Int. J. Electron. Commun. (AEÜ) 65 (2011) 814–826

Contents lists available at ScienceDirect

International Journal of Electronics and Communications (AEÜ)

journal homepage: www.elsevier.de/aeue

Reversible data hiding based on multilevel histogram modification and sequential recovery

Zhenfei Zhao [a,b], Hao Luo [c,*], Zhe-Ming Lu [c], Jeng-Shyang Pan [d]

[a] School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China
[b] Heilongjiang Institute of Science and Technology, Harbin, China
[c] School of Aeronautics and Astronautics, Zhejiang University, No. 38 ZheDa Road, Hangzhou 310027, China
[d] Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan

ARTICLE INFO

Article history:
Received 14 January 2010
Accepted 21 January 2011

Keywords:
Reversible data hiding
Multilevel histogram modification
Sequential recovery

ABSTRACT

This paper proposes a reversible data hiding method for natural images. Due to the similarity of neighbor pixels' values, most differences between pairs of adjacent pixels are equal or close to zero. In this work, a histogram is constructed based on these difference statistics. In the data embedding stage, a multilevel histogram modification mechanism is employed. As more peak points are used for secret bits modulation, the hiding capacity is enhanced compared with those conventional methods based on one or two level histogram modification. Moreover, as the differences concentricity around zero is improved, the distortions on the host image introduced by secret content embedding is mitigated. In the data extraction and image recovery stage, the embedding level instead of the peak points and zero points is used. Accordingly the affiliated information is much smaller than in those methods of the kind. A sequential recovery strategy is exploited for each pixel is reconstructed with the aid of its previously recovered neighbor. Experimental results and comparisons with other methods demonstrate our method's effectiveness and superior performance.

1. Introduction

Data hiding, also called information hiding, plays an important role in information security. It aims at embedding imperceptible confidential information in cover media such as still images, videos, audios, 3D meshes, etc. It consists of several branches such as steganography, watermarking, visual cryptography, etc. The data hiding scheme proposed in this work can be classified into the category of steganography. Steganography is usually used for covert communications. Thus the high embedding capacity is the main concern in this kind of technique. In contrast, watermarking is usually used for copyright protection and announcement. Thus researchers aim at improving the robustness of watermark content against intentional or unintentional attacks. Therefore, most available data hiding methods can provide a higher capacity than that provided by watermarking schemes. This advantage broadens the application scenarios of data hiding.

Nowadays, various data hiding techniques have been reported in literatures [1,2,29–31]. As a burgeoning branch, reversible data hiding has drawn much attention among researchers. Its key property is not only the secret data but also the host image can be accurately recovered in decoder. Therefore, it can be used in those applications where the host images (e.g., military maps, remote sensing images, medical images [11], digitalized art pictures, etc.) must be exactly reconstructed. In contrast, the conventional irreversible data hiding methods are not appropriate any longer.

Available reversible data hiding techniques can be divided into spatial domain, transform domain and compressed domain methods. In the spatial domain based methods [3–20], the secret data is usually embedded by pixels' values modification. In the transform domain methods, some reversibility-guaranteed transforms (e.g., integer discrete cosine transform [21,22], integer wavelet transform [23]) are exploited and the data embedding is reduced to coefficients modulation. In the compressed domain methods, some popular used image compression techniques (e.g., vector quantization [24–26], block truncation coding [27], MPEG coding [28]) are involved.

Most spatial domain reversible data hiding are developed based on two principles, i.e., difference expansion (DE) [3–12] and histogram modification [13–20]. In general, the former kind of methods can provide a higher capacity while the latter can produce a better quality marked image.

This paper proposes a reversible data hiding scheme based on histogram modification. Its principle is to modify the histogram constructed based on the neighbor pixel differences instead of the host image's histogram as in [13]. Many peak points exist around

[*] Corresponding author. Tel.: +86 158 58259064; fax: +86 571 86971612.
E-mail address: luohao@zju.edu.cn (H. Luo).

1434-8411/$ – see front matter © 2011 Elsevier GmbH. All rights reserved.
doi:10.1016/j.aeue.2011.01.014

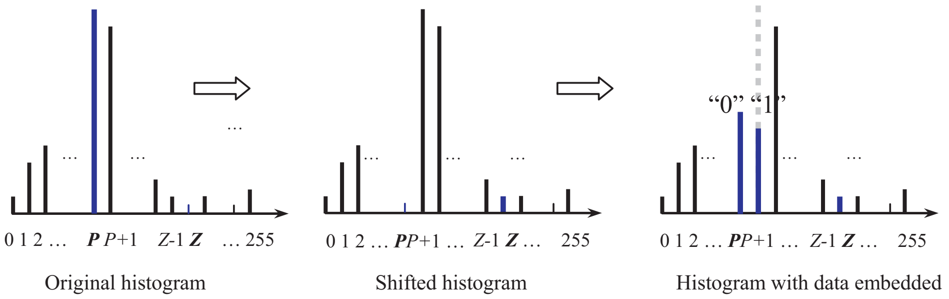
"0"
"1"
…
0 1 2 … P P+1 Z-1 Z … 255
0 1 2 … P P+1 … Z-1 Z … 255
0 1 2 … P P+1 … Z-1 Z … 255
Original histogram
Shifted histogram
Histogram with data embedded

Fig. 1. Principle of reversible data hiding based on histogram modification.

the bin zero in this histogram due to the similarity of adjacent pixel values. Besides, many zero points exist in both sides of the bin zero. Here the peak point refers to the height of histogram bin with the largest statistical value (i.e., the count falling in the corresponding bin), and the zero point means the histogram bin with zero value. In our case, all the differences are classified into levels of [−255, 255] and each level corresponds to a histogram bin. Hence it is reasonable to modify the histogram with a multilevel mechanism for hiding more secret data. In decoder, the host image pixels are recovered one by one. That is, each pixel is recovered aided by its previously recovered neighbor. Meanwhile, the secret data is extracted from the marked adjacent pixels' differences.

This paper is organized as follows. Section 2 reviews the related work. Section 3 describes the proposed scheme including data embedding, extraction and image recovery procedures. Section 4 discusses the capacity estimation, overflow and underflow prevention. Experimental results and performance comparisons with other algorithms are shown in Section 5. Finally, conclusions are given in Section 6.

2. Related work

In [13], Ni et al. proposed a reversible data hiding method based on histogram modification. In the scheme, part of the cover image histogram is shifted rightward or leftward to produce redundancy for data embedding. The principle can be illustrated as shown in Fig. 1. First, the peak and zero point bins of the original histogram are found denoted as $b(P)$ and $b(Z)$, respectively. Then all the bins belonging to $b(P)$ and $b(Z)$ are shifted rightward one level. In this way, the bin of $b(P)$ is emptied and $b(P+1)$ becomes the new peak point. Next, the confidential data can be embedded by modulating the pixel values equaling $P+1$. That is, if encounter a pixel with value equaling $P+1$, then one bit confidential data can be hidden. For example, if the current processing confidential bit is "0", we modify the pixel value as P; whereas if the current processing confidential bit is "1", the pixel with value $P+1$ is kept no changed. In decoder, the data extraction and image recovery is the inverse processing of data embedding.

In [16], Li et al. proposed a reversible data hiding method named adjacent pixel difference (APD) based on the neighbor pixel differences modification. In this method, an inverse "S" order is adopted to scan the image pixels. As shown in Fig. 2, a 3 × 3 image block is used to illustrate this principle. The scan direction is marked as the blue line, and the block can be rearranged into a pixel sequence as $p_1, p_2, \ldots, p_9$.

Suppose the host image I is an 8-bit gray level image sized as $M \times N$. Then a pixel sequence $p_1, p_2, \ldots, p_{M\times N}$ are obtained via the inverse "S" order scan. The differences of adjacent pixels are computed as:

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

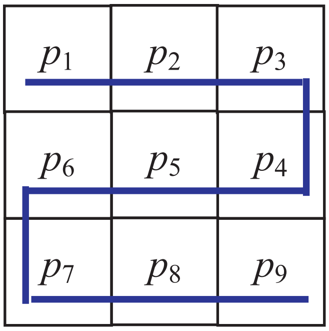

Fig. 2. Inverse "S" scan of a 3 × 3 image block.

Considering the pixel values similarity between p_{i-1} and p_i, a large quantity of d_i $(2 \le i \le M \times N)$ is equal or close to 0. The difference histogram is constructed based on these $M \times N - 1$ difference statistics. Suppose the histogram bins from left to right are denoted by $b(-255)$, $b(-254)$, ..., $b(-1)$, $b(0)$, $b(1)$, ..., $b(254)$, $b(255)$. Fig. 3 shows the 512 × 512 Lena image's difference histogram. Obviously most differences are concentrated around $b(0)$. When the curve spreads away to both sides, it drops dramatically, and no differences fall into those bins far from $b(0)$.

Basically, APD selects one pair of bins $b(p_1)$ and $b(z_1)$ (suppose $p_1 < z_1$) where $b(p_1)$ and $b(z_1)$ denote the peak point and zero point, respectively. Then the bins between $[b(p_1+1), b(z_1-1)]$ are shifted rightward one level. Thus $b(p_1+1)$ are emptied for data embedding. That is, if a secret bit "1" is embedded, the differences equaling p_1 are added by 1. If "0" is embedded, they are not changed.

