Vietnam Journal of Computer Science (2018) 5:185–196
https://doi.org/10.1007/s40595-018-0114-z

REGULAR PAPER

A new multilevel reversible bit-planes data hiding technique based on histogram shifting of efficient compressed domain

Rashid Abbasi[2] · Bin Luo[1] · Gohar Rehman[3] · Haseeb Hassan[2] · Muhammad Shahid Iqbal[2] · Lixiang Xu[2]

Received: 17 May 2017 / Accepted: 12 May 2018 / Published online: 29 May 2018

Abstract
In this paper, we proposed a new technique for reversible data hiding based on efficient compressed domain with multiple bit planes. We conducted a sequence of experiments to use block division scheme to appraise the result with different parameters and amended the probability of zero point in every block of histogram. This scheme attained more embedding capacity and high-quality of stego-image. Experimental consequences prove that the proposed method effectively achieved the objective of high embedding capacity and sustaining the quality of image.

Keywords Reversible data hiding · Histogram shifting · Difference expansion

1 Introduction

Machine learning is a critical technology in computer science which contains pattern recognition [1], image matching [2], data hiding [3] and so on. Data hiding has been extensively used in the fields of fingerprint, authentication, ownership fortification and furtive communication. The focal objective of data hiding is to write the secret data into the host media (called cover media), i.e., image, video, audio, and text. In recent years, countless data hiding techniques have been introduced and mostly are irreversible data hiding [4]. Irreversible data hiding, the cover image is absolutely damaged and cannot be renovated after extraction. Reversible data hiding (*RDH*) recover embedded secret information from stego image that is used in medical imaging [5], image authentication [6,7], military image [3], satellite image [4], multimedia archive management [8] and data coloring in cloud [9]. Two vital requirements of the *RDH* scheme is to preserve the worth of image and provide excessive embedding capacity (*EC*).

RDH [10–12], intentions to inscribe secret information and slightly variates pixel value of original media to entirely recover from marked image. *RDH* [10,11,13–15] has been distributed into three realms, i.e., spatial domain, frequency domain and compressed domain [16]. The spatial domain has been further categorized into *LSB* substitution [13], histogram shifting (*HS*) [11,17–20], and difference expansion (*DE*) [10,13,14,16,21,22]. In 2003, Tian et al. [10] premeditated the variance among two neighbor pair of pixels to embed the information. Thodi and Rodriguez [16] adapted an expansion technique based on error prediction to envisage the pixel value. In 2006, Ni et al. [17] presented the histogram shifting method to get high *PSNR*, but low *EC*. In 2009, Tai et al. [19] proposed the binary tree structure to discover the peak point that is used to write data. Kim et al. [11] employed center pixel as the reference pixel to reckon the difference of sub-image conversely, but center pixel cannot be used to hide the information. Luo et al. [13] enhanced Kim's technique, but the reference pixel does not hold the data. In 2010, Li et al. [18] designed the variation among neighbor pixel based on *HS*. In 2011, Hong and Chen [23] divided the host image into two blocks, but data write

✉ Bin Luo
luobin@ahu.edu.cn

Rashid Abbasi
rashidd.abbasi@gmail.com

1 Key Lab of Intelligent Computing and Signal Processing of MOE and School of Computer and Technology, Anhui University, Hefei 230039, People's Republic of China

2 School of Computer and Technology, Anhui University, Hefei 230039, People's Republic of China

3 College of Computer Science Technology, Chongqing University, Chongqing, People's Republic of China

in only the smooth block region. In 2013, Huang et al. [24] enhanced *EC* to amend Hong and Chen techniques. Lin et al. [25] planned multilevel hiding procedure to obtain greater *EC* and better *PSNR*. Tsai et al. [26] combined *HS* and predictive coding (*PC*) to examine the elementary pixel in every block of residual image, but needed more space to record extra informations.

There were some foremost problems with preceding techniques. *RDH* having great performance on *DE* and *HS* techniques to retain location map, extraction information, image restoration and overflow/underflow problems [17]. Extra bits will increase when smallest point is used to substitute for zero point because location and value for every smallest point is recoded, which consumes significant amount of EC for achieving lossless recovery [27]. *HS* does not work on equal histogram bins of an image and numerous sets of peak points can be employed for hiding information. Previously when *EC* increased the *PSNR* decreased, while *EC* is decreased *PSNR* is increased on blocks but it is time consuming to consider block that have minimum number of zero point to shift histogram. The key motivation is to sustain the stability among the *EC* and image *PSNR* and only consider those blocks that have zero point and otherwise contravene those blocks in the embedding process.

In this paper, we have proposed a new technique for reversible data hiding based on efficient compressed domain with multiple bit planes (*CDMBP*). A collective approach of *RDH* is to delimit a free hiding space then we write the information in an image; consequently, we persuaded the properties of pre-reserving space for data embedding based on efficient compressed domain that play a vital role in data embedding for the reason that they make available additional space for data embedding, because low compression ratio cannot provide much space for the embedded region to control extra bits that increase during small block division and cannot be accommodated into embedded regions. Therefore, the *EC* graph deteriorates on small blocks size. Furthermore, the possession of multiple bit planes extraction of individual pixels to construct a new image is induced and the overflow/underflow problem is overcome at the same time to escalate the hiding capacity to the extreme level and preserve the high quality of the marked image. We ascertained the probability of peak and zero point to improve the hiding capacity in each non-overlapping blocks in the cover image. The rest of this paper is structured as follows. Section 2 describes the earlier techniques. Section 3 conveys an explanation of the proposed hiding method, provides the testing results in Sect. 4 and concludes the observations in Sect. 5.

2 Related work

2.1 Ni et al.'s method

In Ni et al. [17], an embedded process, the algorithm first scansthe whole image to find the peak point (*PP*) (extreme occurrence of pixels assume) and *ZP* (lowest pixel assume) based on the strength of pixels in the histogram.

$$H\left(G'_{(x,y)}\right) = H\left(G_{(x,y)}\right) + 1, \tag{1}$$

where $H(G'_{(x,y)})$ is the modified grayscale value,

$$G'_{(x,y)} = \begin{cases} G_{(x,y)} + 1, & \text{if} \quad PP < G_{(x,y)} < ZP, \\ G_{(x,y)} + m, & \text{if} \quad G_{(x,y)} = PP, \\ G_{(x,y)}, & \text{otherwise.} \end{cases} \tag{2}$$

$G_{(x,y)}$ is the original pixel grayscale value and m is secret information. If $G_{(x,y)}$ pixel value is amid *PP* and *ZP* formerly, all pixels $G_{(x,y)}$ will be shifted one step closer toward the *ZP*. If $G_{(x,y)}$ is equivalent to *PP*, then secret data m will be written in that pixel place. The value of pixels will be unchanged if $G_{(x,y)}$ pixel value is not among *PP* and *ZP*. The number of each pixel corresponds to the number of pixels which are linked with *PP*, for example, the peak-point $PP = 1$ and the zero-point $ZP = 9$. The entire image is scanned from top to bottom and right to left; furthermore, fluctuating the histogram $(1 - 9)$ grayscale values by one step in the right-hand side and leaving vacant *1* grayscale value visible in Fig. 1, the pixel value $G_{(0,1)} = 1$ is identical to *PP* its mean and that pixel value can be set for the secret message.

