Computing and Informatics, Vol. 38, 2019, 321–342, doi: 10.31577/cai_2019_2_321

ADAPTIVE REVERSIBLE DATA HIDING SCHEME FOR DIGITAL IMAGES BASED ON HISTOGRAM SHIFTING

Sonal Kukreja, Geeta Kasana, Singara Singh Kasana

Computer Science and Engineering Department
Thapar Institute of Engineering and Technology
Patiala, Punjab, India-147004
e-mail: kukreja.sonal@gmail.com, {gkasana, singara}@thapar.edu

Abstract. Existing histogram based reversible data hiding schemes use only absolute difference values between the neighboring pixels of a cover image. In these schemes, maxima and minima points at maximum distance are selected in all the blocks of the image which causes shifting of the large number of pixels to embed the secret data. This shifting produces more degradation in the visual quality of the marked image. In this work, the cover image is segmented into blocks, which are classified further into complex and smooth blocks using a threshold value. This threshold value is optimized using firefly algorithm. Simple difference values between the neighboring pixels of complex blocks have been utilized to embed the secret data bits. The closest maxima and minima points in the histogram of the difference blocks are selected so that number of shifted pixels get reduced, which further reduces the distortion in the marked image. Experimental results prove that the proposed scheme has better performance as compared to the existing schemes. The scheme shows minimum distortion and large embedding capacity. Novelty of work is the usage of negative difference values of complex blocks for secret data embedding with the minimal number of pixel shifting.

Keywords: Reversible data hiding, complexity, MSE, PSNR, histogram shifting, BPCS, firefly

1 INTRODUCTION

With the growth in multimedia communication, security of digital data being transferred and stored has been an important concern. Data hiding is one of the approaches used for security of digital data. It allows the sender to embed the information into digital contents like text, images, audio and video. The digital object in which the information is embedded is termed as cover object and the information to be embedded is termed as secret data. The cover object after embedding the secret data is known as marked object. In a Reversible Data Hiding (*RDH*) the secret data is embedded inside some cover media in such a way that it should not be visible with the human eyes. At the receiver side, the scheme should have the ability to restore the cover media as well as the secret data without any distortion [1, 14]. Specific research has been done on the sensitive images like medical images, satellite images, etc. [22].

Histogram modification based schemes are a remarkable contribution in *RDH* schemes. In these approaches, the statistical properties of the image are used to embed the secret data. Ni et al. [14] proposed the *RDH* scheme, in which the zero and peak points of a cover image histogram are recorded. All the pixel values lying in between these zero and peak values are shifted by one to create the empty spaces. These spaces are then used to embed the secret data bits. Peak Signal to Noise Ratio (*PSNR*) of this scheme is around 48 dB with a good embedding capacity. A modification of this scheme was proposed by the same author and it was used as an authentication scheme for semi fragile images [15]. This scheme was further improved by Fallahpour et al. [3] who proposed that instead of hiding the data in the entire image, the image can be divided into blocks and then the same histogram shifting using the peak and the zero points can be repeated for all the blocks, thereby enhancing the embedding capacity. This embedding capacity was further enhanced by Tsai et al. [23] by embedding the secret data in the residual images instead of image histograms. The pixels within the image were classified as wall pixels and non wall pixels, and then the interpolation error of the wall pixels, and the difference values of the non wall pixels and the parent pixels were used for data embedding. The embedding capacity was further enhanced by exploring the prediction values in a rhombus prediction scheme in Chang et al. [2]. Another variation in the histogram based approach was proposed by Wang et al. [24] by introducing the concept of location maps and manipulating the maximum frequency values of the histogram assuming their intensity values and updating these peak values with the other pixel value of the same segment. Hong et al. [7] enhanced the embedding capacity by using the dual binary trees instead of shifting the difference values. To construct a sharper histogram, Lin et al. [11] utilized difference image histograms. The correlation of two adjacent pixels is considered in this scheme. In [10], the host image is divided into sub-images by sampling. One of these sub-images is selected as a reference image to compute differences with others and the secret message is embedded by multilevel histogram modification. Luo et al. [12] improved this scheme by selecting the median of every block to construct a reference image, which leads to a sharper histogram. In Pan et al. scheme [18], histogram is constructed for every block, and the peak point is selected as the reference pixel to compute differences with other pixels.

1.1 Motivation for the Proposed Work

From the literature survey, it is concluded that most of the existing histogram shifting based *RDH* schemes [3, 14, 11] have not considered the following points:

1. The regions within an image are not uniform; some regions are smooth while some are complex. Block based histogram shifting schemes did not consider the complexity of the blocks in the embedding process.
2. Usually in the existing histogram based shifting schemes, maxima and minima are selected to shift the in-between pixels to embed the data. The maxima and the minima point are selected corresponding with the pixels having maximum and minimum frequency value respectively in the image. Out of the various minima (zero) points, any random minima is selected in these schemes. If this is selected judiciously, shifting can be minimized, thereby, minimizing image distortion.
3. In the existing difference image schemes, negative maxima points have not been utilized for data embedding, i.e., in the existing histogram shifting approaches only the absolute differences between the neighboring pixels have been used for data embedding.

1.2 Contribution

The novelty of the proposed scheme is summarized as follows:

1. Complexity of the blocks is evaluated by using Bit Plane Complexity Segmentation (*BPCS*). An optimized threshold value is evaluated by using firefly algorithm and is used to classify a block as smooth or complex. The complex blocks have been used to embed the data, which helps in enhancing the embedding capacity as well as the imperceptibility.
2. Maxima and minima (zero) points of difference blocks are chosen in such a way that the distance between them is least, due to which the shifting of the pixels get minimized, hence reduction in distortion, thereby enhancing the visual quality of the image.
3. In the proposed scheme, the difference blocks contain simple difference values, i.e. positive as well as negative difference values, in which the negative difference value pixels are able to store more than one bit of secret data, which enhances the embedding capacity accordingly maintaining the similar visual quality of the image.

