

Reversible Data Hiding Scheme Based on Maximum Histogram Gap of Image Blocks

Mohammad Arabzadeh and Mohammad Reza Rahimi*

Department of Electrical and Electronic Engineering, Shiraz University of Technology
Modarres Bolvd. P.O. Box 71555-313, Shiraz, Iran
[e-mail: {m.arabzadeh, mr.rahimi }@sutech.ac.ir]
*Corresponding author: Mohammad Reza Rahimi

Received January 13, 2012; revised March 14, 2012; revised June 29, 2012; revised July 28, 2012; accepted August 16, 2012; published August 30, 2012

Abstract

In this paper a reversible data hiding scheme based on histogram shifting of host image blocks is presented. This method attempts to use full available capacity for data embedding by dividing the image into non-overlapping blocks. Applying histogram shifting to each block requires that extra information to be saved as overhead data for each block. This extra information (overhead or bookkeeping information) is used in order to extract payload and recover the block to its original state. A method to eliminate the need for this extra information is also introduced. This method uses maximum gap that exists between histogram bins for finding the value of pixels that was used for embedding in sender side. Experimental results show that the proposed method provides higher embedding capacity than the original reversible data hiding based on histogram shifting method and its improved versions in the current literature while it maintains the quality of marked image at an acceptable level.

Keywords: Reversible Data Hiding, lossless data hiding, histogram shifting

1. Introduction

Increasing popularity of internet applications has led people into digital world. Nowadays information is mainly stored and transmitted in digital form; hence digital media can be easily distributed, tampered and replicated. Therefore, protection of digital data from illegal use and alteration has become a significant issue.

Data hiding is indicated so far as a possible solution that can authenticate the completeness of digital media and also protect intellectual property rights. This is worthy notable that applications of data hiding schemes have not been limited to authentication and also have been used in fingerprinting, information assurance and etc.

Among digital signals that exist, digital images have attracted a large attention of researchers for data hiding. This is because of huge amount of this kind of signals. Besides the methods and algorithms that uses digital images as cover media, can be easily applied to another signals such as audio signals, 3D-meshes and video.

Generally in all of data hiding schemes, by modifying of the cover media's content the to-be-embedded message (or payload) is embedded. The relationship between the to-be-embedded message and the cover media, could divide the data hiding methods into two main groups. In the first group that used in steganographic method cover image has no value to sender and receiver and play the role of a decoy to hide the presence of communication. In this application for covert communication the receiver has no interest in originality of cover image before and after extraction of the embedded message. In the second groups that frequently addressed as digital image watermarking, the embedded message has a close relationship to cover image. In these applications embedded message supplies additional information about the cover image, such as, image caption, ancillary data about the image origin, author signature and so on. In the methods that categorized in this group, meeting a series of requirements such as minimal robustness, sufficient capacity and imperceptibility are desirable.

However, alteration of cover image introduced during embedding process is not acceptable in some applications. Artworks, law enforcement image analysis, military and medical images are good examples where even small modification is not allowed for obvious legal reasons and a potential risk of misinterpreting of image. Therefore, reversible data hiding methods which are also called lossless data hiding or invertible watermarking are explored.

Reversible watermarking methods embed a message (payload) into digital image in a reversible fashion. This means that these methods can recover the host image to its original state after extraction of embedded message. The motivation of reversible watermarking is distortion free data embedding [1]. As a basic requirement in all watermarking methods, the quality degradation of image after embedding in reversible watermarking should be low.

In the past decade many reversible watermarking schemes have been proposed [5][7][9][10][17][19][20][21][23]. In [2] Caldelli et al. divided reversible watermarking algorithms into fragile and semi fragile. In [3] Feng et al. classified the reversible watermarking schemes into three classes: 1) the schemes by applying data compression 2) the schemes by using difference expansion 3) the schemes by using histogram bin shifting. In [4] Shi et al. classified the reversible watermarking algorithms into three classes according to their applications: 1) those for fragile authentication 2) those for high embedding capacity 3) those for semi fragile authentication. Barton [6] patent was the first attempt to losslessly embed a message into a digital image. Honsinger et al. [7] after extraction of embedded message subtracted it from the watermarked image to losslessly recover the original image. Macq [8] used an extension of patchwork algorithm for lossless data embedding. Fridrich [1]

et al. Divide cover image into groups of pixels and then categorized these groups into three types: regular groups, singular groups and useless groups. They embed one bit into every group using type of it and if type of group don't match with to-be-embedded bit, the type of group triggered using a flipping function.

Celic et al. [9] proposed a Generalized-LSB (G-LSB) scheme with low distortion and high capacity that uses compression of quantization residues. De Vleeschouwer et al. [10] proposed a reversible data hiding by circular interpretation of bijective transformation. This scheme suffers from salt and pepper noise and an alternative approach was proposed in [11] to overcome this problem.

Among the reversible watermarking methods that developed so far, the highest capacity provided by ones that uses invertible transformations [12]. The method that proposed by Tian [13] is the main base of these methods. Theoretically, the upper bound of capacity for Tian's algorithm is 0.5 bpp. Alattar [14] extend the Tian's scheme and used vector instead of pixel pairs to increase embedding capacity.

Recently different approaches for reversible data hiding have been proposed by Coltuc [12][15][16]. In [16] Coltuc uses reversible contrast mapping (RCM) of pixel pairs for lossless embedding. For some pixel pairs RCM is invertible even if LSB's of transformed pixels are lost. For such pairs one can embed two bits and without ambiguity restore original value of pixels. Coltuc uses this property and divide pixel pairs into two groups. For the first group that RCM is invertible, one can embed one bit for indicating of initial status of pair and one bit of message. For the second group, there is no space for message bit. In this case one can only reset one of LSBs to indicate original status of pair and save original LSB as overhead information. In [12] Coltuc extend his method for a vector of pixels instead of pixel pairs and improve embedding capacity.

