Multimedia Tools and Applications (2020) 79:26969–26993
https://doi.org/10.1007/s11042-020-09273-y

Two level data encoding approach for reversible data hiding in dual Stego images

C. Shaji[1] · I. Shatheesh Sam[2]

Received: 15 July 2019 / Revised: 25 April 2020 / Accepted: 26 June 2020 /
Published online: 20 July 2020

Abstract

Due to high embedding capacity and security, dual stego-image based data hiding has become so popular. This paper proposes a two-level data encoding approach for reversible data hiding in dual stego-images. In the first level of encoding, the encoded intensities are estimated from the message intensities by assigning lower intensities to higher histogram data and higher intensities to lower histogram data. In the second level of encoding, the encoded intensities are folded by identifying the negative values to obtain the folded intensities. The folded intensities are embedded in the cover image to obtain the dual stego-images. The two-level data encoding process reduces the intensity of secret data which increases the quality of the two stego-images. During the extraction process, the folded intensities are extracted from the dual stego images. The folded intensities are decoded to encoded intensities and then to message intensities to obtain the secret data. This two-level data encoding approach increases the peak signal to noise ratio (PSNR) around 2 dB, and embedding rate (bpp) by 1%when compared to the traditional data hiding approach in dual stego images.

Keywords Reversible data hiding · Histogram · Dual stego images · PSNR and embedding rate

✉ C. Shaji
shaji2012@gmail.com

I. Shatheesh Sam
shatheeshsam@yahoo.com

[1] Department of Computer Science, Nesamony Memorial Christian College affiliated, ManonmaniamSundaranar University, Abishekapatti, Tirunelveli, Tamil Nadu 627 012, India

[2] Department of PG Computer Science, Nesamony Memorial Christian College affiliated, ManonmaniamSundaranar University, Abishekapatti, Tirunelveli, Tamil Nadu 627 012, India

1 Introduction

Reversible data hiding [2, 8, 15, 29, 31] is a technique that not only recovers the original cover image, but also provides the exact secret data. During the extraction process, the pixel of the original cover image is exactly recovered from the stego image without any distortion. So reversible data hiding [7, 9, 14, 16, 28] has very important application in protecting the images that contain highly confidential information such as medical, and military images etc. where there is a need of recovering the extract original cover image and secret data. Data hiding that provides a single stego image can hold fewer amount of secret data, but the data hiding in dual stego image can hold a huge amount of secret data. Moreover the dual stego image approach is more secure than the single stego image approach since the data extraction has been possible only if, both the stego images are to be known (data extraction is not possible from one stego image). Since there are two stego images, it requires more space to store the dual stego images. However, dual stego images have a very positive advantage of having high embedding rate when compared to single image based data hiding.

Different reversible data hiding techniques have been proposed which include histogram shifting, difference expansion and prediction error expansion. Initially, the histogram shifting technique was introduced by Ni et al. [19], where the bins less than a particular intensity are shifted towards the left by one position to create a carrier bin for embedding the message bits. Tian [22] proposed the difference expansion (DE) method to provide high embedding capacity. This DE method estimates the difference between the two adjacent pixels and '1' bit data is embedded in the difference value. Thodi and Rodriguez [21] first proposed the prediction error expansion (PEE) method that combines the advantage of difference expansion and histogram shifting techniques. The PEE method embeds the data on the prediction error estimated between the pixel and the neighborhood pixel.

The DE technique was improved by Weng et al. [26] by using pair-wise difference adjustment (PDA) scheme and the invariability of the sum of pixel pairs. Dragoi and Coltuc [5] proposed local prediction based difference expansion technique where the image has divided into sub-blocks and a predictor of least square was estimated on each sub-block that centered on the pixel. The prediction error was expanded to embed the data. Histogram shifting method was improved by Luo et al. [32] by estimating the difference between the corresponding pixel value and the interpolation value.

Inorder to reduce the distortion between the cover image and stego image, Li et al. [13] proposed multiple histograms based prediction error expansion where the local complexity measurement is used to estimate the multiple histograms. Pixel value ordering (PVO) based prediction error expansion was introduced by Li et al. [12] where the image was first divided into sub-blocks. The pixels belong to a particular sub-block was sorted for embedding the data. This PVO method was further improved by Wang et al. [27] by using local complexity analysis.

Inorder to preserve the privacy of cover image in cloud computing, several RDH techniques in the encrypted domain has been proposed. Since the distortion is not a concern in embedding data on an encrypted image, a high embedding rate is achieved in encrypted images. Li et al. [11] proposed a histogram shifting based scheme by using homomorphic multiplication with a public key histogram of the image. Dual image RDH was introduced to increase the embedding capacity that produces two similar stego images instead of generating a single stego image. Dual stego image based data hiding is highly secure than the single stego image based data hiding, since the unauthorized user cannot extract the data from a single stego image. Exploiting modification direction (EMD) in dual stego image was introduced by Wang et al. [30] which calculates a modulus function for the intensities from 0 *to* 255. The data hiding was done based on the value of the modulus function.

The remainder of the paper is arranged as follows. Few of the related works are shown in section 2. Section 3 shows the proposed two-level data encoding approach. Section 4 discusses the experimental results of the paper and finally, the conclusion is presented in section 5.

2 Related works

This section shows the few related works of dual image RDH such as exploiting modification direction, Authenticable RDH, LSB Matching and our previous embedding method based on Data Encoding.

2.1 EMD scheme

Exploiting modification direction (EMD) [30] for hiding data was initially proposed for single stego image by Zang and Wang (2006). It estimates a modulus function M using the relation in eqn (1),

$$M(x_1, x_2, \ldots\ldots x_l) = \left[\sum_{i=1}^{l} x_i \times i\right] mod\ (2 \times l + 1) \tag{1}$$

Where x_1, x_2, $\ldots\ldots x_l$is the group of l number of pixels. Let d be the data in decimal form. If d and M differs then the group of pixels are modified as eqn (2).

$$r = (d{-}M) mod\ (2 \times l + 1) \tag{2}$$

Where r resembles the position of the pixels to be incremented or decremented.

Figure 1 considers the original pixels {48,86,72,94} where $l = 4$ and message $d = 5_9$. Using eqn (1) $M(48,86,72,94) = (1 \times 48 + 2 \times 86 + 3 \times 72 + 4 \times 94)\ mod\ (2 \times 4 + 1) = 2$. Here the value of d and M differes. Therefore $r = (5 - 2)\ mod\ (2 \times 4 + 1) = 3$. If$r$ is less than l, increment x_r by 1 otherwise decrement $x_{2l, - r + 1}$ by 1. Therefore decrement x_3 by 1, which gives the stego pixels {48,86,73,94}. The EMD scheme was extended on dual stego images [3] by Chang et al. 2007. The modulus function was estimated for all intensities from 0 to 255. A block of 5 × 5 pixel was selected to embed the data. The modulus function was estimated by eqn (3).

$$M(x_i, x_{i+1}) = (x_i + 2 \times x_{i+1}) mod\ 5 \tag{3}$$

Fig. 2 assumes that an original pixel pair {26, 87} and data $d$as {1, 3}. The stego pixel pair corresponding to the first data '1' is {11, 40} and the second data '3' is {9, 42}. The EMD scheme was extended on dual stego images reversibility [4] by Chang et al. 2013. The block size is increased to 9 × 9. Here the modulus function is calculated using eqn (4).

$$M(x_i, x_i) = (x_i + 3 \times x_i) mod\ 9 \tag{4}$$

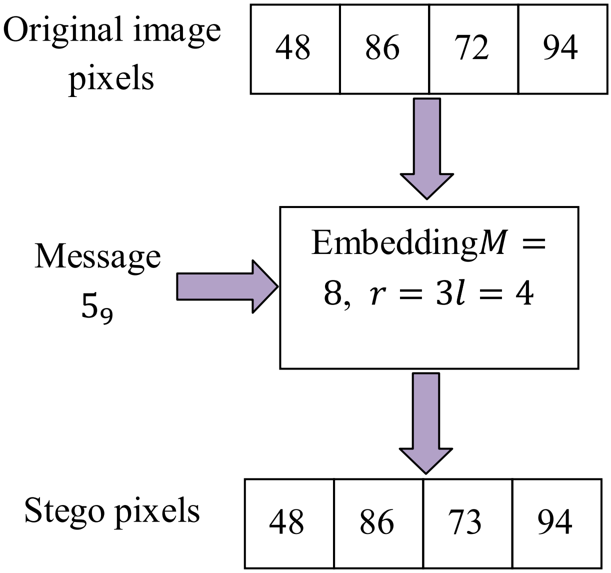
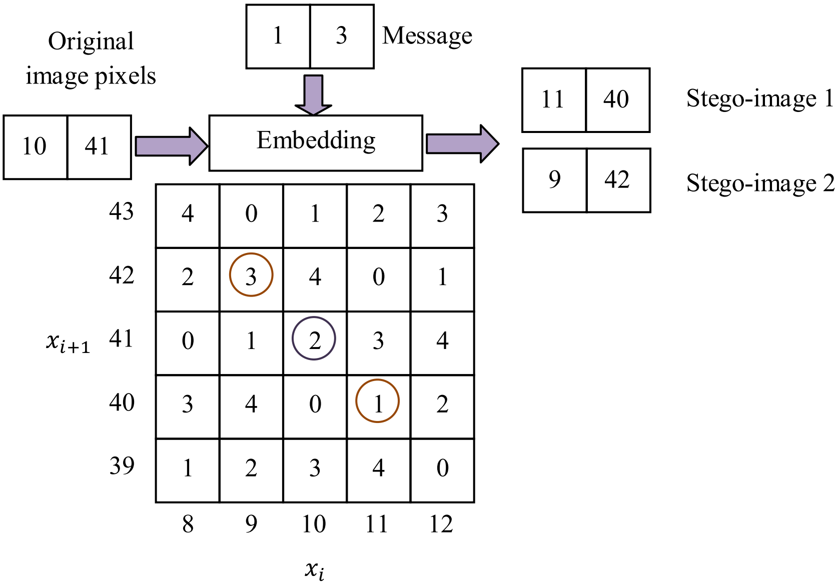