1.3 Organization of the Work

The paper has been organized into following sections. Section 2 discusses the background of the proposed scheme. The proposed scheme itself, which comprises of the schemes of data embedding, and data extraction and handling of the side information are discussed in Section 3. Experimental results and comparisons of the proposed scheme with existing schemes have been illustrated in Section 4 followed by the conclusion in Section 5.

2 RELATED WORK

The concepts of histogram shifting, image complexity and firefly algorithm, used in the proposed scheme, are discussed in this section.

2.1 Histogram Shifting

Reversible Data Hiding using Histogram Shifting was proposed by Ni et al. [14] in 2006. The main concept of the shifted-histogram data hiding method is to find a pair of maxima and minima in the histogram of the image and then shift the intensity of the pixels, which lie between the maxima and minima by one level, towards the minima. If the minima lies to the right of the maxima, all the pixel values between them are incremented by 1, while if the minima lies to the left of the maxima, all the values are decremented by 1, as shown in the Figure 1. This creates an empty space on the shifted histogram around the maxima pixel value. To embed a data stream, the shifted image is re-scanned and when the pixel of maximum frequency is encountered its gray value is incremented by one, if the corresponding bit in the embedding stream is '1' otherwise it remains unaltered. Thus, the largest number of bits which can be hidden into the image is equal to the maximum frequency of the histogram. Owing to the created gap, the data hiding mechanism is reversible. The values of the pixels with maximum and minimum frequency are also recorded as side information. If the minimum frequency is non-zero, then their numbers also need to be embedded as the side information, which reduces the data hiding capacity of the system.

2.2 *RDH* Based on Histogram Modification of Difference Images

In this section, Lin et al. scheme [11] is briefly reviewed in which the cover image is divided into non-overlapping blocks of equal size and difference image is generated for every block. In the histogram based data hiding schemes, the higher the number of peak points, the larger will be its hiding capacity. With the help of the histogram approach, peak values are recorded for every block from its difference image. In the difference image, the grayscale value of the maximum occurrence tends to be around 0. The objective in their scheme is to increase the occurrence of peak point

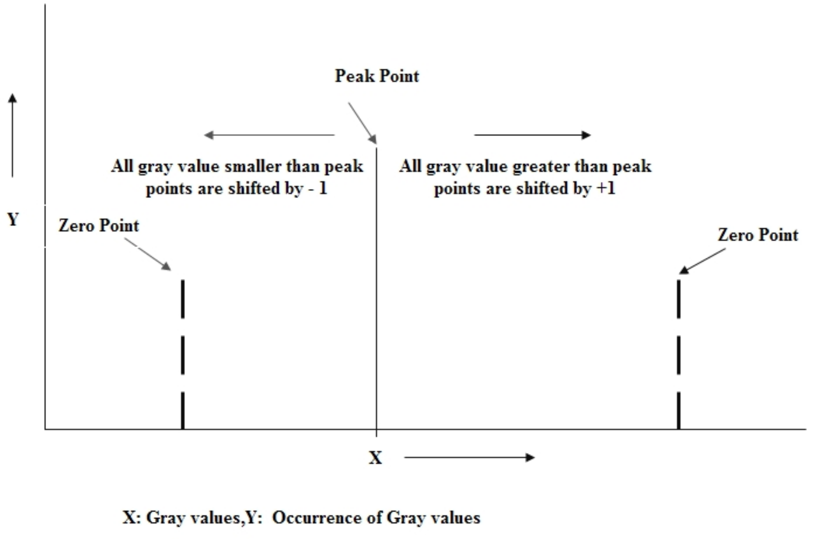


Figure 1. Shifting of pixels in the histogram of an image

and hence the hiding capacity of the proposed scheme. During the hiding phase, empty space is created in the difference image by shifting the histogram and hiding the secret data by using the histogram shifting process. The marked image is generated using original cover image and modified difference image. For the receiver part, same steps are applied in the reverse order to retrieve the original image from the marked image.

2.3 Image Complexity

The data hiding scheme outlined in this paper uses the complex region of the image to embed data. Image complexity has no standard measure. Kawaguchi [9] proposed the measure to evaluate image complexity, which is also known as black-and-white (B-W) border image complexity.

The black-and-white border length is recorded in the binary image to measure the image complexity. The longer border indicates complex image, else it is a simple image. The length of the border is calculated by the summation over the number of times the color changes along all the rows and columns of the image. For example, if a black pixel is surrounded by white background pixels, then it has the border length of 4. The image complexity is defined as follows:

$$\alpha = \frac{k}{m} \tag{1}$$

where m is maximum possible B-W changes in the image and k is the total length of B-W border in the image. Hence the value ranges over, $0 \leq \alpha \leq 1$. The above complexity measure is defined only for binary images. The 8-bit grayscale images are split into 8 binary planes. This splitting of image into its constituent binary planes is called Bit-Plane Slicing. This is performed in Pure-Binary Coding system (PBC) where the intensity values are represented as 8 bit binary numbers, but it suffers from a serious drawback, known as Hamming Cliff [20]. This issue is resolved by using the coding system called Canonical Gray Coding System (CGC), where successive decimal numbers differ in their representation by just one bit. Thus in *BPCS* firstly the absolute intensity values are converted into CGC by *PBC-to-CGC* mapping. This is followed by bit-plane decomposition on the CGC values, and the 8 binary images obtained are called the CGC images. These CGC images do not suffer from Hamming cliffs.

The complexity measure for the grayscale image is explained with the help of an example in Figure 2. An image block and its binary representation is shown in the figure. This block is divided into 8 bit planes. k_i is calculated for every block, where $i \in [1, 8]$. k_i is length of B-W border which equals summation of number of color changes along the rows and columns in the image block. Maximum length (m) = size of image × number of bit planes. For the given image block, size of image = 3×3 and number of bit planes = 8. Then, complexity (α) is calculated using Equation (1).