Ni et al. [17] developed a remarkable histogram based scheme for reversible data hiding. There are several studies that modify Tian or Ni et al.'s schemes to enhance embedding capacity or to reduce degradation of watermarked image in the current literature. Of special interest in this paper is the work done by Ni et al. [17]. Experimental results demonstrate that overall performance of Ni et al.'s scheme is the best among all reversible data hiding methods [2]. The lower bound of PSNR in this method is 48.13 dB for a single pass of algorithm. But the embedding capacity is limited. Increasing the embedding capacity, maintaining reversible characteristics simultaneously and decreasing the degradation of host image are the main challenges for reversible image watermarking.

In this paper, we improve Ni et al.'s scheme [17] and propose a block based reversible data hiding scheme that increases embedding capacity while preserve PSNR of Ni et al.'s scheme. In our understanding the proposed method is the first to do this. The basic idea in this method is based on using all capacity of an image for embedding payload by utilizing blocks instead of entire image. Besides our proposed scheme by utilizing a new strategy to determine the amount of shift for any block, produces smaller amount of overhead information and saves embedding capacity. In other words, in the proposed scheme there is no need to store the value of maximum/minimum point As a result, this paper shows that our algorithm can produce better results than ordinary histogram shifting method and is comparable with existing reversible watermarking methods.

The rest of this paper is organized as follows. Section 2 briefly introduces histogram shifting for reversible data hiding. In Section 3, we present the proposed scheme in details. Experimental results are presented in Section 4 and finally Section 5 concludes the paper.

2. A Brief Look at Histogram Shifting

In Ni et al.'s scheme [17], first a pair of a peak and a zero point in the image histogram is found. Peak and zero points in this scheme referred to the values that have respectively the highest and lowest occurring frequency in image histogram. Next, all of the points that reside between these two points are shifted by 1 unit. This makes a zero point just next to the peak point in the image histogram. Then according to the value of to-be-embedded bits, pixels that take the value of peak are modified to insert data. Embedding is carried out by modifying the peak value by 1 if to-be-embedded bit is “1” or leaving it unchanged if to-be-embedded bit is “0”. We can see embedding procedure of this method schematically in Fig.1.

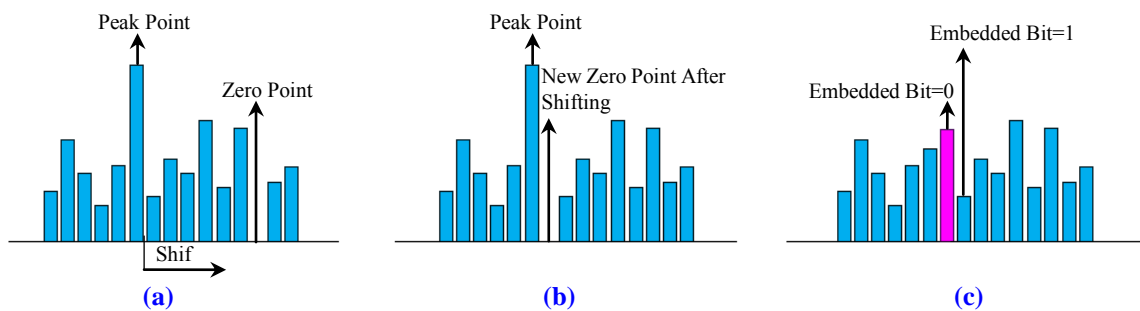


Fig. 1. The Embedding Procedure of Ni et al.'s Method (a) Histogram of Host Image Before Shift (b) Histogram of Host Image After Shift (c) Histogram of Host Image After Embedding.

It is obvious that both peak and zero point values must be known in the receiver. If we choose to apply Ni et al.'s scheme to blocks of the image, then for each block, a pair of peak and zero points is required to be known in the receiver side. Since this information is usually inserted in the host image, therefore much embedding space is used up by this information. The following subsections explain the details of the proposed method for blocking of the host image, watermark embedding and extraction procedures, and the process of restoring the image into its original form.

3. The Proposed Scheme

In this section a detailed explanation of the proposed scheme is presented. We first divide image blocks to 6 categories and then according to this categorization, insertion of payload bits is performed. Insertion scheme is similar to Ni et al.'s method with the difference being the value of shift. This value is chosen in a way that the need for extra information in the receiver is eliminated and payload bits can be extracted in receiving side while original host image is returned into its original form. The main idea of this method is to use the embedding capacity of those pixels that might not be among maximum values in global histogram of image but have the highest frequency in the histogram of a block. After that we look into the embedding procedure where payload bits are inserted into the host image and then we proceed into the extraction procedure where payload is extracted and the host image is fully recovered to its original form, proving the scheme to be reversible.

3.1 Categorization of Image Blocks

After dividing the image into non-overlapping blocks of size $M \times N$ pixels, the first step is to determine the category of each block. For this purpose, we define the following variables for the histogram of an image block that can be seen in Fig.2:

m : Value of pixel of peak point in the histogram of a block.

($m = \arg \{ \max_x (h(x)) \}$, $h(x)$ = frequency of gray level x in the histogram).

l : The pixel value related to the first non-zero bin in the right side of m .

($l = \{ \min(x) \mid x > m, h(x) \neq 0 \}$)

d : The free gap between the bin that corresponds to peak point and the first non-zero bin to its right side ($d = l - m - 1$).

Our purpose from “free gap” is distance between two non-zero bars in histogram that there are no pixel with middle values in the mentioned histogram. In other words if we have at least one pixel with gray level ‘ a ’ and at least one pixel with gray level ‘ b ’ in the histogram and $a < b$, free gap between them is $b - a - 1$ if don’t exist any pixel with gray level x in the histogram that $a < x < b$.

d_{\max} : Maximum free gap that exists between bins in a histogram of a block

($d_{\max} = \max(d_i), i = 1, 2, \dots, N_d, N_d = \text{Number of free gaps in a histogram}$).

n : The number of free gaps of block's histogram that is equal to d_{\max} .

$d_{s-\max}$: The second largest free gap between bins of block's histogram ($d_{s-\max} < d_{\max}$).

l_{bar} : The value of non-zero bar in the left side of d_{\max} .

r_{bar} : The value of non-zero bar in the right side of d_{\max} . ($r_{bar} = l_{bar} + d_{\max} + 1$).