Fig. 1 Data embedding using EMD in the single stego image

Fig. 3 assumes that the original image pixel is 23 and the data is 4_9. The modulus function for the pixel 23 is calculated as $M(23,23) = (23 + 3 \times 23)\, mod\ \ 9 = 2$. The



x_{i+1} \ x_i	8	9	10	11	12
43	4	0	1	2	3
42	2	3	4	0	1
41	0	1	2	3	4
40	3	4	0	1	2
39	1	2	3	4	0

Fig. 2 Data Embedding using EMD in the dual stego image

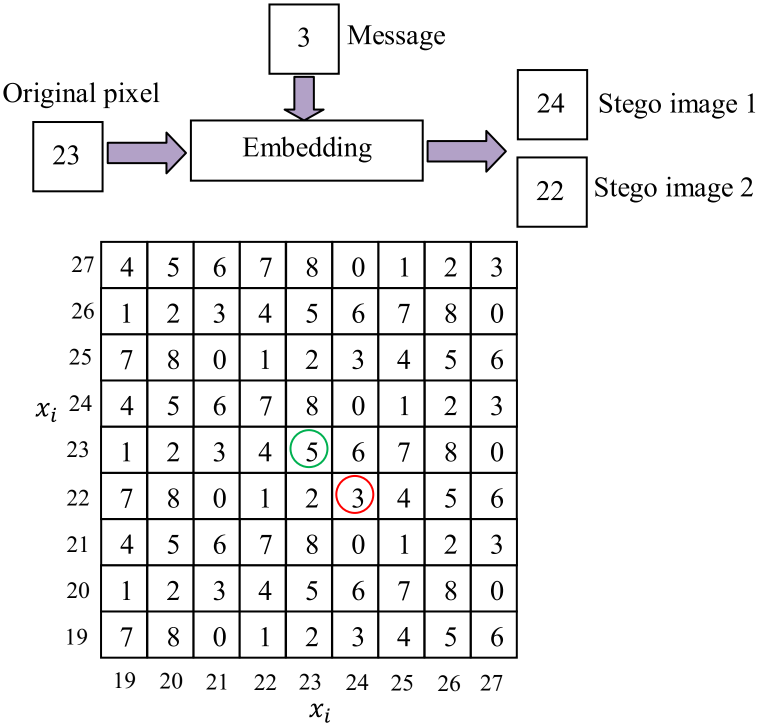
stego pixel for the original pixel 23 is calculated as 24 and 22. Qin et al. (2015) [17] converted the binary data to base 5 values before embedding. The stego pixel pair x_i^1 and x_{i+1}^1 for the first stego image is obtained by embedding two symbols d_1 and d_2 on the original image pixel pairs x_i and x_{i+1} by using EMD embedding scheme.

The procedure for estimating the second stego image pixel pair x_i^2 and x_{i+1}^2 is as follows.

Step (i): The second stego image pixel pairs x_i^2 and x_{i+1}^2 was estimated by EMD embedding process if $x_i^1 = x_i$ and $x_{i+1}^1 = x_{i+1.}$

Step (ii): Find the value of W_1 and W_2 as eqns (4) and (5)

$$W_1 = \begin{cases} 5, if\ d_2 = M\left[x_i - 5 \times sign\left(x_i^1 - x_i\right), x_{i+1}\right] \\ 4, if\ d_2 = M\left[x_i - 4 \times sign\left(x_i^1 - x_i\right), x_{i+1}\right] \\ 3, if\ d_2 = M\left[x_i - 3 \times sign\left(x_i^1 - x_i\right), x_{i+1}\right] \\ 2, if\ d_2 = M\left[x_i - 2 \times sign\left(x_i^1 - x_i\right), x_{i+1}\right] \\ 1, if\ d_2 = M\left[x_i - 1 \times sign\left(x_i^1 - x_i\right), x_{i+1}\right] \end{cases} \tag{4}$$



x_i	19	20	21	22	23	24	25	26	27
27	4	5	6	7	8	0	1	2	3
26	1	2	3	4	5	6	7	8	0
25	7	8	0	1	2	3	4	5	6
24	4	5	6	7	8	0	1	2	3
23	1	2	3	4	5	6	7	8	0
22	7	8	0	1	2	3	4	5	6
21	4	5	6	7	8	0	1	2	3
20	1	2	3	4	5	6	7	8	0
19	7	8	0	1	2	3	4	5	6

x_i

Fig. 3 Data embedding using EMD with 9 × 9 modulus function

$$W_2 = \begin{cases} 5, if\ d_2 = M\left[x_i, x_{i+1} - 5 \times sign\left(x_{i+1}^1 - x_{i+1}\right)\right] \\ 4, if\ d_2 = M\left[x_i, x_{i+1} - 4 \times sign\left(x_{i+1}^1 - x_{i+1}\right)\right] \\ 3, if\ d_2 = M\left[x_i, x_{i+1} - 3 \times sign\left(x_{i+1}^1 - x_{i+1}\right)\right] \\ 2, if\ d_2 = M\left[x_i, x_{i+1} - 2 \times sign\left(x_{i+1}^1 - x_{i+1}\right)\right] \\ 1, if\ d_2 = M\left[x_i, x_{i+1} - 1 \times sign\left(x_{i+1}^1 - x_{i+1}\right)\right] \end{cases} \tag{5}$$

Where $M[a, b]$ is the modulus function which is calculated using the expression in eqn (6)

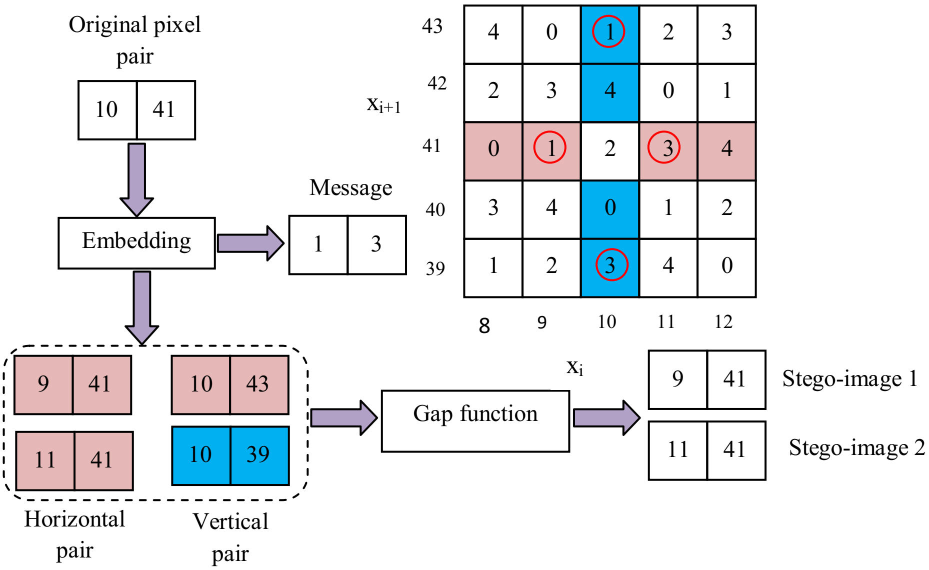
$$M[a, b] = (a + 2b) mod\ 5 \tag{6}$$

Step (iii): If $x_i^1 \neq x_i$ and $x_{i+1}^1 = x_{i+1}$ then $x_i^2 = x_i - W_1 \times sign\left(x_i^1 - x_i\right)$ and $x_{i+1}^2 = x_{i+1}$

Step (iv): If $x_i^1 = x_i$ and $x_{i+1}^1 \neq x_{i+1}$ then $x_i^2 = x_i$ and $x_{i+1}^2 = x_{i+1} - W_2 \times sign\left(x_{i+1}^1 - x_{i+1}\right)$

2.2 Authenticable reversible data hiding

An Authenticable Reversible data hiding was proposed by K.H Jung (2017) [10] using the same modulus function is shown in eqn (3). Instead of estimating the stego pixels in a diagonal direction, this method estimates the stego pixels in the horizontal and vertical direction. Figure 4 considers an example, where the message **7** s $\{1, 3\}$ and the original



x_{i+1} \ x_i	8	9	10	11	12
43	4	0	1	2	3
42	2	3	4	0	1
41	0	1	2	3	4
40	3	4	0	1	2
39	1	2	3	4	0

Fig. 4 Data embedding using Authenticable RDH

pixel pair be {10,41}. The vertical and horizontal pairs are estimated and one pair is chosen from it, which has less distortion using a gap function.

2.3 LSB matching

LSB matching based data hiding was proposed by Y.L.Wang et al. (2018) [23]. In LSB matching scheme, the cover pixel is converted to 8-bit binary and it is replicated to obtain two sets of eight-bit numbers. Let the eight bit binary numbers be $B_7B_6B_5B_4B_3B_2B_1B_0$. Let the 3-bit message be $m_2m_1m_0$. The message m_1m_0 is embedded in such a way that B_1B_0 are replaced by m_1m_0 to obtain the binary sequence $B_7B_6B_5B_4B_3B_2m_1m_0$. The message $m_2$is embedded in the 3rd position of the 8-bit binary $B_7B_6B_5B_4B_3m_2B_1B_0$. The two sets of eight bit binary numbers are converted to decimal to obtain the two stego pixels.

2.4 Data encoding approach

In our previous paper [18], a new data encoding method for reversible data hiding in dual stego images based on maximum to the minimum histogram is introduced. This method works on the principle that the conversion of higher histogram data from the secret image to lower intensity and conversion of lower histogram data from the secret image to higher histogram provides quality stego images. While embedding, the data from the secret image are grouped in K bits and converted to decimal. This decimal value forms the decimal intensities. The message intensities are converted to encoded intensities using a data encoding process which works on maximum to minimum histogram values. In the data encoding process, the histograms of message intensities are estimated and the histogram is sorted from maximum to minimum. By using the histogram values, assigned intensities are estimated. Based on the initial intensity, assigned intensity and sorted histogram index, encoding is performed by traversing the message intensity sequence in the form of an adjacent pair. The encoded intensity thus obtained after data encoding is embedded in the cover pixel to obtain the stego pixel pair.

