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

Mohammad Reza Rahimi, Habibollah Danyali, Mohammad Sadegh Helfroush, Mohammad Arabzadeh
Department of Electrical and Electronic Engineering
Shiraz University of Technology
Modarres Blvd. P.O. Box 71555-313, Shiraz, Iran
Email: {mr.rahimi, danyali, ms_helfroush, m.arabzadeh}@sutech.ac.ir

Abstract—In this paper a reversible data hiding scheme based on shifting the histogram 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) are used in order to extract payload and recover the block to its original state. A method to eliminate the need for this extra information in this paper is presented. 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 while this method has higher embedding capacity than the original histogram method and their improved schemes, it maintains the quality of watermark image at an acceptable level.

I. INTRODUCTION

In the context of data hiding, it is an interesting and challenging issue to be able to recover the host image from the watermarked image without any loss. Generally in all image watermarking applications it is preferable to be able to recover the original image without any degradation in quality but it is of crucial importance in some applications where high image fidelity is required, such as medical and military images and law enforcement image analysis.

Reversible watermarking schemes can be divided into three categories [1], [2]: reversible watermarking based on lossless compression, reversible watermarking based on difference expansion and reversible watermarking based on histogram shifting.

Reversible watermarking schemes based on lossless data compression use the coding redundancy in images. They compress image data so that it takes less space and use the remaining space to embed watermark data. These schemes generally require high computational complexity and their capacity is relatively small [3]–[5].

Difference expansion-based schemes pioneered by Tian [6] use the interpixel redundancy that exist in natural images. Most of these schemes generate a small value based on original image and then expand the value to embed the watermark bits. Generally their capacity is higher than the previous scheme and they have less computational complexity [7]–[9].

Histogram shifting method was first introduced by Ni et al. [10]. This scheme uses maximum and zero (or minimum if no zero points are available) points of histogram of image and shifts the values between these points. Although this method was an effective technique, it required that some additional information be transmitted to receiver separately from watermarked image [11]. This technique is very simple however it has a low capacity.

Lately some schemes have been proposed that use a mixture of both DE and histogram shifting. These schemes mainly use a difference histogram to expand the difference values. Kim et al. [12] and Luo et al.’s [3] schemes are good examples of these schemes. But these schemes suffer from overflow/underflow issue. In Kim’s scheme modulo 64 addition is used to circumvent overflow/underflow problem. Although it is better than using modulo 256 addition, it generates noise in watermarked image. In Luo’s scheme to prevent overflow/underflow the boundary pixels (pixels with value 0 or 255) are not used for embedding. But in this way not only we lose the capacity of pixels with values of 0 and 255, but also we need to use a boundary map to distinguish the pixels that originally was 0 or 255 before watermark embedding from those that was 1 or 254 and change into 0 or 255 after watermark embedding. Even in some test images like “tiffany” the number of pixels with 0 or 255 values is relatively high and severely hurts the capacity of watermark embedding.

The situation even gets worse in the case of medical images that are mainly composed of black backgrounds and white foregrounds.

In this paper we proposed a reversible data hiding method based on histogram shifting. That is applied to a block of the host image instead of the whole image for increasing the embedding capacity. We also changed the embedding rule of Ni et al.’s method. With this new method we eliminate the need for overhead information that is produced by ordinary histogram shifting method. This technique increases the embedding capacity in two ways. First, by dividing the host image into non-overlapping blocks we use all the capacity for data embedding. For example the peak point of a block’s histogram may not be the peak point of image’s histogram and it is a local feature of the block as opposed to the global feature of image’s peak point. Second, by preprocessing the histogram in receiving side, we can find the value of peak point of the original block without the need for extra information. We achieve this by always putting the maximum gap between bins.
of histogram exactly after the original peak point. Another improvement over Ni et al.’s method is that in the proposed scheme we do not need to store the value of maximum/minimum points. This can help us save capacity for embedding payload and thus increasing capacity. The rest of the paper is organized as follows. Histogram shifting is briefly introduced in section II. Details of the proposed method are given in section III. Experimental results are presented in section IV. And finally the conclusion is given in section V.