To enhance the capacity, APD can also select two pairs of peak-zero points, e.g. $[b(p_1), b(z_1)]$ and $[b(z_2), b(p_2)]$ (suppose $p_1 < z_1$

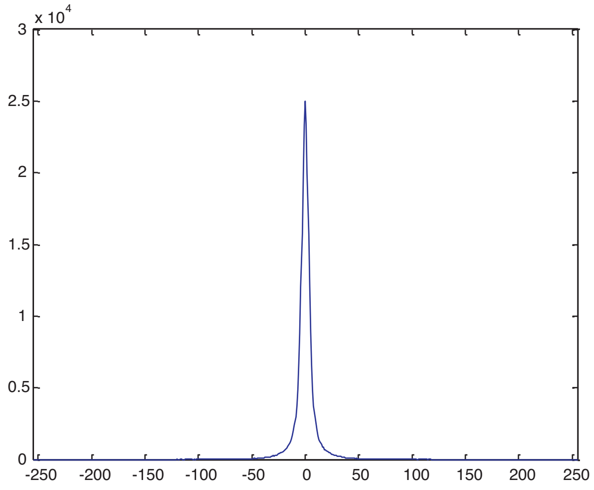

Fig. 3. The difference histogram of 512 × 512 Lena image.

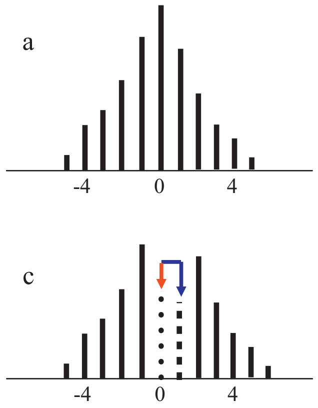

Fig. 4. Histogram modification for *EL* = 0.

and $z_2 < p_2$). Then the bins between $[b(p_1+1), b(z_1-1)]$ are shifted rightward one level, and those between $[b(z_2+1), b(p_2-1)]$ are shifted leftward one level. Thus $b(p_1+1)$ and $b(p_2-1)$ are emptied for data embedding. The secret bits modulation is similar as that in one pair of peak-zero points embedding. Note the ranges of $[b(p_1), b(z_1)]$ and $[b(z_2), b(p_2)]$ must not be overlapped.

3. Proposed scheme

3.1. Motivation

However, the disadvantage of APD method is that the provided capacity is not very high due to only two pairs of peak-zero points at most are employed for data hiding. This limits its scope of application where a large quantity of data is to be embedded. In fact, more pairs of peak-zero points can be utilized. Motivated from this, this work designed a multilevel histogram modification mechanism for a large capacity data hiding.

3.2. Data embedding

In our scheme, the inverse "S" order is adopted to scan the image pixels for difference generation. The secret data are binary sequences produced by pseudo random number generator. In the data embedding stage, a multilevel histogram modification strategy is utilized. An integer parameter called embedding level *EL* ($EL \geq 0$) is involved to control the hiding capacity. A larger *EL* indicates more secret data can be embedded. As the embedding operations for $EL > 0$ are more complicated than those of $EL = 0$, we describe them for $EL = 0$ and $EL > 0$ separately.

Step 1. Inverse "S" scan *I* into a pixel sequence $p_1, p_2, \ldots, p_{M\times N}$.
Step 2. Compute the differences d_i $(1 \leq i \leq M\times N)$ according to Eq. (1) and construct a histogram based on d_i $(2 \leq i \leq M\times N)$.
Step 3. Select an *EL*. If $EL = 0$, execute *Step 4*. If $EL > 0$, go to *Step 5*.
Step 4. Data embedding for $EL = 0$.
Step 4.1. Shift the right bins of $b(0)$ rightward one level as:

$$d'_i = \begin{cases} p_1 & \text{if} \quad i = 1 \\ d_i & \text{if} \quad d_i \leq 0, 2 \leq i \leq M\times N \\ d_i + 1 & \text{if} \quad d_i > 0, 2 \leq i \leq M\times N \end{cases} \qquad (2)$$

Step 4.2. Examine $d'_i = 0$ $(2 \leq i \leq M\times N)$ one by one. Each difference equaling 0 can be used to hide one secret bit. If the current processing secret bit $w = 0$, it is not changed. If $w = 1$, it is added by 1. The operation is as:

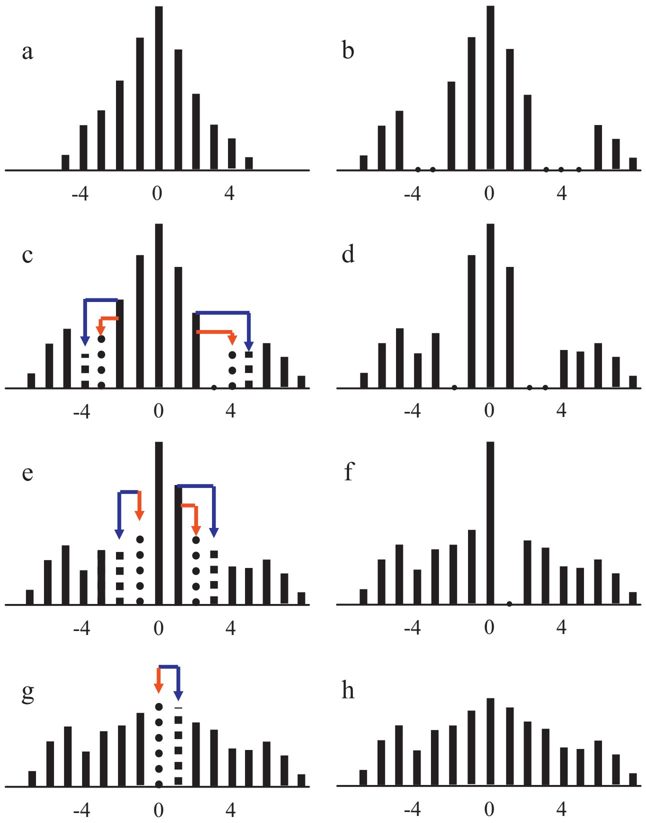

Fig. 5. Histogram modification for *EL* = 2.

$$d''_i = \begin{cases} p_1 & \text{if} \quad i = 1 \\ d'_i + w & \text{if} \quad d'_i = 0, 2 \leq i \leq M\times N \\ d'_i & \text{if} \quad d'_i \neq 0, 2 \leq i \leq M\times N \end{cases} \qquad (3)$$

The histogram modification strategy for $EL = 0$ is shown in Fig. 4(a)–(d) where the red and blue arrows indicate embedding "0" and "1", respectively. After that, go to *Step 6*. (For interpretation of the references to colour in the text, the reader is referred to the web version of the article.)
Step 5. Data embedding for $EL > 0$.
Step 5.1. Shift the right bins of $b(EL)$ rightward $EL + 1$ levels, and shift the left bins of $b(-EL)$ leftward *EL* levels as:

$$d'_i = \begin{cases} p_1 & \text{if} \quad i = 1 \\ d_i & \text{if} \quad -EL \leq d_i \leq EL, 2 \leq i \leq M\times N \\ d_i + EL + 1 & \text{if} \quad d_i > EL, 2 \leq i \leq M\times N \\ d_i - EL & \text{if} \quad d_i < -EL, 2 \leq i \leq M\times N \end{cases} \qquad (4)$$

Step 5.2. Examine $d'_i = 0$ $(2 \leq i \leq M\times N)$ in the range of $[-EL, EL]$ one by one. The multilevel data embedding strategy is described as follows.
Step 5.2.1. Embed the secret data as:

$$d''_i = \begin{cases} p_1 & \text{if} \quad i = 1 \\ d'_i & \text{if} \quad -EL < d'_i < EL, 2 \leq i \leq M\times N \\ 2\times EL + w & \text{if} \quad d'_i = EL, 2 \leq i \leq M\times N \\ -2\times EL - w + 1 & \text{if} \quad d'_i = -EL, 2 \leq i \leq M\times N \end{cases} \qquad (5)$$

Step 5.2.2. *EL* is decreased by 1.

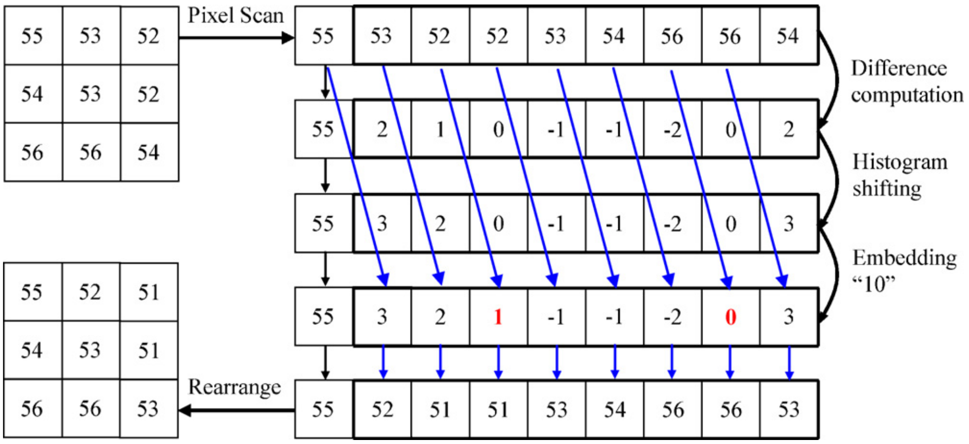

Fig. 6. Example of data embedding for *EL* = 0.

