Reversible Image Hiding for High Image Quality based on Histogram Shifting and Local Complexity

Chin-Chen Chang[1,2], Thai-Son Nguyen[1,4], and Chia-Chen Lin[3]
(Corresponding author: Chia-Chen Lin)

Department of Information Engineering and Computer Science, Feng Chia University[1]
No. 100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan, R.O.C
Department of Biomedical Imaging and Radiological Science, China Medical University[2]
No.91 Hsueh-Shih Rd., Taichung 40402, Taiwan, R.O.C
Department of Computer Science and Information Management, Providence University [3]
No. 200 Chung-chi Rd., Taichung 43301, Taiwan, R. O. C
Department of Information Technology, Tra Vinh University [4]
No. 126 National Road 53, Ward 5, Tra Vinh city, Tra Vinh Province, Vietnam
(Email: mhlin3@pu.edu.tw)

(Received Jan. 20, 2013; revised and accepted May 16, 2013)

Abstract

Reversible data hiding is a technique that can restore the original cover image without any distortion after the embedded secret message is extracted. In this paper, a novel reversible data hiding method is proposed in the spatial domain. The proposed scheme is based on a local complexity function to classify a cover image block into the smooth blocks or complex blocks to increase embedding capacity and enhance image quality. Then, the complex blocks are preserved during the embedding process so that a high visual quality is obtained. The average *PSNR* of the stego-image was 49.83 dB for single-level embedding. For pixels in the smooth block, a new scan path was designed to compute the difference values for data embedding while using histogram shifting. The experimental results demonstrated that our scheme provided larger embedding capacity and better image quality than some existing schemes. In addition, the proposed scheme also obtained excellent performance in terms of embedding capacity and image quality for the Kodak test images.

Keywords: *Data hiding, high image quality, histogram shifting, local complexity, reversible*

1 Introduction

With the rapid development of various technologies, the Internet has become an increasingly important part of the lives of most people. Rather than running the software on a desktop computer or server, Internet users are now able to use the "Cloud" service [8, 12], also referred as cloud computing, which is a network collection of servers, storage systems, and devices. It is used to combine software, data, and computing power in multiple locations through the network. There are many cloud services provided by various service providers on the Internet. Each cloud service will exchange digital data with other cloud services. Unfortunately, the digital data that are exchanged between these cloud services can be intercepted by malicious users of the Internet. Therefore, the development of techniques for ensuring the secrecy and security of data during exchange has become a top priority. Thus, researchers have introduced various algorithms, such as encryption [7] and data hiding [1, 3, 4, 9, 15, 18, 19, 25, 29, 30]. In encryption, data are transformed into meaningless, undecipherable form. Then, only an authorized user can recover the original form of these meaningless data by using a secret key that is supplied in advance through the security channel. However, the secret data in the meaningless form are obvious to malicious users, and they may try to decrypt the messages to get information. To avoid unnecessarily attracting the attention of malicious users, data hiding, also called information hiding, is a technique that aims to conceal the secret information into digital data, i.e., video, images, text, and audio [3, 15, 18, 25 29, 30], as well as software, natural language text, and relational databases [1, 4, 9].

Most of the proposed data hiding schemes are based on irreversible data hiding [6, 13, 20]. In such schemes, the cover image is distorted permanently and cannot be restored correctly after the secret data have been extracted from the stego-image. However, in some fields, e.g., military, medical, legal investigation, and certain satellite applications, it is essential to be able to reconstruct the original cover image without any distortion after the secret information has been extracted. As a result, various reversible data hiding techniques have been proposed to

deal with this issue [5, 10, 16, 17, 19, 21, 24, 26, 27, 28]. These schemes allow receivers to extract the secret information and then reconstruct the cover image without any distortion.

Most reversible data hiding schemes for the spatial domain are based on difference expansion (DE) [2, 17, 19, 24, 26, 27] and histogram shifting [10, 14, 16, 19, 21, 25, 28-30]. In general, the schemes that are based on DE provide higher embedding capacity, whereas the schemes based on histogram shifting offer better visual quality of the stego-image. In the past five years, some reversible data hiding schemes based on histogram shifting have been developed. The first well-known reversible data hiding scheme based on histogram shifting was proposed by Ni et al. [21] in 2006. The basic idea of their scheme is to shift each pixel only one grayscale value after data embedding so that the visual quality of the stego-image can be retained. Ni et al.'s scheme provides a much higher image quality than Tian's reversible data hiding scheme [27], but it does so at the cost of limited embedding capacity. To further improve embedding capacity, in 2009, Tsai et al. [28] divided the image into blocks and modified the differences between the reference pixel and other pixels in each block to hide secret data. In the same year, Kim et al. [14] proposed a reversible data hiding scheme based on difference histogram shifting. They partitioned a cover image into several sub-images and used the correlation among these sub-images to embed more secret data. To further improve Kim et al.'s scheme, in 2010, Li et al. [16] presented a new, reversible data hiding scheme based on the differences of adjacent pixels. In this scheme, the difference between adjacent pixels was used to conceal the secret data. In 2011, Hong and Chen [10] combined the interpolation technique and a reference pixel distribution mechanism to enhance image quality instead of using Li et al.'s technique. Later, Zhao et al. [30] proposed a new reversible data hiding scheme based on histogram shifting and sequential recovery. The principle of their scheme was to shift the histogram constructed from the neighboring difference instead of shifting the histogram of the cover image. To improve Kim et al.'s scheme [14] in terms of embedding capacity and visual quality of the stego-image, Luo et al. [19] selected the median pixel of each block to structure the reference sub-image. However, since the reference pixels or reference sub-images were not used for embedding data, the embedding capacity of their scheme was limited in that its average embedding capacity was less than 0.1 bpp for a single embedding level. In 2013, Wang et al. introduced a new, reversible data hiding scheme [29] that classified all pixels into wall pixels and non-wall pixels. For a wall pixel, the interpolation error was used to embed secret data over the interpolation prediction algorithm. In addition, the direction order was proposed to compute the difference values of non-wall pixels for histogram shifting to obtain better embedding capacity while maintaining the good visual quality of the stego-image.

In this paper, a novel, reversible data hiding scheme based on histogram shifting and local complexity is presented to further improve embedding capacity and the quality of the stego-image. In the proposed scheme, the local complexity of the current block is calculated based on the reference pixels of adjacent blocks, and it is used to determine an image block into a smooth or a complex block. Then, the complex blocks are preserved during the embedding process so that a high visual quality is obtained. A new scan path for calculating the difference value in the smooth blocks also was designed in order to obtain a higher frequency of the peak histogram bins for the smooth blocks. The experimental results confirmed that our proposed scheme offers better embedding capacity than the previous schemes, while maintaining the excellent quality of the visual image.