In the extraction process, the encoded intensities are obtained from the two stego images using the difference between pixels from stego image1 and stego image 2. The data decoding process converts the encoded intensities, which is extracted from the stego images are converted back to message intensities. The encoded intensities are traversed from the starting value in pair to decode the message intensity. The message intensity thus obtained are converted to K bit binary numbers to obtain the secret data which forms the secret image.

In the proposed work, we aim to improve the PSNR and embedding capacity further by including one more level encoding process to our previous work. For the first level, we uses the algorithm of our previous work, but for the second level the encoded intensities of first level are folded by identifying the negative values to obtain the folded intensities. The folded intensities are embedded in the cover image to obtain the dual stego-images.

3 Proposed method

This section shows the working of proposed two-level data encoding for reversible data hiding in dual stego image. The proposed method has two phases ie., data embedding and data extraction. The data embedding phase aims to produce the two stego images using the cover

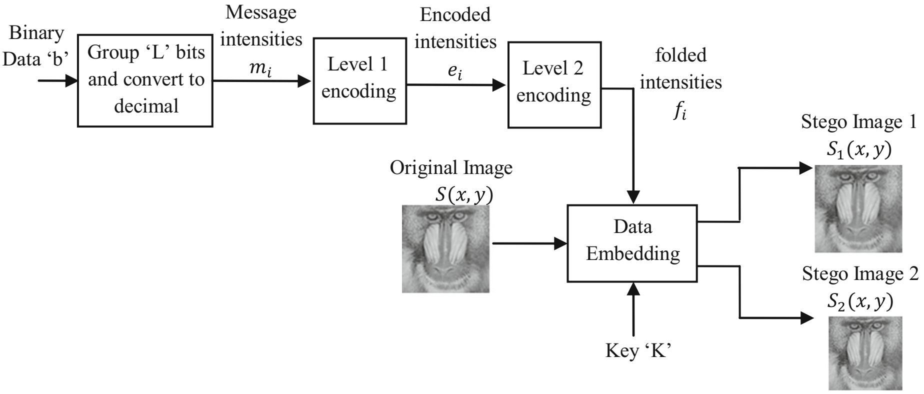


Fig. 5 Block diagram of data embedding in the proposed algorithm

image and secret data, while the data extraction phase aims to recover the cover image and secret data using the two stego images.

3.1 Data embedding

The data embedding is performed in two levels of encoding such as Level 1 encoding and Level 2 encoding. The block diagram of the proposed data embedding process is shown in Fig. 5. Let $S(x, y)$ be the cover image and ‘b’ be the binary data. Let $S_1(x, y)$ and $S_2(x, y)$ are the two stego images.

Initially, the binary data b is grouped into L bits and are converted to decimal to obtain the message intensities m_i, where m_i lies in the interval $[0, 2^L - 1]$. The Fig. 6 message intensities m_i are encoded using Level 1 encoding to obtain the encoded intensities e_i. The encoded intensities are further encoded using Level 2 encoding to obtain the folded intensities f_i.

3.1.1 Level 1 encoding

Let $m_i = \{m_1, m_2, \ldots\ldots m_n\}$ be the ‘n’ number of data in the decimal form obtained by grouping L bits from the binary data b. The histogram of the message intensity m_i is estimated and it is

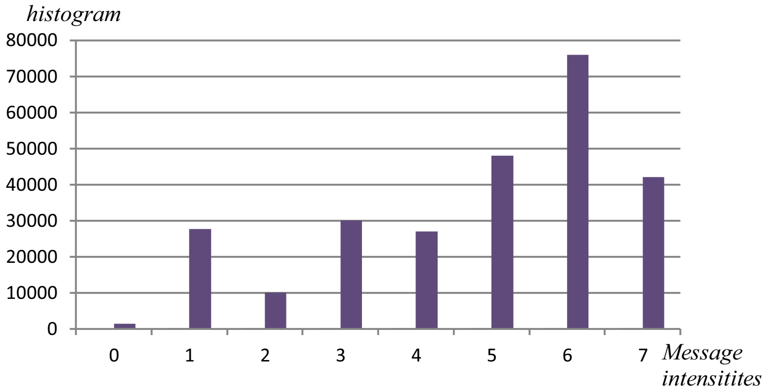
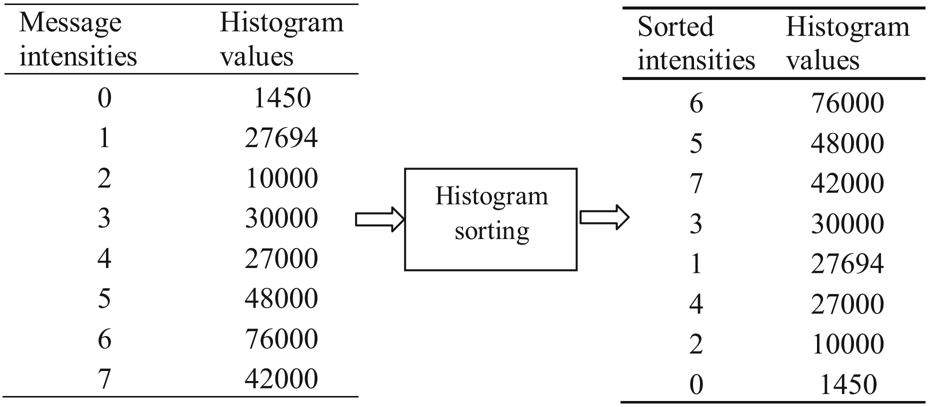


Fig. 6 Histogram for message intensities m_i



Message intensities	Histogram values
0	1450
1	27694
2	10000
3	30000
4	27000
5	48000
6	76000
7	42000

Sorted intensities	Histogram values
6	76000
5	48000
7	42000
3	30000
1	27694
4	27000
2	10000
0	1450

Fig. 7 An example of Histogram sorting

sorted in descending order to obtain the sorted index C_i which is represented as eqn (7),

$$C_i = \{C_0, C_1, C_2, \ldots\ldots C_{N-1}\} \qquad 0 \le C_i \le N-1 \tag{7}$$

The sorted index C_i contains Nnumber of samples,where$N = 2^L$. The sorted indices C_0 to C_{N-1} are the message intensities that has maximum to minimum histogram values. Consider a histogram plot for the message intensities m_i is shown in Fig. 6.

In the example, shown in Fig. 7 the sorted index is represented as $C_i = \{6, 5, 7, 3, 1, 4, 2, 0\}$.After estimating the sorted index, Form the assigned intensity table B_i as eqn (8),

$$B_i = \left\{0, 1, -1, 2, -2, 3, -3, \ldots\ldots\left(\frac{N}{2}-2\right), -\left(\frac{N}{2}-2\right), \left(\frac{N}{2}-1\right), -\left(\frac{N}{2}-1\right)\right\} \tag{8}$$

Where, B_i contains alternate positive and negative numbers starting from zero. From, initial intensity table A_i as eqn (9),

$$A_i = \left\{\frac{N}{2}, -\frac{N}{2}\left(\frac{N}{2}+1\right), -\left(\frac{N}{2}+1\right)\ldots\ldots(N-2), -(N-2), (N-1), -(N-1)\right\} \tag{9}$$

The assigned intensity table B_i and initial intensity table A_i have N number of elements. The procedure for Level 1encoding is shown below.

Step 1: Traverse any message intensity m_i from starting to end of the sequence.

Table 1 Level 1 Encoding table

Sorted index C_i	C_0	C_1	C_2	C_3	....	C_{N-3}	C_{N-2}	C_{N-1}
Message intensity d_i	0	1	2	3	....	$(N-3)$	$(N-2)$	$(N-1)$
Initial intensity A_i	$\frac{N}{2}$	$-\frac{N}{2}$	$\left(\frac{N}{2}+1\right)$	$-\left(\frac{N}{2}+1\right)$	....	$-(N-2)$	$(N-1)$	$-(N-1)$
Assigned intensity B_i	0	1	−1	2	....	$\left(\frac{N}{2}-1\right)$	$-\left(\frac{N}{2}-1\right)$	–

Step 2: While traversing m_0 through m_n replace lonely message intensity m_i by its corresponding initial intensity A_i using Table 1 till pair of m_i is found.

Step 3: While traversing m_0 through m_n if there is a pair of m_i ie. (m_i, m_i) is found, replace the first element of the pairm_i by its corresponding initial. Intensity A_i also replace the second element of the pair m_i by the assigned intensity B_i corresponding to the sorted index of d_i. ie., the pair (m_i, m_i) will be replaced as (A_i, B_i) if the sorted index of m_i is other than $n-1$. The pair (m_i, m_i) will be replaced by $(A_i, -(N-1))$ if the sorted index of m_i is $N-1$.

Step 4: While traversing m_0 through m_n, the message intensity m_i found after the first pair must be replaced by B_i, if the sorted index m_i is other than $N-1$ and replaced by $-(N-1)$ if the sorted index of m_i is $N-1$.

Step 5: Finally, encoded intensitye_i is obtained by repeating step 1 to 4 for all message intensity.

3.1.2 Level 2 encoding

The level 2 encoding converts the encoded intensities e_i to folded intensities f_i. Let the encoded intensity be $e_i = \{e_0, e_1, \ldots\ldots e_N\}$. Note the position of negative numbers. Let the position of negative numbers be represented as $l_i = \{l_0, l_1, \ldots\ldots l_{u-1}\}$. Therefore the number of negative numbers in $e_i$is u. Estimate the magnitude intensity (Ma_i) of encoded intensities (e_i) as an eqn (10)

$$Ma_i = |e_i| \tag{10}$$

Estimate the new sorted index of Ma_i as shown in Fig. 7. Let the new sorted index be g_i.