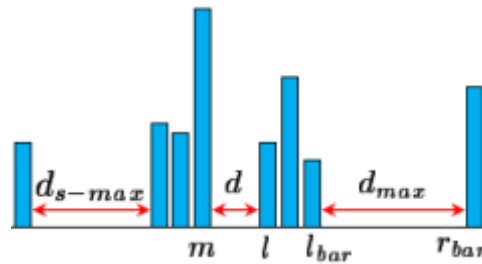


Fig. 2. The Visual Definition of the Defined Variables

Note that in this paper if more than one d_{\max} in the histogram of host image exists, we take the rightmost of them as d_{\max} . The blocks are then categorized into 6 category based on the following conditions in Table 1 and after this the value of *Shift* and *LM* (Location Map) for each block are determined. These blocks that satisfy at least one condition of Table 1 are considered as embeddable blocks. The *LM* is used in the extractor side to categorize the blocks uniquely. In other word *LM* is a bit stream for remembering the initial status of blocks and $LM = \{b_j \mid j = 1, 2, \dots, N_B\}, b_j \in \{0, 1\}$ here N_B is the number of host image blocks. The value of *LM* is 0 for blocks that belong to categories 1,2,3,5 and blocks that don't satisfy any

condition in **Table 1**. A series of visual examples for each category is shown in Appendix **Table A-1**.

Table 1. Essential conditions for categorizing blocks in embedder side

Category	Condition in Embedder Side	LM	$Shift$
1	All pixels in the block are equal	0	0
2	$d_{\max} = 0$, m =Maximum value in the block	0	0
3	$d = d_{\max} = 0$	0	2
4	$d < d_{\max} < T, d_{\max} > 0$	1	$d_{\max} + 2$
5	$0 < d = d_{\max} < T, n \geq 1$	0	2
6	$d_{\max} < T$, m =Maximum value in the block	1	0

In **Table 1**, T is a threshold to control the capacity and quality of the marked image and the variable $Shift$ is the amount of shift of the pixels that are greater than m . The blocks that do not belong to any of the above categories are non-embeddable blocks and used for embedding of LM bits.

Table 2 present the conditions that used for determination of blocks category in the extractor side. In this Table d'_{\max} and $d'_{s-\max}$ indicates the value of d_{\max} and $d_{s-\max}$ for the modified histogram of a block after embedding. The value of LM bits are required in the extractor for extraction of hidden data and recovering the marked image into its original form. The LM are embedded in the blocks that have $d_{\max} > 2T$ with simple LSB substitution. The original LSB 's of these blocks are also embedded alongside the actual payload bits.

Table 2. Essential conditions for categorizing blocks in extractor side

Category	Condition in Extractor Side and LM value
1	We have only two non-zero bin and $d'_{\max} = 0, LM = 0$
2	$d'_{\max} = 0, LM = 0$
3	$d'_{\max} = 1, LM = 0$
4	$1 < d'_{\max} < 2T, LM = 1$
5	$d'_{\max} \leq T, d'_{\max} \geq d'_{s-\max} + 1, LM = 0$
6	$d'_{\max} = 2T, LM = 1$

3.2 Embedding Procedure

After each block's category is determined and LM is constructed, it is time for actual data embedding. First, LM bits that are losslessly compressed using either an arithmetic coding (or JBIG2) are inserted into LSB of the blocks with $d_{\max} > 2T$ and original LSB 's are gathered into a sequence and appended to payload and therefore are considered as a part of payload. These blocks that contains LM bits and during the next steps are considered as non-embeddable blocks. Now depending on the block category, a shift is applied to the blocks according to the **Table 1**; that is, pixels with values higher than m are increased by value of the shift depending on their category. By doing so the histogram is displaced to right and we

can start the embedding of payload data by scanning pixels of block in raster scan or any space filling curve. The embedding method is to keep the pixels with the value of peak point (m) unchanged if the corresponding payload bit is "0" and to increase it by 1 if the payload bit is "1". **Fig. 3** presents the embedding procedure. Embedding procedure could be explained with the running steps below:

- Step 1: Divide host image to $M \times N$ non-overlapping blocks.
- Step 2: Generate LM vector using characteristics of blocks and pre-defined value of T and compress LM using arithmetic coding to obtain LM_C .
- Step 3: Capture $\eta(LM_C)$ of LSB 's of blocks with $d_{\max} > 2T$ and replace them with LM_C bits using simple LSB substitution ($\eta(\cdot)$ denotes the number of bits). The first 16-bit that replaced with LSB 's determine the value of $\eta(LM_C)$.
- Step 4: Concatenate captured LSB 's and actual payload and construct final payload.
- Step 5: Scan each block and if this block satisfactory one of the conditions listed in **Table 1** regard it as embeddable blocks and embeds $\eta(C_B)$ bits using proposed scheme (C_B denotes the embedding capacity of each block).
- Step 6: Finally increase the value of d_{\max} for blocks that are non-embeddable. This is done simply by increasing the pixel values greater than r_{bar} by one.

