the interpolation-error e using (9) and put the extracted bit into W_1, else remove the second bit from B, here, if it is equal to 0, move to the next one, else operate like $x_1 \in [1, 254]$.
7) Decode overhead information and restore the pixels in marginal area once their LSBs are extracted.
8) Go to Step 6) if the former part of payload is not completely extracted.
9) Merge the bits in W_1 and W_2 to form the watermark information.

IV. Experimental Results

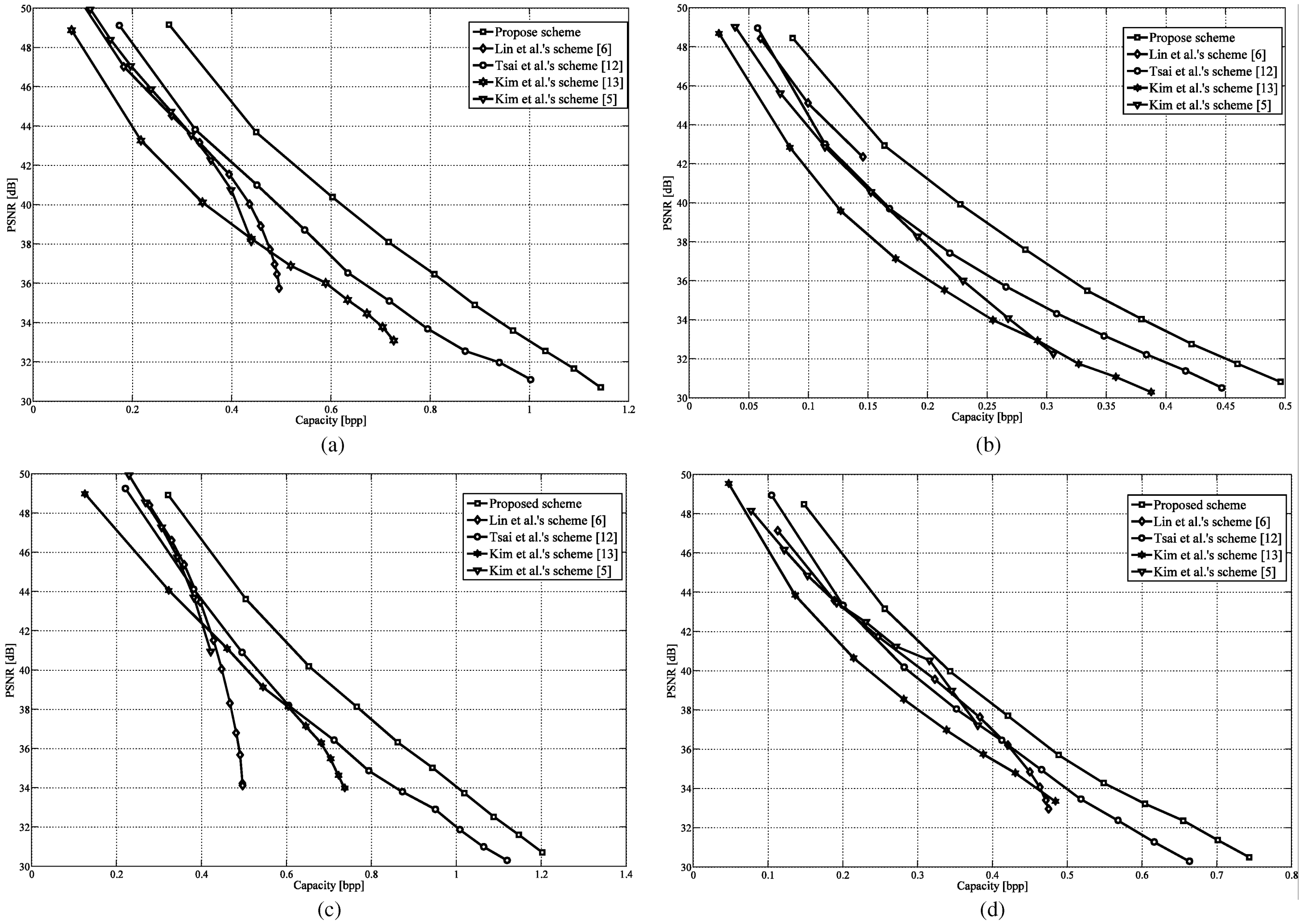
We have implemented the proposed reversible watermarking scheme using MATLAB, and have successfully applied it to standard test images varying from complex (Baboon) to smooth (Plane). Equality between the original images and restored images has proved the reversibility of the proposed scheme in experiments. In our experiments, we take a random bit string as the watermark message and adopt peak signal-to-noise ratio (PSNR) value and bit number (or bpp) as measurements of image quality and embedding capacity, respectively. The watermarked versions of test images (sized 512 × 512, 8-bit grayscale) are placed as examples in Fig. 4, where the visual qualities of them are satisfactory.

For single-layer embedding, since the proposed scheme alters each pixel at most by 1, the PSNR value of the watermarked image is guaranteed to be larger than $10 \times log_{10} \, (255 \times 255) = 48.13$ dB. Table I summarizes comparison results with other conventional schemes [7], [9]–[11] for four test images: *Lena*, *Baboon*, *Plane*, and *Sailboat*. Schemes [9]–[11] achieve reversible watermarking through histogram-shifting, whereas scheme [7] is based on bit-shifting prediction-error expansion and its capacity is controlled by the threshold T. As shown in Table I, the single-layer embedding of the proposed scheme outperforms other algorithms and its capacity achieves about from 6% to 1213% performance enhancement for the test images.