Ni et al.'s scheme *EC* of image "Lena" is 2787 pixels which can be used to implant the information; however, it

	0	1	2	3	4	5	6	7	8	9
0	4	1	5	1	4	2	1	2	1	7
1	8	2	4	2	3	1	2	0	4	2
2	1	2	1	4	5	6	0	1	3	1
3	3	1	4	7	2	1	2	4	1	3
4	1	3	0	3	5	0	2	1	2	1
5	1	2	0	1	3	0	1	3	0	7
6	3	1	2	5	4	2	4	1	6	3
7	1	7	0	1	3	0	5	1	0	1
8	1	4	2	2	1	2	3	1	2	6
9	0	2	1	3	5	1	4	1	0	1

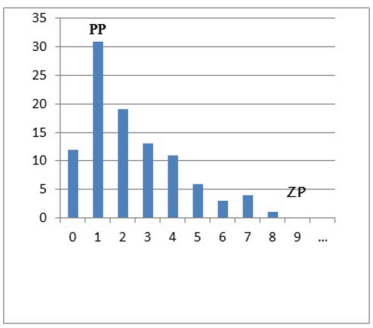


Fig. 1 Original histogram of $G(x, y)$

cannot provide an adequate free space. Thus, several couples of the least and utmost *PP* and *ZP* are needed to embed data and need to conserve the spare information like the value of *PP*, *ZP* and location information of overflow/underflow pixels.

2.2 Kim et al.'s method

The author achieved high embedding capacity in adjacent pixels by means of high spatial associations. The described data embedding algorithm is as follows.

input cover image D size $x \times y$ output stego image D.

Step 1. Using Eq. (3), the cover image allocated to many sub-image equals size $D_1, D_2, D_3, \ldots D_\Delta$

$$D_s(i,j) = D\left(i.\Delta + \lfloor \frac{S-1}{\Delta} \rfloor, j.\Delta + ((S-1)\mathrm{mod}\Delta)\right). \tag{3}$$

The sub-image rows and columns are denoted by (I, j), while intervals of sub-sampling is $m/\Delta = 1, 2, \ldots, n/\Delta.\Delta$

Step 2. Compute the difference image and sub-image selected as reference image in Eq. (4) $dS(i,j)(s \neq$ reference-image(r)).

$$d_S(i,j) = D_S(i,j) - r(i,j)(S = 1, 2, \ldots, \Delta^2, S \neq r. \tag{4}$$

Step 3. Histogram shifting using Eq. (5).

$$d_S^{'}(i,j) = \begin{cases} d_S(i,j) + L + 1, & \text{if} \quad d_S(i,j) \geq L+1 \\ d_S(i,j) - L - 1, & \text{if} \quad d_S(i,j) \leq -L-1. \\ d_S(i,j), & \text{otherwise} \end{cases} \tag{5}$$

Step 4. Altering the histogram to embed the secret bit b using Eq. (6).

$$d_S^{''}(i,j) = \begin{cases} d_S^{'}(i,j) + L + b, & \text{if} \quad d_S^{'}(i,j) = L \neq 0 \\ d_S^{'}(i,j) - L - b, & \text{if} \quad d_S^{'}(i,j) = -L \neq 0. \\ d_S^{'}(i,j) + b, & \text{if} \quad d_S^{'}(i,j) = -L = 0 \end{cases} \tag{6}$$

Step 5. Using Eq. (7) compute marked sub-image.

$$D_s^{'}(i,j) = D_r(i,j) + D_s^{''}(i,j). \tag{7}$$

2.3 Che et al.'s method

Fallahpour et al.'s [28] block division method evolved the *EC* of Ni et al.'s method. Kuo et al. [29] further distributed the cover image into four blocks size of 512×512. Che-Wei Lee [30] divided the cover image into supplementary sub-blocks 256×256 to 2×2, 256×256 to 8×8, and 4×4 to 2×2 shown in Fig. 2. The authors [13,31] by mentioning

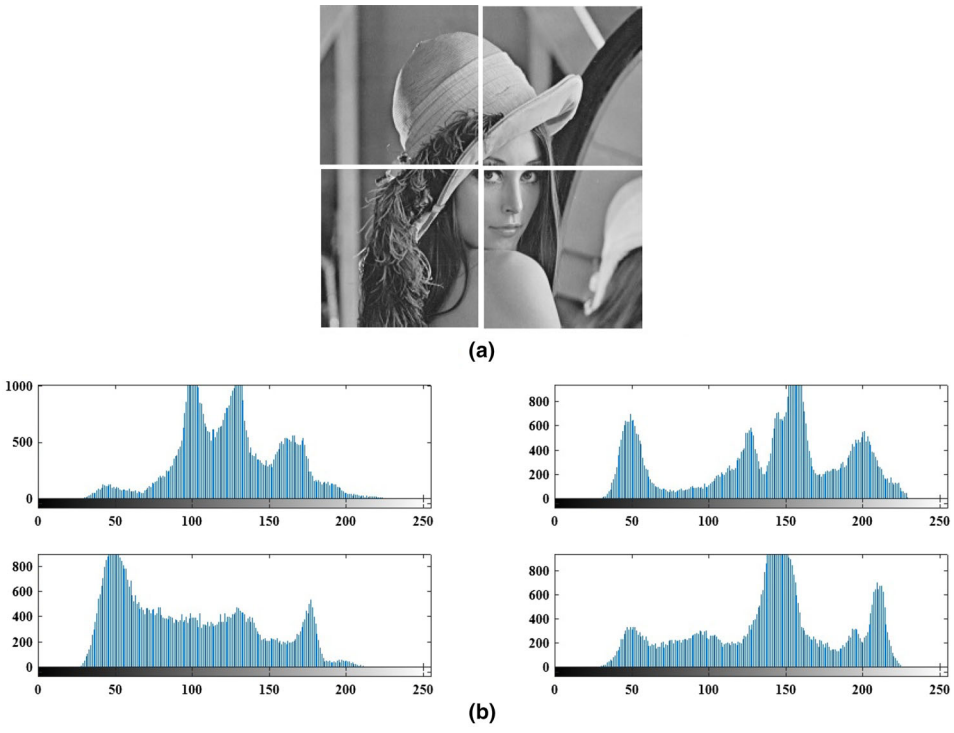


Fig. 2 **a** Block division of "Lena"; **b** Histogram of image "Lena" for each block

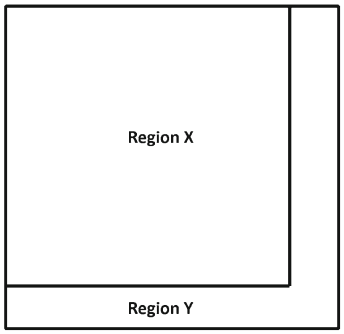


Fig. 3 Cover image division

the median value of each block, embedded the secret data to produce the difference histogram among a high block of pixel correlation. Block division effectively enriched the *EC*, since the whole capacity of data can be hidden in numerous blocks and is typically more than the single block. In the study, *EC* growth increased by block division, but all *EC* was contingent upon zero point [32].