The rest of this paper is organized as follows. Section 2 reviews Zhao et al.'s reversible data hiding scheme [30]. Section 3 provides an extensive description of our scheme, including the phases in which data are embedded and extracted. Our experimental results are presented and discussed in Section 4, and our conclusions are presented in Section 5.

2 Related Work

In 2011, Zhao et al. [30] proposed a new, reversible data hiding scheme based on multi-level histogram shifting and sequential recovery to obtain a larger embedding capacity. In their scheme, the inverse "S" order, as shown in Figure 1, the scan path was used to examine a cover image while computing the difference of adjacent pixels. In the data embedding phase, a multi-level histogram shifting strategy was used, and the parameter L denoted the embedding level that was used to control the embedding capacity. The embedding capacity was directly proportional to the value of L. The details of Zhao et al.'s data embedding phase are presented below.

Data embedding algorithm:

Input: Grayscale cover image I sized of $M \times N$ and secret data W, where M and N are the height and width, respectively, of cover image I.

Output: Stego-image I' with a size of $M \times N$.

Step 1: Inverse "S" scans a cover image I into a pixel sequence P (p_1, p_2, ..., $p_{M\times N}$).

Step 2: Compute the differences d_i according to Equation (1), and generate a histogram based on d_i ($2 \le i \le M\times N$). Assume that the histogram bins from left to right are denoted by b(-255), b(-254),..., b(-1), b(0), b(1),..., b(254), b(255), sequentially, as shown in Figure 2.

$$d_i = \begin{cases} p_1 & if \ \ i = 1 \\ p_{i-1} - p_i & if \ \ 2 \le i \le M \times N \end{cases} \quad (1)$$

Step 3: Select embedding level L. If $L = 0$, go to Step 4. If $L > 0$, go to Step 5.

Step 4: Data embedding for level $L = 0$.

Step 4.1: Shift the right bins of b(0) to the right side by 1

using Equation (2):

$$d_i' = \begin{cases} p_1 & if\ i = 1 \\ d_i & if\ d_i \le 0\ and\ 2 \le i \le M \times N \\ d_i + 1 & if\ d_i > 0\ and\ 2 \le i \le M \times N \end{cases} \quad (2)$$

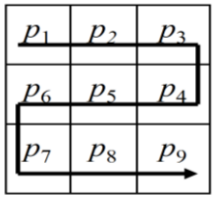


Figure 1: Example of inverse "S" scan path of a 3 × 3 image block

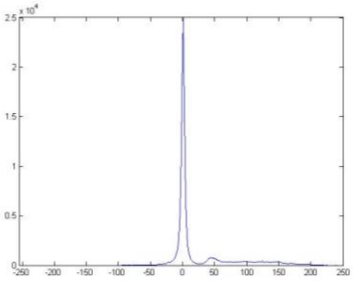


Figure 2: Difference histogram of the "Lena" image

Step 4.2: Scan the difference value d'_i ($2 \le i \le M \times N$) one by one. For every difference value that equals 0, one secret bit w is embedded. If the secret bit w = 0, the difference value d'_i remains unchanged. If the current secret bit w = 1, 1 is added to the difference value d'_i using Equation (3).

$$d_i'' = \begin{cases} p_1 & if\ i = 1 \\ d_i' + w & if\ d'_i = 0\ and\ 2 \le i \le M \times N \\ d_i' & if\ d'_i \ne 0\ and\ 2 \le i \le M \times N \end{cases} \quad (3)$$

Step 5: Data embedding for level $L > 0$.

Step 5.1: Shift the right bins of $b(L)$ to the right side by L+1, and shift the left bin of $b(-L)$ to the left side by L using Equation (4).

$$d_i'' = \begin{cases} p_1 & if\ i = 1 \\ d_i' + w & if\ d'_i = 0\ and\ 2 \le i \le M \times N \\ d_i' & if\ d'_i \ne 0\ and\ 2 \le i \le M \times N \end{cases} \quad (4)$$

Step 5.2: Scan d_i' in the range [-L, L] one by one. The multi-level data embedding strategy is presented as follows:

Step 5.2.1: Embed secret bit w by utilizing Equation (5)

$$d_i'' = \begin{cases} p_1 & if\ i = 1 \\ d_i' & if\ -L \le d_i' \le L\ and\ 2 \le i \le M \times N \\ 2 \times L + w & if\ d_i' = L\ and\ 2 \le i \le M \times N \\ -2 \times L - w + 1 & if\ d_i' = -L\ and\ 2 \le i \le M \times N \end{cases} \quad (5)$$

Step 5.2.2: Decrease L by 1.

Step 5.2.3: If $L \ne 0$, repeat Steps 5.2.1 and 5.2.2. If L = 0, perform Equation (6), and go to Step 6.

$$d_i'' = \begin{cases} p_1 & if\ i = 1 \\ d_i' + w & if\ d_i' = 0\ and\ 2 \le i \le M \times N \\ d' & if\ d_i' \ne 0\ and\ 2 \le i \le M \times N \end{cases} \quad (6)$$

Step 6: Generate the stego-pixel sequence P' by using Equation (7).

$$p_i' = \begin{cases} p_1 & if\ i = 1 \\ p_{i-1} - d_i'' & if\ 2 \le i \le M \times N \end{cases} \quad (7)$$

Step 7: Rearrange the stego-pixel sequence P' to generate the stego-image I'.

3 Proposed Scheme

The proposed scheme also uses the histogram-shifting technique. To enhance the embedding capacity while maintaining acceptable quality of the stego-image, the blocks are classified as either complex or smooth. The complex blocks are not used for embedding secret data to guarantee the quality of the stego-images. This is because the complex blocks usually provide less embedding capacity than the smooth blocks. However, they generate more distortion than smooth blocks when carrying secret data. A new scan path was designed for the smooth blocks to increase their embedding capacity. The details of the proposed scheme are described in the following sections.

3.1 Local Complexity Measurement and Block Classification

In the proposed scheme, the cover image I with the size of $M \times N$, where M and N are the height and width of the cover image, respectively, is divided into non-overlapping blocks with sizes of 3 × 3 pixels. For each block, the center pixel C_i is selected as the reference pixel, and other eight pixels are referred to as non-reference pixels. For the non-border block with the reference pixel C_i, there are four other reference pixels, i.e., C_i^l, C_i^r, C_i^u, and C_i^d, which are located directly left, right, up above, and down below pixel C_i, respectively. These four reference pixels are defined as the satellite reference pixels of the reference pixel C_i. Figure 3 gives an example of an image block with one reference pixel C_i and its four corresponding satellite reference pixels.