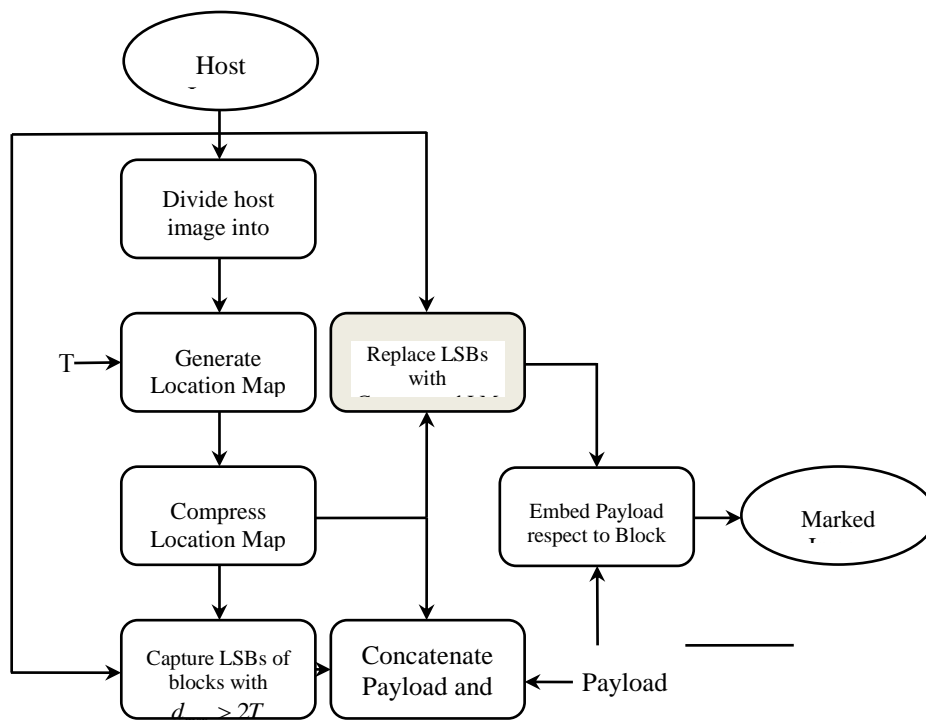


Fig. 3. The Embedding Procedure

During implementation of the proposed method we discovered that if a lot of 0's (or 1's) occurs after each other in the payload none of the pixels that take the value of maximum is changed. Hence the category to which a block belongs cannot be uniquely determined in the receiver and an ambiguity arises. We resolved this problem by changing the first pixel with value of m into $m + 1$ and keep the last one unchanged regardless of payload bits for all categories except category 6. This modification permits the receiver to uniquely determine block's category under all conditions but decreases embedding capacity by 2 bits for each block. Fig. 4 depicts this trick. For category 6, we additionally change the second pixel with value of m into $m + 2T + 2$ and therefore 3 bits are spent for preventing ambiguity in the receiver. It is worthy notable that for blocks that neither are a member of any category nor contains LM information, the pixels that have a value greater than or equal to r_{bar} must be increased one unit.

Another important note is that during all the aforementioned operations if there is more than one d_{max} in a block, the rightmost one should be selected in order to minimize the distortion by shifting a smaller portion of image histogram. In this way the quality of the host image is considerably retained during embedding process.

It is possible that after the above operations some pixels exceed the allowed range for gray scale images [0,255]. In this case the coordinate of these pixels must be appropriately sent to the receiver. But it is notable that this situation is rare in natural.

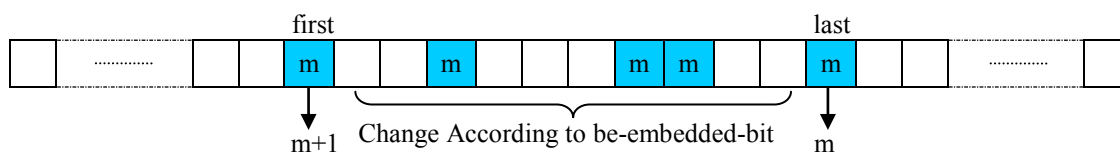


Fig. 4. The Trick Used for Preventing Ambiguity in the Extractor Side

3.3 Extraction Method and Recovery Procedure

In this method, the value m corresponds to the point with the highest value of frequency in histogram of each block. Obviously, m is between 0 and 255 for a grayscale image with a bit depth of 8. After embedding, peak point of histogram is no longer the original peak point because of the fact that a number of pixels with the value of peak point (i.e. m) are now changed to $m + 1$. The question that arises here is that how can we find the actual value of m from an already marked block whose peak point is probably changed during embedding.

To answer this question, we must say that this method works in such a way that the biggest gap between bins in histogram (d_{max}) is always positioned right after $m + 1$. At receiver, the biggest gap in histogram is found and then the first bin to the left of this gap (l'_{bar}) is actually $m + 1$. This is how the value of m is found for each block. In other words the value of m corresponds to peak point in the histogram of block *before embedding* and does not necessarily correspond to the peak point *after embedding*. Hence, care must be taken when dealing with block histogram before embedding and after embedding. Therefore, m and $m + 1$ is defined in original block's histogram and l'_{bar} and $l'_{bar} - 1$ is defined in marked image block's histogram.

In receiver side, LM_C bits are first extracted from the blocks with $d_{max} > 2T$ and are decompressed. Then according to the characteristics of the blocks and LM , the category of

each block is determined and finally after extraction of the payload bits, the histogram of the block is shifted back to have the original block. Extraction and recovery operations can be expressed in the 4 steps:

- Step 1: Extract LM_C bits from the LSB 's of blocks with $d_{\max} > 2T + 1$ and decompress it. For these blocks first the pixels with value of r'_{bar} should be replaced with $r'_{bar} - 1$.
- Step 2: Extraction the payload according to LM bits and characteristics of each block using [Table 2](#).
- Step 3: Extract payload bits "0" and "1" from the pixels with value l'_{bar} and $l'_{bar} - 1$ respectively.
- Step 4: Shift back the pixels that are greater than l'_{bar} according to [Table 1](#).
- Step 5: Replace the original LSB 's of the blocks that contain LM_C instead of LM_C bits.
- Step 6: Finally decrease the value of d'_{\max} by one for non-embeddable blocks that in Step 1 don't have been changed.

It is obvious that according to the method used in the embedding phase, the value of l'_{bar} in the marked block is certainly the same as m for unmarked block. Note that first and last bits that are being extracted from each block in categories 1-5 and the first, second, and the last bits that are being extracted from category 6 do not carry any information and are discarded. The extraction and recovery procedure is depicted in [Fig. 5](#).