Since it is completely lossless, our scheme can easily achieve higher capacities through multilayer embedding when larger payloads are required. For better evaluation of the performance of multilayer embedding, we compare our results with other reversible watermarking algorithms which are proposed by Kim *et al.* [5], Lin *et al.* [6], Tsai *et al.* [12], and Kim *et al.* [13], respectively, for the *Lena*, *Baboon*, *Plane*, and *Sailboat*. In Fig. 5, for all four images, the top curve is the proposed scheme, and at the same PSNR value, the embedding capacity of the

TABLE I
COMPARISON RESULTS IN TERMS OF THE CAPACITY (bits) AND THE PSNR VALUE (dB) FOR LENA, BABOON, PLANE, AND SAILBOAT

Schemes	Lena image		Baboon image		Plane image		Sailboat image	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Ni et al.[9]	5,460	48.20	5,421	48.20	16,171	48.20	7,301	48.20
Hwang et al.[10]	5,336	48.22	5,208	48.20	15,300	48.40	7,051	48.25
Lin et al.[11]	59,900	46.60	19,130	47.61	80,006	47.99	37,644	47.60
Hu et al.[7]	60,241	48.69	21,411	48.34	77,254	48.86	28,259	48.40
Proposed	71,674	48.82	22,696	48.36	84,050	48.94	38,734	48.50



(a)

(b)

(c)

(d)

Fig. 5. Performance evaluation of multilayer embedding over standard test images. (a) Lena; (b) Baboon; (c) Plane; (d) Sailboat.

proposed scheme is about 0.15–0.3 bpp higher than those of Kim *et al.*'s [5] and Lin *et al.*'s [6], which are based on interpixel DE. For Tsai *et al.*'s scheme [12] which achieves large capacities through multilayer embedding, its PSNR value is about 1.5–3.5 dB lower than that of the proposed scheme when the same amount of payloads are embedded. Fig. 5 also dedicates that the proposed method achieves higher embedding capacity with lower image distortion than Kim *et al.*'s [13], especially for *Baboon* which represents images with large areas of complex texture.

In addition, the experimental results of single-layer and multilayer embedding also verify the interpretation of the superiority of interpolation-error to prediction-error or interpixel difference, which is described in Section II-B.

V. CONCLUSION

In this paper, a novel reversible watermarking scheme has been presented. Different from the latest schemes using prediction or wavelet techniques, the proposed scheme uses an interpolation technique to generate residual values named interpolation-errors, which are demonstrated to be of greater decorrelation ability. By applying additive expansion to these interpolation-errors, we achieve a highly efficient reversible watermarking scheme, which can guarantee high image quality without sacrificing embedding capacity.

According to the experimental results, the proposed reversible scheme provides a higher capacity and achieves better image quality for watermarked images. In addition, the computational cost of the proposed scheme is small.

REFERENCES

[1] J. M. Barton, "Method and Apparatus for Embedding Authentication Information Within Digital Data," U.S. Patent 5 646 997, 1997.

[2] J. B. Feng, I. C. Lin, C. S. Tsai, and Y. P. Chu, "Reversible watermarking: Current status and key issues," *Int. J. Netw. Security*, vol. 12, no. 3, pp. 161–171, 2006.

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

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

[5] 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. Forensic Security*, vol. 3, no. 3, pp. 456–465, Sep. 2008.

[6] C. C. Lin, S. P. Yang, and N. L. Hsueh, "Lossless data hiding based on difference expansion without a location map," in *2008 Congress on Image and Signal Processing*, 2008, pp. 8–12.

[7] Y. Hu, H.-K. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 250–260, Feb. 2009.

[8] C. D. Vleeschouwer, J. F. Delaigle, and B. Macq, "Circular interpretation of bijective transformation in lossless watermarking for media as management," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 97–105, Mar. 2003.

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

[10] J. Hwang, J. W. Kim, and J. U. Choi, "A reversible watermarking based on histogram shifting," in *Int. Workshop on Digital Watermarking, Lecture Notes in Computer Science*, Jeju Island, Korea, 2006, vol. 4283, pp. 348–361, Springer-Verlag.

[11] C. C. Lin and N. L. Hsueh, "A lossless data hiding scheme based on three-pixel block differences," *Pattern Recogni.*, vol. 41, no. 4, pp. 1415–1425, Apr. 2008.

[12] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Process.*, vol. 89, pp. 1129–1143, 2009.

[13] K.-S. Kim *et al.*, "Reversible data hiding exploiting spatial correlation between sub-sampled images," *Pattern Recognit.*, 2009, DOI: 10.1016/j.patcog.2009.04.004.

[14] T. Kalker and F. M. J. Willems, "Capacity bounds and construction for reversible data-hiding," in *Proc. Int. Conf. DSP*, Santorini, Greece, 2002.

[15] D. Maas, T. Kalker, and F. M. J. Willems, "A code construction for recursive reversible data-hiding," in *Pro. ACM Workshop Multimedia*, Juan-les-Pins, France, Dec. 2002, pp. 15–18.

[16] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.