In Liu et al. [27], the hiding rate is escalated by the division of image from 256×256 ,16×16 , 8×8, 2×2, but *PSNR* persistently decreased from the size of 256×256 to 2×2 blocks division.

3 The proposed method

Liu et al. [27] take out the n-bit planes from 8-bit for every pixel to make new bit-plane truncation image *(BPTI)* to write the information further, allocate the cover image into two regions, region X (use for secret bit) and region Y (use for extra bit), and both regions are used for data write shown in Fig. 3. Hiding capacity increasing on small block size while *EC* graph is deteriorates on small block size conversely, extra bits are also increasing on small block size that cannot be accommodate into region Y.

In the histogram of image's "Lena" 512×512 pixel size, the peak point corresponds to grayscale values, the highest hiding capacity of image "Lena" which can be used to embed the data but cannot provide large enough free space to hide the information.

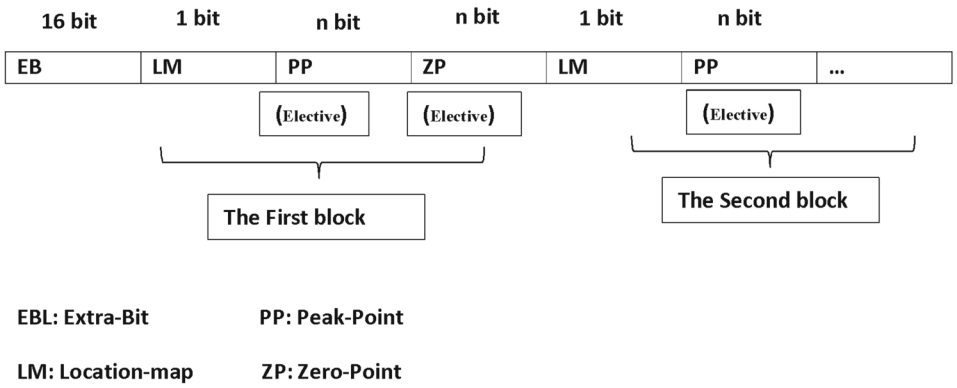
The low compression ratio for the intention cannot provide much space for data embedding so that many early schemes were based on lossless compression technique [31,33]. We have proposed a new technique for reversible data hiding based on efficient compression algorithm, which is slightly better as compared to binary arithmetic coding method with multiple bit planes (*CDMBP*).

CDMBP divides the size $A \times A$ into non-overlapping blocks with respect to zero point. A large number of blocks do not have zero point and have nothing to do with that block because we need some space to shift the histogram. First, we have mined the maximum number of zero point *ZP* with respect to the high probability in each *CDMBP* block. If *ZP* exists, then we set the value of the current block $CB = 1$; otherwise, we set $CB = 0$. We used embedding algorithm where $CB = 1$. Our detailed portrayal is separated into two sections: hiding data phase and recovering and extracting stage in Sects. 3.1 and 3.2. These phases are described in more detail in further subsections.

3.1 Data embedding algorithm

We recode the block that can be used for secret bits to differentiate the location map (*LM*) to embed the data in embedding and no embedding blocks. The value of *LM* is 1-bit for each block. If the value of *LM* is one in this case, we hide the secret data on that block, or else that block will not be used for the data embedding process shown in Fig. 4. Our proposed *RDH* technique, the sender part, embedded information (*I*) and proposed conversion (*PT*) based on efficient compressed domain with multiple bit-plane (*CDMBP*) image is employed to construct the free hiding space for embedded information *I*. Afterward, the original image is acquired during converse conversion of $PT - 1$. On the other side, the receiver can employ the same conversion to coalface the embedded data and recover the original image. *CDMBP*

Fig. 4 Extra bit structure



method in hiding phase is used to make a free giant space for information hiding. The grayscale image $I_{(i,j)}$, $N \times M$ pixel in size, the image $I_{(i,j)}$, and $N_x(M-1)$ in size of pixel are generated from the original image I. The proposed data hiding phase is described as follows.

In the *CDMBP* method, cover image I size 512×512 is split into two regions X and Y(embedded region and non-embedded region. The region of X, we extracted from 8-bit plane to generate a *CDMBP* bit-plane image furthermore, *CDMBP* image divided into non-overlapping blocks with respect of zero point (*ZP*). If *ZP* occurs, then we will set the value of the current block $CB = 1$; or else, we amend to check the *ZP* yet again. The data will be embedded on blocks when *CB* is one or else that block skips throughout the embedding process. The whole blocks are skimmed from left to right and top to bottom. The peak point has comparable value with $G_{(x,y)}$, then *PP* is modified to hide the secret bit M (generate randomly), and the value of *PP* will be *0* or *1*. We repeat all steps till all blocks have been skimmed; afterward, we generate a stego image $I^{'}$ in the succeeding embedding algorithm.

Input. Original image I, serect bit S

Output. Stego image $I^{'}$

Step 1. The 512×512 size cover image alienated into two regions X and Y; the region was employed to embed the extra bits into the last eight columns and rows of pixels. Moreover to compress the extra bits, we used efficient lossless compression algorithm in the subsequent section.

Compression algorithm plays a vital role in data embedding for the reason that they make available additional space for data embedding.

Definition We employed effectual compression [33] D-biased algorithm, where $D\epsilon[0, 1/2]$ is D-biased bitstream $B\epsilon(0,1)^n$ in case of fraction lower than D or greater than $1-D$ and D is less than $1/2$ positive constant.

$X = (x_1, x_2 x_n)$ and $v = (v_1, v_2 v_n)$ string two n-bit in subsequent order

$$x_1.2^0 + x_2.2x_n.2^{n-1} < v_1.2^0 + v_2.2v_n.2^{n-1}$$

Bitstream $B = (B_1, B_2, \ldots, B_n)\epsilon(0,1)^n$ by means of L ones. Computed e of B ordinal number in ensuing equation.

$$e = 1 + \sum_{m=0}^{n-1} s_{n-m} \left(\begin{matrix} n-m-1 \\ L - \sum_{j=0}^{m} B_n - j + 1 \end{matrix} \right). \tag{8}$$

Input: Bitstream D-biased $B = (B_1, B_2, \ldots, B_n)\epsilon(0,1)^n$, while $f_i(B)$ is compressed bitstream attained in succeeding steps:

Step1. Compute ordinal number N of B in the n-bit strings a set with L ones.