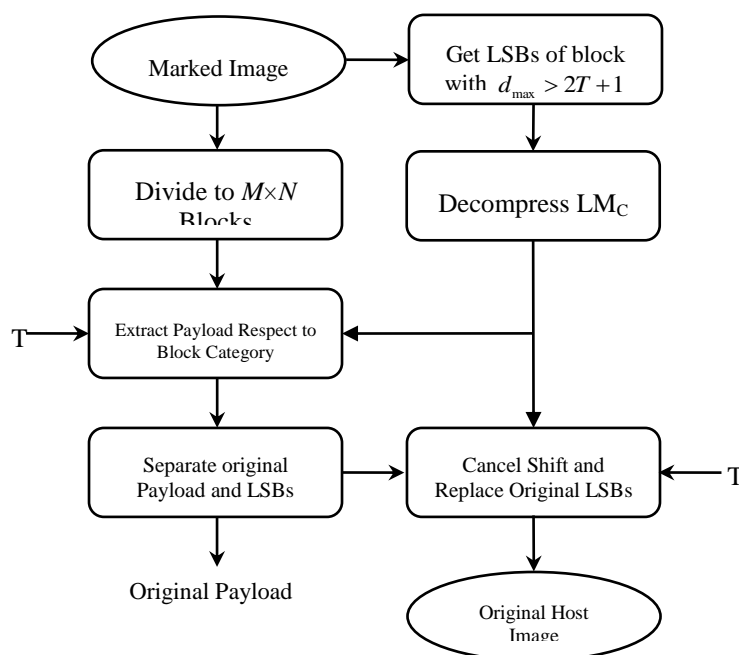


Fig. 5. The Extraction and Recovery Procedure

4. Simulation Results

In this section, we will show the feasibility and the performance of the proposed method in terms of pure payload and image quality over the relevant techniques proposed by Ni et al. and other researchers. The proposed scheme was successfully applied to a number of standard test images and equality between the recovered and original images proved the reversibility of the scheme. In this experiment 512×512 pixels, grayscale test images were used and payload was chosen to be a binary random sequence with the same size of the corresponding image's embedding capacity. **Fig. 6** presents the test images that are used in our simulations. As it is common in the literature, we adopted PSNR as imperceptibility measure to compare our scheme with other schemes.



Fig. 6. Test Images, First Row Left to Right: Lena, F16, Candy, Aerial; Second Row: Elaine, Splash, Pepper, Sailboat; Third Row: Man, Tiffany, House, Couple

In **Fig. 7** and **Fig. 8** the embedding capacity in *bpp* and PSNR (peak-signal-to-noise-ratio) in *dB* using this method for various *T* values are shown for four standard test images (F16, Couple, Lena, Tiffany).

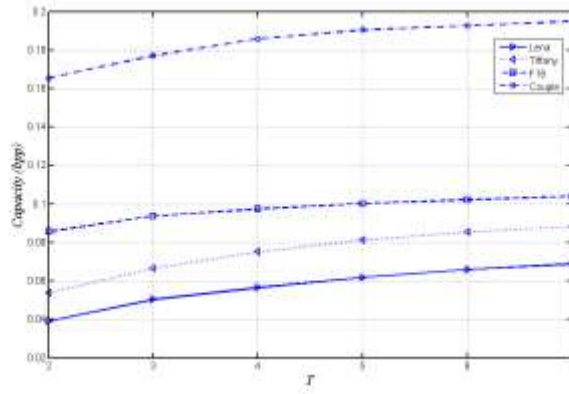


Fig. 7. Embedding capacity vs. T for four test images

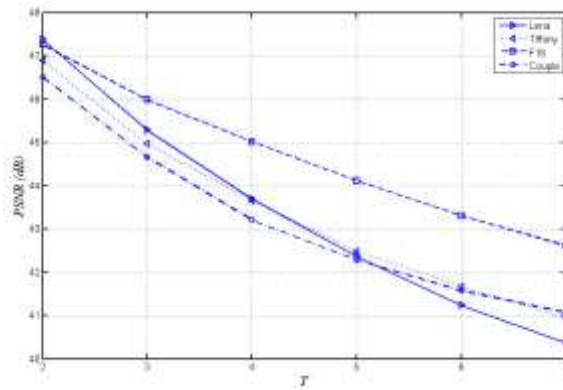


Fig. 8. PSNR vs. T for four test images

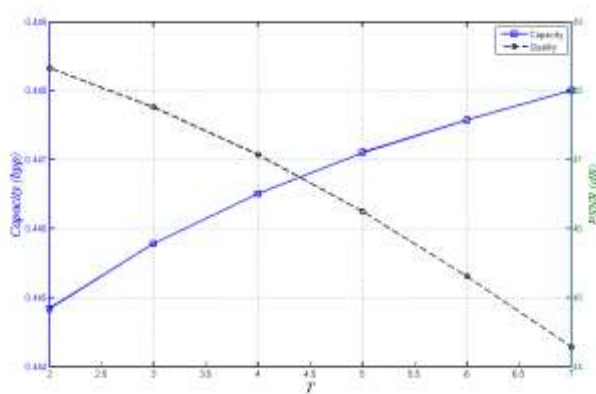


Fig. 9. Relationship between visual imperceptibility and capacity vs. T for Candy image

Fig. 9 demonstrates the relationship between visual imperceptibility and capacity vs. T for Candy. The pure payload, PSNR and overhead information size for test images using the proposed scheme is presented in Table 3. In this table the size of block is 8x8 (M=N=8). Note that in all tables the listed payload is pure payload (i.e., the amount of overhead information has been excluded). Table 4 presents the effect of size of blocks. In the Table 4 we take T=7.