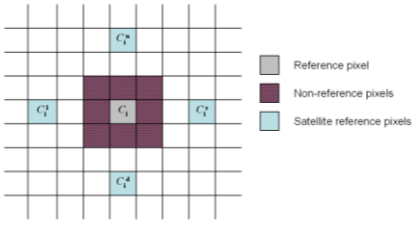


Figure 3: Example of an image block with one reference pixel C_i and its four corresponding satellite reference pixels

Regions	Pixels	Embedding capacity (bpp)
Smooth	166,815	0.241
Complex	93,285	0.071

Figure 4: Distribution of smooth and complex regions of the "Lena" image

For a given block, based on its satellite reference pixels, its local complexity, which is calculated by Equation (8), is used to evaluate whether the pixels are located in a smooth or a complex region.

$$LC_i = Range(C_i^l, C_i^r, C_i^u, C_i^d) \quad (8)$$

where the range function $Range(C_i^l, C_i^r, C_i^u, C_i^d)$ is defined as the absolute difference between the maximum and minimum values of the four satellite reference pixels, i.e., C_i^l, C_i^r, C_i^u, and C_i^d, of the current processing block. If the complexity value is less than the threshold T, the current processing block likely is located in the smooth region. In this case, we obtain one or more pixels that can be used for carrying secret bits. Figure 4 shows the distribution of smooth regions and complex regions of the "Lena" image when we use threshold $T = 20$.

Figure 4 shows that the embedding capacity by using histogram shifting for smooth regions is significantly greater than the embedding capacity in the complex regions. Based on this property, to obtain higher embedding capacity while maintaining the quality of the stego-image, we use the local complexity measurement as shown in Equation (8) to classify image blocks into complex blocks or smooth blocks. For the complex blocks, all the pixels will be marked as reference pixels to prevent them from being modified in the embedding process.

3.2 Data Embedding Phase

During the data embedding phase, the histogram shifting technique also is used in the proposed scheme; therefore, the pixel values might be shifted by 1. Pixels that have values of 0 and 255 might be changed to -1 and 256 after data embedding, meaning that the overflow/underflow problem has occurred. To avoid this problem, the positions of the pixels that have values of 0 and 255 are recorded into location map L with same strategy mentioned in [11], and, in advance, the pixel values are changed to 1 and 254, respectively. In the data extracting phase, the extra information, including location map L and side information, should be reconstructed in advance before extracting data. To obtain reversibility of the proposed scheme, the cover image I is partitioned into two areas, i.e., the embeddable area and the non-embeddable area, which are shown in Figure 5. The embeddable area is used for embedding secret data and the least significant bits (LSBs) of the pixels in the non-embeddable area, and the non-embeddable area is used to record the information of location map L and side information, which should be compressed losslessly by using JBIG-kit in [22]. Therefore, the LSBs of a pixel in the non-embeddable area must be extracted and concatenated into secret data B to generate the embedded data, S. Figure 6 shows the flowchart of our proposed embedding phase.

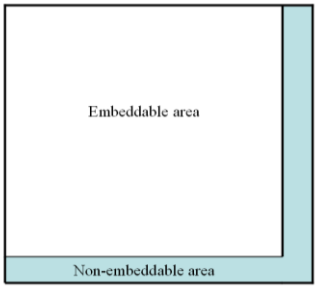


Figure 5: Embeddable and non-embeddable areas in a cover image

Since pixel values in many natural images are highly correlated with their adjacent pixels, such as upper, left, right, and bottom pixels, as shown in Figure 7(a). That is, the differences between the current pixel and its adjacent pixel could be equal to or close to zero. Therefore, we proposed a new scan path for each block to compute the difference values as shown in Figure 7(b). The new scan path contains four sub-scan paths, i.e., $C_i - P_2 - P_1$, $C_i - P_4 - P_3$, $C_i - P_6 - P_5$, and $C_i - P_8 - P_7$. To compute the difference value of no-reference pixels of each smooth block, these four sub-scan paths are processed, respectively, instead of computing the difference value based on the scan path of the inverse "S" order, as was the case in Zhao et al.'s scheme

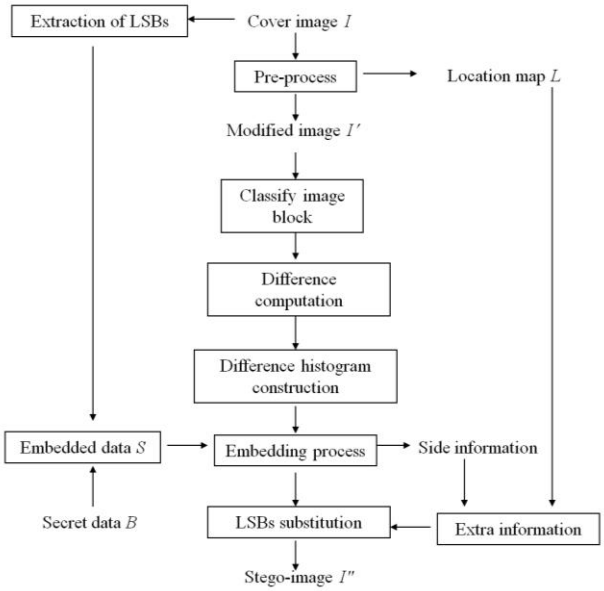


Figure 6: Flowchart of the data embedding phase

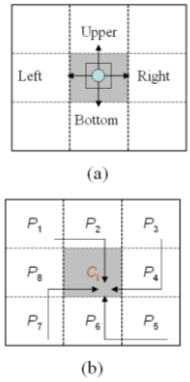


Figure 7: New scan path for the proposed scheme

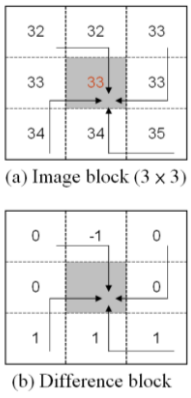


Figure 8: Example of computing the difference values for eight non-reference pixels

[30]. The difference values d_i can be calculated by Equation (9). In this way, more secret data can be embedded due to the fact that higher peak points are obtained in the proposed scheme than in existing schemes. Figure 8 shows a simple example that demonstrates clearly how to compute the difference values of the eight non-reference pixels. In our proposed scheme, the reference pixel must remain unchanged. Thus, the reference pixel of each block is not used for embedding data.

The algorithm of the proposed data embedding phase is listed below:

Input: Cover image I of sized $M \times N$, secret data B, threshold T.

Output: Stego-image I''.

Step 1: The cover image I is partitioned into two areas, i.e., the embeddable area and the non-embeddable area, as shown in Figure 5. The non-embeddable area consists of the two lowest rows and the two right-most columns of the cover image. The LSBs of the pixels in the non-embeddable area are used to record the information of the location map L and the side information. Therefore, the LSBs of the pixels in the non-embeddable area must be extracted and concatenated into the secret data B to generate the embedded data S.