2.4 Firefly Algorithm

Firefly Algorithm (*FA*), proposed by Yang et al. [25], is a swarm intelligence optimization technique. *FA* is inspired by flashing behavior of fireflies. Two basic functions of the flash light are to attract mating partners and to attract potential prey. *FA* is based upon the assumption that solution of an optimization problem can be perceived as fireflies whose brightness is proportional to the value of its objective function within a given problem space. In the *FA*, there are three idealized rules:

1. All fireflies are unisexual, so that any individual firefly will be attracted to all other fireflies.
2. Attractiveness is proportional to their brightness, and for any two fireflies, the less bright one will be attracted by (and thus move towards) the brighter one; however, the intensity (apparent brightness) decreases as their mutual distance increases.
3. If there are no fireflies brighter than a given firefly, it will move randomly.

3 PROPOSED SCHEME

In this section, the data embedding algorithm, data extraction algorithm and prevention of overflow or underflow has been discussed.

167	133	111
144	140	135
159	154	148

Image Block

10100111	10000101	01101111
10010000	10001100	10000111
10011111	10011010	10010100

Binary Representation of Image Block

1	1	0
1	1	1
1	1	1

$k_1 = 2$

0	0	1
0	0	0
0	0	0

$k_2 = 2$

1	0	1
0	0	0
0	0	0

$k_3 = 4$

0	0	0
1	0	0
1	1	1

$k_4 = 4$

0	0	1
0	1	0
1	1	0

$k_5 = 7$

1	1	1
0	1	1
1	0	1

$k_6 = 6$

1	0	1
0	0	1
1	1	0

$k_7 = 8$

1	1	1
0	0	1
1	0	0

$k_8 = 6$

Bit Planes and their corresponding B-W changes

Calculation for complexity of a block:

$$m = 72$$
$$k = k_1 + k_2 + \ldots + k_8$$
$$= 2 + 2 + 4 + 4 + 7 + 6 + 8 + 6$$
$$= 39$$

$$\text{complexity} = k / m$$
$$= 39 / 72$$
$$= 0.541$$

Figure 2. Representation of bit plane slicing and complexity measure

Algorithm 1 Firefly algorithm for maximum optimization

Objective function: $f(x)$, where $x = (x_1, x_2, x_3, \ldots, x_t)$;
Generate Initial Population of fireflies x_i where $i = 1, 2, 3, \ldots, t$;
Formulate light intensity I in association with $f(x)$
Define absorption coefficient γ
while ($t <$ MaxGeneration) **do**
  for $i \leftarrow 1$ to t **do**
    for $j \leftarrow 1$ to t **do**
      if $(I_i > I_j)$ **then**
        Vary attractiveness with distance r *via* $exp(-\gamma r)$;
        move firefly i towards j
        Evaluate new solutions and update light intensity
  Rank fireflies and find current best

3.1 Data Embedding

Let *CI* be the cover image of size $M \times N$ and *SI* be the Secret Data. A functional block diagram of the embedding procedure is represented in Figure 3 and discussed in Algorithm 2.

Algorithm 2 Embedding Algorithm

Input: Cover Image *CI* of size $M \times N$, Secret Data Image *SI*
Output: Marked Image *MI*

1: The cover image *CI* is divided into blocks CI_b of size $r \times c$. The number of blocks n created for the image would be: $n = \dfrac{M \times N}{r \times c}$
2: Complexity value is calculated for all the blocks using Kawaguchi et al. scheme [9], described in Section 2.3. Then blocks are categorized into two categories: Complex and Smooth. The set of blocks having complexity value greater than the threshold are considered as Complex Blocks while the set of blocks having complexity value lesser than the threshold are considered as Smooth Blocks. An optimized threshold value is evaluated by using *FA*, mentioned in Algorithm 1. The set of complex blocks is used for data embedding.

An example representing embedding algorithm has been shown in Figure 4.

3.2 Data Extraction

In this procedure, the embedded secret data bits are extracted from the marked image *MI* and it is restored to its original image without any distortion. The steps within the extraction procedure are discussed in Algorithm 3:

3: From every selected complex block CI_b, absolute difference image block CAD_b of size $r \times (c-1)$ is evaluated by using the formula:

$$CAD_b(i,j) = |CI_b(i,j) - CI_b(i,j+1)|, \tag{2}$$

for $0 \leq i \leq r-1$, $0 \leq j \leq c-2$, $1 \leq b \leq n$

4: A simple difference image CSD_b of size $r \times (c-1)$ of each block b is evaluated by using the formula:

$$CSD_b(i,j) = CI_b(i,j) - CI_b(i,j+1), \tag{3}$$

for $0 \leq i \leq r-1$, $0 \leq j \leq c-2$, $1 \leq b \leq n$

5: For all the absolute difference image blocks CAD_b, histograms are generated and maxima $\max_b$ and minima $\min_b$ are recorded for every block. The minima is chosen in such a way that the value of $|\max_b - \min_b|$ should be minimum. All the maximas are copied from CSD_b to CAD_b so that CAD_b should consist of positive and negative maximas and other pixels as absolute values.

6: All the pixel values between $\max_b$ and $\min_b$ are incremented or decremented by 1 depending on the case that the minima $\min_b$ is present on the right or left of the corresponding maxima $\max_b$. Rest all the pixel values $CAD_b(i,j)$ remain unchanged. Both cases can be mathematically described in Equations (4) and (5) as follows:

$$CAD'_b(i,j) = \begin{cases} CAD_b(i,j)+1, & \text{if } CAD_b(i,j) > \max_b, \\ CAD_b(i,j), & \text{otherwise,} \end{cases} \tag{4}$$

$$CAD'_b(i,j) = \begin{cases} CAD_b(i,j)-1, & \text{if } CAD_b(i,j) < \max_b, \\ CAD_b(i,j), & \text{otherwise,} \end{cases} \tag{5}$$

for $0 \leq i \leq r-1$, $0 \leq j \leq c-2$, $1 \leq b \leq n$.