Step2. The binary illustration output is (L, N), whereas $\lceil log_{2^n} \rceil$ has bits of L put in first.
Extraction algorithm for $f_i(B)$

Input: n and $f_i(B)$ attained original bitstream $S\epsilon(0,1)^n$ using subsequent steps.

Step 1. The value of (n, L, N) is calculated from $f_i(B)$.

Step 2. Recreate B from (n, L, N).

Step 2. For eight-bit plane, we extracted n-bit plane to generate 512×512 size *CDMBP image* while embedded bits E are generated randomly. According to previous study, many authors used the block division technique [27–30] to expand the hiding capacity of the image. The hiding proportion rises up by division of image from 256×256 to 2×2; consequently, we divided the *CDMBP* image into non-overlapping blocks.

Step 3. Generate the current block histogram; additionally, find the peak point *PP* and *ZP* of the current block. If *ZP* occurs, then the current block $CB = 1$; otherwise, we will set $CB = 0$. If the pixel value $G_{(x,y)}$ of the block is greater than *PP*, alter the pixel value of block $G_{a(x,y)}$ to $G_{a(x,y)} + 1$; else, the value of $G_{a(x,y)}$ will remain unchanged. The alteration procedure is defined as,

$$G_{s(x,y)} = \begin{cases} G_{d(x,y)} + 1, & \text{if} \quad G_{d(x,y)} > PP \\ G_{d(x,y)}, & \text{otherwise.} \end{cases} \tag{9}$$

Step 4. Look over the entire block from left to right and top to bottom. The peak point has comparable value with $G_{(x,y)}$; then *PP* can be amended to hide the secret bit M and the value of *PP* will be 0 or 1. This process will be continued till the last block to embed the data.

For example, the secret bit is 01101, and the whole block is skimmed from top to bottom and left to right. In the embedding block, *PP* is 24 and *ZP* is 26. If the values of *PP* are less than *ZP*, in this case all 25 values will be increased by one. Embed the secret bit and increase pixel value by one in case the value of *PP* is 25; if the secret bit is one, no alteration will occur in Fig. 5b, c.

$$G_{m(x,y)} = \begin{cases} G_{m(x,y)} + m, & \text{if} \quad G_{s(x,y)} = PP \\ G_{s(x,y)}, & \text{otherwise.} \end{cases} \tag{10}$$

Step 5. We use n-bit planes to substitute in cover image to transformed *CDMBP*, and the extra bits are embedded in region Y. Eventually the stego image $I^{'}$ is generated. Finally, the stego image has been engendered. To attain the reversibility, the pixel LSBs are mined earlier and fixed to secret bits; likewise, on the other side converse transformation of data embedding algorithm will be executed to regain the secret message.

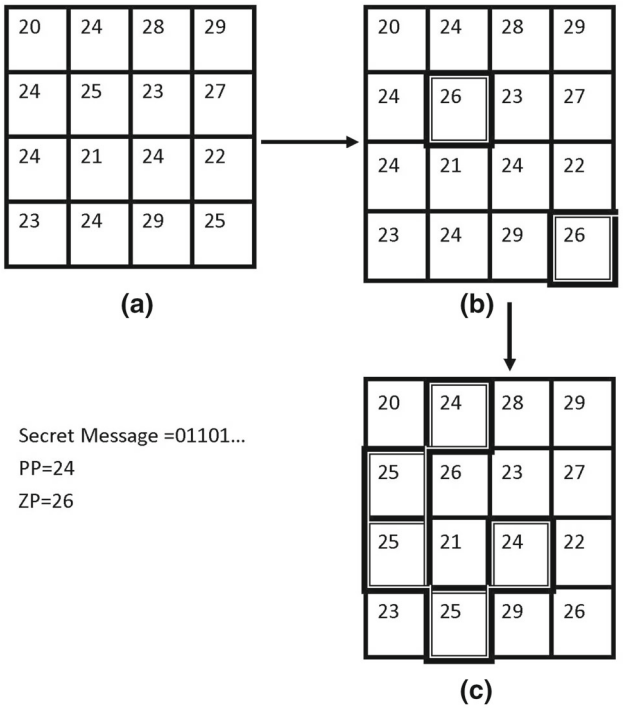


Fig. 5 Histogram modification example

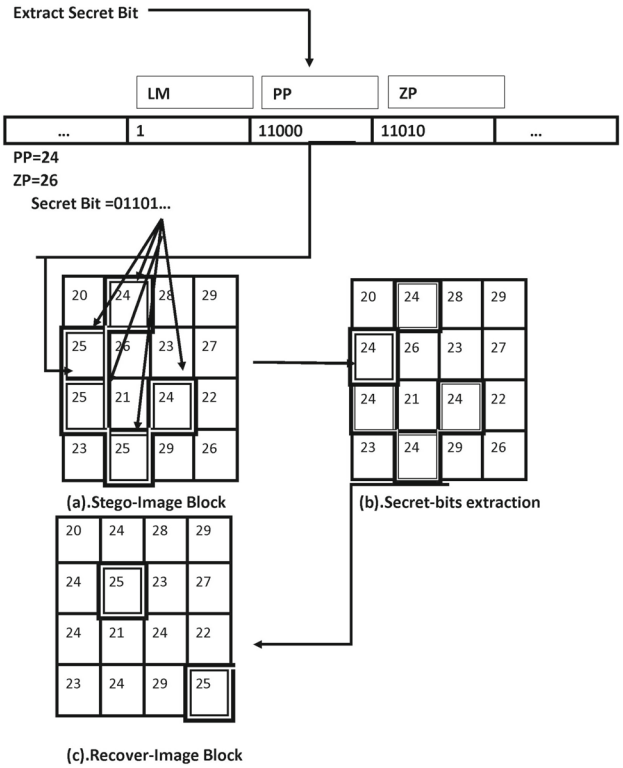


Fig. 6 Data extraction

3.2 Data extraction and cover image retrieval algorithm

In this phase, the secret bits are extracted from stego image correctly, whereas the original media are recovered from stego image entirely without any distortion. Data extraction phase is the opposite operation and is articulated in subsequent steps.

Step 1. The eight-bit planes extract n-bit planes of the respective pixels of stego image embedded in region X, while extra bits extract from non-embedding region Y and are distributed into non-overlapping blocks.

Step 2. The embedded message presume $m = 01101$ extract from $E_{SI(x,y)}$ for the respective block. First, the whole block is scanned from left to right and top to bottom and its extra bits congruently; also, PP having akin value with $E_{SI(x,y)}$ is then extracted and the hidden secret bit $M = 0$. If the value of $PP + 1$ is encountered, then secret bit $M = 1$ will be extracted and there is a further shift in the value decrement by 1 and the original pixel value is recovered by inverse transformation.