Step 2: Process the original cover image I by scanning the entire cover image; then, record the positions of those pixels that have values of 0 and 255 into location map L. The location map L is concatenated with side information to generate the extra information that will be processed later in this section. Then, pixels that have values of 0 and 255 are modified to 1 and 254, respectively. The modified cover image is denoted as I'.

Step 3: The embeddable area of the modified cover image I' is divided into blocks with sizes of 3×3 pixels. Then, the blocks are classified as smooth blocks (S-blocks) and complex blocks (C-blocks) using Equation (8).

Step 4: For each block $B_i \in$ S-block, compute the difference values d_i of the non-reference pixels in B_i with Equation (9).

$$\begin{aligned} d_2 &= P_2 - C_i \\ d_1 &= P_1 - P_2 \\ d_4 &= P_4 - C_i \\ d_3 &= P_3 - P_4 \\ d_6 &= P_6 - C_i \\ d_5 &= P_5 - P_6 \\ d_8 &= P_8 - C_i \\ d_7 &= P_7 - P_8 \end{aligned} \tag{9}$$

Step 5: The difference histogram of the non-reference pixels is constructed. Then, two pairs, i.e., (PP_1, ZP_1) and (PP_2, ZP_2), of the peak and zero points of the histogram are obtained. Without loss of generality we assume that $ZP_1 < PP_1 < PP_2 < ZP_2$. Note that $(PP_1, ZP_1) \cap (PP_2, ZP_2) = \varnothing$.

Step 6: Scan the difference values d_i sequentially. If $d_i = PP_1$ or $d_i = PP_2$, a secret bit s, extracted from the embedded data S, can be embedded by modifying the difference value d_i to d_i' using Equation (10). Otherwise, no secret bit can be embedded, and the scan difference value d_i has to be modified to d_i' using Equation (11).

$$d_i' = \begin{cases} d_i & if\ s = 0 \\ d_i - 1 & if\ s = 1\ and\ d_i = PP_1 \\ d_i + 1 & if\ s = 1\ and\ d_i = PP_2 \end{cases} \quad (10)$$

$$d_i' = \begin{cases} d_i - 1 & if\ ZP_1 < d_i < PP_1 \\ d_i + 1 & if\ PP_2 < d_i < ZP_2 \\ d_i & otherwise \end{cases} \quad (11)$$

Step 7: Repeat Step 6 until the embedded data S are embedded completely.

Step 8: Construct the stego-image block B'_i using Equation (12).

$$\begin{aligned} P_1' &= d_1' + P_2 \\ P_3' &= d_3' + P_4 \\ P_5' &= d_5' + P_6 \\ P_7' &= d_7' + P_8 \\ P_2' &= d_2' + C_i \\ P_4' &= d_4' + C_i \\ P_6' &= d_6' + C_i \\ P_8' &= d_8' + C_i \end{aligned} \quad (12)$$

After all of the steps have been processed completely, the stego blocks are grouped to obtain the stego-image *I''*, and the extra information, including the location map *L* and side information, i.e., ZP_1, ZP_2, PP_1, PP_2, and the threshold *T*, will be compressed losslessly by using JBIG-kit, as mentioned in [22]. The compressed bitstream is denoted as *E*, which is embedded into the lowest two rows and the two right-most columns of the cover image, known as the non-embeddable area, by using LSBs substitution. Finally, the stego-image *I''* is sent to the receiver.

3.3 Data Extracting Phase

In this section, we also divide the stego-image into two areas, i.e., embeddable and non-embeddable areas, according to the rules mentioned in above subsection. Then, the embeddable area is partitioned into blocks that have sizes of 3×3 pixels. Since the reference pixels are preserved in the embedding process, the image block will be re-classified into smooth blocks or complex block in the same manner that was used in the data-embedding phase. Subsequently, the secret data can be extracted correctly, and the cover image *I* can be reconstructed. The detailed extracting algorithm is described below:

Input: Stego-image *I''* of sized $M \times N$.

Output: Secret data *B*, cover image *I*.

Step 1: The stego-image *I''* is divided into two areas, i.e., the embeddable area and the non-embeddable area, in the same manner used in the data embedding phase. Then, the LSBs of the pixels in the non-embeddable area are extracted to retrieve the compressed bitstream *E*, which is decoded by using JBIG-kit in [22] to reconstruct the extra information, including the location map *L* and the side information, i.e., ZP_1, ZP_2, PP_1, PP_2, and the threshold *T*.

Step 2: Divide the embeddable area of the stego-image into blocks with sizes of 3×3 pixels. Then, classify these blocks into smooth blocks (S-blocks) or complex blocks (C-blocks) using Equation (8).

Step 3: For each block $B'_i \in$ S-block, compute the difference values d'_i of the non-reference pixels in B'_i using Equation (9).

Step 4: Scan the difference values d'_i sequentially by using the same order that was used in the data embedding phase. If the coordinate of d'_i is recorded in the location map *L*, then skip d'_i, and proceed to the next difference value. Otherwise, a secret bit *s* can be extracted according to Equation (13).

$$s = \begin{cases} 0 & if\ d_i' = PP_1\ or\ d_i' = PP_2 \\ 1 & if\ d_i' = PP_1 - 1\ or\ d_i' = PP_2 + 1 \end{cases} \quad (13)$$

The original difference value d_i can be reconstructed using Equation (14).

$$d_i = \begin{cases} d_i' + 1 & if\ ZP_1 < d_i < PP_1 \\ d_i' - 1 & if\ PP_2 < d_i < ZP_2 \\ d_i' & otherwise \end{cases} \quad (14)$$

Step 5: Repeat Step 4 until all embedded data *S* have been extracted.

Step 6: Construct the modified image block B_i using Equation (15). Then, the modified image *I'* can be obtained by grouping the modified image blocks.

$$\begin{aligned} P_1 &= d_1 + P_2 \\ P_3 &= d_3 + P_4 \\ P_5 &= d_5 + P_6 \\ P_7 &= d_7 + P_8 \\ P_2 &= d_2 + C_i \\ P_4 &= d_4 + C_i \\ P_6 &= d_6 + C_i \\ P_8 &= d_8 + C_i \end{aligned} \quad (15)$$

After finishing the six steps above, the modified cover image *I'* is recovered completely, and the embedded data *S* also are extracted correctly. To reconstruct the cover image *I*, the LSBs of the pixels in the non-embeddable area are recovered according to the embedded data *S* by substituting their LSBs.