Table 3. Pure payload, PSNR and overhead information size using proposed scheme for $M=N=8$

		T					
		7	2	3	4	5	6
Lena	Capacity (bpp)	0.039	0.050	0.057	0.062	0.066	0.068
	PSNR (dB)	47.34	45.30	43.68	42.39	41.25	40.40
	$\eta(LM_C)$ (bit)	3,013	3,768	4,052	4,096	4,067	3,990
F16	Capacity (bpp)	0.086	0.094	0.097	0.100	0.102	0.104
	PSNR (dB)	47.23	46.00	45.04	44.18	43.35	42.67
	$\eta(LM_C)$ (bit)	2,663	3,317	3,596	3,775	3,903	3,977
Candy	Capacity (bpp)	0.444	0.445	0.446	0.447	0.448	0.450
	PSNR (dB)	48.27	47.88	47.08	46.30	45.24	44.40
	$\eta(LM_C)$ (bit)	665	1,020	1,399	1,707	1,998	2,215
Aerial	Capacity (bpp)	0.011	0.013	0.015	0.017	0.019	0.021
	PSNR (dB)	48.99	47.79	46.40	44.93	43.36	42.14
	$\eta(LM_C)$ (bit)	1,157	1,887	2,459	2,887	3,300	3,568
Elaine	Capacity (bpp)	0.001	0.017	0.025	0.034	0.040	0.045
	PSNR (dB)	48.71	45.59	43.24	42.43	41.20	40.02
	$\eta(LM_C)$ (bit)	2,134	3,508	4,037	4,091	3,877	3,578
Splash	Capacity (bpp)	0.056	0.072	0.082	0.089	0.093	0.098
	PSNR (dB)	46.79	44.39	42.87	41.92	41.12	40.46
	$\eta(LM_C)$ (bit)	3,238	3,970	4,096	4,091	4,035	3,966
Pepper	Capacity (bpp)	0.022	0.032	0.040	0.046	0.050	0.053
	PSNR (dB)	47.55	45.19	43.37	42.05	41.10	40.23
	$\eta(LM_C)$ (bit)	2,874	3,782	4,074	4,096	4,040	3,939
Boat	Capacity (bpp)	0.012	0.020	0.025	0.030	0.035	0.038
	PSNR (dB)	48.36	46.03	44.05	42.38	41.19	40.15
	$\eta(LM_C)$ (bit)	2,177	3,249	3,806	4,055	4,096	4,068
Man	Capacity (bpp)	0.025	0.032	0.036	0.040	0.046	0.050
	PSNR (dB)	48.65	46.20	44.33	42.76	41.23	40.06
	$\eta(LM_C)$ (bit)	1,998	3,099	3,643	3,945	4,085	4,096
Couple	Capacity (bpp)	0.166	0.178	0.185	0.190	0.192	0.194
	PSNR (dB)	46.50	44.68	43.27	42.34	41.64	41.13
	$\eta(LM_C)$ (bit)	2,962	3,690	3,971	4,065	4,096	4,096
House	Capacity (bpp)	0.145	0.150	0.153	0.155	0.157	0.158
	PSNR (dB)	46.30	45.2	44.36	43.61	42.81	42.22
	$\eta(LM_C)$ (bit)	2,886	3,382	3,628	3,775	3,879	3,941
Tiffany	Capacity (bpp)	0.054	0.068	0.075	0.080	0.085	0.088
	PSNR (dB)	46.89	44.98	43.67	42.52	41.72	41.00
	$\eta(LM_C)$ (bit)	3,087	3,866	4,081	4,096	4,056	3,980

Table 4. The effect of size of blocks in proposed scheme for $T=7$

M=N		8×8	16×16	32×32	64×64
Lena	Capacity (bpp)	0.068	0.060	0.047	0.034
	PSNR (dB)	40.40	39.25	38.12	39.03
	$\eta(LM_C)$ (bit)	3,990	966	209	59
F16	Capacity (bpp)	0.104	0.092	0.074	0.052
	PSNR (dB)	42.67	40.78	40.80	40.71
	$\eta(LM_C)$ (bit)	3,977	1,024	249	59
Candy	Capacity (bpp)	0.450	0.378	0.285	0.173
	PSNR (dB)	44.40	43.05	43.72	46.32
	$\eta(LM_C)$ (bit)	2,215	722	216	50
Aerial	Capacity (bpp)	0.021	0.019	0.017	0.027
	PSNR (dB)	42.14	41.92	40.11	38.29
	$\eta(LM_C)$ (bit)	3,568	907	256	58
Elaine	Capacity (bpp)	0.045	0.042	0.030	0.020
	PSNR (dB)	40.02	38.64	38.12	38.47
	$\eta(LM_C)$ (bit)	3,578	880	210	56
Splash	Capacity (bpp)	0.098	0.074	0.062	0.035
	PSNR (dB)	40.46	40.06	39.50	41.61
	$\eta(LM_C)$ (bit)	3,966	1,016	256	64
Pepper	Capacity (bpp)	0.053	0.043	0.032	0.032
	PSNR (dB)	40.23	39.33	38.73	38.37
	$\eta(LM_C)$ (bit)	3,939	1,017	244	57
Boat	Capacity (bpp)	0.038	0.040	0.032	0.020
	PSNR (dB)	40.15	38.89	38.58	38.49
	$\eta(LM_C)$ (bit)	4,068	1,015	254	64
Man	Capacity (bpp)	0.050	0.043	0.034	0.032
	PSNR (dB)	40.06	38.98	38.08	38.05
	$\eta(LM_C)$ (bit)	4,096	1,020	246	57
Couple	Capacity (bpp)	0.194	0.167	0.135	0.110
	PSNR (dB)	41.13	40.49	41.20	39.17
	$\eta(LM_C)$ (bit)	4,096	1,024	256	64
House	Capacity (bpp)	0.158	0.140	0.114	0.082
	PSNR (dB)	42.22	41.17	41.15	40.29
	$\eta(LM_C)$ (bit)	3,941	1,017	256	64
Tiffany	Capacity (bpp)	0.088	0.078	0.057	0.038
	PSNR (dB)	41.00	40.52	40.92	41.27
	$\eta(LM_C)$ (bit)	3,980	981	251	64

As we can see in **Table 4** for this method, lower capacity does not always result in higher quality or PSNR. The reason resides within the heart of this method. What happens to those blocks that are not embeddable and do not carry LM bits can clarify the answer.

In this method, lower capacity means that there are more non-embeddable blocks in image. These blocks do not belong to any of our 6 categories. Because maximum gap (i.e. d_{\max}) for these blocks needs to be incremented by 1 in order to prevent ambiguity, there might be cases where such blocks go through more changes and hence the quality decreases.