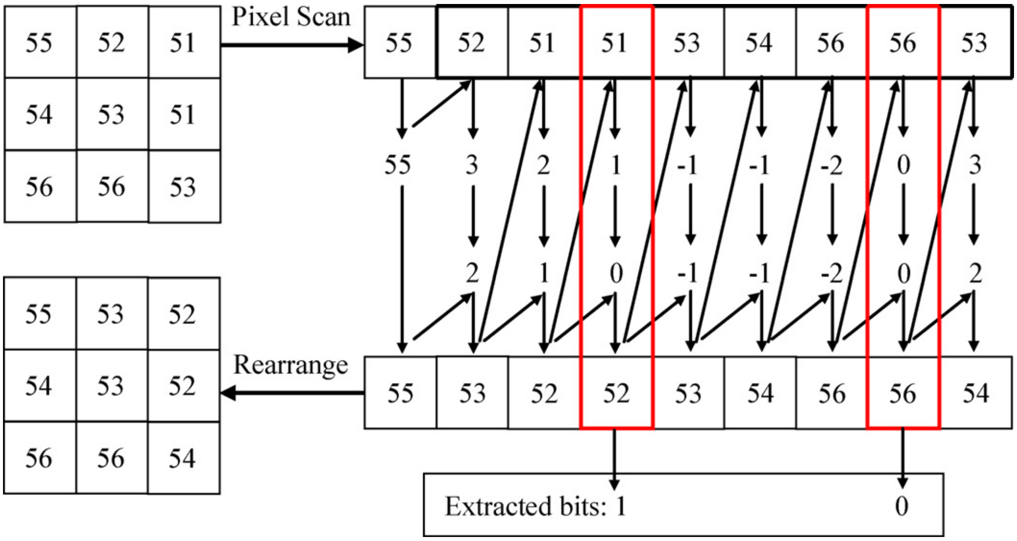

Fig. 7. Example of data extraction and image recovery for *EL* = 0.

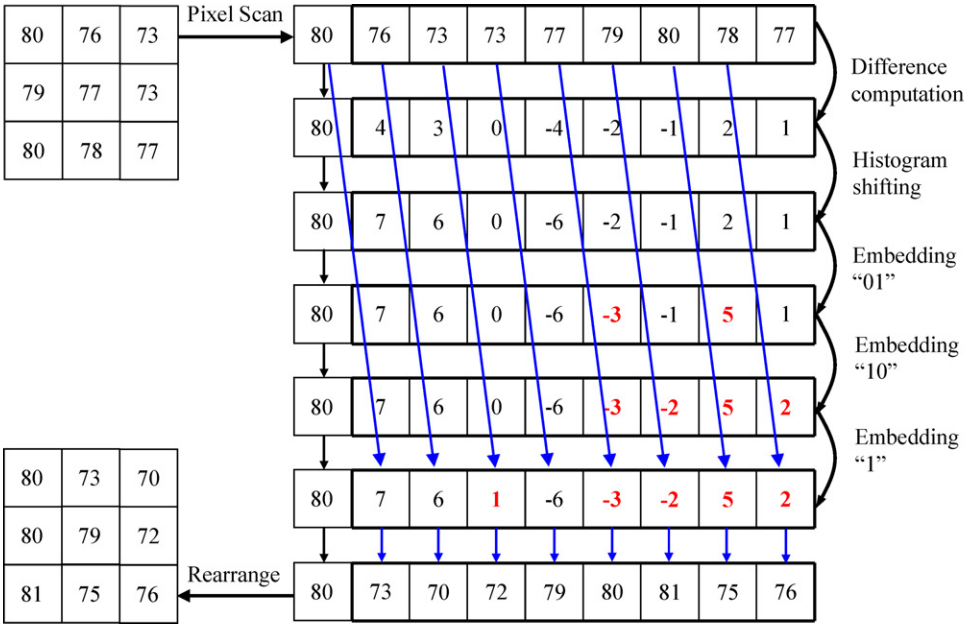

Fig. 8. Example of data embedding for *EL* = 2.

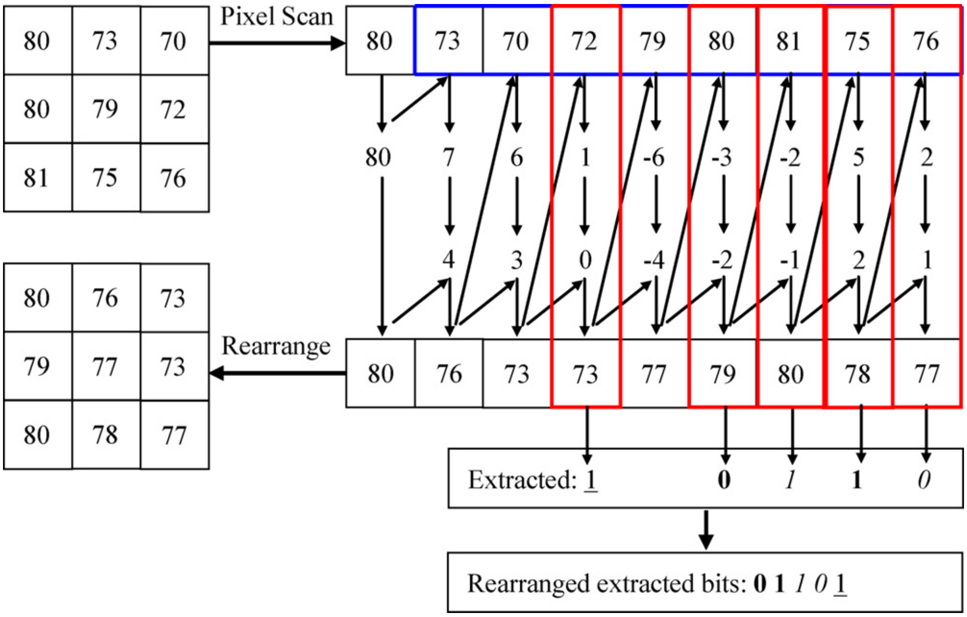

Fig. 9. Example of data extraction and image recovery for $EL = 2$.

Step 5.2.3. If $EL \neq 0$, execute *Step 5.2.1* and *Step 5.2.2* repeatedly. If $EL = 0$, execute Eq. (6) and then go to *Step 6*:

$$d''_i = \begin{cases} p_1 & \text{if} \quad i = 1 \\ d'_i + w & \text{if} \quad d'_i = 0, 2 \le i \le M \times N \\ d'_i & \text{if} \quad d'_i \ne 0, 2 \le i \le M \times N \end{cases} \tag{6}$$

The histogram modification strategy for $EL = 2$ is shown in Fig. 5(a)–(h), where the red and blue arrows correspond to embedding "0" and "1" respectively.

Step 6. Generate the marked pixels sequence p' as:

$$p'_i = \begin{cases} p_1 & \text{if} \quad i = 1 \\ p_{i-1} - d''_i & \text{if} \quad 2 \le i \le M \times N \end{cases} \tag{7}$$

Step 7. Rearrange p' and the marked image I' is obtained.

3.3. Data extraction and image recovery

The data extraction and image recovery is the inverse process of data embedding, and the details are described below.

Step 1. Inverse "S" scan I' into a pixel sequence p'_i $(1 \le i \le M \times N)$.

Step 2. Receive the EL parameter from the encoder via a secure channel. If $EL = 0$, then execute *Step 3* and *Step 4*. If $EL > 0$, execute *Step 5* and *Step 6*.

Step 3. For $EL = 0$, the host image pixels are recovered as:

$$p_i = \begin{cases} p'_1 & \text{if} \quad i = 1 \\ p'_i & \text{if} \quad p_{i-1} - p'_i \le 0, 2 \le i \le M \times N \\ p'_i + 1 & \text{if} \quad p_{i-1} - p'_i \ge 1, 2 \le i \le M \times N \end{cases} \tag{8}$$

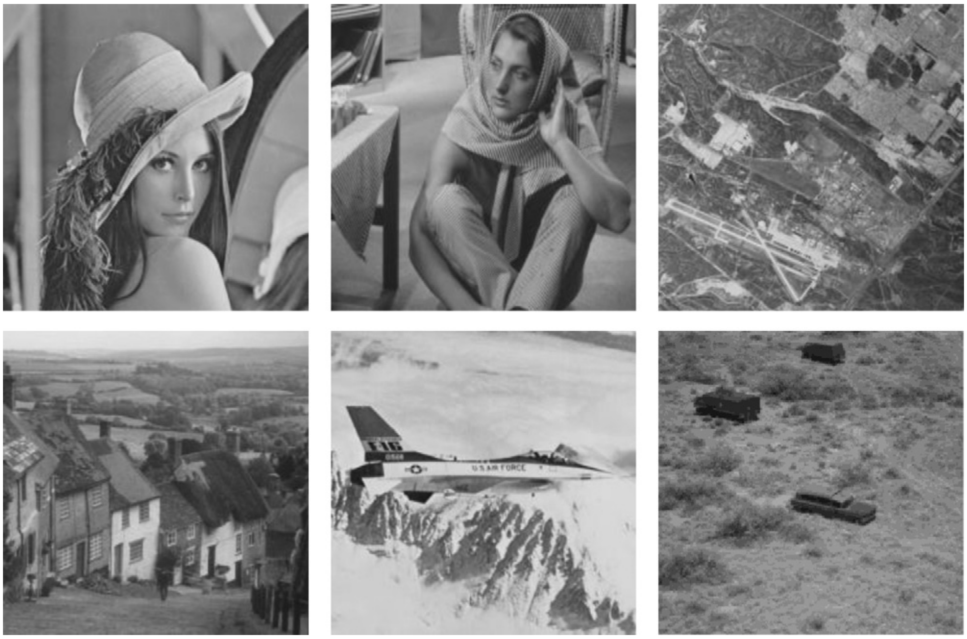

Fig. 10. Test images, Lena, Barbara, Aerial, Goldhill, Airplane, Car (from left to right, from top to bottom).