The peak signal-to-noise ratio (PSNR) was used to estimate the difference between the quality of the original cover image and its stego-image. A large PSNR value indicates that the visual quality of the stego-image is good because it means that a small amount of distortion has occurred. In contrast, a small PSNR value denotes that the stego-image has poor visual quality due to its large distortion. The equation for calculating PSNR is:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (16)$$

where the MSE is the mean square error for an M × N

(a) Lena (b) Airplane (c) Baboon

(d) Peppers (e) Barbara (f) Zelda

Figure 9: Six 512×512-pixel grayscale test images

grayscale image; MSE is defined as:

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(I_{ij} - I_{ij}^{'} \right)^2 \quad (17)$$

where I_{ij} and I'_{ij} are pixel values of the original cover image and the stego-image, respectively.

Embedding capacity (*EC*) indicates the number of secret bits that can be embedded into the cover image. We used bits per pixel (bpp) as the unit of embedding capacity, which is defined in Equation (18):

$$EC = \frac{\|B\|}{M \times N} (bpp) \quad (18)$$

where $\|B\|$ is the length of the secret data B that can be embedded into a cover image that has a size of $M \times N$ pixels.

Figure 9 shows the difference histograms of six standard test images computed by the proposed scheme and Luo et al.'s scheme [19]. Figure 10 shows 24 grayscale images obtained from Kodak Lossless True Color Image Suite (sizes of 768×512 pixels) [23]. The proposed scheme is represented by the blue curve, and the Luo et al.'s scheme is represented by the red curve. The decision was made to compare the proposed scheme with Luo et al.'s scheme [19] in Figure 11 because both schemes used the reference pixel to compute the difference values for embedding secret data. However, our proposed scheme is based on the new scan path instead of block-median selection in [19]. Figure 11 shows that, in our proposed scheme, there is a higher concentration of the difference values around bin 0 than in Luo et al.'s scheme [19], and there are fewer difference values in the bins that are far removed from bin 0. Table 1 shows a detailed comparison of several peak histogram bins that were obtained by our proposed scheme and Luo et al.'s scheme [19]. It is apparent that, on average, the bins around 0 (from -2 to 2) resulting from the proposed scheme are larger than those resulting from Luo et al.'s scheme [19].

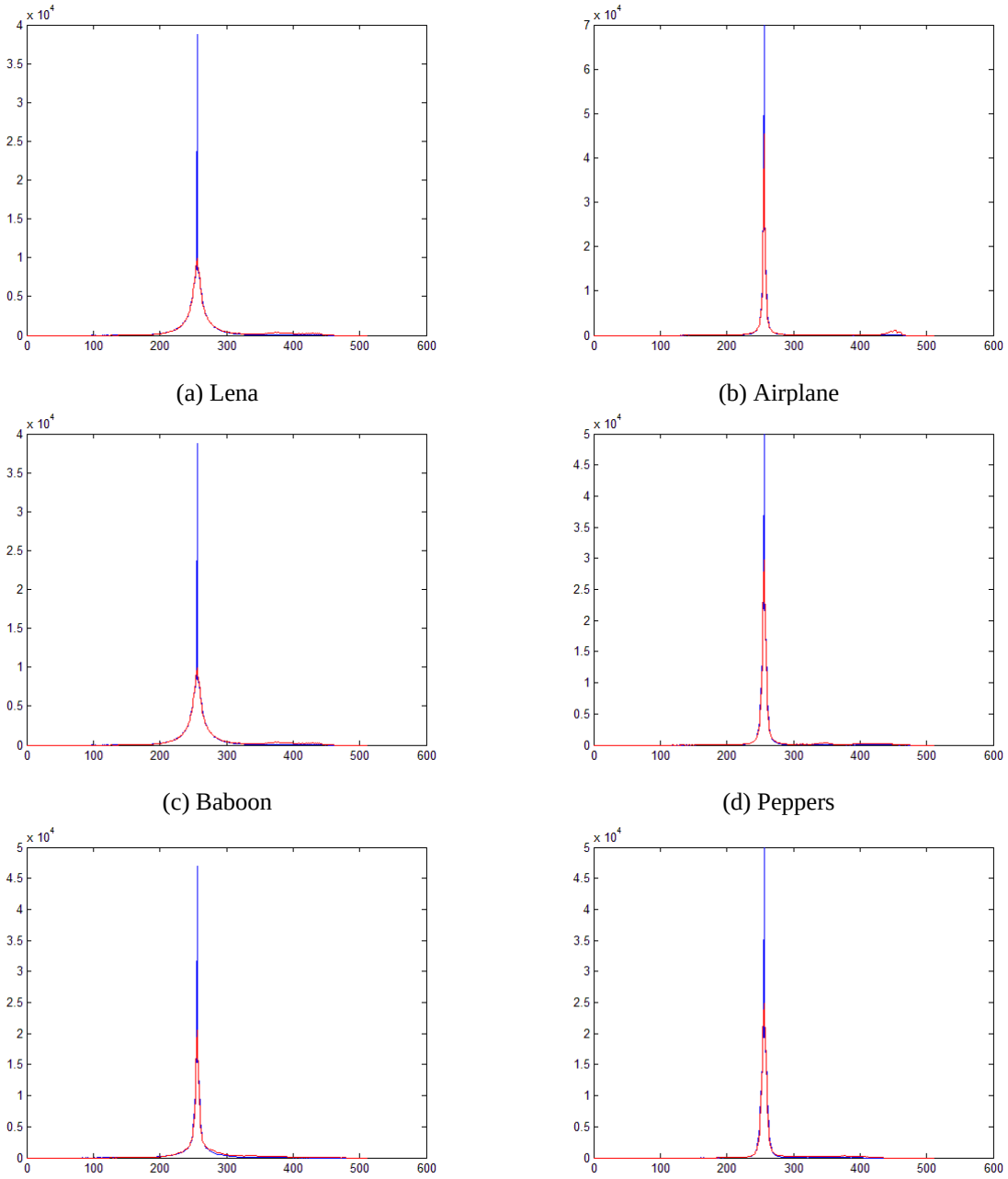
Table 2 presents the comparison results of maximum embedding capacity (EC) and corresponding *PSNR*s among six schemes for a single layer embedding when the first set of test images was used. In Kim et al.'s scheme [14], four sub-sample images were used, and the embedding level was set to 0. In Li et al.'s scheme [16], we used the APD2 scheme. In the Zhao et al.'s scheme [30], the embedding level *EL* was set to 0. In Luo et al.'s scheme [19], the block size and embedding level were set to 3 × 3 and 0, respectively. In Wang et al.'s scheme [29], the distance value Δ was set to 2 as described in their scheme. In our proposed scheme, we set threshold T to 30. The above settings ensure that the best performance of each previous scheme is obtained for making a fair comparison. As shown in Table 2, the embedding capacity of the proposed scheme was better than that of each of the other five schemes. This was because the local complexity of each image block in our scheme was computed based on the satellite reference pixel to classify the image blocks as smooth and complex blocks. For the smooth blocks, the new scan path was used to compute the difference value for data embedding. With the new scan path, the higher frequency of difference values was concentrated around bin 0. This means that the proposed scheme provides a better embedding capacity than the other schemes. In terms of image quality, Zhao et al.'s scheme had a better performance than our proposed scheme because the average embedding capacity (0.09 bpp) of their scheme was significantly lower than that of our proposed scheme. However, the average *PSNR* of the