7: The secret data bits m of the secret data, *SI*, are embedded in the shifted difference image blocks CAD'_b by modifying their pixels having grayscale equal to the maxima $\max_b$, with the help of the following principle:

$$CAD''_b(i,j) = \begin{cases} CAD'_b(i,j)+m, & \text{if } CAD'_b(i,j) = \max_b, \\ |CAD'_b(i,j)|+m_1, & \text{if } CAD'_b(i,j) = -\max_b, \\ CAD'_b(i,j), & \text{otherwise,} \end{cases} \tag{6}$$

for $0 \leq i \leq r-1$, $0 \leq j \leq c-2$, $1 \leq b \leq n$, $m \in \{0,1\}$ and $m_1 \in \{0, 2^L\}$ where $L = \log_2(d)$, d is the difference between negative and positive maximas of the block.

8: The blocks of marked image MI are created with the help of blocks of the original cover image *CI* and the blocks of the updated difference image CAD'' with the help of the following transformations:

$$MI_b(i,0) = \begin{cases} CI_b(i,0), & \text{if } CI_b(i,0) < CI_b(i,1), \\ CI_b(i,1) + CAD''_b(i,0), & \text{otherwise.} \end{cases} \quad (7)$$

$$MI_b(i,1) = \begin{cases} CI_b(i,0) + CAD''_b(i,0), & \text{if } CI_b(i,0) > CI_b(i,1), \\ CI_b(i,1), & \text{otherwise,} \end{cases} \quad (8)$$

for $0 \le i \le r-1$, $0 \le j \le c-2$, $1 \le b \le n$.

9: For other remaining pixels, the following operation is performed

$$MI_b(i,j) = \begin{cases} MI_b(i,j-1) + CAD''_b(i,j-1), & \text{if } CI_b(i,j-1) \ge CI'_b(i,j), \\ MI_b(i,j-1) - CAD''_b(i,j-1), & \text{otherwise,} \end{cases} \quad (9)$$

for $0 \le i \le r-1$, $0 \le j \le c-2$, $1 \le b \le n$.

3.3 Preventing Possible Overflow or Underflow

In the marked image, the grayscale values of some pixels of the cover may exceed the upper bound ($2^{bd}-1$ for cover image having *bd* depth pixels) or the lower bound (0 for cover image having *bd* depth pixels). This is possible due to the shifting operations performed on pixel values that are close to $2^{bd}-1$ or 0. To overcome the overflow or underflow problem, the modulo operation proposed by Goljan et al. [4] and Honsinger et al. [5], is adopted. For the marked image, we define each pixel $S_b(i,j)$ as

$$S_b(i,j) = S_b(i,j) \bmod 256.$$

On the receiver side, whether the received pixel, for example, $S_b(i,j) = 255$, was derived from 255 or -1, must be distinguished. Considering the characteristics of an image, no tremendous variations exist for adjacent pixels. Therefore, in case of a significant difference between $S_b(i,j-1)$ and $S_b(i,j)$, $S_b(i,j)$ was conducted by a modulo operation.

Two evaluations are presented here to restore the original value of $S_b(i,j)$ after the modulo operation is performed. If $S_b(i,j-1)$ is larger than TH_1, $S_b(i,j)$ is restored as

$$S_b(i,1) = \begin{cases} S_b(i,1) + 256, & \text{if } |S_b(i,j-1) - S_b(i,j)| \ge TH_2, \\ S_b(i,1), & \text{otherwise,} \end{cases}$$

where TH_1 and TH_2 are the threshold values.

Algorithm 3 Extraction Algorithm

Input: Marked Image MI

Output: Original image CI and secret data image SI.

1: The received marked image is divided into n blocks of size $r \times c$.

2: The complexity values are calculated for all the blocks using scheme described in Section 2.3. These blocks are classified into complex and smooth on the basis of their comparison with the threshold value. An optimized threshold value is evaluated using *FA* described in Algorithm 1. The set of blocks having complexity value greater than the threshold are considered as Complex Blocks while the set of blocks having complexity value lesser than the threshold are considered as Smooth Blocks. Only the complex blocks are used for extraction as the data was embedded only in the complex blocks.

3: Difference image complex blocks $RCAD_b$ are calculated from the received marked image by using the formula:

$$RCAD_b(i,j) = |MI_b(i,j) - MI_b(i,j+1)| \tag{10}$$

for $0 \leq i \leq r-1$, $0 \leq j \leq c-2$, $1 \leq b \leq n$.

4: Values of maxima $\max_b$ and minima $\min_b$ are recieved as side information.

5: The difference image generated in Step 3 is traversed and the embedded secret data bits m are extracted by using the following rules.

$$m = \begin{cases} 0, & \text{if } RCAD_b(i,j) = \max_b, \\ 1, & \text{if } RCAD_b(i,j) = \max_b + 1, \\ de2bi(RCAD_b(i,j)), & \text{if } RCAD_b(i,j) \in [-\max_b, \max_{b-1}), \end{cases} \tag{11}$$

for $0 \leq i \leq r-1$, $0 \leq j \leq c-2$, $1 \leq b \leq n$ where *de2bi* is the conversion function of any decimal number to its equivalent binary number. $\max_b$ is the received maxima of block b and m is the array of bits extracted. The entire difference image blocks b are scanned and bit 0 is extracted, if the pixel value $\max_b$ is encountered and 1 is extracted if the pixel value $\max_b + 1$ is encountered. If pixels within the range $[-\max_b, \max_{b+1})$ are encountered, L number of bits are retrieved by following Equation (11), where

$$L = \lfloor \log_2(d) \rfloor \tag{12}$$

L is the number of bits embedded in every negative maxima and

$$d = \max_b - (-\max_b) + 2 \tag{13}$$

$\max_b$ is the positive peak value and $-\max_b$ is the negative peak value.