Table 1
Performance comparison of Li et al.'s method [16] and our scheme.

Method		Lena		Barbara		Aerial		Goldhill		Airplane		Car	
		Cap	PSNR	Cap	PSNR	Cap	PSNR	Cap	PSNR	Cap	PSNR	Cap	PSNR
APD$_1$	[16]	24976	51.14	16845	51.14	9247	51.13	18233	51.15	39621	51.14	30381	51.14
APD$_2$	[16]	48383	48.55	33113	48.41	18110	48.28	35868	48.44	70741	48.76	52311	48.72
Our	*EL* = 0	24976	51.14	16845	51.14	9247	51.15	18233	51.14	39621	51.14	30381	51.15
	EL = 1	71727	44.84	49062	44.61	27183	44.40	53255	44.65	101504	45.20	44859	44.68
	EL = 2	111629	41.38	78018	40.90	44532	40.49	84434	40.99	142567	42.00	88731	40.98
	EL = 3	143182	39.20	102547	38.49	60684	37.87	111683	38.62	170683	39.99	110976	38.63
	EL = 4	167336	37.68	122019	36.74	75864	35.95	134487	36.93	189356	38.58	140380	36.97
	EL = 5	185073	36.56	137108	35.38	90160	34.45	153114	35.65	201727	37.50	161904	35.75
	EL = 6	198326	35.68	148842	34.28	103499	33.23	168866	34.64	210377	36.63	180307	34.81
	EL = 7	208079	34.97	157873	33.34	115700	32.21	181954	33.83	217249	35.90	197015	34.08
	EL = 8	215528	34.37	165031	32.54	126803	31.35	192858	33.15	222450	35.28	209511	33.51
	EL = 9	221413	33.85	170783	31.82	137316	30.60	201810	32.57	226497	34.73	220801	33.05

Step 4. For *EL* = 0, the secret data is extracted as:

$$w = \begin{cases} 0 & \text{if} \quad p_{i-1} - p'_i = 0, 2 \le i \le M \times N \\ 1 & \text{if} \quad p_{i-1} - p'_i = 1, 2 \le i \le M \times N \end{cases} \tag{9}$$

That is, if coming across $p_{i-1} - p'_i = 0$ ($2 \le i \le M \times N$), a secret bit "0" is extracted. If $p_{i-1} - p'_i = 1$ ($2 \le i \le M \times N$), a "1" is extracted. Rearrange these extracted bits and the original secret sequence is obtained. After that, go to *Step 7*.

Step 5. For *EL* > 0, obtain the first host pixel as $p_1 = p'_1$. The marked differences are computed as:

$$d''_i = \begin{cases} p'_1 & \text{if} \quad i = 1 \\ p_{i-1} - p'_i & \text{if} \quad 2 \le i \le M \times N \end{cases} \tag{10}$$

Then the original differences are obtained as:

$$d_i = \begin{cases} d''_i - EL - 1 & \text{if} \quad d''_i > 2 \times EL + 1, 2 \le i \le M \times N \\ d''_i + EL & \text{if} \quad d''_i < -2 \times EL, 2 \le i \le M \times N \\ r & \text{if} \quad d''_i \in \{2 \times r, 2 \times r + 1\}, r = 0, 1, \ldots EL, 2 \le i \le M \times N \\ -r & \text{if} \quad d''_i \in \{-2 \times r, -2 \times r + 1\}, r = 1, \ldots EL, 2 \le i \le M \times N \end{cases} \tag{11}$$

Next the host pixel sequence is recovered as:

$$p_i = \begin{cases} p'_1 & \text{if} \quad i = 1 \\ p'_{i-1} - d_i & \text{if} \quad 2 \le i \le M \times N \end{cases} \tag{12}$$

Note Eqs. (10)–(12) are executed repeatedly, i.e., p_i ($2 \le i \le M \times N$) is recovered in advance, and then p_{i+1} is recovered with the aid of p_i. In other words, a sequential recovery strategy is utilized.

Step 6. For *EL* > 0, the secret data extraction is associated with *EL* + 1 rounds. First set the round index *R* = 1.

Step 6.1. Extract the data as:

$$w_R = \begin{cases} 0 & \text{if} \quad d''_i = 2 \times EL, 2 \le i \le M \times N \\ 0 & \text{if} \quad d''_i = -2 \times EL + 1, 2 \le i \le M \times N \\ 1 & \text{if} \quad d''_i = 2 \times EL + 1, 2 \le i \le M \times N \\ 1 & \text{if} \quad d''_i = -2 \times EL, 2 \le i \le M \times N \end{cases} \tag{13}$$

Step 6.2. *EL* is decreased by 1 and *R* is increased by 1.

Step 6.3. If $EL \neq 0$, execute *Steps 6.1 and 6.2* repeatedly. If *EL* = 0, execute Eq. (14) as:

$$w_R = \begin{cases} 0 & \text{if} \quad d''_i = 0, 2 \le i \le M \times N \\ 1 & \text{if} \quad d''_i = 1, 2 \le i \le M \times N \end{cases} \tag{14}$$

In Eq. (14) *R* is increased as *EL* + 1.

Step 6.4. Rearrange and concatenate the extracted data w_R ($1 \le R \le EL + 1$) as:

$$w = cat(w_1, w_2, \ldots, w_{EL+1}) \tag{15}$$

Hence, the hidden secret bits are obtained, and then go to *Step 7*.

Step 7. Rearrange the recovered sequence p_i ($1 \le i \le M \times N$) into the host image *I*.

3.4. *Examples*

Two examples for *EL* = 0 and *EL* = 2 are given to explain the above principles with a 3 × 3 block investigated.

3.4.1. *EL* = 0

3.4.1.1. Data embedding. The data embedding principle for *EL* = 0 is shown in Fig. 6. First, the 3 × 3 block is inverse "S" scanned and the difference histogram is constructed. Next, the histogram shifting is performed. Suppose the secret bits are "10". Thus the "1" can be hidden by changing the first difference 0–1, and the "0" is hidden by kept the second difference 0 not changed (marked as red). In this way, each marked pixel can be produced by its left neighbor subtracting the modified difference, as indicated by the blue arrows. Finally, rearrange these marked pixels into the marked block.

3.4.1.2. Data extraction and image recovery. As shown Fig. 7, the marked block is also inverse "S" scanned into a sequence first. As the first pixel is not changed during embedding, we have $p_1 = p'_1 = 55$. Second, the difference $d''_2 = p_1 - p'_2 = 3$. Obviously, its counterpart $d'_2 = 2$. Thus the original pixel associated with p'_2 is $p_2 = p_1 - d'_2 = 53$. Next, we obtain $d''_3 = p_2 - p'_3 = 2$, and its counterpart $d'_3 = 1$. Then $p_3 = p_2 - d'_3 = 52$. Repeat these operations for the remained marked pixels and all the host pixels are recovered. In this example, p_1 is obtained first, and then $p_2, p_3, \ldots, p_9$ are recovered consecutively.

As marked as red in Fig. 7, one bit secret data "1" is extracted from $p_3 - p'_4 = 1$, and a "0" is extracted from $p_7 - p'_8 = 0$.

3.4.2. *EL* = 2

3.4.2.1. Data embedding. As shown in Fig. 8, *EL* is set as 2 to describe the data embedding operations for *EL* > 0. The inverse "S" scan and the difference histogram construction are the same as those in *EL* = 0. Next, the histogram is shifted as follows. As *EL* = 2, the differences larger than 2 is added by 3 (i.e., *EL* + 1), and those smaller than −2 are subtracted by 2 (i.e., *EL*). For example, $d'_2 = d_2 + 3 = 7$, $d'_5 = d_5 - 2 = -6$.

Now the secret data can be embedded. Suppose the secret bits are "01101". In the first round, only $d'_6 = -2$ and $d'_8 = 2$ are investigated for falling into [−2, 2] (i.e., [−*EL*, *EL*]). As "0" and "1" are embedded in d'_6 and d'_8 respectively, we obtain the marked