proposed scheme, 49.83 dB, was superior to those of the other four schemes, i.e., Wang et al.'s scheme (48.65 dB), Luo et al.'s scheme (48.59 dB), Li et al.'s scheme (48.58 dB), and Kim et al.'s (44.10 dB). The proposed scheme obtained better image quality and higher embedding capacity than the other four schemes, as shown in Table 2. This is because the designs of the other four schemes did not allow exploring the properties of the blocks while embedding data. In contrast, the proposed scheme uses the local complexity function to identify smooth blocks and also incorporated a new scan path for embedding data to get higher frequencies of the peak histogram bins for embedding data. In addition, the complex blocks were excluded during the embedding process to maintain the high quality of the stego-image. Notably, by using the above strategies, our proposed scheme achieved higher embedding capacity and less visual distortion for different types of test images.

Figures 12(a)-(f) show the multi-level embedding performance of our proposed scheme and five other schemes for different test images. For the first set of test images, the top curve is our proposed scheme. As was the case for *PSNR* value, the embedding capacity of our proposed scheme was higher than those of the other five schemes. Figure 12 shows that the proposed scheme achieved a larger embedding capacity and less image distortion than the other five schemes for all of the images that were tested. This is because the proposed scheme used a new scan path to compute the difference values so that the higher frequency of the peak histogram bins can be

Figure 10: Twenty-four grayscale images obtained from Kodak Lossless True Color Image Suite

(sizes of 768×512 pixels)



(a) Lena (b) Airplane

(c) Baboon (d) Peppers

(e) Barbara (f) Zelda

Figure 11: Comparison of the histograms from the proposed scheme (blue curve) and Luo *et al.*'s scheme [19] (red curve)

found for data embedding. As shown in Table 1, our proposed scheme always obtained higher peak histogram bins than Luo et al.'s scheme for all of the test images. When the higher frequency of peak histogram bins is generated, larger embedding capacity is obtained. In addition, the embedding process in the proposed scheme was implemented only in the smooth blocks, so it always provided the best quality of the six stego-images while maintaining the same embedding capacity.

To further demonstrate the performance of our proposed scheme in image quality and embedding capacity, the second set of test images, i.e., the 24 colors images in the Kodak set, which were taken with a digital camera, were tested. For each of the images, the grayscale version was generated in advance, and, then, the proposed algorithm was used to embed the secret data. Table 3 provides the average *PSNR*s and embedding capacity of the grayscale versions of the 24 test images with different threshold *Ts* for a single

embedding level. The results confirmed that the proposed scheme provided excellent performance on the Kodak test images with different threshold values.

4 Conclusions

In this paper, a new, reversible data hiding scheme was proposed by considering the reference pixels in the adjacent area to identify complex image blocks and to design a new scan path. With the new scan path, a higher frequency of peak histogram bins was obtained from the smooth blocks of the original cover image. A higher visual quality of the stego-image can be obtained by identifying complex blocks so they can be ignored during the process of embedding data. Six standard test images and 24 grayscale versions from the Kodak set were used in the experiments to prove the performance of the proposed scheme in embedding capacity and image quality. In comparisons with five other existing schemes, our proposed scheme provided better performance in both embedding capacity and image quality.

Table 1: Comparison of several peak histogram bins obtained by the proposed scheme and Luo *et al*.'s scheme [19]

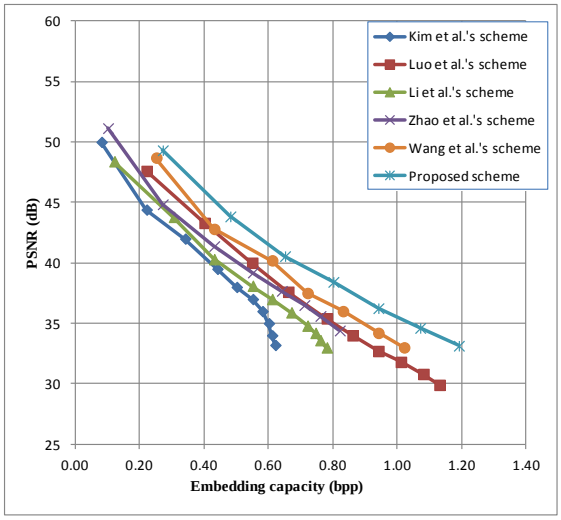
Test images	Schemes	Histogram bins								
		-4	-3	-2	-1	0	1	2	3	4
Lena	Luo et al.	11,316	15,828	21,561	26,799	26,465	18,505	12,091	7,302	4,319
	Proposed	9,380	13,559	19,258	24,750	57,879	24,527	18,990	13,748	9,245
Airplane	Luo et al.	8,493	12,815	19,291	31,552	36,221	18,263	9,935	6,163	3,713
	Proposed	7,243	11,682	18,077	29,156	69,932	29,565	17,863	11,741	7,229
Baboon	Luo et al.	7,095	8,009	8,758	9,173	9,942	9,077	8,542	7,919	7,011
	Proposed	6,938	7,706	8,202	8,621	38,805	8,490	8,114	7,750	6,884
Barbara	Luo et al.	9,332	12,603	15,983	18,620	18,281	13,817	9,719	6,582	4,456
	Proposed	8,044	10,867	13,874	16,602	46,921	16,630	14,080	10,843	8,018
Peppers	Luo et al.	9,526	14,011	19,813	25,991	29,770	25,943	19,647	14,057	9,657
	Proposed	10,697	14,559	19,680	23,872	55,536	23,703	19,479	14,636	10,786
Zelda	Luo et al.	11,668	15,560	19,435	22,899	24,821	22,677	19,524	15,171	11,432
	Proposed	12,211	15,472	18,271	20,193	49,969	20,457	18,381	15,613	12,240

Table 2: Comparison of embedding capacities and PSNR values achieved by the six schemes for six test images