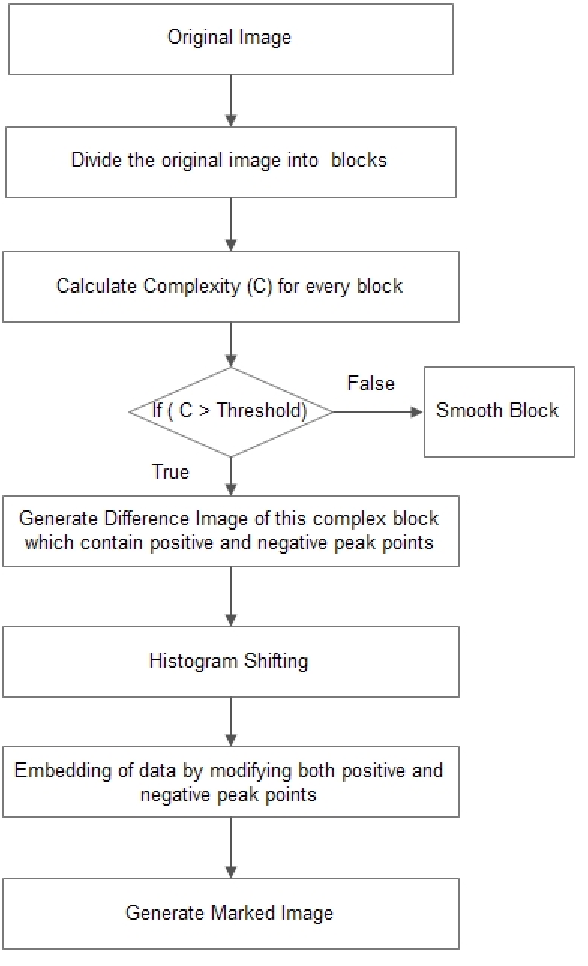


Figure 3. Flowchart of data embedding procedure

6: The difference image blocks $RCAD'_b$ are retrieved back by using the formula:

$$RCAD'_b = \begin{cases} RCAD_b(i,j) - 1, & \text{if } RCAD_b(i,j) = \max_b + 1, \\ RCAD_b(i,j), & \text{if } RCAD_b(i,j) = \max_b, \\ -\max_b, & \text{if } RCAD_b(i,j) \in [-\max_b, \max_{b-1}), \end{cases} \tag{14}$$

for $0 \leq i \leq r-1$, $0 \leq j \leq c-2$, $1 \leq b \leq n$ where $\max_b$ is the maxima of block b and $-\max_b$ is the negative maxima of block b.

7: The original image CI is recovered back by applying the inverse shifting operations.

62	56	63	60
61	59	58	59
60	61	59	55
58	58	56	57

(a) Complex Block

0	1	1	0	1	1	0

(b) Secret Data Bits

6	7	3
2	1	1
1	2	4
0	2	1

(c) Absolute Difference Image Block

6	-7	3
2	1	-1
-1	2	4
0	2	-1

(d) Simple Difference Image Block

6	7	3
2	1	-1
-1	2	4
0	2	-1

(e) Simple Difference Image Block copied to Absolute Difference Image Block

7	8	4
3	1	-1
-1	3	5
0	3	-1

(f) Difference Image Block after Shifting

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

(g) Data embedded in Difference Image Block

63	56	64	60
62	59	58	60
60	60	59	55
58	58	56	57

(h) Marked Image Block

Figure 4. Example for data embedding in proposed scheme

If $S_b(i, j-1)$ is smaller than TH_1, $S_b(i, j)$ is restored as

$$S_b(i,1) = \begin{cases} S_b(i,1) - 256, & \text{if } |S_b(i,j-1) - S_b(i,j)| \geq TH_2, \\ S_b(i,1), & \text{otherwise.} \end{cases}$$

This approach is an efficient approach to prevent underflow and overflow and is also used by Lin et al. [11].

4 EXPERIMENTAL RESULTS

Figure 5. a)–e) Original cover images; f)–j) marked images after embedding 5 000 bits; k)–o) marked images after embedding 10 000 bits; p)–t) marked images after embedding 25 000 bits

Image	Capacity → / Threshold ↓	5 000	10 000	20 000	50 000
Lena	0.0983	64.65	61.43	58.61	54.86
Boat	0.0635	65.68	62.65	59.83	56.10
Baboon	0.065	67.42	63.84	60.67	55.94
Barbara	0.0762	48.40	45.26	42.11	37.81
Peppers	0.0683	64.59	61.46	58.40	54.57

Table 1. *PSNR* (dB) values at different capacities (in bits) for different images at optimized threshold

Proposed scheme is implemented using *MATLAB*. The input images considered for this work are uncompressed grayscale images named Lena, Boat, Peppers, Baboon and Barbara. These images are generally used by the researchers to compare their results with existing schemes. Each of the image is 512×512 size and they are shown in Figure 5. Figures 5 a)–e) show the original images and their marked images at different embedding data, are shown in Figures 5 f)–t). Block sizes considered in

Image →	Lena		Barbara		Boat		Baboon		Peppers	
Scheme ↓	Capacity	*PSNR*	Capacity	*PSNR*	Capacity	*PSNR*	Capacity	*PSNR*	Capacity	*PSNR*
Tsai et al. [23]	38 310	49.11			25 370	48.97	12 739	48.85		
Kim et al. [10]	78 071	41.09			52 924	40.65			65 293	40.84
Hong et al. [6]	54 457	48.17			34 148	48.17			34 025	48.77
Ni et al. [14]	5 446	48.21			11 473	48.26			5 447	48.21
Ou et al. [16]	60 000	50.5	47 500	51.1	–	–	20 000	50.1		
Ma el al. [13]	50 000	50.5	40 000	51.5			17 000	50	38 000	50.5
Qu and Kim el al. [19]	49 000	50.5	38 000	50.5			15 000	50.5	34 000	51.5
Ou et al. [17]	54 000	49	43 000	49			18 000	49	43 000	49
Hong et al. [8]	46 839	49.19			29 824	49.02	14 154	48.86		
Proposed	102 431	51.85	154 237	33.01	91 687	53.23	88 807	53.28	97 767	51.60

Table 2. *PSNR* (dB) vs. Capacity (in bits) comparison of proposed scheme with the existing schemes