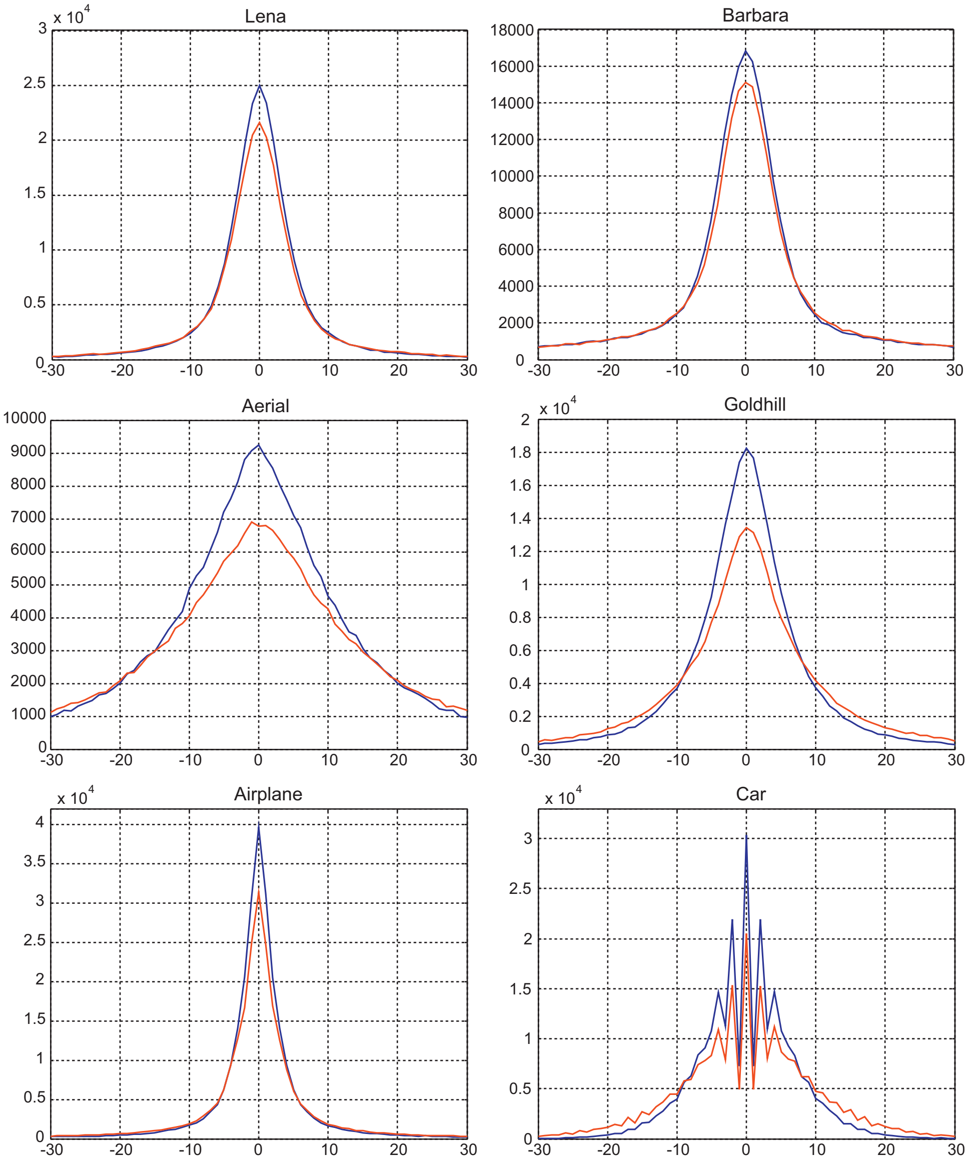

Fig. 11. Histogram bins of ($b(-30)$, $b(30)$) obtained by Kim et al.'s scheme [17] (marked as red) and our scheme (blue) on Lena, Barbara, Aerial, Goldhill, Airplane, Car (from left to right and from top to bottom). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

differences $d''_6 = -2 - 1 = -3$ and $d''_8 = 2 + 3 = 5$. In the second round, only $d'_7 = -1$ and $d'_9 = 1$ are investigated for falling into $[-1, 1]$ (i.e., $[-EL+1, EL-1]$). As "1" and "0" are embedded in d'_7 and d'_9 respectively, we obtain $d''_7 = -1 - 1 = -2$ and $d''_9 = 1 + 1 = 2$. In the third round, only d'_4 is investigated for equaling 0. As "1" is embedded in d'_4, we obtain $d''_4 = 0 + 1 = 1$. All the marked differences are marked as red. Now each marked pixel can be produced by its left neighbor host pixel subtracting the modified difference, as indicated by the blue arrows. That is, as $d'' = [80, 7, 6, 1, -6, -3, -2, 5, 2]$ are produced, the marked pixels are obtained as $p'_1 = p_1$, $p'_2 = p_1 - d''_2 = 80 - 7 = 73$, $p'_3 = p_2 - d''_3 = 76 - 6 = 70, \ldots, p'_9 = p_8 - d''_9 = 78 - 2 = 76$. At last, the marked block is obtained by rearranging these marked pixels.

3.4.2.2. Data extraction and image recovery. The image pixel recovery for $EL=2$ is shown in Fig. 9. The marked block is also inverse "S" scanned into a sequence first. Obviously, the first pixel is not changed during embedding. Second, the difference $d''_2 = p_1 - p'_2 = 80 - 73 = 7$. Its counterpart $d'_2 = 7 - (EL + 1) = 4$. Thus the second host pixel is recovered as $p_2 = p_1 - d'_2 = 76$. Next, we obtain $d''_3 = p_2 - p'_3 = 6$, and its counterpart $d'_3 = 3$. Then $p_3 = p_2 - d'_3 = 73$. Sequentially repeat these operations for the remained marked pixels and all the host pixels are recovered.

For $EL=2$, the secret data is extracted with three (i.e., $EL+1$) rounds. In the first round, EL is set as 2. As $p_5 - p'_6 = -2 \times EL + 1 = -3$, a secret bit "0" is extracted from p_5 and p'_6. Besides, as $p_7 - p'_8 = 2 \times EL + 1 = 5$, a "1" is extracted from p_7 and p'_8. These two secret bits $\mathrm{w}_1 =$ "**01**" are indicated as bold.

In the second round, EL is decreased by 1 and thus $EL=1$. As $p_6 - p'_7 = -2 \times EL = -2$, a "1" is extracted from p_6 and p'_7. Besides, as $p_8 - p'_9 = 2 \times EL = 2$, a "0" is extracted from p_8 and p'_9. The $w_2 =$ "10" are indicated as italic.

In the third round, EL is further decreased by 1 and thus $EL=0$. As $p_3 - p'_4 = 1$, a "1" is extracted from p_3 and p'_4. The $w_3 =$ "1" is indicated as underlined.

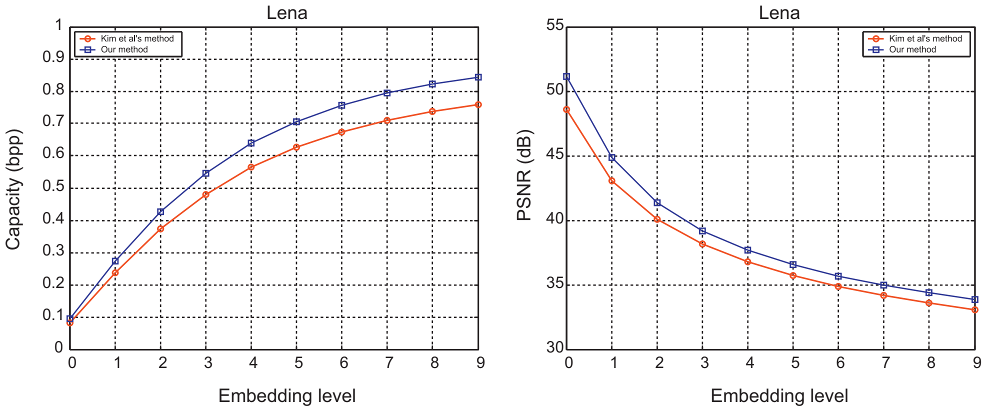

Fig. 12. Comparison of Kim et al.'s scheme [17] and our scheme on Lena image.

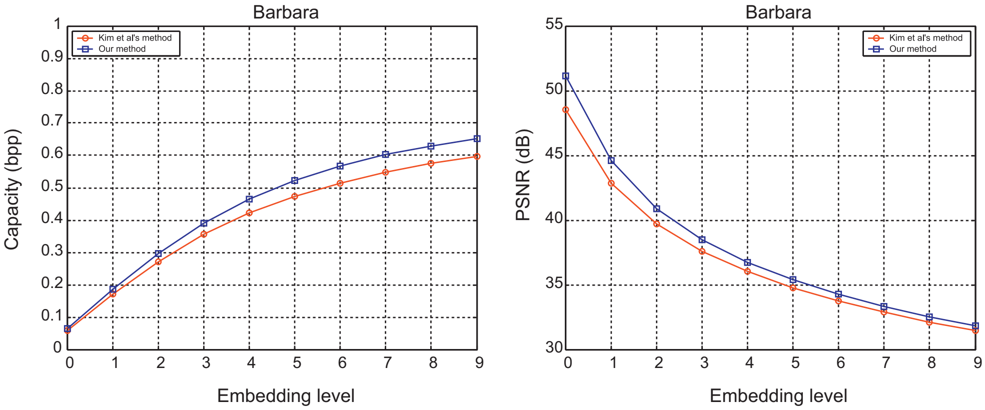

Fig. 13. Comparison of Kim et al.'s scheme [17] and our scheme on Barbara image.

The last step is to rearrange all the extracted bits as $w = cat(w_1, w_2, w_3) =$ "01101". It is exactly the same as the original secret data.