Based on the New sorted index (g_i) of the magnitude intensity (Ma_i) replace the magnitude intensity (Ma_i) by the New Assigned intensity (V_i) to obtain the transformed intensity (T_i) as shown in Table 2. After estimating the transformed values (T_i) insert '1' just after the elements present in the location l_i to obtain the folded intensity f_i which are embedded in the cover image $S(x, y)$ as eqns (11) and (12),

$$S_1(x,y) = S(x,y) + \left\lfloor \frac{f_i}{2} \right\rfloor \tag{11}$$

$$S_2(x,y) = S(x,y) - \left\lceil \frac{f_i}{2} \right\rceil \tag{12}$$

The accessorial information is $AI \in \{0, 1, -1, 2, -2, 3, -3\ldots\ldots\ldots\}$which corresponds to the new sorted index $\{0,1,2,3,4,5,6\ldots\ldots..\}$. For example if the new sorted index- is$g_i = \{7, 6, 3, 2,$

Table 2 Level 2 Encoding table

New Sorted index g_i	g_0	g_1	g_2	g_3	….	g_{N-3}	g_{N-2}	g_{N-1}
New Assigned intensity V_i	0	−1	2	−2	….	$-\left(\frac{N}{2}-1\right)$	$\frac{N}{2}$	$-\frac{N}{2}$

$5, 1, 0, 4\}$ then the accessorial information is $AI = \{4, -3, 2, -1, 3, 1, 0, -2\}$ as eqn (13).

$$AI = \begin{cases} \frac{-g_i}{2} & if\ g_i\ is\ even \\ \frac{g_i+1}{2} & if\ g_i\ is\ odd \end{cases} \tag{13}$$

The procedure for embedding the data m_i using the proposed two-level data encoding algorithm is summarized as follows.

Input: Original image $S(x, y)$, data b, Key 'K' and Group size L.

Output: Stego images $S_1(x, y)$ and $S_2(x, y)$

Step 1: Select the embeddable pixels whose intensity lies between (2^{L-1}) and $(256-2^{L-1})$.

Step 2: Group *'L'* data bits from b and convert to decimal to obtain decimal intensity(m_i).

Step 3: Apply level 1 encoding to convert the decimal intensity (m_i) to encoded intensity (e_i).

Step 4: Apply level 2 encoding to convert the encoded intensity (e_i) to folded intensity (f_i).

Step 5: Shuffle the folded intensity (f_i) using key 'K'.

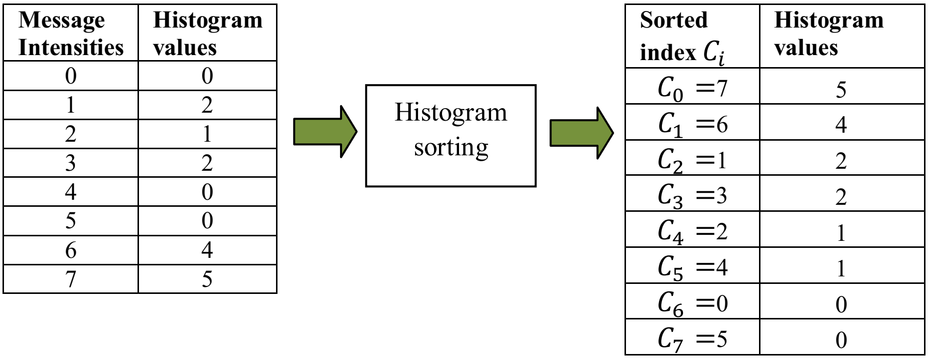
Step 6: Embed the Accessorial information (AI) on first 2^L embeddable pixels and embed the shuffled folded intensity (f_i) on remaining embeddable pixels using eqn (11) and (12)

Consider an example where the message intensity $m_i = \{7, 6, 3, 7, 1, 7, 7, 6, 6, 3, 7, 2, 6, 1\}$. The histogram of the message intensity m_i is tabulated and it is sorted as shown in Fig. 8.

Since $L = 3$, the assigned intensity B_i, initial intensity $A_i$is tabulated as $B_i = \{0, 1, -1, 2, -2, 3, -3\}$ and $A_i = \{4, -4, 5, -5, 6, -6, 7, -7\}$ in Table 3.

Figure 9 shows consider the message intensity $m_i = \{7, 6, 3, 7, 1, 7, 7, 6, 6, 3, 7, 2, 6, 1\}$. From the message m_i, traverse the intensity *'7'* from left to right. There are lonely $7's$ in position 1 and 4. So replace the lonely 7's by the corresponding initial intensity (−7). There is an adjacent pair of 7 in position 6 and 7. Therefore encode the pair (7, 7) as (−7, 0). i.e. replace first element of pair by initial intensity A_i and second element of pair by assigned intensity B_i. After first pair, replace all $7's$ by its assigned intensity '0'. Repeat the same process until all the message intensities are encoded.

Figure 10 shows he encoded intensities e_i isestimated as {−7,7,-5,-7,-4,-7,0,7,1,-5,0, 5,1,-4}. Estimating the location of negative elements to get $l_i = \{1, 3, 4, 5, 6,10,14\}$. The magnitude intensity of e_i is calculated as $Ma_i = \{7, 7, 5, 7, 4, 7, 0, 7, 1, 5, 0, 5, 1, 4\}$. The histogram of the magnitude intensityMa_i is calculated as,



Message Intensities	Histogram values
0	0
1	2
2	1
3	2
4	0
5	0
6	4
7	5

Histogram sorting

Sorted index C_i	Histogram values
C_0 =7	5
C_1 =6	4
C_2 =1	2
C_3 =3	2
C_4 =2	1
C_5 =4	1
C_6 =0	0
C_7 =5	0

Fig. 8 Estimation of sorted index

Table 3 An example of Level 1 Encoding

Sorted index C_i	7	6	1	3	2	4	0	5
Initial intensity A_i	4	−4	5	−5	6	−6	7	−7
Assigned intensity B_i	0	1	−1	2	−2	3	−3	–
Message intensity m_i	0	1	2	3	4	5	6	7

Table 4 shows the transformed values (T_i) is calculated by replacing the sorted magnitude by new assigned intensity(V_i). Therefore the transformed values, $T_i = \{0, 0, -1, 0, 3, 0, 2, 0, -2, -1, 2, -1, -2, 3\}$. Inserting '1' just after the elements present in the location $l_i = \{1, 3, 4, 5, 6,10,14\}$. Figure 11 shows the estimation of the folded intensities $f_i = \{0, 1, 0, -1, 1, 0, 1, 3, 1, 0, 1, 2, 0, -2, -1, 2, -1, -2, 3\}$, and the accessorial information AI is obtained from the new sorted index $g_i = \{7, 5, 0, 1, 4, 2, 3, 6\}$ as $AI = \{4, 3, 0, 1, -2, -1, 2, -3\}$ using eqn (13) which is embedded in first L embeddable pixels. The first four folded intensities $f_i = \{0, 1, 0, -1\}$ is embedded in the pixels $S(x, y) = \{93,124,19,87\}$ by using eqn (11) and (12) as $S_1(x, y) = \{93,124,19,86\}$and $S_2(x, y) = \{93,123,19,87\}$.

3.2 Data extraction

Figure 12 shows the block diagram of data extraction in the proposed algorithm. The data extraction algorithm is inverse to data embedding, where level 2 data decoding and level 1 decoding are performed to extract the hidden data.

Similar to the data embedding process, level 2 decoding converts the folded intensities f_i to encoded intensities e_i, while level 1 decoding converts the encoded intensities e_i to message intensities e_i. The folded intensities is extracted by estimating the difference between the two stego image as eqn (14)

$$f_i = S_1(x,y) - S_2(x,y) \tag{14}$$

Similarly, the Accessorial information (AI) is extracted from the first 2^L embeddable pixels. From the accessorial information (AI) construct the new sorted index g_i using eqn (15). For example, if the accessorial information (AI) is $AI = \{4, -3, 2, -1, 3, 1, 0, -2\}$ then the new sorted index is calculated using the relation $g_i = \{7, 6, 3, 2, 5, 1, 0, 4\}$

$$g_i = \begin{cases} 2 \times |AI| & if\, AI \leq 0 \\ 2 \times AI - 1 & if\, AI > 0 \end{cases} \tag{15}$$

Note the position of 1's in the folded intensity (f_i), also replace the 1's by N. Let the position of 1's present in the folded intensity (f_i) be represented as $l_i = \{l_0, l_1, \ldots\ldots l_{u-1}\}$. Replace the

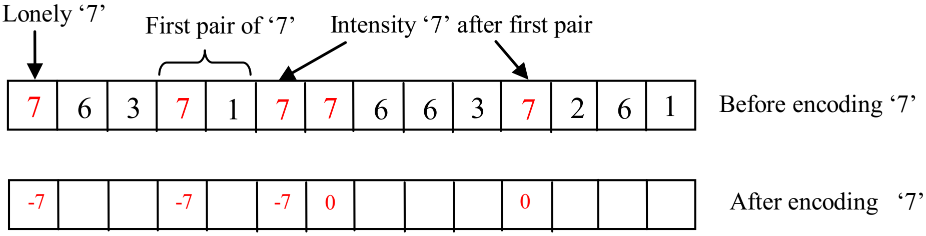
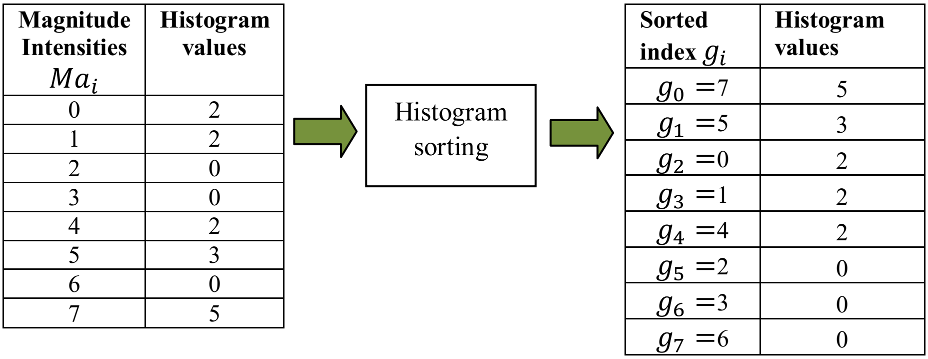


Fig. 9 Encoding of '7' using level 1 encoding



Magnitude Intensities Ma_i	Histogram values
0	2
1	2
2	0
3	0
4	2
5	3
6	0
7	5

Histogram sorting

Sorted index g_i	Histogram values
g_0 =7	5
g_1 =5	3
g_2 =0	2
g_3 =1	2
g_4 =4	2
g_5 =2	0
g_6 =3	0
g_7 =6	0

Fig. 10 Histogram sorting for level 2 encoding

folded intensity (f_i) by their corresponding new sorted index (g_i) shown in Table 5. Also, put a negative sign for the intensities just before the position l_i i.e. in position $\{l_0 - 1, l_1 - 1, \ldots\ldots l_{u-1} - 1\}$. After placing the negative sign discard the intensity N to obtain the encoded intensities (e_i). Let the encoded intensities be $e_i = \{e_1, e_2, \ldots\ldots. e_n\}$.