Block Size →	4×4		8×8		16×16	
Image ↓	Capacity	*PSNR*	Capacity	*PSNR*	Capacity	*PSNR*
Lena	102 431	51.85	77 206	47.97	65 496	43.30
Baboon	88 807	53.28	54 817	50.54	40 932	46.58
Peppers	97 767	51.60	72 332	47.89	61 611	43.35
Barbara	154 237	33.01	134 238	27.57	113 465	23.18
Boat	91 687	53.23	62 294	49.96	40 086	45.77

Table 3. *PSNR* (dB) vs. Capacity (in bits) comparison of proposed scheme for different block sizes

the proposed work are 4×4, 8×8 and 16×16. In this experiment, a binary image is taken as secret data. *PSNR* is taken as one of the quality parameters which is defined as

$$PSNR = 10 \times \log_{10} \frac{(2^{bd} - 1)^2}{MSE} \tag{15}$$

where bd is the bit depth of the image and MSE is defined as

$$MSE = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{\delta(i,j)^2}{M \times N}, \tag{16}$$

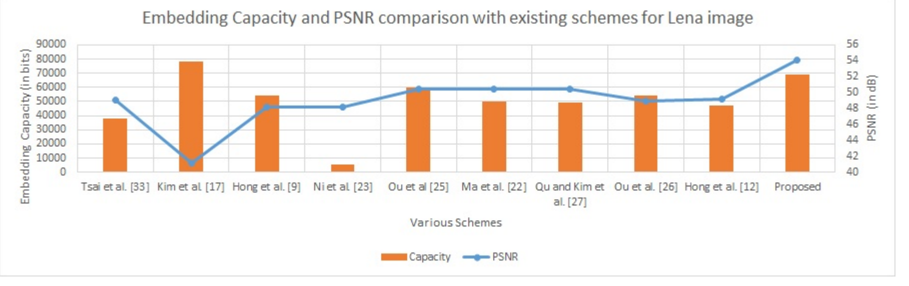
$$\delta(i,j) = MI(i,j) - CI(i,j) \tag{17}$$

Schemes	Sachnev et al. [21]	Ma et al. [13]	Qu et al. [19]	Ou et al. [16]	Proposed Scheme 4×4	Proposed Scheme 8×8	Proposed Scheme 16×16
Lena	20 785	144 390	10 370	10 290	4 758	3 027	1 440
Baboon	60 312	53 715	57 549	52 008	4 266	4 257	7 821
Barbara	21 068	16 177	11 512	10 223	2 093	798	420
Peppers	42 296	23 239	17 741	16 369	5 048	4 347	4 895

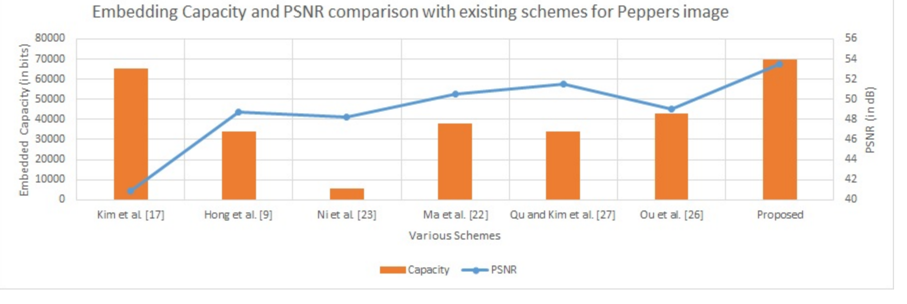
Table 4. Shifting (in pixels) comparison for different images for 10 000 embedding capacity

Schemes	Sachnev et al. [21]	Ma et al. [13]	Qu et al. [19]	Ou et al. [16]	Proposed Scheme 4 × 4	Proposed Scheme 8 × 8	Proposed Scheme 16 × 16
Lena	43 420	33 795	27 176	26 825	9 879	7 085	3 540
Barbara	43 420	34 371	29 644	28 420	4 027	1 636	909
Peppers	90 388	52 460	43 250	43 191	10 012	8 675	5 089

Table 5. Shifting (in pixels) comparison for different images for 20 000 embedding capacity



a)



b)

Figure 6. Embedding capacity (in bits) and *PSNR* (dB) comparison of proposed scheme with existing schemes for a) Lena image and b) Peppers image

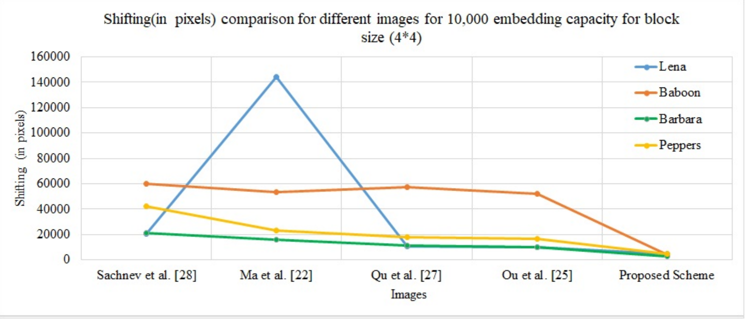
where $MI(i, j)$ is the pixel of marked image and $CI(i, j)$ is the pixel of cover image, M and N are the height and width of image, respectively.

Objective function used to optimize the threshold is

$$f_{obj} = \text{embedding\_capacity} \times PSNR \tag{18}$$

where embedding_capacity is the total amount of secret data embedded and *PSNR* is taken between original and marked image.