4. Discussion

4.1. Capacity estimation

The embedding capacity of our scheme is determined by two factors, the embedding level and the peak points around $b(0)$. If no overflow or underflow occurs, the capacity *Cap* (bit) can be computed as:

$$Cap = \begin{cases} b(0) & \text{if} \quad EL = 0 \\ \displaystyle\sum_{k=-EL}^{EL} b(k) & \text{if} \quad EL > 0 \end{cases} \tag{16}$$

4.2. Overflow and underflow prevention

Given a large *EL*, the operations of histogram bins empty and shifting may cause overflow (i.e., $p' > 255$) or underflow (i.e., $p' < 0$). Actually, we can predict when they appear. The overflow or underflow first appears on the pixels with values near 255 or 0. In

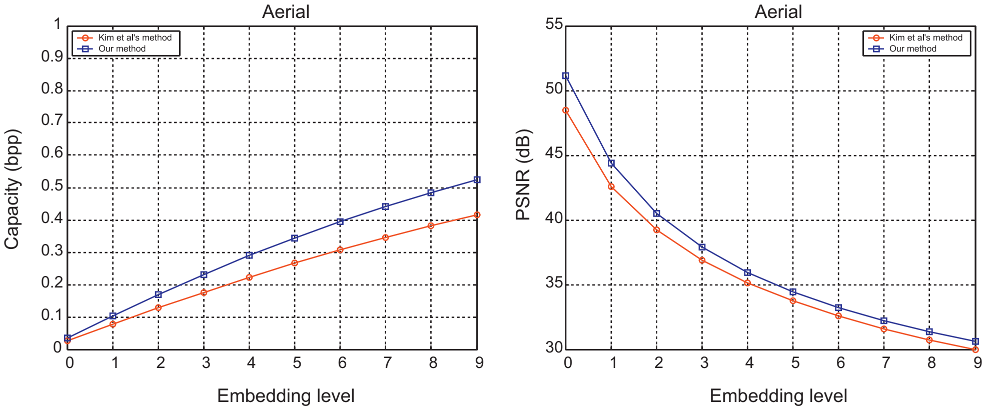

Fig. 14. Comparison of Kim et al.'s scheme [17] and our scheme on Aerial image.

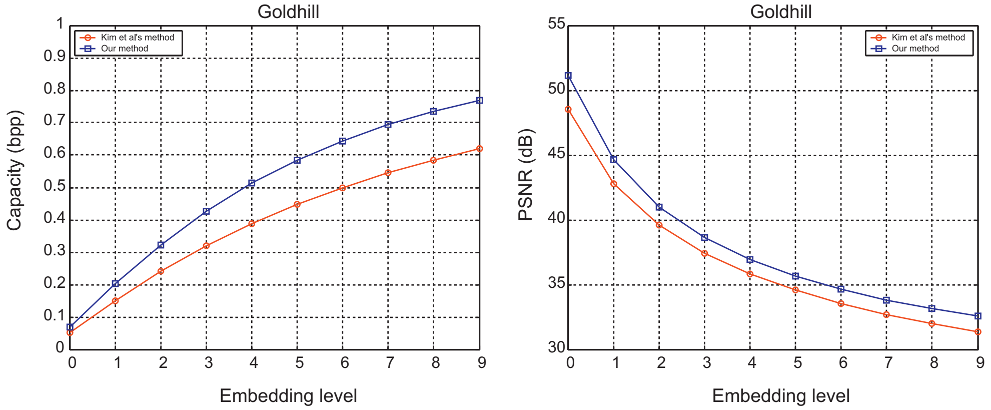

Fig. 15. Comparison of Kim et al.'s scheme [17] and our scheme on Goldhill image.

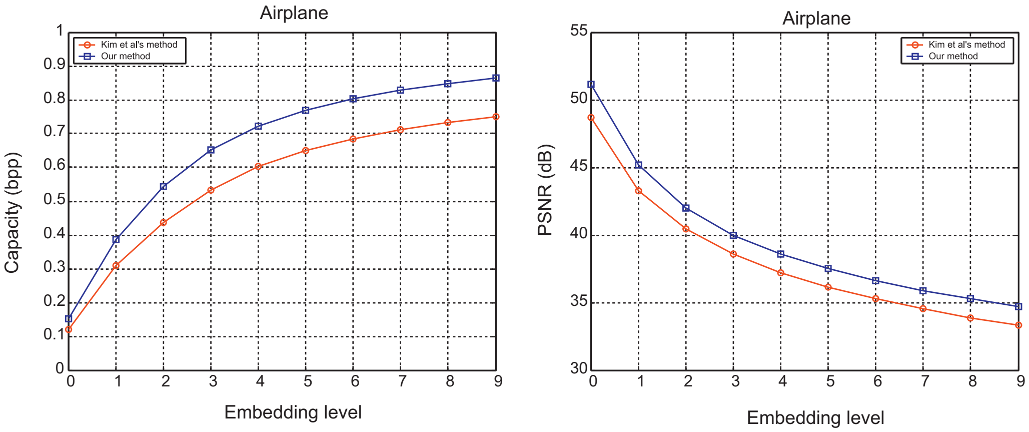

Fig. 16. Comparison of Kim et al.'s scheme [17] and our scheme on Airplane image.

particular, suppose $p_{\max}$ and $p_{\min}$ represent the maximum and minimum of the pixel values in I respectively. In the worst case, the distortions on $p_{\max}$ and $p_{\min}$ can be computed as:

$$\begin{cases} p'_{\max} = p_{\max} + EL + 1 \\ p'_{\min} = p_{\min} - EL \end{cases} \tag{17}$$

where $p'_{\max}$ and $p'_{\min}$ represent the marked pixels. When $p'_{\max} \leq 255$ and $p'_{\min} \geq 0$, no overflow or underflow occurs, and consequently EL must be set as:

$$EL \leq \mathbf{min}(254 - p_{\max}, p_{\min}) \tag{18}$$

That is, EL should be no larger than the minimum of $254 - p_{\max}$ and $p_{\min}$. In other words, if a host pixel with value belonging to [0, $EL - 1$], the underflow may appear on it. If belonging to [$255 - EL$, 255], the overflow may appear.

In this work, the embedding level is tested as integers which are less than 10. This is because with the EL increasing, the overflow and underflow problem becomes increasingly prevalent. As a result, a lot of pixels with boundary values (i.e., near 255 or 0) in the cover image cannot be used for data embedding. Hence the capacity will not be enhanced any longer. Moreover, the length of the compressed location map is increased at the same time. As the

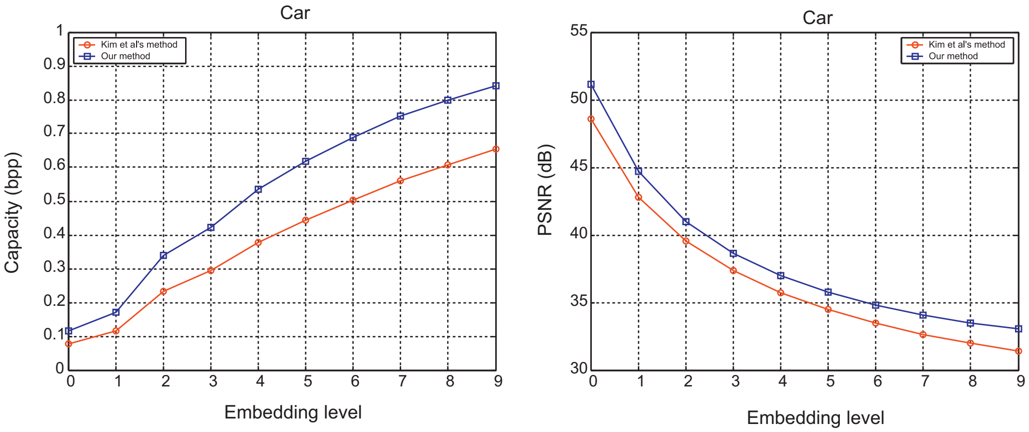

Fig. 17. Comparison of Kim et al.'s scheme [17] and our scheme on Car image.

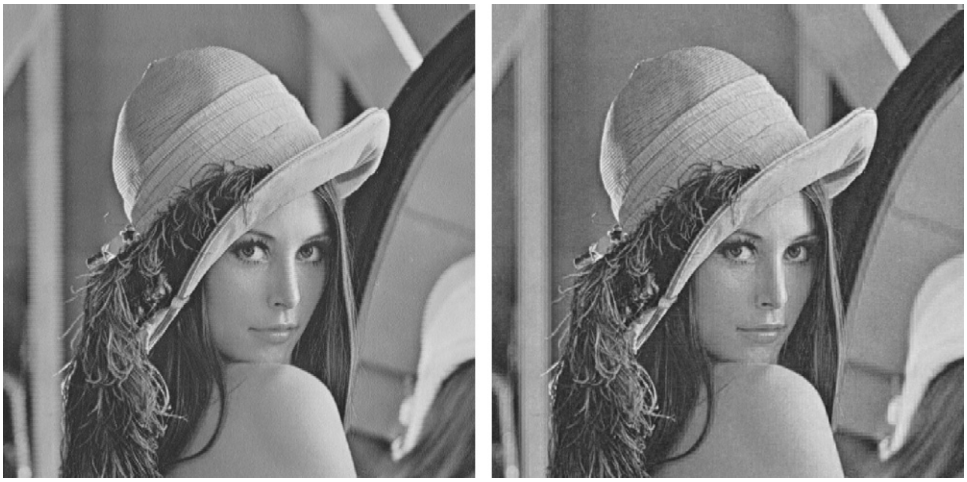

Fig. 18. Marked Lena images obtained by our scheme (left: 221413 bits hidden, 33.85 dB) and Kim et al.'s scheme [17] (right: 198838 bits hidden, 33.06 dB) with $EL = 9$.