The procedure for obtaining the folded intensity (f_i) from the encoded intensity (e_i) is shown below.

Step 1: Traverse the encoded intensity (e_i) from the starting value.

Step 2: While traversing, form a pair of intensity (e_i,e_{i+1}), where e_{i+1} is the next adjacent element of e_i.

Step 3: If both the intensities e_i ande$_{i+1}$ belongs to the category initial intensity A_i, replace e_i by the corresponding message intensity m_i.

Step 4: If the first intensity e_i of the pair belongs to the category initial intensity A_i and the second intensity e_{i+1} of the pair belongs to the category assigned intensity B_i. Then replace e_i by its corresponding message intensity m_i. Also, replace all the element of the encoded intensity with the intensity value e_{i+1} by the message intensity m_i.

Step 5: Step 1 to Step 4 is repeated until all the encoded intensities are replaced, which gives the complete message intensity m_i.

The original image $S(x, y)$ is recovered from the stego images $S_1(x, y)$ and $S_2(x, y)$ using the relation as eqn (16).

$$S(x,y) = \left\lceil \frac{S_1(x,y) + S_2(x,y)}{2} \right\rceil \tag{16}$$

The data extraction algorithm is summarized as follows.

Input: Stego images $S_1(x, y)$and $S_2(x, y)$, Key K, Group size L

Output: Binary data, Original image $S(x, y)$

Table 4 Example of level 2 encoding

New Sorted index g_i	7	5	0	1	4	2	3	6
New assigned intensity B_i	0	−1	2	−2	3	−3	4	−4

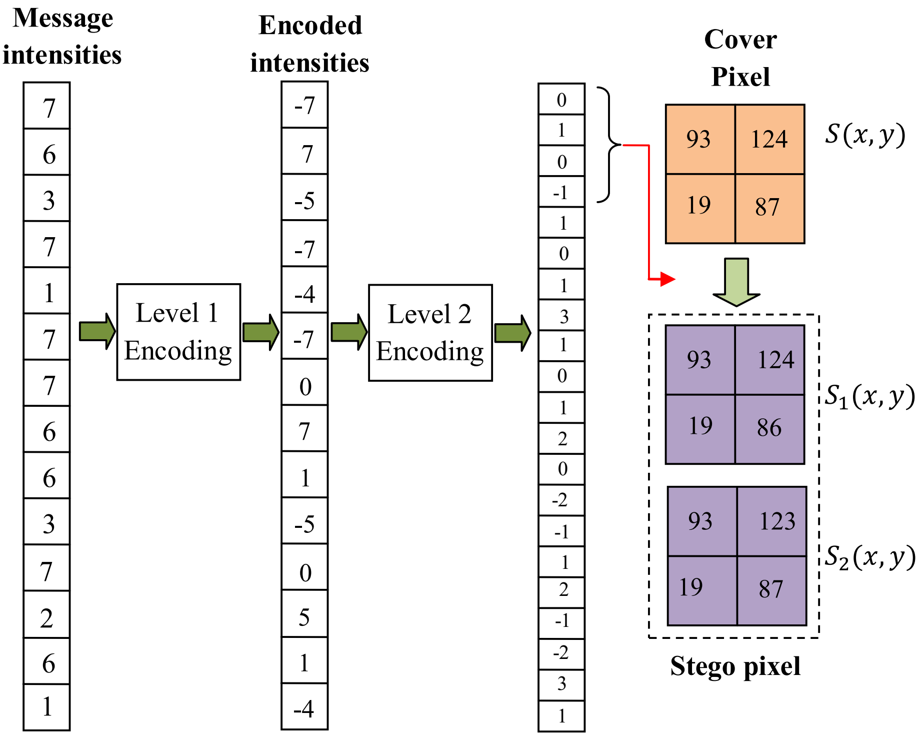


Fig. 11 An example of two-level encoding

Step 1: Identify the embedded stego pixels. Embedded pixels have intensities between $(2^{L-1})$and $(256-2^{L-1})$ in both the stego images $S_1(x,y)$ and $S_2(x,y)$.

Step 2: Find the difference between the first 2^L embedded pixels of the stego images $S_1(x,y)$and $S_2(x,y)$, using eqn (18) to extract the accessorial information.

Step 3: Find the difference between the remaining embedded pixels (other than first 2^L embedded pixels) of the stego images $S_1(x,y)$ and $S_2(x,y)$ using eqn (14) to extract folded intensity f_i. Using the key K, rearrange the folded intensity f_i.

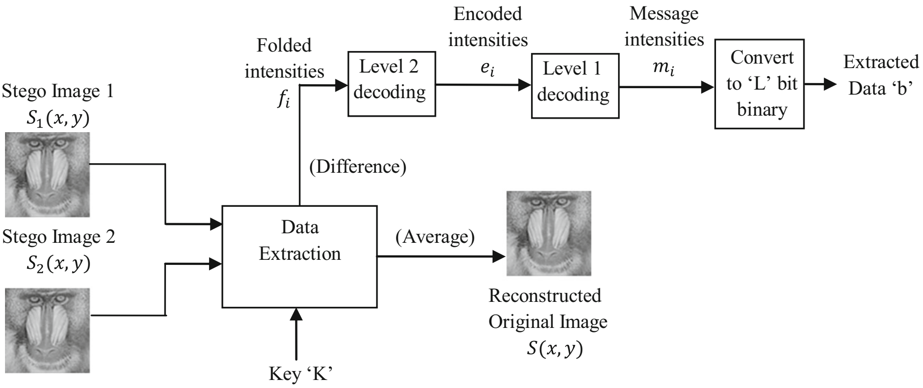


Fig. 12 Block diagram of data extraction in the proposed algorithm

Table 5 Level 2 Decoding table

Folded intensity f_i	0	−1	2	−2	….	$-\left(\frac{N}{2}-1\right)$	$\frac{N}{2}$	$-\frac{N}{2}$
New Sorted index g_i	g_0	g_1	g_2	g_3	….	g_{N-3}	g_{N-2}	g_{N-1}

Step 4: Find the encoded intensity $e_i$from the folded intensity f_i and the accessorial information (AI) using the Level 2 decoding.

Step 5: Apply level 1 decoding to obtain the message intensity m_i from the encoded intensity e_i.

Step 6: Find the secret data by converting the message intensity to L bit binary.

Step 7: Recover the original image $S(x, y)$ from $S_1(x, y)$ and $S_2(x, y)$ using eqn (16).

Fig. 13 shows consider the two stego images, $S_1(x, y) = \{93,124,19,86\}$ $S_2(x, y) = \{93,123,19,87\}$and $L = 4$. The folded intensity f_i is obtained using eqn (14) as $f_i = \{0, 1, 0, -1\}$. Let $f_i = \{0, 1, 0, -1, 1, 0, 1, 3, 1, 0, 1, 2, 0, -2, -1, 1, 2, -1, -2, 3, 1\}$ be the folded intensity obtained from the entire stego images. Let the accessorial information extracted from the first L embedded pixels

are $=\{4, 3, 0, 1, -2, -1, 2, -3\}$. Using eqn (15), the new sorted index $g_i = \{7, 5, 0, 1, 4, 2, 3, 6\}$. Note the position of 1's in the folded intensity f_i, Let the position of 1's be $l_i = \{2, 5, 7, 9, 11,16,21\}$, Also replace the 1's by $N = 8$, to obtain $seq = \{0, 8, 0, -1, 8, 0, 8, 3, 8, 0, 8, 2, 0, -2,$