In Table 1, *PSNR* values have been given for the proposed scheme for different embedding capacities for different images at their respective optimized thresholds.



a)

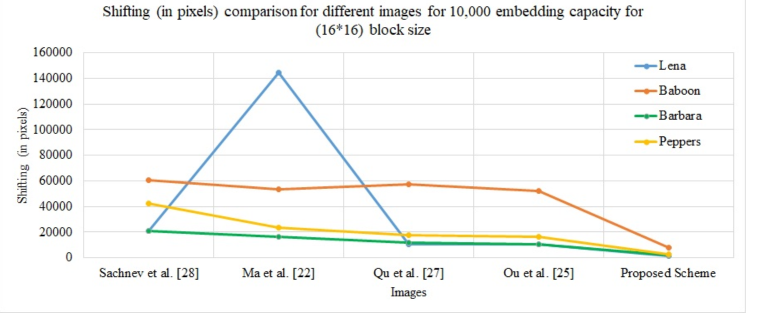
Minimum bits embedded are 5 000 and maximum bits embedded are 50 000 for all images, considered in this work. One can observe that *PSNR* values decrease with the increase in the embedded bits. This is due to the reason as number of embedded bits decreases the distortion in the marked image, which further decreases *PSNR* between cover and marked images.

In Table 2, the *PSNR* vs. the maximum embedding capacity of the proposed scheme and various existing schemes have been presented. It can be inferred that the embedding capacity and *PSNR* of the proposed scheme are better when compared to the existing schemes. The embedding capacity has been increased using negative pixels of the difference image for embedding. *PSNR* has been maintained by choosing the closest minima to the maxima of every difference block of the image and also by using just the complex blocks for the host image. In case of Barbara image, *PSNR* value is lesser than the existing schemes, but the embedding capacity has been increased remarkably as compared to the existing schemes. Table 3 shows *PSNR* and embedding capacity for the proposed scheme for different block sizes, i.e. 4×4, 8×8 and 16×16. In Tables 4 and 5, the comparison of the shifting of pixels of the proposed scheme with the existing schemes at 10 000 and 20 000 embedding capacities has been shown. It can be seen that the shifting of pixels in the proposed scheme has been reduced remarkably in comparison to the shifting of the existing schemes.

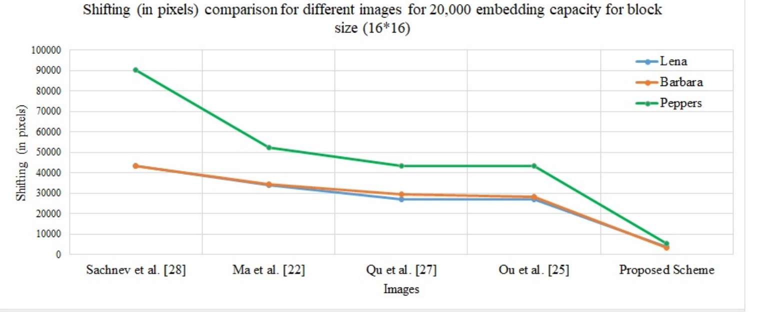
PSNR vs. embedding capacity comparison of the proposed scheme with the existing schemes has been shown in Figure 6 for Lena and Pepper images. It can be observed from this figure that the proposed scheme maintains the highest *PSNR* vs. embedding capacity ratio for both the images. Though Kim et al. [10] manage to have larger embedding capacity for Lena image, but its *PSNR* for the same is quite low as compared to the proposed scheme.

Comparison of the proposed scheme with the existing schemes in terms of shifting of pixels has been shown in Figure 7. It can be observed that the proposed scheme has minimum shifting of pixels, which helps in reducing the distortion, thereby enhancing the visual quality of the image.

b)



c)



d)

Figure 7. Shifting (in pixels) comparison of the proposed scheme with the existing schemes for different images for a) block size 4×4 and embedding capacity 10 000, b) block size 4×4 and embedding capacity 20 000, c) block size 16×16 and embedding capacity 10 000, and d) block size 16×16 and embedding capacity 20 000

5 CONCLUSION

In this paper, an adaptive reversible data hiding scheme for digital images has been proposed. The blocks of the image have been categorized into complex and smooth blocks on the basis of their complexity, and only complex blocks have been used for embedding, which helped in maintaining the *PSNR* value. Secret data bits are embedded in the difference blocks of the image, in which both the positive and negative difference maxima have been utilized for data embedding, as they are able to embed more than one bit, thereby increasing the embedding capacity. Histogram shifting has been minimized by choosing the closest pairs of maxima and minima which further minimizes the distortion in the marked image. The embedding capacity and the *PSNR* of the proposed scheme have proven to be better as compared to the existing schemes.

REFERENCES

[1] Celik, M. U.—Sharma, G.—Tekalp, A. M.—Saber, E.: Reversible Data Hiding. Proceedings IEEE International Conference on Image Processing, Rochester, NY, 2002, pp. 157–160.

[2] Chang, Y.-F.—Tai, W.-L.: Histogram Based Reversible Data Hiding Based on Pixel Differences with Prediction and Sorting. KSII Transactions on Internet and Information Systems, Vol. 6, 2012, No. 2, pp. 3100–3116, doi: 10.3837/tiis.2012.12.004.

[3] Fallahpour, M.—Sedaaghi, M. H.: High Capacity Lossless Data Hiding Based on Histogram Modification. IEICE Electronics Express, Vol. 4, 2007, No. 7, pp. 205–210, doi: 10.1587/elex.4.205.

[4] Goljan, M.—Fridrich, J. J.—Du, R.: Distortion-Free Data Embedding for Images. In: Moskowitz, I. S. (Ed.): Information Hiding (IH 2001). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2137, 2001, pp. 27–41, doi: 10.1007/3-540-45496-9_3.

[5] Honsinger, C. W.—Jones, P. W.—Rabbani, M.—Stoffel, J. C.: Lossless Recovery of an Original Image Containing Embedded Data. United States Patent No. US6278791B1, August 2001.

[6] Hong, W.—Chen, T.-S.—Shiu, C.-W.: Reversible Data Hiding for High Quality Images Using Modification of Prediction Errors. Journal of Systems and Software, Vol. 82, 2009, No. 11, pp. 1833–1842, doi: 10.1016/j.jss.2009.05.051.

[7] Hong, W.—Chen, T.-S.: A Local Variance-Controlled Reversible Data Hiding Method Using Prediction and Histogram-Shifting. Journal of Systems and Software, Vol. 83, 2010, No. 12, pp. 2653–2663, doi: 10.1016/j.jss.2010.08.047.