A look at **Fig. 7** and **Fig. 8** shows that for a single image, as it is expected, an increase in T results in an increase in capacity and a decrease in quality. But in **Table 4**, the value of T is fixed at 7 and block size is being increased. For the case that a block has the size of 64×64 pixels, the probability of the block being in one of 6 categories has obviously decreased and therefore there might be the possibility that an increase in d_{\max} by 1 causes abrupt changes.

Table 5 shows a performance comparison between the proposed method and the method proposed in [17] for single-layer and multi-layer embedding respectively. In this Table we used arithmetic coding for compression of LM . Although the best embedding capacity for Ni et al.'s method is 0.04 *bpp* [3], our proposed method can achieve 0.4 *bpp* for single layer embedding for some test images.

It is obvious that by utilizing more efficient coding schemes we can achieve more available capacity for data embedding. As shown in **Table 5** the proposed method achieved higher embedding capacity while maintaining the quality of the marked image. Besides, in Ni et al.'s method, once the peak and minimum points have been chosen, regardless of the size of to-be-embedded message, the pixels with gray value between the peak and minimum points will be shifted. Therefore, Ni et al.'s method suffers from undesirable distortion at low embedding rates, and lack of the capacity-PSNR control. On the contrary, the proposed method modifies just required number of blocks for smaller amount of payload, and uses a parameter T to control the image quality. It is worth mentioning that block-based characteristic of the proposed method increases the robustness and can prevent burst error. In other hand, occurrence of error in some blocks can't affect on the bits that are extracted from other ones. Obtained values for PSNR (above 40 dB) for single layer embedding proves high visual quality of the marked image.

Chung et al. [18] proposed a new method for optimization Ni et al.'s method. Instead of selecting the peak-zero(or minima) pairs in a greedy way, their method presents a dynamic programming based reversible data hiding algorithm to determine the most suitable peak-zero(or minima) pairs such that the embedding capacity can be maximized. Based on some artificial map images, experimental results demonstrate that this algorithm has 9% embedding capacity improvement ratio although it has some execution-time degradation. This method for standard test images such as Lena can't improve capacity. In contrast, experimental results shows our method improve embedding capacity more than 40% averagely.

Overall comparison between the existing reversible watermarking techniques and the proposed technique in terms of pure payload and the PSNR is presented in **Table 6**.

Table 5. Comparison between Ni et al.'s method and proposed method

Image	Pure Capacity (bits)		PSNR (dB)
	Ni et al.'s Method	Proposed	
Lena	5,460	10,222	48.2
Airplane	16,171	22,466	48.2
Tiffany	8,782	14,139	48.2
Jet	59,979	114,223	48.2
Baboon	5,421	8,223	48.2
Boat	7,301	3,221	48.2
House	14,310	37,994	48.2
Bacteria	13,579	20,272	48.2
Blood	79,460	90,466	48.2

Table 6. Overall comparison between other reversible watermarking methods and proposed method

Method	Pure Payload (bits)	PSNR (dB)
Honsinger et al.'s [7]	<1,024	Not mentioned
Macq and Deweyand [19]	<2,046	Not mentioned
Fridrich et al.'s [20]	1,024	Not mentioned
Goljan et al.'s [21]	3k-41k	39
Vleeschouwer et al.'s [10]	<4,096	<35
Xuan et al.'s [22]	15k-94k	24-36
Celick et al.'s [9]	15k-143k	38
Ni et al.'s [17]	5k-80k	>48
Proposed	3k-180k	>40

That is obvious that the field of image processing sometimes makes extensive use of the characteristic of natural images. Perhaps a large number of image processing algorithms cannot be proven using strict mathematics but the interesting characteristics of natural images let these algorithms work very well. For the proposed method, authors did not face any cases during their simulations and tests in which there are no embeddable blocks. But if this situation happens, payload cannot be embedded. This fact holds true even for well-known algorithms like difference expansion (DE) and original histogram shifting. For example, in DE method, if there are no expandable and shiftable pair in the image, the payload cannot be inserted and DE fails to work or if location map is not compressible then it again fails to work. But the point is that most natural images have a characteristic that let these algorithms work well and without problems.

5. Conclusion

A reversible data hiding scheme was presented in this paper that aimed for increasing the embedding capacity. By dividing host image into non-overlapping blocks and categorizing them in the embedding phase, the total capacity of data embedding for single-layer embedding is increased compared to the single-layer histogram shifting method. Also by utilizing the new strategy, large overhead information for recovering the host image was prevented and instead a much smaller location map was used. This method enabled us to determine the size of overhead data prior to embedding. Finally the efficiency of the scheme was shown in experimental results and visual examples were given. Based on these results visual quality of the scheme is satisfactory while its data embedding capacity is larger than other methods. Beside this method is robust against burst error.