For example, if the secret bit is 01101 , the whole block is skimmed from top to bottom and left to right. We extract the value from the block where PP is 24 and ZP is 26. If the value of PP is less than that of ZP and CB or LM is one, in this case 0 will be extracted if PP is 24. We extract the secret bit and the recovered pixel value decreased by one in case the value of $PP + 1 = 25$, as shown in Fig. 6b, c.

$$E_{SB(x,y)} = \begin{cases} \text{extract the 0 bit, if } E_{SI(x,y)} = PP \\ \text{extract the 1 bit, if } E_{SI(x,y)} = PP + 1, \end{cases} \tag{11}$$

$$E_{RI(x,y)} = \begin{cases} E_{SB(x,y)} - 1, & \text{if } E_{SB(x,y)} = PP + 1 \\ E_{SB(x,y)}, & \text{otherwise.} \end{cases} \tag{12}$$

Step 3. Repeat this process to recover till the end of the last blocks.

Step 4. We use n-bit planes to substitute in stego image to reinstate $CDMBP$, and the extra bits are embedded in region Y. Eventually without any distortion, reinstate the original image I.

4 Experiments

In this slice, we directed several trials to exhibit our proposed method presentation demonstrated by peak-signal-to-noise-ratio ($PSNR$) and hiding capacity of stego image. We pick six grayscale images in Fig. 2, "Lena", "Boat", "Baboon", "Jet", "Tiffany" and "Pepper", all 512×512 pixels, serving as cover images and sum of all pixels allied with the PP in the image.

Algorithm 1 Secret bits extraction algorithm

Input: Stego image $I^{'}$
Output: Cover image I
Step 1: Extract n-bit-planes of respectively pixel of stego image in embedded region X while extra bits extract from non-embedding region Y
Step 2: The *CDMBP* distribute into auxiliary non-overlapping blocks
Step 3: if $CB = 1$ then find PP and ZP of block
Step 4: if $PP < ZP$ then if $E_{SI(x,y)} = PP$
Step 5: Extract zero bit
Step 6: else if $E_{SI(x,y)} = PP + 1$, then extract 1 secret bit
Step 7: end
Step 8: Move the value to left $[PP + 1]$ and $[PP - 1]$ by 1
Step 9: else if $PP > ZP$
Step 10: if $E_{SI(x,y)} = PP$ then extract zero bit
Step 11: else if $E_{SI(x,y)} = PP - 1$
Step 12: Extract 1 secret bit
Step 13: end
Step 14: Move the value to right $[PP + 1]$ and $[PP - 1]$ by 1
Step 15: end
Step 16: else do zilch
Step 17: end

Grayscale value adjacent to *PP* in *CDMBP* image block has the uppermost number of pixel values; consequently, due to this, we can hide colossal amount of data in the cover image.

4.1 Performance comparisons

We compared our proposed schemes with those of Ni et al. [17], Luo et al. [13], He et al. [32], Kim et al. [11], Lin et al. [25], Liu et al. [27], He et al. [31] methods in term of *PSNR* and *EC*. Ni et al.'s *EC* was restricted due to the number of peak bins, but has high *PSNR* at 48 dB; this scheme employs one grayscale to reform the utmost pixels and preserve the extra information like the value of *PP*, *ZP* and position information of those pixels which have overflow/underflow during the embedding process. In Lin et al. [25], the average *PSNR* is 35 db, which also decreases if the embedding level *(EL)* increases because of the high proportion of distortion due to high embedding capacity in marked images. The embedding capacity is also very low, because peak information capacity exceeds the whole embedding capacity for each block.

In Kim et al. [11] image Lena, the *PSNR* is 35.5 db for *EC* 0.73, while the maximum *PSNR* value is 48.9 db; on the other hand, *EC* is very low. For image Baboon, the *PSNR* is 44.23 db for *EC* 0.09, where the block size is 4×4. For Luo et al. five test image, the lowest *PSNR* values are 40.43, 43.68, 49.48 db, for $EL = 2, 1, 0$ while the maximum *PSNR* is 49.68 db on block division 2×2 where $EL = 0$, despite the highest embedding capacity on block division 3×3 where $EL = 2$, the *PSNR* is less than 40 db. For He et al. [32] image baboon, the *PSNR* is 44.23 db for *EC* 0.09, while the maximum *EC* is 0.25 bpp, but *PSNR* decreases to 37.36 bpp, where *EL* is 7.

Liu et al. [27] has high *EC* and low *PSNR*; however, there is a need to uphold the more extra information on the small block size.The hiding rate rises by division of image from 256×256, 16×16, and 8×8, 2×2, but *PSNR* persists downward from the size of 256×256 to 2×2, whereas the extra bit increases simultaneously and cannot be accommodated on region Y due to less space. On large block size, the embedded capacity is decreased and the number of extra bits also decreases. In He et al. [31], the Lena image embedding capacity is 0.38 for 43.9 *PSNR*; however, for image Barbara, the *EC* is 0.39 for 40.59 *PSNR* , where multiple embedding level *(EL)* is 5. In image Lena, the maximum *PSNR* is 50.62 db, but very low *EC* which is 0.143 bpp. As the *EC* is increases near 0.261 bpp, the *PSNR* gradually drops down near 47.01.

CDMBP scheme has high *PSNR* and more *EC* on small block size, as in other methods such as in Ni et al. [17], Luo et al. [13], He et al. [32], Kim et al. [11], Lin et al. [25], Liu et al. [27] and He et al. [31]. Secret bits are randomly generated and embedded on region X of size 504×504 represented by *EC*, while extra bits are embedded in region *Y*. We have gained high embedding capacity and image quality on 16×16 block division with high *PSNR* of 50.51. Likewise, we controlled the growth proportion of extra bits to expand more *EC* besides obtaining high *PSNR* value of 51.03, by using different bit-pane values. We have gained high *PSNR* and embedding capacity on block division 2×2 and sustained the *PSNR* value that is 49.18.

To check the authenticity of our scheme, we conducted many experiments using multi-embedding. In multi-embedding, stego image can be used as a new cover image for embedding secret data with repetition by using the same embedding algorithm as we recover the cover image reversibly. Multi-embedding is harmless for extracting secret data and similarly will not affect the cover image. Comparing our scheme with previous schemes clearly indicates betterment. After comparing and analyzing the results of the proposed scheme, which is lossless compression algorithm, a slightly better compression ratio as compared to binary arithmetic coding algorithm is revealed and higher hiding capacity with better image quality of stego images is obtained, providing strong evidence for higher capacity, quality and sensitivity.