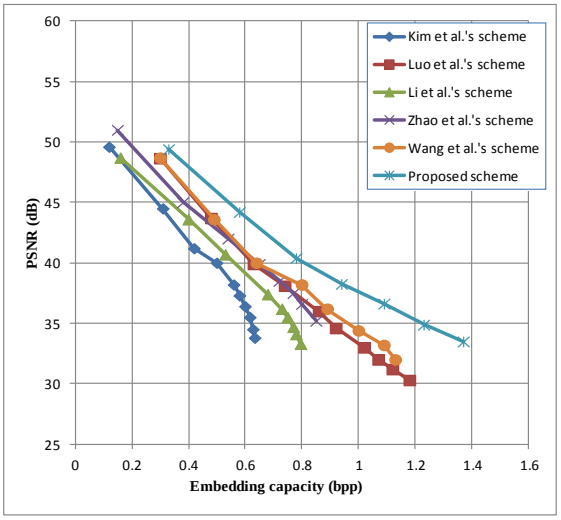
Schemes	Metrics	Lena	Airplane	Baboon	Peppers	Zelda	Barbara	Average
Kim et al.'s scheme	EC (bpp)	0.24	0.31	0.08	0.23	0.24	0.15	0.21
	PSNR (dB)	44.20	44.55	43.63	44.16	44.20	43.85	44.10
Li et al.'s scheme	EC (bpp)	0.23	0.30	0.09	0.24	0.21	0.13	0.20
	PSNR (dB)	48.67	48.75	48.33	48.68	48.60	48.42	48.58
Zhao et al.'s scheme	EC (bpp)	0.10	0.15	0.05	0.13	0.08	0.06	0.09
	PSNR (dB)	51.14	51.14	50.02	51.10	51.14	51.14	50.95
Luo et al.'s scheme	EC (bpp)	0.10	0.14	0.04	0.11	0.10	0.07	0.09
	PSNR (dB)	48.54	48.64	48.50	48.67	48.65	48.56	48.59
Wang et al.'s scheme	EC (bpp)	0.25	0.32	0.09	0.25	0.23	0.14	0.21
	PSNR (dB)	48.75	48.92	48.38	48.72	48.67	48.45	48.65
Proposed scheme	EC (bpp)	0.26	0.33	0.10	0.26	0.23	0.18	0.23
	PSNR (dB)	49.31	49.37	51.05	49.18	49.04	51.04	49.83

Table 3: Performance of the proposed scheme on 24 grayscale versions of the Kodak test images with different thresholds

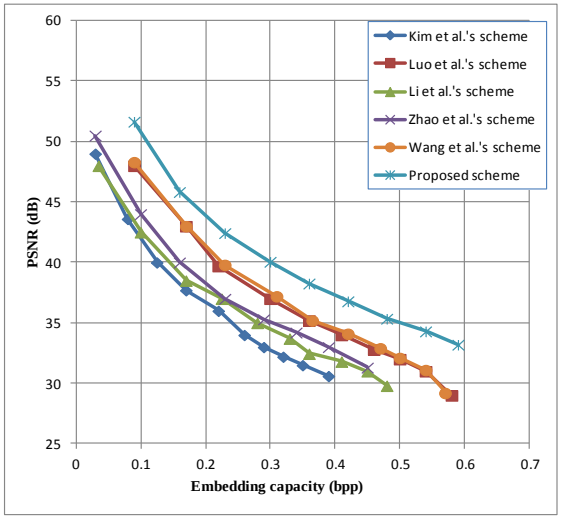
Threshold *Ts* / Metrics	10	30	50	70
Average EC (bpp)	0.26	0.38	0.42	0.44
Average *PSNR* (dB)	52.78	49.73	48.85	48.46



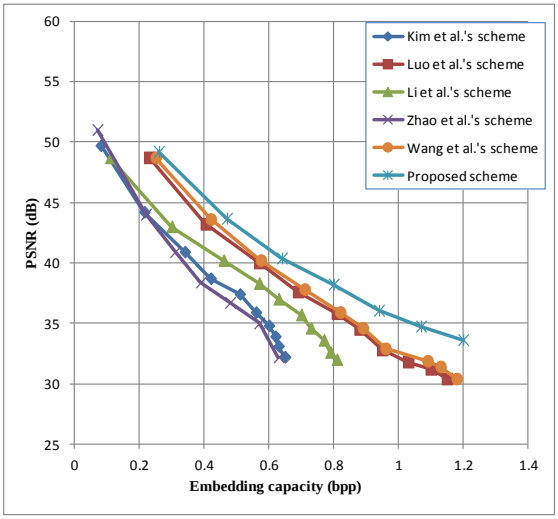
(a) Lena



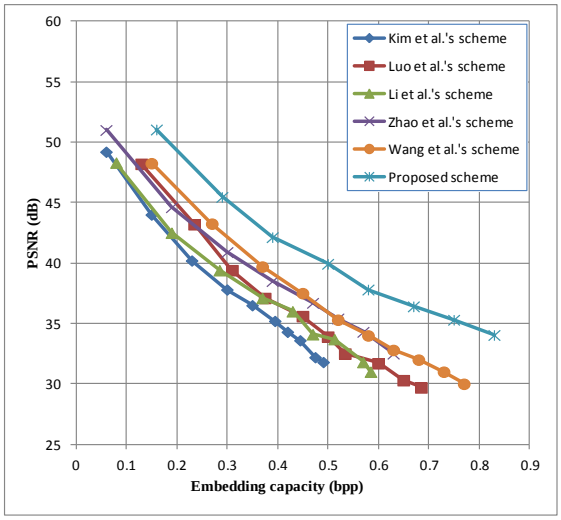
(b) Airplane



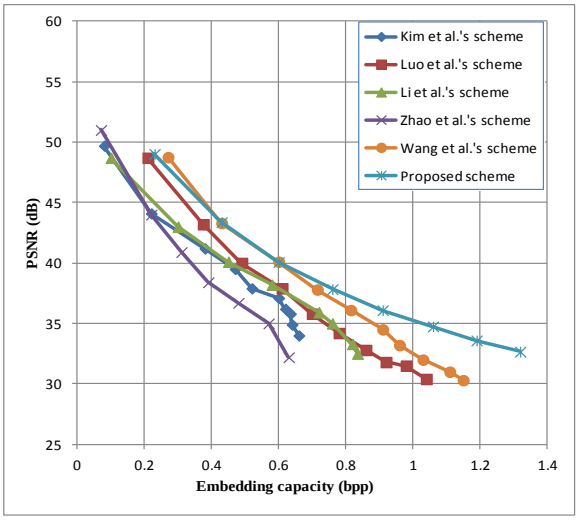
(c) Baboon



(d) Peppers



(e) Barbara



(f) Zelda

Figure 12: Performance evaluation of multilayer embedding over six standard testing images

References

[1] M. Atallah, V. Raskin, C. Hempelman, M. Karahan, R. Sion, K. Triezenberg, and U. Topkara, "Natural language watermarking and tamperproofing," in *Proceedings IH'02 Revised Papers from the 5th International Workshop on Information Hiding*, 2002, pp. 196-212.