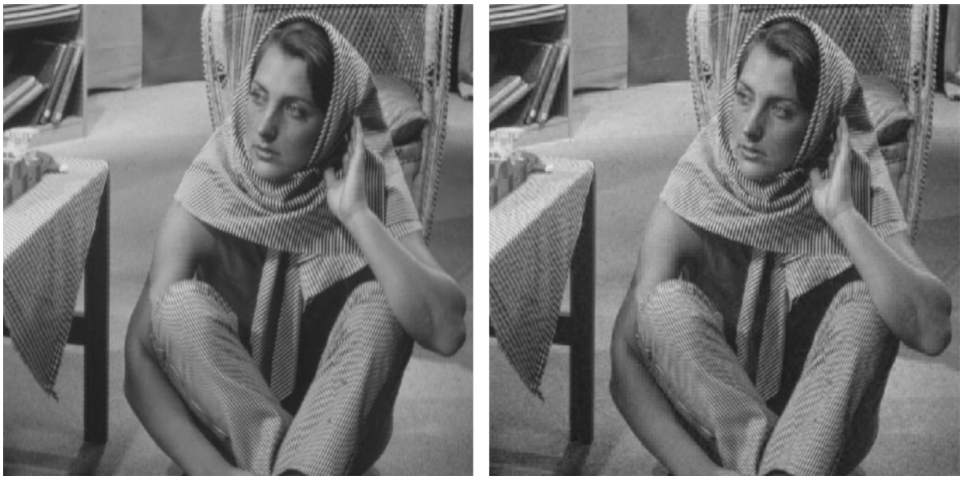

Fig. 19. Marked Barbara images obtained by our scheme (left: 170783 bits hidden, 31.82 dB) and Kim et al.'s scheme [17] (right: 156648 bits hidden, 31.45 dB) with $EL = 9$.

compressed location map is also hidden the cover image, the valid capacity for confidential data hiding is also decreased. Actually, *EL* selection depends on the cover image content. According to the experimental results, *EL* can be set as an integer less than 10 on the average for providing a maximum value of valid capacity.

In this work, an $M \times N$ location map *LM* is used for overflow and underflow prevention. Before data embedding, *I* is preprocessed. If a pixel with value falling into $[0, EL - 1]$ or $[255 - EL, 255]$, it is excluded for data embedding and a “1” is recorded in the *LM*, otherwise a “0” recorded. After all pixels processed, a binary *LM* is generated. Obviously, a larger *EL* corresponds to more “1”s and

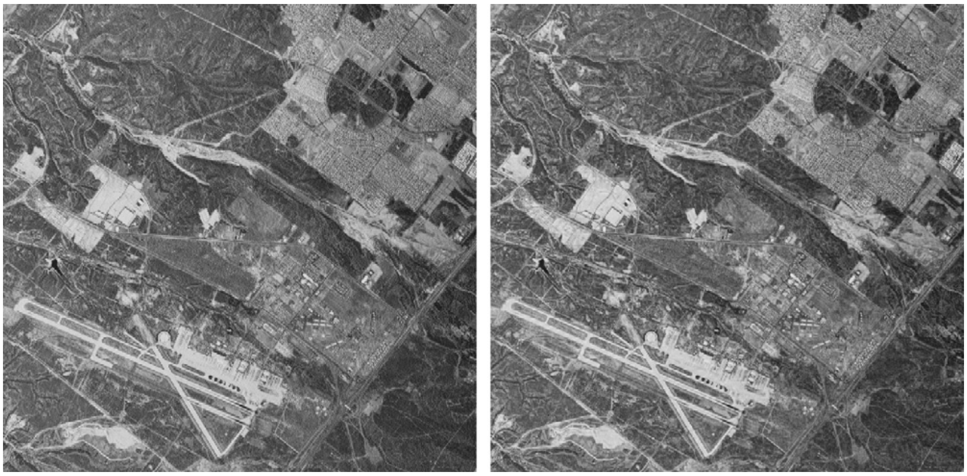

Fig. 20. Marked Aerial images obtained by our scheme (left: 137316 bits hidden, 30.60 dB) and Kim et al.'s scheme [17] (right: 108956 bits hidden, 29.99 dB) with $EL = 9$.

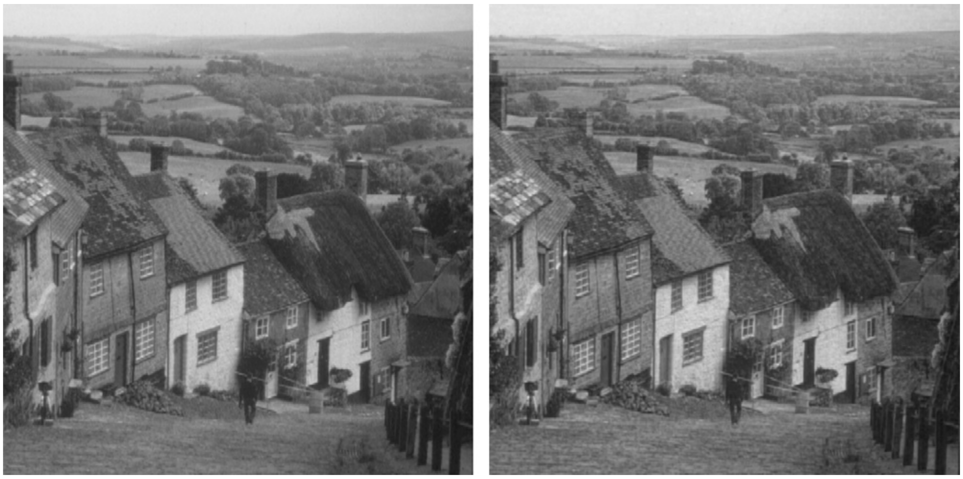

Fig. 21. Marked Goldhill images obtained by our scheme (left: 201810 bits hidden, 32.57 dB) and Kim et al.'s scheme [17] (right: 162518 bits hidden, 31.35 dB) with $EL = 9$.

fewer "0"s in it. Next, the LM is losslessly compressed. In our case, the arithmetic coding is used for its high efficiency. The compressed map LM_c can be also hidden in I. In particular, I is segmented into two parts, I_1 and I_2 for embedding w and LM_c respectively. That is, some pixels in I_2 are selected according to a secret key, and their least significant bits (LSB) are replaced by LM_c, and these LSB bits are hidden in I_1 concatenated with w. In decoder, the same key is used to retrieve the selected pixels' LSB bits in I_2 and thus LM_c is reconstructed. After lossless decompression, LM is further obtained. Then we can extract w and the original LSB bits of the selected pixels in I_2. Finally, the host image can be recovered by removing w from I_1 and replacing LSB bits of the selected pixels in I_2 with the latter part of data extracted from I_1.

5. Experimental results and comparison with other schemes

As shown in Fig. 10, six 512 × 512 gray level images are selected as test images. Table 1 lists the capacity (bit) and PSNR (dB) values of the proposed scheme with various ELs. The PSNR (peak signal to noise ratio) is computed as:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{\mathrm{MSE}} (\mathrm{dB}) \tag{19}$$

where MSE (mean square error) between I and I' is defined as:

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (I(i,j) - I'(i,j))^2 \tag{20}$$

From Table 1, we find a larger EL leads to a larger capacity. Even for $EL = 9$, the average PSNR is higher than 30 dB. As human eyes are not sensitive to distortions when PSNR > 30 dB, the marked images' visual qualities are acceptable.

Our scheme is compared with two state-of-the-art methods proposed by Li et al.'s method [16] and Kim et al.'s method [17]. The reason to compare with the method of [16] lies in that both methods are based on inverse scan order. That is, the difference histogram is exactly the same whereas a multilevel histogram modification strategy is used in our scheme. Hence the performance improvement achieved by this novel strategy is validated in our work. In addition, the reason to compare with the method of [17] lies in that both methods adopt similar multilevel histogram modification strategies. However, as our scheme is based on a pixel-wise differential mechanism instead of block-wise processing in [17], the capacity and the quality of marked images are enhanced simultaneously as proved in the following experimental results.

First, our scheme is compared with Li et al.'s method [16] for both are based on APD histogram modification. In [16], only two cases, APD_1 and APD_2 are provided for data embedding. Here APD_1

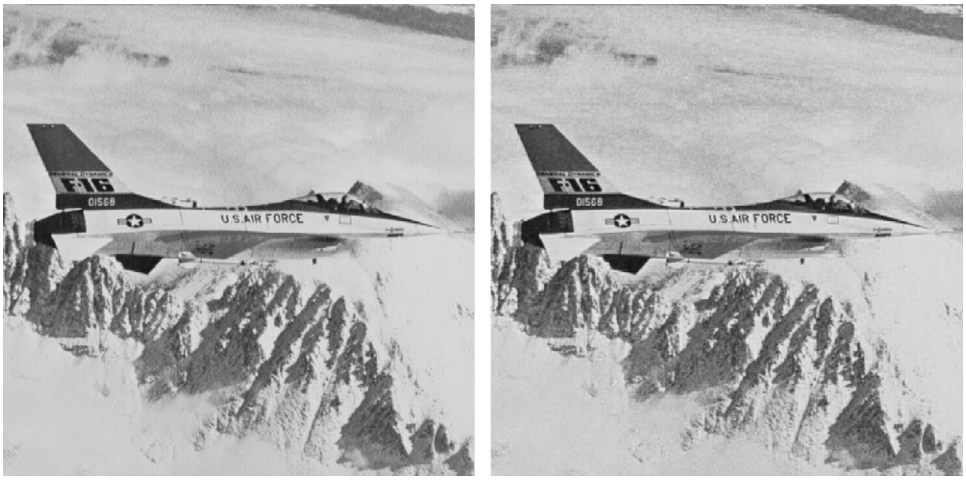

Fig. 22. Marked Airplane images obtained by our scheme (left: 226497 bits hidden, 34.73 dB) and Kim et al.'s scheme [17] (right: 196634 bits hidden, 33.31 dB) with $EL = 9$.