4.2 Block divisions

According to preceding study, many authors employed the block division technique [27–30,32], to progress the hiding capacity; likewise, hiding rate is obtained by division of image from 256×256 to 2×2. The valuation of the proposed scheme block base division, *CDMBP* separated into non-overlapping blocks, is shown in Table 1. If a large number of blocks have an extreme number of zero point, then embed-

Table 1 Evaluation of the proposed scheme recital with different size block divisions

Different size of block divisions of Lena image								
No. of block	256 × 256	128 × 128	64 × 64	32 × 32	16 × 16	8 × 8	4 × 4	2 × 2
EC	25,432	26,542	27,833	30,634	38,342	66,748	90,234	143,673
PSNR	51.03	50.97	50.81	50.70	50.51	49.77	49.75	49.18
Extra bit	432	732	2143	3422	6163	38,1654	144,026	684,372

Table 2 Evaluation of the proposed scheme with different sizes of bit planes and block division; MBP signifies multiple bit planes, Cap represents embedding capacity and EB depicts extra bits

Different size of block division and bit planes of images						
MBP	Cap, PSNR, EB	Lena	Boat	Baboon	Jet	Tiffany
2–5	Cap	920,856	1,016,452	33,864	117,620	118,453
2–5	PSNR	37.41	38.81	40.77	38.30	37.88
2–5	Extra bit	5231	5102	3425	5435	6235
2–6	Cap	64,132	78,723	22,814	89,834	79,860
2–6	PSNR	43.37	44.92	45.66	44.50	44.44
2–6	Extra bit	7412	7122	5133	7712	8361
2–7	Cap	39,128	49,133	16,621	59,336	48,912
2–7	PSNR	49.33	50.35	50.15	51.46	50.74
2–7	Extra bit	9645	9622	8865	9845	9842
3–6	Cap	50,845	61,245	9623	74,635	67,524
3–6	PSNR	44.85	44.72	48.70	45.30	45.96
3–6	Extra bit	4156	3245	1625	4325	5132
3–7	Cap	33,845	42,564	9632	54,328	45,332
3–7	PSNR	50.23	49.81	53.08	50.72	51.31
3–7	Extra bit	6022	5912	2721	6521	7123
3–8	Cap	22,645	27,761	8932	35,420	29,120
3–8	PSNR	55.65	55.45	56.11	55.81	56.90
3–8	Extra bit	9465	9425	7012	9771	9541
4–7	Cap	21,354	31,358	1856	43,962	29,321
4–7	PSNR	53.17	53.33	64.46	52.89	53.01
4–7	Extra bit	2550	3052	445	3521	3021
4–8	Cap	14,954	20,998	1450	29,231	21,854
4–8	PSNR	58.70	58.71	65.98	58.08	58.28
4–8	Extra bit	4021	4632	1203	5123	4932
5–8	Cap	5121	10,450	1080	16,741	10,321
5–8	PSNR	64.16	62.23	68.23	61.23	61.63
5–8	Extra bit	1380	1832	621	2130	1902

ding size will be high; in case of the tiniest zero point, then embedded capacity will be low because we cannot enlarge the space to shift the histogram. The *CDMBP* method has high prospect of zero point with diverse block division to evaluate the *EC* and *PSNR* ratio of the image.

Selection of n bit planes and size can affect the performance of our scheme, as from small blocks we can get more hiding capacities. We assume 2 × 2, 3 × 3, 4 × 4, 6 × 6, 8 × 8, 14 × 14, 18 × 18, and21 × 21 blocks in our method. For 2 × 2 block, extra bits is five times as compared to the hiding capacity. It is sure that a large number of excessive bits would be difficult to be hidden in region B. Upon increasing the size of block, the hiding capacity will be lower, which will result in extra bits to decrease .

As we know, with small blocks size we can get more hiding capacities. But in 2 × 2 block, extra bits are five times as compared to the hiding capacity. It is sure that a large number of excessive bits are difficult to be hidden in region *Y*; con-

Table 3 Evaluation of *CDMBP* with block division 16 × 16 and different bit planes; (Ni et al. [17], Luo et al. [13], Kim et al. [11], Lin et al. [25], Liu et al. [27], He et al. [32])

Methods	Lena		Boat		Baboon		Jet		Pepper		Tiffany	
	Cap	PSNR	Cap	PSNR	Cap	PSNR	Cap	PSNR	Cap	PSNR	Cap	PSNR
[17]	6215	48.20	8101	48.23	5921	48.23	18, 079	48.27	–	–	9882	48.19
[13]	29,253	49.01	29,253	49.01	8927	49.02	40, 805	49.23	–	–	34,906	49.28
[11]	24,573	49.02	22,515	49.01	8062	48.86	36, 097	49.14	–	–	28,208	49.05
[25]	115,963	43.02	101,866	43.02	70,730	43.02	123, 660	43.02	110,929	43.02	–	–
[27]	33,499	50.19	42,292	49.77	9486	53.01	54, 027	50.67	–	–	45,297	51.27
Proposed	33,845	50.23	42,564	49.81	9632	53.08	54, 328	50.72	45632	49.02	45,332	51.31

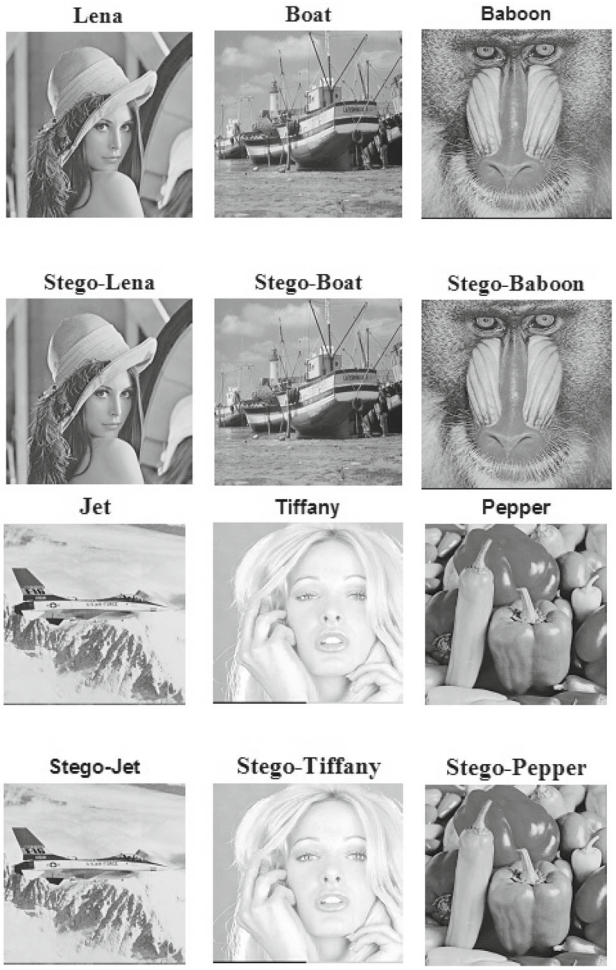


Fig. 7 Visual representations among the stego images and cover images

sequently, *CDMBP* controls the growth proportion of extra bits by expanding more *EC* besides maintaining high *PSNR* value.