[8] Hong, W.—Chen, T.-S.—Wu, M.-C.: An Improved Human Visual System Based Reversible Data Hiding Method Using Histogram Modification. Optics Communications, Vol. 291, 2013, pp. 87–97, doi: 10.1016/j.optcom.2012.10.081.

[9] Kawaguchi, E.—Taniguchi, R.-I.: Complexity of Binary Pictures and Image Thresholding – An Application of DF-Expression to the Thresholding Problem. Pro-

ceedings of 8th International Conference on Pattern Recognition (ICPR), Vol. 2, 1986, pp. 1221–1225.

[10] Kim, K.-S.—Lee, M.-J.—Lee, H.-Y.—Lee, H.-K.: Reversible Data Hiding Exploiting Spatial Correlation Between Sub-Sampled Images. Pattern Recognition, Vol. 42, 2009, No. 11, pp. 3083–3096, doi: 10.1016/j.patcog.2009.04.004.

[11] Lin, C.-C.—Tai, W.-L.—Chang, C.-C.: Multilevel Reversible Data Hiding Based on Histogram Modification of Difference Images. Pattern Recognition, Vol. 41, 2008, No. 12, pp. 3582–3591, doi: 10.1016/j.patcog.2008.05.015.

[12] Luo, H.—Yu, F.-X.—Chen, H.—Huang, Z.-L.—Li, H.—Wang, P.-H.: Reversible Data Hiding Based on Block Median Preservation. Information Sciences, Vol. 181, 2011, No. 2, pp. 308–328, doi: 10.1016/j.ins.2010.09.022.

[13] Ma, X.—Pan, Z.—Hu, S.—Wang, L.: High-Fidelity Reversible Data Hiding Technique Based on Multi-Predictor Sorting and Selecting Mechanism. Journal of Visual Communication and Image Representation, Vol. 28, 2015, pp. 71–82, doi: 10.1016/j.jvcir.2015.01.012.

[14] Ni, Z.—Shi, Y. Q.—Ansari, N.—Su, W.: Reversible Data Hiding. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, 2006, No. 3, pp. 354–362, doi: 10.1109/TCSVT.2006.869964.

[15] Ni, Z.—Shi, Y. Q.—Ansari, N.—Su, W.—Sun, Q.—Lin, X.: Robust Lossless Image Data Hiding Designed for Semi-Fragile Image Authentication. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, 2008, No. 4, pp. 497–509, doi: 10.1109/TCSVT.2008.918761.

[16] Ou, B.—Li, X.—Wang, J.: Improved PVO-Based Reversible Data Hiding: A New Implementation Based on Multiple Histograms Modification. Journal of Visual Communication and Image Representation, Vol. 38, 2016, pp. 328–339, doi: 10.1016/j.jvcir.2016.03.011.

[17] Ou, B.—Li, X.—Wang, J.—Peng, F.: High-Fidelity Reversible Data Hiding Based on Geodesic Path and Pairwise Prediction-Error Expansion. Neurocomputing, Vol. 226, 2017, pp. 23–34, doi: 10.1016/j.neucom.2016.11.017.

[18] Pan, Z. B.—Hu, S.—MA, X. X.—Wang, L. F.: Reversible Data Hiding Based on Local Histogram Shifting with Multilayer Embedding. Journal of Visual Communication and Image Representation, Vol. 31, 2015, pp. 64–74, doi: 10.1016/j.jvcir.2015.05.005.

[19] Qu, X.—Kim, H. J.: Pixel-Based Pixel Value Ordering Predictor for High-Fidelity Reversible Data Hiding. Signal Processing, Vol. 111, 2015, pp. 249–260, doi: 10.1016/j.sigpro.2015.01.002.

[20] Richard, E.: A Tutorial on BPCS Steganography and Its Applications. Proceedings of Pacific Rim Workshop on Digital Steganography, Kitakyushu, Japan, 2003, pp. 18–31 (invited paper).

[21] Sachnev, V.—Kim, H. J.—Nam, J.—Suresh, S.—Shi, Y. Q.: Reversible Watermarking Algorithm Using Sorting and Prediction. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19, 2009, No. 7, pp. 989–999, doi: 10.1109/TCSVT.2009.2020257.

[22] Sencar, H. T.—Ramkumar, M.—Akansu, A. N.: Data Hiding Fundamentals

and Applications: Content Security in Digital Multimedia. Elsevier Academic Press, 2004.

[23] Tsai, P.—Hu, Y.-C.—Yeh, H.-L.: Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting. Signal Processing, Vol. 89, 2009, No. 6, pp. 1129–1143, doi: 10.1016/j.sigpro.2008.12.017.

[24] Wang, Z.-H.—Lee, C.-F.—Chang, C.-Y.: Histogram Shifting Imitated Reversible Data Hiding. Journal of Systems and Software, Vol. 86, 2013, No. 2, pp. 315–323, doi: 10.1016/j.jss.2012.08.029.

[25] Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press, 2008. ISBN: 978-1-905986-10-6.

Sonal Kukreja graduated and received her M.Tech. degree in 2015 in the field of computer science and applications from the Thapar Institute of Engineering and Technology, Patiala. In 2016, she enrolled as doctoral student at the Thapar Institute of Engineering and Technology, Patiala, Punjab, India. Her research interests are focused on data hiding, visual cryptography and watermarking.

Geeta Kasana is working as Assistant Professor in Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, India. She received her Ph.D. degree in information security from the Thapar University. Her research interests include image processing and information security. She has published many research papers in reputed international journals and conferences. She is currently guiding Ph.D. students on information security.

Singara Singh Kasana is working as Associate Professor in Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, India. He has fifteen years of teaching and research experience. He received his Ph.D. degree in image compression from the Thapar University. His research interests include image processing, wireless networks, and information security. He has published many research papers in reputed international journals and conferences. He is currently guiding Ph.D. students on information security, image and video watermarking, cryptography and remote sensing.