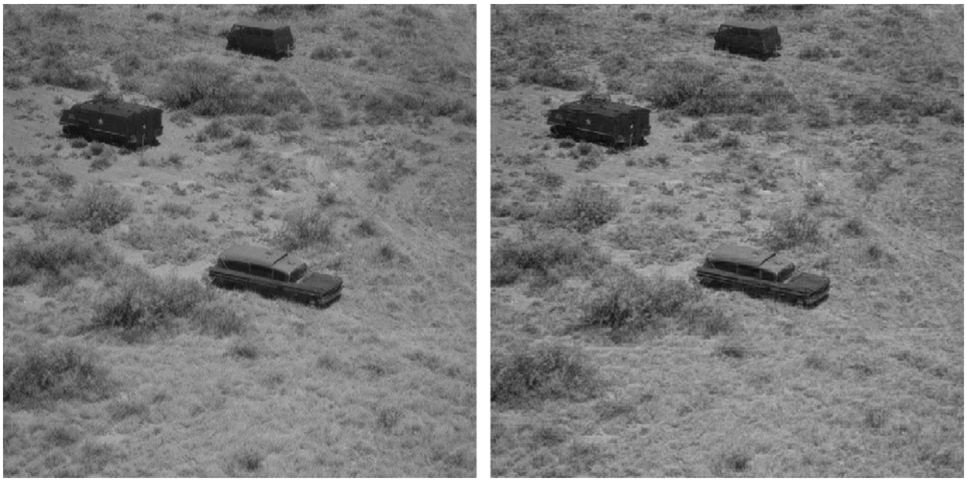

Fig. 23. Marked Car images obtained by our scheme (left: 220801 bits hidden, 33.05 dB) and Kim et al.'s scheme [17] (right: 171244 bits hidden, 31.40 dB) with $EL = 9$.

and APD_2 denote one and two pairs of peak-zero points are used respectively. The comparison results are also given in Table 1. Our scheme can provide a higher capacity than Li et al.'s method with good marked images quality.

Next, our scheme is compared with Kim et al.'s method [17]. Although both are based on multilevel histogram modification, the histogram construction mechanisms are different. In general, the capacity of difference histogram modification is jointly affected by the total number of differences and the their concentricity to $b(0)$. In [17], the differences are computed based on subimages' correlation and hence the number of differences is determined by the number of subimages. For example, if a 512×512 host image is subsampled into 16 equal-sized subimages, there are $512 \times 512 \times 15/16 = 245760$ differences produced. In contrast, there are $512 \times 512 - 1 = 262143$ differences produced in our scheme. The histogram bins belonging to $[b(-30), b(30)]$ obtained by [17] and our scheme are shown in Fig. 11. Obviously, more differences in our histograms are concentrated around $b(0)$. As a result, a larger capacity can be provided in our scheme than in [17]. Moreover, $b(-1)$ is emptied after embedding in [17] for the shifting leftward is one level farther than that in our scheme, and consequently the introduced distortions are more serious.

The performance comparisons of our scheme (marked as blue) and Kim et al.'s method (marked as red) are shown in Figs. 12–17. The horizontal axis denotes the EL set from 0 to 9. The vertical axes of capacity and PSNR are normalized as [0, 1.0] bpp (bit per pixel) and [30, 55] dB, respectively. In these experiments, the host images are partitioned into 16 equal-sized subimages in [17]. The six marked images obtained by our scheme and Kim et al.'s scheme are shown in Figs. 18–23. All these results demonstrate not only the capacities but also the PSNRs in our method are improved. In other words, even though more secret data embedded in our scheme, the marked images quality is still better than those in [17].

6. Conclusions

A reversible data hiding scheme is proposed in this paper. The multilevel histogram modification is employed for data embedding. On one hand, a higher capacity is provided compared with one or two level histogram modification based methods. On the other hands, as secret data is embedded in differences of adjacent pixels values, the marked images quality is improved compared with that in previous multilevel histogram modification based work.

Acknowledgments

The authors thank the editor and the anonymous reviewers for their constructive comments and valuable suggestions for readability improvement. This work is financially supported by the National Scientific Fund of China (Nos. 61003255 and 61071128).

References

[1] Pan JS, Huang HC, Jain LC. Information hiding and applications. Berlin-Heidelberg, Germany: Springer; 2009.
[2] Cheddad A, Condell J, Curran K, Kevitts PM. Digital image steganography: survey and analysis of current methods. Signal Processing 2010;90(3):727–52.
[3] Tian J. Reversible data embedding using a difference expansion. IEEE Transactions on Circuits and Systems for Video Technology 2003;13(8):890–6.
[4] Alattar AM. Reversible watermark using the difference expansion of a generalized integer transform. IEEE Transactions on Image Processing 2004;13(8):1147–56.
[5] Weng S, Zhao Y, Pan JS. A novel reversible data hiding scheme. International Journal of Innovative Computing, Information and Control 2008;4(2):351–8.
[6] Chang CC, Lu TC. A difference expansion oriented data hiding scheme for restoring the original host images. The Journal of Systems and Software 2006;79(12):1754–66.
[7] Lee CC, Wu HC, Tsai CS, Chu YP. Adaptive lossless steganographic scheme with centralized difference expansion. Pattern Recognition 2008;41(6):2097–106.
[8] Lin CC, Hsueh NL. A lossless data hiding scheme based on three-pixel block differences. Pattern Recognition 2008;41(4):1415–25.
[9] Lu TC, Chang CC. Lossless nibbled data embedding scheme based on difference expansion. Image and Vision Computing 2008;26(5):632–8.
[10] Hsiao JY, Chan KF, Chang JM. Block-based reversible data embedding. Signal Processing 2009;89(4):556–69.
[11] Lou DC, Hu MC, Liu JL. Multiple layer data hiding scheme for medical images. Computer Standards & Interfaces 2009;31(2):329–35.
[12] Weng S, Zhao Y, Pan JS, Ni R. Reversible data hiding using the companding technique and improved DE method. Circuits Systems Signal Process 2008;27(2):229–45.
[13] Ni Z, Shi Y, Ansari N, Su W. Reversible data hiding. In: IEEE proceedings of ISCAS'03, vol. 2. 2003. p. II-912–915.
[14] Huang HC, Fang WC, Tsai IT. Reversible data hiding using histogram-based difference expansion. IEEE International Symposium on Circuits and Systems 2009:1661–4.
[15] Tsai P, Hu YC, Yeh HL. Reversible image hiding scheme using predictive coding and histogram shifting. Signal Processing 2009;89(6):1129–43.
[16] Li YC, Yeh CM, Chang CC. Data hiding based on the similarity between neighboring pixels with reversibility. Digital Signal Processing 2009, doi:10.1016/j.dsp.2009.10.025.
[17] Kim KS, Lee MJ, Lee HY, Lee HK. Reversible data hiding exploiting spatial correlation between sub-sampled images. Pattern Recognition 2009;42(11):3083–96.
[18] Lin CC, Tai WL, Chang CC. Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recognition 2008;41(12):3582–91.
[19] Gao X, An L, Li X, Tao D. Reversibility improved lossless data hiding. Signal Processing 2009;89(10):2053–65.
[20] Tai WL, Yeh CM, Chang CC. Reversible data hiding based on histogram modification of pixel differences. IEEE Transactions on Circuits and Systems for Video Technology 2009;19(6):906–10.
[21] Yang B, Schmucker M, Funk W, Brush C, Sun S. Integer DCT-based reversible watermarking for images using companding technique. Proceedings of

SPIE, Security, Steganography and Watermarking of Multimedia Contents 2004;5306:405–15.

[22] Yang B, Schmucker M, Niu X, Busch C, Sun SH. Integer-DCT-based reversible image watermarking by adaptive coefficient modification. Proceedings of SPIE, Security, Steganography, and Watermarking of Multimedia Contents 2005;5681:218–29.

[23] Xuan G, Shi YQ, Yao Q, Ni Z, Yang C, Gao J, et al. Lossless data hiding using histogram shifting method based on integer wavelets. International Workshop on Digital Watermarking, Lecture Notes in Computer Science 2006;4283: 323–32.

[24] Lu ZM, Wang JX, Liu BB. An improved lossless data hiding scheme based on image VQ-index residual value coding. Journal of Systems and Software 2009;82(6):1016–24.

[25] Wang JX, Lu ZM. A path optional lossless data hiding scheme based on VQ index table joint neighboring coding. Information Sciences 2009;179(19):3332–48.

[26] Yang B, Lu ZM, Sun SH. Reversible watermarking in the VQ-compressed domain. In: Proceedings of the fifth international conference on visualization, imaging, and image processing. 2005. p. 289–303.

[27] Chang CC, Lin CY, Fan YH. Lossless data hiding for color images based on block truncation coding. Pattern Recognition 2008;41(7):2347–57.

[28] Mobasseri BG, Cinalli D. Lossless watermarking of compressed media using reversibly decodable packets. Signal Processing 2006;86(5):951–61.

[29] Li LD, Guo BL. Localized image watermarking in spatial domain resistant to geometric attacks. International Journal of Electronics and Communications (AEU) 2009;63(2):4. p. 123–31.

[30] Wang ZH, Kieu TD, Chang CC, Li MC. A novel information concealing method based on exploiting modification direction. Journal of Information Hiding and Multimedia Signal Processing 2010;1(1):1–9.

[31] Besdok E. Hiding information in multispectral spatial images. International Journal of Electronics and Communications (AEU) 2005;59:15–24.