Efficient compressed domain with multiple bit plane to construct *CDMBP image* and further split it into non-coinciding blocks with reverence of high probability of zero point (*ZP*) , shown in Table 2. We seasoned diverse bit planes and block division size from 256 × 256 to 2 × 2. For instance, we reconnoiter higher to lower bit planes with unlike block division to evaluate the *EC* and *PSNR* of image. On larger bit planes, embedding capacity is very high but in this case image quality was debauched and extra bit also increased in higher bit planes from two to six, while five to eight bit planes have high image quality but low extra bits and embedding capacity. Three to seven and three to eight have high embedding capacity and image quality. We have gained high embedding capacity and image quality on 16 × 16 and three to seven and three to eight bit planes. Likewise, control of the growth proportion of extra bits to expand more *EC* besides high *PSNR* value equate with Ni et a.l's, Luo et al.'s, Kim et al's, Lin et al.'s , Liu et al.'s and He et al's methods as in Table 3.

4.3 Computational complexity

Our proposed scheme's computational complexity is low, as we do not assume frequency domain like *DCT*, *DWT*, etc. It takes a short time for execution for the proposed scheme to perform an operation for generating image histogram for a cover image and hiding messages with no overflow or underflow problem. Beside this, it is also capable of inverse transformation in the spatial domain.

4.4 Lower bound PSNR

Image quality can be measured from *PSNR*. So in our proposed system, we also considered *PSNR* for measuring quality of the marked image. After getting the marked image, we investigated lower bound for quality of the marked image.

Our tryout assumption attests that the proposed method effectively performed the objective of high *PSNR* and *EC* on small blocks size of image. We have employed six grayscale images 512 × 512 as shown in Fig. 7. All measurements were performed on 1.80 GHz Intel core i-5 CPU with 4GB RAM. Peak signal-to-noise-ratio measured the quality of stego image (visual representation shown in Fig. 8). High *PSNR* signpost shows that stego image has virtuous visual

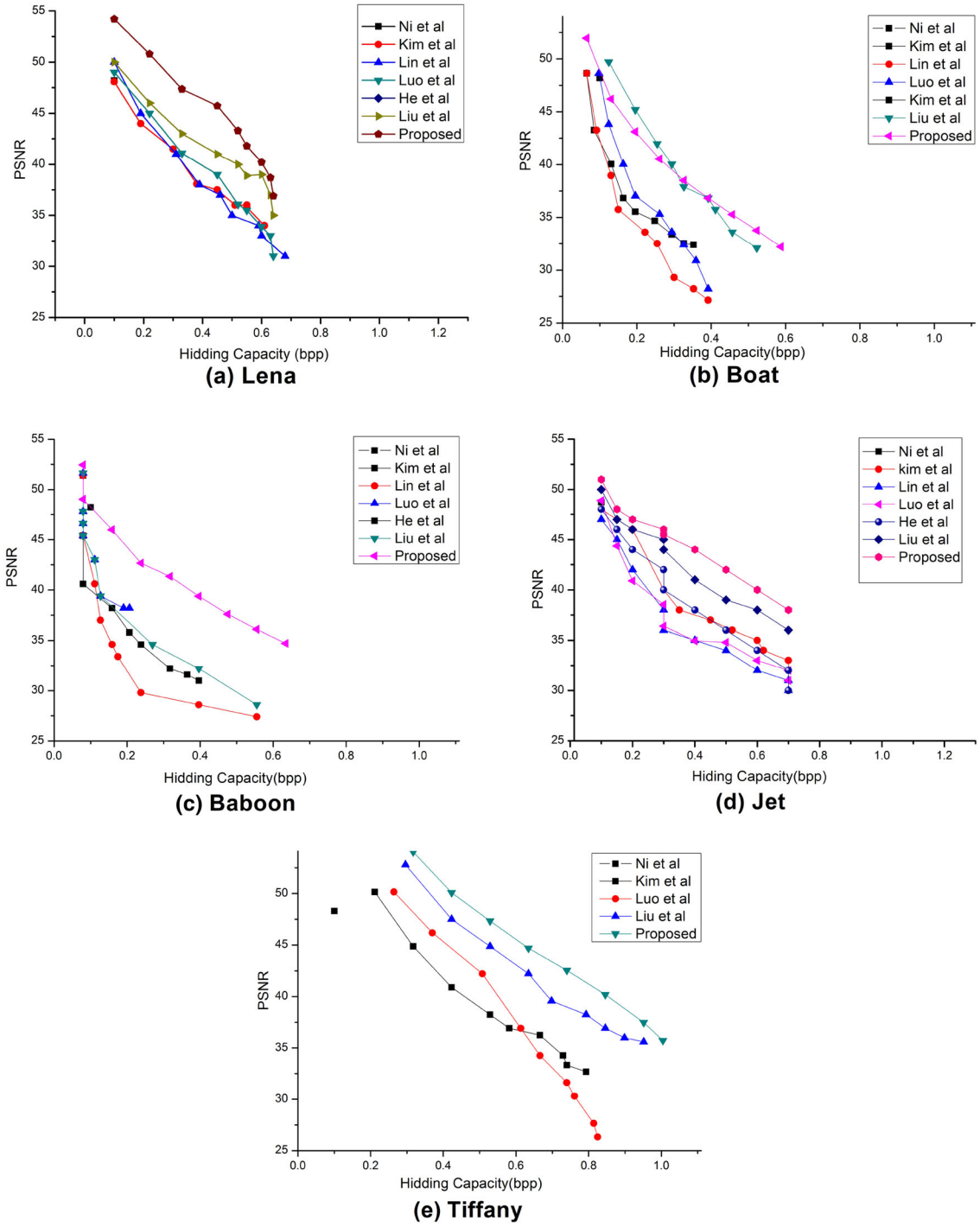


Fig. 8 Evaluation of our multi-embedding process with regard to *EC* and *PSNR* with other techniques

quality in addition to lowest number of distortions in the stego image.

$$PSNR = 10x\log 10\frac{255^2}{\text{MSE}}(\text{db}) \tag{13}$$

5 Conclusions

It is durable to sustain a balance among hiding capacity and image *PSNR* in the marked image. If *EC* is high, then *PSNR* will be low. Large number of blocks that have the tiniest number of zero points at that juncture *EC* will be low and it is also time consuming to cogitate a block that does not have zero

point. Our proposed scheme's efficiently compressed domain with multiple bit planes (*CDMBP*) attained more *EC* in small block size and gained high reversibility. During the embedding process, we mined those blocks that have zero point, and the blocks that do not have zero point were not considered for the embedding process and overwhelmed the problem of extra bits on small block size and overflow/underflow problem. In future, we would try to provide higher embedding capacity with low distortion based on localized bit plane multilayer embedding and try to reduce more extra bits, while increase *EC* and inflate broadcast quality of image.