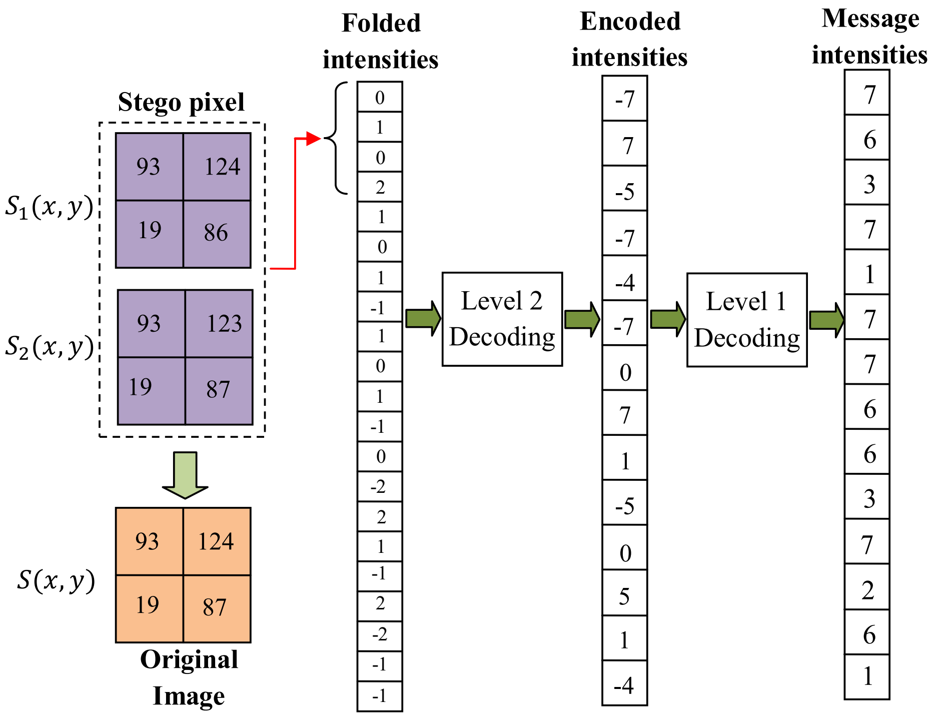


Fig. 13 An example of the data extraction process

Table 6 Example of Level 2 decoding

Folded intensityf_i	0	−1	2	−2	3	−3	4	−4
New sorted index g_i	7	5	0	1	4	2	3	6

$-1, 8, 2, -1, -2, 3, 8\}$. Using the Table 6, transform the folded intensity present in the sequence $seq$to $seq_1 = \{7, 8, 7, 5, 8, 7, 8, 4, 8, 7, 8, 0, 7, 1, 5, 8, 0, 5, 1, 4, 8\}$. Place a negative sign before the elements $N = 8$, To get $seq_2 = \{-7, 8, 7, -5, 8, -7, 8, -4, 8, -7, 8, 0, 7, 1, -5, 8, 0, 5, 1, -4, 8\}$. Discarding $N = 8$, on seq_2 we can obtain the encoded intensity $e_i = \{-7, 7, -5, -7, -4, -7, 0, 7, 1, -5, 0, 5, 1, -4\}$.

Table 7 shows the level 1 decoding table is formed as shown in table. From the encoded intensity $e_i = \{-7, 7, -5, -7, -4, -7, 0, 7, 1, -5, 0, 5, 1, -4\}$, form the pair (−7,7). Here both the elements belongs to initial intensity category, therefore replace −7 by its message intensity 7, Again form a pair (7,-5), both 7 and − 5 belongs to initial intensity category, therefore replace 7 by its message intensity 6. Repeat the same process till the pair (−4,7) is reached. Again form the next pair (−7,0), here −7 belongs to initial intensity category, but 0 belongs to assigned intensity category, therefore replace −7 by its message intensity 7 and all 0 by the same message intensity 7. Repeat the same process to obtain the complete message intensity $m_i = \{7, 6, 3, 7, 1, 7, 7, 6, 6, 3, 7, 2, 6, 1\}$. The original image $S(x, y)$ are obtained from the stego images $S_1(x, y) = \{93,124,19,86\}$, $S_2(x, y) = \{93,123,19,87\}$ using eqn (20) as $S(x, y) = \{93,124,19,87\}$.

4 Experimental results

The data embedding and data extraction of the proposed two-level data hiding algorithm were verified using the 10 standard test images shown in Figs. 14 and 5 rarely used test images shown in Fig. 17. We have used Fig. 14(a)-(f) as cover images and Fig. 14(g)-(j) as secret images. The algorithm was implemented using MATLAB 2018a. The test ‘Original images’ are gray scale images [20, 24] each having a size of 512 × 512 and the test secret image [1, 6, 25], are gray scale images having a size of 216 × 216. The performance of the proposed algorithm was evaluated using the parameters such as embedding rate (bpp), peak signal to noise ratio (PSNR). The embedding rate (bpp) resembles the number of bits hidden in a pixel which is estimated by eqn (17),

$$bpp = \frac{t}{2 \times h \times w} \tag{17}$$

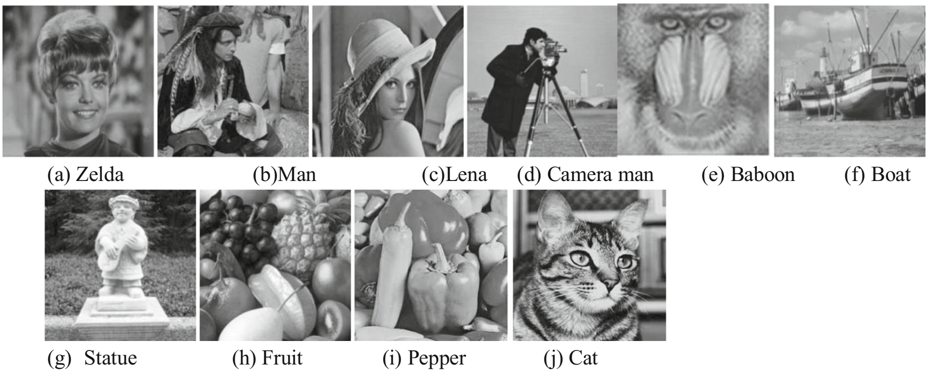
Here $h \times w$ is the size of the original image and t is the capacity of data (in bits) hidden in two stego images. The parameter PSNR shows the distortion level of two stego images from

Table 7 Example of Level 1 decoding

Initial intensity A_i	4	−4	5	−5	6	−6	7	−7
Assigned intensity B_i	0	1	−1	2	−2	3	−3	–
Message intensity d_i	0	1	2	3	4	5	6	7

Table 8 PSNR and Capacity comparison for different test images with a different value of *L*

L	Metrics	Zelda	Man	Lena	Cameraman	Baboon	Boat
3	$PSNR_1$	48.692	48.9004	48.692	48.7018	48.6920	48.692
	$PSNR_2$	50.5036	50.7149	50.5036	50.5141	50.5036	50.5036
	$PSNR_{avg}$	49.5978	49.80765	49.5978	49.60795	49.5978	49.5978
	Capacity	540,121	499,175	539,335	522,453	541,065	541,013
4	$PSNR_1$	46.9756	46.9756	46.9756	46.9756	46.9756	46.9756
	$PSNR_2$	47.8104	47.8104	47.8104	47.8104	47.8104	47.8104
	$PSNR_{avg}$	47.393	47.393	47.393	47.393	47.393	47.393
	Capacity	804,782	690,540	802,003	759,117	809,920	809,291
5	$PSNR_1$	43.4907	43.4907	43.4907	43.4907	43.4907	43.4907
	$PSNR_2$	43.8714	43.8714	43.8714	43.8714	43.8714	43.8714
	$PSNR_{avg}$	43.68105	43.68105	43.68105	43.68105	43.68105	43.68105
	Capacity	1,021,365	840,696	1,002,911	879,965	1,053,714	1,036,255


(a) Zelda (b)Man (c)Lena (d) Camera man (e) Baboon (f) Boat

(g) Statue (h) Fruit (i) Pepper (j) Cat

Fig. 14 Test Images (*Cover Images are (a) – (f) and secret images are (g) – (j)*)

the original image which is calculated using the eqns (18) and (19),

$$PSNR_1 = 10 \times \log_{10}\left[\frac{255^2}{\frac{1}{h \times w}\sum_{x=1}^{h}\sum_{y=1}^{w}\left(S(x,y)-S_1(x,y)\right)^2}\right] \tag{18}$$

$$PSNR_2 = 10 \times \log_{10}\left[\frac{255^2}{\frac{1}{h \times w}\sum_{x=1}^{h}\sum_{y=1}^{w}\left(S(x,y)-S_2(x,y)\right)^2}\right] \tag{19}$$

The average PSNR ($PSNR_{avg}$) and the PSNR of the two stego images $PSNR_1$and $PSNR_2$ are related as eqn (20),

$$PSNR_{avg} = \frac{(PSNR_1 + PSNR_2)}{2} \tag{20}$$

Table 8. The performance of the proposed algorithm was evaluated for different values of group size 'L' ($L = 3$, $L = 4$, $L = 5$) for different cover images. For $L = 3$, the embedding capacity is around $t = 54{,}000$ bits, while the PSNR of the first stego image is around 48.7 dB, while the PSNR of the second image is around 50.6 dB. The average PSNR ($PSNR_{avg}$) is around 49.6 dB. As the value of L is increased to 4 and 5, the embedding capacity is increased, but the PSNR reduces. However, there is a high increase in capacity at slight reduction in PSNR. The embedding capacity at $L = 4$ and $L = 5$ are around$t = 80{,}000$ bits and $t = 10{,}00{,}000$ bits respectively. The average PSNR for $L = 4$ is 47.393 dB and the average PSNR for $L = 5$ is 43.68105 dB. The embedding capacity is not constant for a particular value of L for different cover images. For example for $L = 4$, the cover image 'Zelda' has a embedding capacity of $t = 10{,}21{,}365$ bits, while the cover image 'Man'has a embedding capacity of $t = 8{,}40{,}696$ bits. The reason is the number of embeddable pixels varies in different cover images. i.e. The cover image 'Zelda' has more embeddable pixels when compared to the cover image 'Man'.

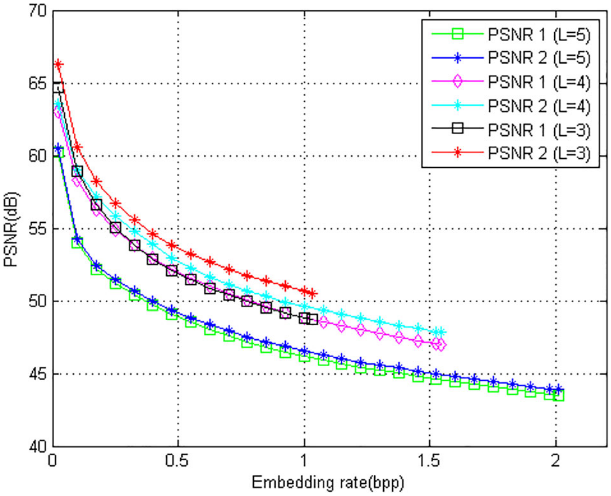


Fig. 15 $PSNR_1$ and $PSNR_2$ Comparison of the cover image 'Baboon'

Figure 15 shows the $PSNR_1$ and $PSNR_2$ Comparison of the cover image 'Baboon' for different values of L. The difference between $PSNR_1$ and $PSNR_2$ for $L = 3$ is high when compared to $L = 4$and $L = 5$. For a embedding rate of $bpp = 1$ and $L = 3$, the PSNR difference between the two stego image is around$2dB$. For $L = 4$ and $L = 5$, the PSNR difference between the two stego images is less than 1. As the value of L increases the PSNR reduces around 3 dB, but the embedding rate increases around 0.5bpp.