References

- [1] J. Fridrich, M. Goljan and R. Du, "Lossless data embedding-new paradigm in digital watermarking," *EURASIP Journal on Applied Signal Processing*, vol. 2, no. 2, pp. 185-196, 2002. [Article \(CrossRef Link\)](#).
- [2] R. Caldelli, F. Filippini and R. Becarelli, "Reversible watermarking techniques: An overview and a classification," *EURASIP Journal on Information Security*, 2010. [Article \(CrossRef Link\)](#).
- [3] J. B. Feng, I. C. Lin, C. S. Tsai and Y. P. Chu, "Reversible watermarking: Current status and key issues," *International Journal of Network Security*, vol. 2, no. 3, pp. 161-171, May. 2006. [Article \(CrossRef Link\)](#).
- [4] Y. Q. Shi, Z. Ni, D. Zou, C. Liang and G. Xuan, "Lossless data hiding: Fundamentals, algorithms and applications," in *IEEE International Symposium on Circuits and Systems*, pp. 33-36, May. 2004. [Article \(CrossRef Link\)](#).
- [5] H. J. Hwang, H. J. Kim, V. Sachnev and S. H. Joo, "Reversible watermarking method using optimal histogram pair shifting based on prediction and sorting," *KSII Transactions on Internet and Information Systems*, vol. 4, no. 4, Aug. 2010. [Article \(CrossRef Link\)](#).
- [6] J. M. Barton, "Method and apparatus for embedding authentication information within data," U.S. Patent 5646997, 1997. [Article \(CrossRef Link\)](#).
- [7] C. W. Honsinger, P. Jones, M. Rabbani and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," U.S. Patent 6278791 B1, Aug. 2001. [Article \(CrossRef Link\)](#).
- [8] B. Macq, "Lossless multiresolution transform for image authentication watermarking," in *Proc. EUSIPCO*, pp. 533-536, Sep. 2000. [Article \(CrossRef Link\)](#).
- [9] M. U. Celik, G. Sharma, A. M. Tekalp and E. Saber, "Reversible data hiding," in *Proc. of International Conference on Image Processing*, pp.157-160, Sep. 2002. [Article\(CrossRef Link\)](#).
- [10] C. D. Vleeschouwer, J. F. Delaigle and B. Macq, "Circular interpretation on histogram for reversible watermarking," in *IEEE International Workshop on Multimedia and signal Processing*, pp.345-350, Oct. 2001. [Article \(CrossRef Link\)](#).
- [11] C. D. Vleeschouwer, J. F. Delaigle and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 16, no. 11, pp. 1423-1429, 2006. [Article \(CrossRef Link\)](#).
- [12] D. Coltuc and J. M. Chassery, "Simple reversible watermarking schemes: Further results," in *Proc. of SPIE-IS&T Electronic Imaging*, 2006. [Article \(CrossRef Link\)](#).
- [13] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 13, no. 8, pp. 890-896, Aug. 2003. [Article\(CrossRefLink\)](#).
- [14] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transaction on Image processing*, vol. 13, no. 8, pp. 1147-1156, 2004. [Article\(CrossRef Link\)](#).
- [15] D. Coltuc, "Improved capacity reversible watermarking," in *Proc. Of International Conference on*

- Image Processing (ICIP)*, pp. 249-252, 2007. [Article \(CrossRef Link\)](#).
- [16] D. Coltuc and J. M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255-258, Apr. 2007. [Article \(CrossRef Link\)](#).
- [17] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 16, no. 3, pp. 356-361, 2006. [Article \(CrossRef Link\)](#).
- [18] K. L. Chung, Y. H. Huang, W. N. Yang, W. C. Hsu and C. H. Chen, "Capacity maximization for reversible data hiding based on dynamic programming approach," *Journal of Applied Mathematics and Computation*, vol. 208, issue. 1, pp. 284-292, 2009. [Article \(CrossRef Link\)](#).
- [19] B. Macq and F. Deweyand, "Trusted headers for medical images," *Presented at the DFG VIII-DII Watermarking Workshop*, Oct. 1999. [Article \(CrossRef Link\)](#).
- [20] J. Fridrich, M. Goljan and R. Du, "Invertible authentication," in *Proc. of SPIE Security Watermarking Multimedia Contents*, pp. 197-208, Jan. 2001. [Article \(CrossRef Link\)](#).
- [21] M. Goljan, J. Fridrich and R. Du, "Distortion free data embedding," in *Proc. of 4th Information Hiding Workshop*, pp. 27-41, Apr. 2001. [Article \(CrossRef Link\)](#).
- [22] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni and W. Su, "Distortionless data hiding based on integer wavelet transform," *IEEE Electronic Letters*, vol. 38, no. 25, pp. 1646-1648, Dec. 2002. [Article \(CrossRef Link\)](#).
- [23] M. Arabzadeh, M. S. Helfroush, H. Danyali and K. Kasiri, "Reversible watermarking based on generalized histogram shifting," *18th IEEE International on Image Processing*, pp. 2741-2744, Sep. 2011. [Article \(CrossRef Link\)](#).



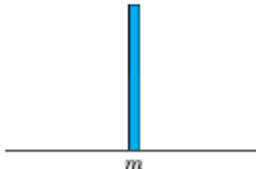
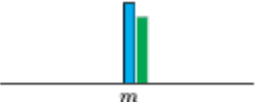
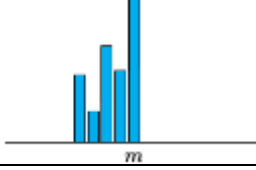
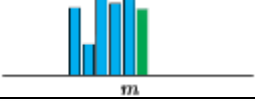
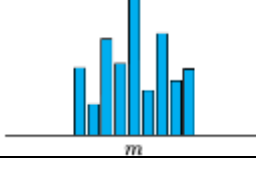
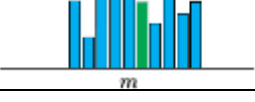
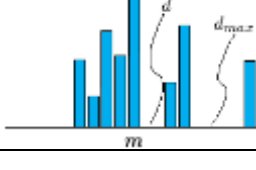

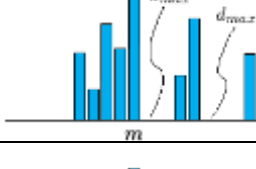
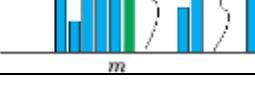
Mohammad Arabzadeh received both his B.S. and M.S. degrees from the department of electrical and electronic engineering of Shiraz university of technology, Shiraz, Iran in the field of communications. His research interests include, data hiding, watermarking and steganography and he more specifically works on reversible data hiding algorithms.



Mohammad Reza Rahimi was born May 19, 1985 in Galougah, Iran. He received his B.S degree in electrical and electronic engineering from the Urmia University, Urmia, Iran, in 2008 and his M.S degree in Telecommunication Engineering Systems from Shiraz University of Technology, Shiraz, Iran, in 2011. From September 2010 to June 2011, he was an instructor in Shahid Bahonar Institute of Technology, Shiraz, Iran. During 2011, he was an instructor in Payam-e-Noor University of Behshahr. His research interests include data hiding, digital watermarking, Blind image quality assessment and image processing.

Appendix

Table A-1. Visual examples of categorization used in this paper

Category	Before Embedding	After Embedding
1		
2		
3		
4		
5		
6	