[2] C. C. Chang and T. C. Lu, "A difference expansion oriented data hiding scheme for restoring the original host images," *The Journal of Systems and Software*, vol. 79, no. 12, pp. 1754-1766, 2006.

[3] S. Chen, H. Leung, "Chaotic watermarking for video authentication in surveillance applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 704-709, 2008.

[4] C. Collberg and C. Thomborson, "Software watermarking: Model and dynamic embeddings," in *Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 1999, pp. 311-324.

[5] D. Coltuc, "Improved embedding for prediction-based reversible watermark," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 873-882, 2011.

[6] I. J. Cox, J. Kilian, F. T. Leighton, T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, pp. 1673-1687, 1997.

[7] R. M. Davis, "The data encryption standard in perspective," *IEEE Communications Magazine*, vol. 16, no. 6, pp. 5–9, 1978.

[8] F. Doelitzscher, C. Reich, A. Sulistio, "Designing Cloud Services Adhering to Government Privacy Laws," in *IEEE 10th International Conference on Computer and Information Technology*, pp. 930-935, 2010.

[9] M. E. Farfoura, S. J. Horng, J. L. Lai, R. S. Run, R. J. Chen, and M. K. Khan, "A blind reversible method for watermarking relational databases based on a time-stamping," *Expert Systems with Applications*, vol. 39, pp. 3185-3196, 2012.

[10] W. Hong, and T. S. Chen, "Reversible data hiding for high quality images using interpolation and reference pixel distribution mechanism," *Journal of Visual Communication and Image Representation*, vol. 22, pp. 131-140, 2011.

[11] Y. Hu, H. K. Lee, amd J. Li, "De-based reversible data hiding with improved overflow location map," *IEEE Transactions on Circuit System and Video Technology*, vol. 19, no.2, pp. 250-260, 2009.

[12] W. Itani, A. Kayssi, and A. Chehab, "Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures," in *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, 2009 pp. 711-716.

[13] M. Iwata, K. Miyake, A. Shiozaki, "Digital steganography utilizing features of JPEG images," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E87-A (2004), pp. 929–936, 2004.

[14] K. S. Kim, M. J Lee, H. Y. Lee, H. K. Lee, "Reversible data hiding exploiting spatial correlation between sub-sampled images," *Pattern Recognition*, vol. 42, no. 11, pp. 3083-3096, 2009.

[15] G. Langelaar, I. Setyawan, and R. Lagendijk, "Watermarking digital image and video data: A state-of-the-art-overview," *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20-46, Sept. 2000.

[16] Y. C. Li, C. M. Yeh, C. C. Chang, "Data hiding based on the similarity between neighboring pixels with reversibility," *Digital Signal Processing*, vol. 20, no. 4, pp. 1116-1128, 2010.

[17] D. C. Lou, M. C. Hu, J. L. Liu, "Multiple layer data hiding scheme for medical images," *Computer Standards & Interfaces*, vol. 31, pp. 329-335, 2009.

[18] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong "Reversible image watermarking using interpolation technique," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, 2011, pp. 187-193.

[19] H. Luo, F. X. Yu, H. Chen, Z. L. Huang, H. Li, and P. H. Wang, "Reversible data hiding based on block median preservation," *Information Sciences*, Vol. 181, pp. 308–328, 2011.

[20] J. Mielikainen, "LSB matching revisited," *IEEE Signal Processing Letters*, vol. 13, pp. 285–287, 2006.

[21] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, 2006.

[22] [Online]. Available: http://www.cl.cam.ac.uk/~mgk25/jbigkit

[23] [Online]. Available: http://ww.r0k.us/graphics/kodak/

[24] C. Qin, C. C. Chang, and L. T. Liao, "An adaptive prediction-error expansion oriented reversible information hiding scheme," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2166-2172, 2012.

[25] W. L. Tai, C. M. Yeh, C. C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, 2009.

[26] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, pp. 721-730, 2007.

[27] J. Tian, "Reversible data hiding using difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 890-896, 2003.

[28] P. Tsai, Y. C. Hu, H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram

shifting," *Signal Processing*, vol. 89, no. 6, pp.1129-1143, 2009.

[29] X. T. Wang, C. C. Chang, T. S., Nguyen, and M. C. Li, "Reversible data hiding for high quality images exploiting interpolation and direction order mechanism," *Digital Signal Processing*, vol. 23, no. 2, pp. 569-577, 2013.

[30] Z. Zhao, H. Luo, Z. M. Lu, J. S. Pan, "Reversible data hiding based on multilevel histogram modification and sequential recovery," *International Journal of Electronics and Communications*, vol. 65, pp. 814-826, 2011.

Chin-Chen Chang received the B.S. degree in applied mathematics and the M.S. degree in computer and decision sciences from National Tsing Hua University, Hsinchu, Taiwan, China, in 1977 and 1979, respectively. He received the Ph.D degree in computer engineering from National Chiao Tung University, Hsinchu, in 1982. During the academic years 1980-1983, he was on the faculty of the Department of Computer Engineering, National Chiao Tung University. From 1983 to 1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From August 1989 to July 1992, he was Head of, and a Professor in, the Institute of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan. From August 1992 to July 1995, he was the Dean of the College of Engineering at the same university. From August 1995 to October 1997, he was the Provost of National Chung Cheng University. From September 1996 to October 1997, he was Acting President of National Chung Cheng University. From July 1998 to June 2000, he was Director of the Advisory Office, Ministry of Education, China. From 2002 to 2005, he was a Chair Professor at National Chung Cheng University. From February 2005, he has been a Chair Professor at Feng Chia Univeristy. In addition, he was severed as a Consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression, and data structures.

Thai-Son Nguyen received the bachelor's degree in information technology from the Open University, HCM city, Vietnam, in 2005. From December 2006, he has been lecturer of TraVinh University, TraVinh, Vietnam. In 2011, he received M.S. degree in computer sciences from Feng Chia University, TaiChung, Taiwan. He is currently pursuing the Ph.D. degree with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. His current research interests include data hiding, image and signal processing, image compression, multimedia security, information security.

.

Chia-Chen Lin received the B. S. degree in information management from the Tamkang University, Taipeo, Taiwan, R.O.C., in 1992. She received the M.S. degree in information management and the Ph.D. degree in information management from Chiao Tung University, Hsinchu, Taiwan, in 1994 and 1998, respectively. She was a Visiting Associate Professor at Business School, University Illinois at Urbana Champain, during August 2006 to July 2007. She is currently a Professor in the Department of Computer and Information Management, Providence University, Sha-Lu, Taiwan. Her research interests include image and signal processing, image data hiding, mobile agent, and electronic commerce.