The PSNR and embedding capacity of the proposed method was compared with the traditional methods such as Chang et al. (2007) [3],Chang et al. (2013) [4],Qin et al. (2015) [17], Ki-Hyun Jung (2017) [10],Wang et al. (2018) [23], Data Encoding approach (2019) [18]. The proposed method provides a low PSNR when compared to our previous Data Encoding approach (2019), however, the embedding capacity of the proposed method is around 8,00,000 bits which is very high than the traditional methods. The comparison of PSNR and Embedding capacity is tabulated in Table 9. Fig. 16 shows a graphical comparison of proposed and traditional methods for different embedding rate. While embedding the data at different embedding rate, the PSNR is found almost the same in different test images. But the maximum embedding capacity differs in different cover images because of the change in a number of embeddable pixels. The graph depicted in Fig. 16 concludes that the proposed method shows a high PSNR than the traditional methods for $L = 3$ and $L = 4$ for different test images. We have also evaluated the quality of the stego image using SSIM (structural similarity index measurement). The average SSIM ($SSIM_{avg}$) can be estimated as,

$$SSIM_{avg} = (SSIM_1 + SSIM_2)/2 \tag{21}$$

Where $SSIM_1$ and $SSIM_2$ can be estimated between the cover image $S(x, y)$ and stego images $S_1(x, y)$ and $S_2(x, y)$ respectively using the relation,

$$SSIM_i = L(S_i, S)\ C\ (S_i, S)\ Sa(S_i, S) \quad i \in \{1, 2\} \tag{22}$$

Here $L(S_i, S)$, $C\ (S_i, S)$ and $Sa(S_i, S)$ are the Luminance, contrast and saturation comparison function respectively. We have also used the rarely used test images shown in Fig. 17 taken from (http://decsai.ugr.es/cvg/CG/base.htm) to evaluate SSIM performance.

Table 9 PSNR and Embedding capacity comparison of proposed and traditional methods

Method	Metric	Zelda	Man	Lena	Cameraman	Baboon	Boat
Chang et al. (2007) [3]	$PSNR_1$	45.13	45.14	45.13	45.12	45.58	45.11
	$PSNR_2$	45.11	45.14	45.13	45.13	45.13	45.13
	$PSNR_{avg}$	45.12	45.14	45.13	45.36	45.13	45.12
	Capacity	524,196	524,288	524,288	524,288	524,288	524,148
Chang et al.(2013) [4]	$PSNR_1$	39.89	39.94	39.9	39.89	39.88	39.91
	$PSNR_2$	39.89	39.9	39.89	39.88	39.89	39.91
	$PSNR_{avg}$	39.89	39.9	39.89	39.88	39.89	39.91
	Capacity	802,535	802,698	802,895	802,789	802,895	802,524
Qin et al.(2015) [17]	$PSNR_1$	52.06	52.12	52.11	51.72	52.11	52.04
	$PSNR_2$	41.57	41.58	41.58	41.75	41.58	41.56
	$PSNR_{avg}$	46.82	46.85	46.85	46.74	46.85	46.8
	Capacity	557,129	557,194	557,052	557,264	557,052	557,096
Ki-Hyun Jung (2017) [10]	$PSNR_1$	48.03	48.12	48.18	48.15	48.18	48.18
	$PSNR_2$	47.04	47.18	47.26	47.19	47.26	47.2
	$PSNR_{avg}$	47.535	47.68	47.705	47.685	47.705	47.69
	Capacity	519,180	519,180	519,180	519,180	519,180	519,180
Wang et al. (2018) [23]	$PSNR_1$	42.69	42.72	42.71	42.7	42.71	42.63
	$PSNR_2$	38.99	39.01	38.09	39.01	38.09	38.01
	$PSNR_{avg}$	40.84	40.86	40.395	40.86	40.395	40.32
	Capacity	786,432	786,432	786,432	786,432	786,432	786,432
Data Encoding Approach [18] (2019)	$PSNR_1$	48.5341	48.5341	48.5341	48.4341	48.5341	48.5341
	$PSNR_2$	48.5711	48.5711	48.5711	48.5711	48.5711	48.5711
	$PSNR_{avg}$	48.5526	48.5526	48.5526	48.5526	48.5526	48.55255
	Capacity	629,146	629,146	629,146	629,146	629,146	629,146
Proposed (L = 4)	$PSNR_1$	46.9756	46.9756	46.9756	46.9756	46.9756	46.9756
	$PSNR_2$	47.8104	47.8104	47.8104	47.8104	47.8104	47.8104
	$PSNR_{avg}$	47.393	47.393	47.393	47.393	47.393	47.393
	Capacity	804,782	690,540	802,003	759,117	809,920	809,291

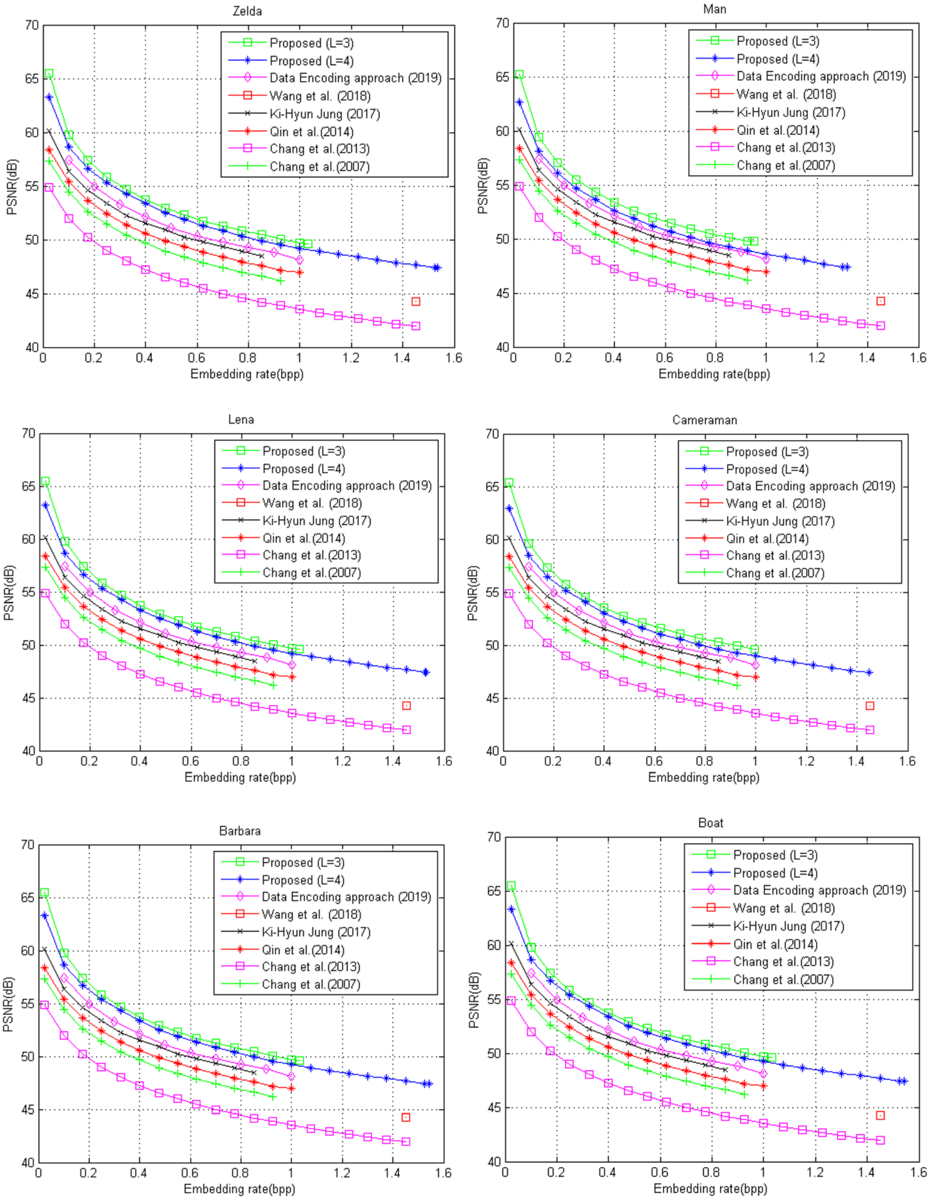
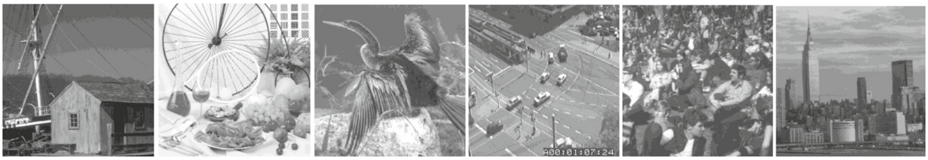


Fig. 16 PSNR comparison for different test images



(a)House (b) Wheel (c) Bird (d) Traffic (e) Crowd (f) Tower

Fig. 17 Rarely used test images from (http://decsai.ugr.es/cvg/CG/base.htm)