Acknowledgements This work is partially supported by the National High Technology Research and Development program of China (863 program) under Grant 2014AA012204, the NSFC under Grant 61671018 and Chinese Government Scholarship (CSC) for international Scholar.

Compliance with ethical standards

Conflict of interest Rashid Abbasi , Bin Luo , Gohar Rehman , Haseeb Hassan , Muhammad Shahid Iqbal and Lixiang Xu declare that they have no conflict of interest.

Informed consent All procedures followed were in accordance with the ethical standards of the responsible committee on human experimentation (institutional and national) and with the Helsinki Declaration of 1975, as revised in 2008 (5). Additional informed consent was obtained from all patients for which identifying information is included in this article. Human and Animal Rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Zhao, H., Wang, Z., Nie, F.: Orthogonal least squares regression for feature extraction. Neurocomputing **216**, 200–207 (2016)
2. Zhao, M., Jiang, B., Luo, B., et al.: Common visual patterns discovery with an elastic matching model. Cogn. Comput **8**(5), 839–846 (2016)
3. Li, J., Li, X., Yang, B., et al.: Segmentation-based image copy-move forgery detection scheme. IEEE Trans. Inf. Forensics Secur. **10**(3), 507–518 (2015)
4. Cox, I.J., Kilian, J., Leighton, F.T., et al.: Secure spread spectrum watermarking for multimedia. Secure spread spectrum watermarking for images, audio and video. **3**, 243-246 (1996)
5. Coatrieux, G., Guillou, C.L., Cauvin, J.M., et al.: Reversible watermarking for knowledge digest embedding and reliability control in medical images. IEEE Trans. Inf. Technol. Biomed. Publ. **13**(2), 158–165 (2009)
6. Barton, J.M.: Method and apparatus for embedding authentication information within digital data: US, US 6047374 A[P] (2000)
7. Yin, Z., Niu, X., Zhou, Z, et al.: Improved reversible image authentication scheme. Cogn. Comput. 1–10 (2016)
8. Lee, S., Yoo, C.D., Kalker, T.: Reversible image watermarking based on integer-to-integer wavelet transform. IEEE Trans. Inf. Forensics Secur. **2**(3), 321–330 (2007)
9. Hwang, K., Li, D.: Trusted cloud computing with secure resources and data coloring. IEEE Internet Comput. **14**(5), 14–22 (2010)
10. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Technol. **13**(8), 890–896 (2003)
11. Kim, K.S., Lee, M.J., Lee, H.Y., et al.: Reversible data hiding exploiting spatial correlation between sub-sampled images. Pattern Recognit. **42**(11), 3083–3096 (2009)
12. Li, X., Zhang, W., Gui, X., et al.: Efficient reversible data hiding based on multiple histograms modification. IEEE Trans. Inf. Forensics Secur. **10**(9), 2016–2027 (2015)
13. Luo, H., Yu, F.X., Chen, H., et al.: Reversible data hiding based on block median preservation. Inf. Sci. **181**(2), 308–328 (2011)
14. Lou, D.C., Hu, M.C., Liu, J.L.: Multiple layer data hiding scheme for medical images. Comput. Stand. Interfaces **31**(2), 329–335 (2009)
15. Chang, C.C., Nguyen, T.S., Lin, C.C.: A reversible data hiding scheme for VQ indices using locally adaptive coding. J. Vis. Commun. Image Represent. **22**(7), 664–672 (2011)
16. Thodi, D.M., Rodrguez, J.J.: Expansion embedding techniques for reversible watermarking. IEEE Trans. Image Process. **16**(3), 721–730 (2007)
17. Ni, Z., Shi, Y.Q., Ansari, N., et al.: Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. **16**(3), 354–362 (2006)
18. Li, Y.C., Yeh, C.M., Chang, C.C.: Data hiding based on the similarity between neighboring pixels with reversibility. Digit. Signal Process. **20**(4), 1116–1128 (2010)
19. Tai, W.L., Yeh, C.M., Chang, C.C.: Reversible data hiding based on histogram modification of pixel differences. IEEE Trans. Circuits Syst. Video Technol. **19**(6), 906–910 (2009)
20. Li, X., Li, B., Yang, B., et al.: General framework to histogram-shifting-based reversible data hiding. IEEE Trans. Image Process. **22**(6), 2181–2191 (2013)
21. Qin, C., Chang, C.C., Liao, L.T.: An adaptive prediction-error expansion oriented reversible information hiding scheme. Pattern Recognit. Lett. **33**(16), 2166–2172 (2012)
22. Gui, X., Li, X., Yang, B.: A high capacity reversible data hiding scheme based on generalized prediction-error expansion and adaptive embedding. Signal Process. **98**, 370–380 (2014)
23. Hong, W., Chen, T.S.: Reversible data embedding for high quality images using interpolation and reference pixel distribution mechanism. J. Vis. Commun. Image Represent. **22**(2), 131–140 (2011)
24. Lu, T.C., Chang, C.C., Huang, Y.H.: High capacity reversible hiding scheme based on interpolation, difference expansion, and histogram shifting. Multimed. Tools Appl. **72**(1), 417–435 (2014)
25. Lin, C.C., Tai, W.L., Chang, C.C.: Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recognit. **41**(12), 3582–3591 (2008)
26. Tsai, P., Hu, Y.C., Yeh, H.L.: Reversible image hiding scheme using predictive coding and histogram shifting. Signal Process. **89**(6), 1129–1143 (2009)
27. Liu, L., Chang, C.C., Wang, A.: Reversible data hiding scheme based on histogram shifting of n-bit planes. Multimed. Tools Appl. 1–16 (2015)
28. Fallahpour, M., Sedaaghi, M.H.: High capacity lossless data hiding based on histogram modification. IEICE Electron Express **4**(7), 205–210 (2007)

29. Kuo, W.C., Jiang, D.J., Huang, Y.C.A.: Reversible data hiding scheme based on block division. In: IEEE Congress on Image and Signal Processing, CISP'08, vol. 2008, no. (1), pp. 365–369 (2008)
30. Lee, C.W., Tsai, W.H.: A Lossless Data Hiding Method by Histogram Shifting Based on an Adaptive Block Division Scheme. Pattern Recognition and Machine Vision, pp. 1–14. River Publishers, Aalborg (2010)
31. He, W.: Improved block redundancy mining based reversible data hiding using multi-sub-blocking. Signal Process. Image Commun. **60**, 199–210 (2018)
32. He, W., Xiong, G., Zhou, K., et al.: Reversible data hiding based on multilevel histogram modification and pixel value grouping. J. Vis. Commun. Image Represent. **40**, 459–469 (2016)
33. Chang, J.C., Lu, Y.Z., Wu, H.L.: A separable reversible data hiding scheme for encrypted JPEG bitstreams. Signal Process. **133**, 135–143 (2017)