Table 10 PSNR, SSIM and Capacity comparison for the test images shown in Fig. 18

L	Metrics	House	Wheel	Bird	Traffic	Crowd	Boat
3	$PSNR_1$	48.7218	48.692	48.9004	48.692	48.792	48.792
	$PSNR_2$	50.5141	50.5136	50.7049	50.5136	50.5736	50.5336
	$PSNR_{avg}$	49.61795	49.6028	49.80265	49.6028	49.6828	49.6628
	Capacity	522,581	541,001	499,932	540,103	539,423	541,026
	$SSIM_{avg}$	0.9988	0.9986	0.9990	0.9986	0.9997	0.9997
4	$PSNR_1$	46.9556	46.9756	46.9856	46.9256	46.9756	46.9356
	$PSNR_2$	47.8304	47.8004	47.8152	47.8134	47.8104	47.8164
	$PSNR_{avg}$	47.393	47.388	47.4004	47.3695	47.393	47.376
	Capacity	759,232	809,093	690,780	804,342	802,012	809,302
	$SSIM_{avg}$	0.9888	0.9883	0.9886	0.9883	0.989	0.9891
5	$PSNR_1$	43.5007	43.4807	43.4917	43.4932	43.4827	43.4807
	$PSNR_2$	43.8814	43.8723	43.8797	43.8734	43.8764	43.8734
	$PSNR_{avg}$	43.6911	43.6765	43.6857	43.6833	43.6796	43.6771
	Capacity	879,996	1,036,001	840,856	1,021,002	1,002,101	1,053,121
	$SSIM_{avg}$	0.9788	0.9779	0.9781	0.9779	0.9782	0.9785

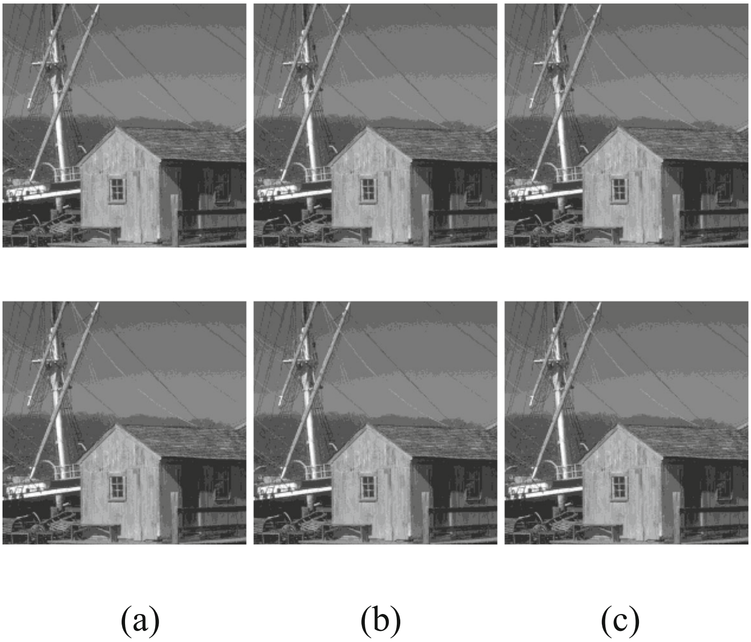


Fig. 18 Dual stego images for different values of L (a) $L = 3$ (b) $L = 4$ (c) $L = 5$ (*First row:* Stego image 1, *Second row:* Stego image 2)

Table 10 shows the PSNR, SSIM and capacity comparison for the test images shown in Fig. 18 with different values of L. Similar to the test images shown in Fig. 15, the average PSNR of the images shown in Fig. 18 decreases as the value of L increases. The capacity increases as L increases from 3 to 5. The SSIM for $L = 4$, $L = 5$, and $L = 6$ is around 0.9988, 0.9888 and 0.9788 respectively. Higher values of SSIM indicates that the proposed method produces high quality stego images.

Figure 18 shows the visual appearance of the stego images obtained for the cover image shown in Fig. 18 (a). There is only a slight change in visual quality as the value of L increases from 3 to 5. The next section shows the conclusion of the proposed work.

5 Conclusion

This paper proposed a two-level data encoding approach for reversible data hiding in dual stego images. The two-level data encoding includes level 1 encoding that converts message intensities to encoded intensities and level 2 encoding converts encoded intensities to folded intensities. This proposed method assigns lower intensities to intensities that have higher histogram and higher intensities to intensities that have a lower histogram. The aim of the proposed method is to represent the most repeated secret information by lower intensity. The experimental verification was done on standard test images for different values of group size 'L' and comparison were made with the traditional RDH methods. The proposed method provides a high PSNR for $L = 3$. As the value of L increases to 4 and 5, the PSNR reduces around 3 dB, but the embedding rate increases around 0.5 bpp. The experimental results shows that the proposed method outperforms than the traditional dual stego image based reversible data hiding schemes in terms of image quality and embedding capacity.

References

1. (Archive) CS103 S15 — Picture class (n.d.) http://bits.usc.edu/cs103-sp15/picture/ (accessed July 22, 2019).
2. X. Cao, S. Member, L. Du, X. Wei, D. Meng, High Capacity Reversible Data Hiding in Encrypted Images by Patch-Level Sparse Representation, (2015) 1–12.
3. Chang CC, Kieu TD, Chou YC (2007) Reversible Data Hiding Scheme Using Two Steganographic Images, TENCON 2007–2007 IEEE Reg. 10 Conf. 1–4. https://doi.org/10.1109/TENCON.2007.4483783.
4. Chang CC, Lu TC, Horng G, Huang YH, Hsu YM (2013) A high payload data embedding scheme using dual stego-images with reversibility, ICICS 2013 - Conf. Guid. 9th Int. Conf. Information, Commun. Signal Process. 1–5. https://doi.org/10.1109/ICICS.2013.6782790.
5. Dragoi I, Coltuc D (2014) Local-prediction-based difference expansion reversible watermarking. IEEE TransImage Process 23:1779–1790
6. Emmanuel AUBERT - LCM3B UHP Nancy 1, (n.d.) http://crm2.univlorraine.fr/pages_perso/Aubert/FTenglish/TheoIm/theoim.html (accessed July 22, 2019).
7. Hong W, Chen M, Chen TS (2017) An efficient reversible image authentication method using improved PVO and LSB substitution techniques. Signal Process Image Commun 58:111–122. https://doi.org/10.1016/j.image.2017.07.001
8. Huang F, Qu X, Kim HJ, Huang J (2015) Reversible Data Hiding in JPEG Images 8215:1–12. https://doi.org/10.1109/TCSVT.2015.2473235
9. J.V.C.I.R, Yao H, Qin C, Tang Z, Tian Y (2017) Guided filtering based color image reversible data hiding q. J Vis Commun Image Represent 43:152–163. https://doi.org/10.1016/j.jvcir.2017.01.004
10. Jung K (2017) Authenticable reversible data hiding scheme with less distortion in dual stego-images. Multimed Tools Appl 77:6225–6241. https://doi.org/10.1007/s11042-017-4533-0
11. Li M, Li Y (2016) Histogram shifting in encrypted Images with public key cryptosystem for reversible data hiding. Signal Process. https://doi.org/10.1016/j.sigpro.2016.07.002
12. Li X, Li J, Li B, Yang B (2013) High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. Signal Process 93:198–205. https://doi.org/10.1016/j.sigpro.2012.07.025
13. Li X, Zhang W, Gui X, Yang B (2016) Efficient reversible data hiding based on Multiple Histograms Modification 10:2016–2027
14. Lo C, Hu Y (2014) A novel reversible image authentication scheme for digital images. Signal Process 98: 174–185. https://doi.org/10.1016/j.sigpro.2013.11.028
15. Nikolaidis A (2015) Reversible data hiding in JPEG images utilising zero quantised coefficients. Reversible data hiding in JPEG images utilising zero quantised coefficients 9:560–568. https://doi.org/10.1049/iet-ipr.2014.0689
16. Ou B, Li X, Zhao Y, Ni R (2015) Efficient color image reversible data hiding based on channel-dependent payload partition and adaptive embedding 108:642–657. https://doi.org/10.1016/j.sigpro.2014.10.012
17. Qin C, Chang C, Hsu T (2014) Reversible data hiding scheme based on exploiting modification direction with two steganographic images. Multimed Tools Appl 74:5861–5872. https://doi.org/10.1007/s11042-014-1894-5
18. C. Shaji, I.S. Sam, A new data encoding based on maximum to minimum histogram in reversible data hiding, Imaging Sci J 0 (2019) 1–13. https://doi.org/10.1080/13682199.2019.1592892, 67.
19. Shi YQ, Reversible Data Hiding, (2005) 1–12. https://doi.org/10.1109/ICIP.2002.1039911.
20. Test Images (n.d.) https://homepages.cae.wisc.edu/~ece533/images/ (accessed July 22, 2019).
21. Thodi DM, Rodríguez JJ, Member S (2007) Expansion embedding techniques for Reversible Watermarking 16:721–730
22. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol 13:890–896. https://doi.org/10.1109/TCSVT.2003.815962
23. Wang Y, Shen J, Hwang M (2018) A Novel Dual Image-based High Payload Reversible Hiding Technique Using LSB Matching 20:801–804. https://doi.org/10.6633/IJNS.201807
24. Weblet Importer (n.d.) https://www.hlevkin.com/06testimages.htm (accessed July 24, 2019).
25. Weblet Importer (n.d.) https://www.cs.montana.edu/courses/spring2004/430/lectures/02/lect02.html (accessed July 22, 2019).
26. Weng S, Zhao Y, Pan J, Ni R (2008) Reversible Watermarking Based on Invariability and Adjustment on Pixel Pairs 15:721–724
27. Weng S, Pan J, Li L, Zhou L (2016) Reversible data hiding based on an adaptive pixel-embedding strategy and two-layer embedding. Inf. Sci. (NY). https://doi.org/10.1016/j.ins.2016.05.030
28. Wu H, Dugelay J, Shi Y (2015) Reversible Image Data Hiding with Contrast Enhancement 22:81–85
29. Zhang X (2011) Reversible data hiding in encrypted image. IEEE Signal Process 18:255–258

30. Zhang X, Wang S (2006) Efficient Steganographic Embedding by Exploiting Modification Direction, 10 781–783.
31. Zhang W, Zhao X, Yu N, Li F (2013) Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption 8:553–562
32. Luo ZXL, Chen Z, Chen M, Zeng X (2010) Reversible image watermarking using interpolation technique. IEEE Trans Inf Forensics Secur 5:187–193

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.