II. HISTOGRAM SHIFTING

In Ni et al.’s scheme [10], first a pair of a maximum and a minimum point in the image histogram are found. Maximum and minimum points in this scheme referred to the values that have respectively the highest and lowest occurring frequency in image histogram. Next, all the points that resides between these two points are shifted by 1 unit. This makes a zero point just after maximum point in the image histogram. Then according to the value of watermark bits, pixels that take the value of maximum are modified to insert data. Embedding is carried out by modifying the maximum value by 1 if watermark bit is 1 or leaving it unchanged if watermark bit is 0. It is obvious that both minimum and maximum values must be known in receiver. If we choose to apply Ni et al.’s scheme to blocks of the image, we need to at least make maximum and minimum points known to receiver and since this information needs to be inserted in the host image, therefore much embedding space is used up by this information. The following subsections detail the method for blocking of image, watermark embedding and extraction procedures, and the process of restoring the image into its original form.

III. PROPOSED SCHEME

In this section a detailed explanation of the proposed scheme is presented. We first divide image blocks to 7 categories and then according to this categorization, insertion of watermark bits is performed. Insertion scheme is very similar to Ni et al.’s method [10] 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 watermark 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. First we look into embedding procedure where watermark bits are inserted into the host image and then we proceed into extraction procedure where watermark is extracted and the host image is fully recovered to its original form, proving the scheme to be reversible.

A. Categorization of Image Blocks

After dividing the image into non-overlapping blocks of size $M \times N$, 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.1:

- $d_{\text{max}}$: Maximum free gap between bins in a histogram of a block.
- $m$: Value of pixel of peak point in the histogram of a block.
- $l$: The smallest value higher than $m$ which exists in the block.
- $d$: The free gap between the bin that corresponds to peak point and the first non-zero bin to its right side.
- $n$: The number of free gaps between bins of block’s histogram that is equal to $d_{\text{max}}$.
- $d_{\text{smax}}$: The second largest free gap between bins of block’s histogram ($d_{\text{smax}} < d_{\text{max}}$).
- $l_{\text{bar}}$: The value of non-zero bar in left side of $d_{\text{max}}$.
- $r_{\text{bar}}$: The value of non-zero bar in right side of $d_{\text{max}}$.

The blocks are then categorized into 7 category based on the following conditions in Table I and after this the value of shift and LM for each block are determined. The LM (Location Map) are used in the extractor to categorize the blocks uniquely.

In the Table I conditions, the value $T$ is a threshold to control the capacity and quality of watermarking and also 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 II present the conditions that used for determination of blocks in the extractor side. In this Table $d_{\text{max}}^{\text{LM}}$ and $d_{\text{smax}}^{\text{LM}}$ indicates the value of $d_{\text{max}}$ and $d_{\text{smax}}$ for modified histogram of block after embedding. The value of LM bits are required in the extractor for extraction of hidden data and recovering the watermarked image into its original form. The LM are embedded in the blocks that have $d_{\text{max}} > T$ with simple LSB substitution. The original

![Fig. 1. The Defined Variables](image-url)

<table>
<thead>
<tr>
<th>Category</th>
<th>Conditions in Embedder</th>
<th>LM</th>
<th>shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All pixel in block are equal</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$m = \text{Max. value in block}$, $d_{\text{max}} = 0$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>$d = d_{\text{max}} = 0$</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>$d &lt; d_{\text{max}} &lt; T$, $d_{\text{max}} &gt; 0$</td>
<td>1</td>
<td>$(d_{\text{max}} + 2)$</td>
</tr>
<tr>
<td>5</td>
<td>$0 &lt; d = d_{\text{max}} &lt; T$, $n &gt; 1$</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>$0 &lt; d = d_{\text{max}} &lt; T$, $n = 1$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>$m = \text{Max. value in block}$, $0 &lt; d_{\text{max}} &lt; T$</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Essential Conditions for Categorizing Blocks in Embedder Side
We have only two non-zero bin and $k = k'$, point procedure could be explained with the running steps below:

1. 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 arithmetic coding (or JPEG2) are inserted into LSB of the blocks with $d_{max} > 2T$ and original LSB’s are gathered into a sequence and appended to watermark and therefore are considered as a part of payload. These blocks that contains $LM$ bits are kept completely unchanged during next steps. Now depending on the block category, a shift is applied to the blocks according to Table I; that is, pixels with values higher than $m_i$ are increased by a value of shift depending on the category of which they are a member. By doing so the histogram is displaced to right and we can start the embedding of payload data by scanning pixels of block is raster scan or any space filling curve. The embedding method is to keep the pixels with the value of peak point ($m_i$) unchanged if the corresponding payload bit is ”0” and to increase it by 1 if the payload bit is ”1”. Embedding procedure could be explained with the running steps below:

   - Divide host image to $M \times N$ non-overlapping blocks.
   - Generate $LM$ vector using characteristics of blocks and pre-defined value of $T$ and compress $LM$ using arithmetic coding to obtain $LM_{C}$.
   - Capture $\eta(LM_{C})$ of $LM'$s of blocks with $d_{max} > 2T$ and replace them with $LM_{C}$ bits($\eta(.)$ denotes the number of bits).
   - Concatenate captured $LM'$s and actual watermark and construct final payload.
   - Scan each block and if this block satisfactory one of the conditions listed in Table I regard it as embeddable block and embed $\eta(C_{B})$ bits using proposed scheme($C_{B}$ denotes the embedding capacity of each block).
   - Finally increase the value of $d_{max}$ for blocks that are neither embeddable block and $LM$ carrying. This is done simply by increasing the pixel values greater than $r_{bar}$ by one.

During implementation of the proposed method we discovered that if a lot of 0’s (or 1’s) occurs after each other in payload none of pixels that take the value of maximum is changed. Hence the category to which a block belongs cannot be uniquely determined in receiver and ambiguity arises. We resolved this issue by changing the first pixel with value of $m_i$ into $m_i + 1$ and keep the last one unchanged regardless of watermark bits for all the categories expect category 7. This modification permits the receiver to uniquely determine block’s category under all conditions but decreases embedding capacity by 2 bits for each block. For category 7, we additionally change second pixel with value of $m_i$ into $m_i + 2T + 2$ and so 3 bits are spent over preventing ambiguity in the receiver. 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 $l$ must be increased one unit.

Another important note is that during all the aforementioned operations if there are more than one $d_{max}$ in a block, the rightmost one is to be selected in order to minimize distortion by shifting a smaller portion of image histogram. In this way the quality of image is retained by a considerable amount. It is possible that after the above operations the value of some pixels exceed allowed range for gray scale images [0,255] in which case the coordinate of these pixels must be appropriately sent to receiver. But it is notable that this situation is rare in natural images and does not introduce much problem.

### C. Extraction Method and Recovery Procedure

In this phase, $LM$ bits are first extracted from the blocks with $d_{max} > T$ and are decompressed. Then according to the characteristics of the blocks and $LM$, the category to which it belongs are identified and finally after extraction of watermark, the histogram of the block is shifted back to get the original block. Extraction and recovery operations can be expressed in the 4 steps:

- **Extraction of compressed $LM$ bits from the $LSB$’s of blocks with $d_{max} > T$ and decompress it.**
- **Extraction of watermark according to these decompressed bits and characteristics of each block.** Extract watermark bits “0” and “1” from the pixels with value $l'_{bar}$ and $l'_{bar} + 1$ respectively.
- **After extraction of watermark from each block shift back pixels that are greater than $l'_{bar}$, according to Table I.**
- **Replace original $LSB$’s of blocks that contain $LM$ instead of $LM$ bits.**

It is obvious that according to the method used in embedding phase, the value of $l'_{bar}$ in watermarked block is certainly the same as $m_i$ for unwatermarked block. Note that first and last bits that are being extracted from each block in categories 1-6 and first, second, and last bits that are being extracted from category 7 do not carry any information and are discarded.

### IV. Simulation Results

In this section, we will show the feasibility and the performance of proposed method in terms of pure payload and image quality over the relevant techniques proposed by Ni et al. [10] and other researchers. The proposed scheme was successfully applied to a number of standard test images and equality between recovered and original images proved the reversibility of the scheme. In this experiment $512 \times 512$ test images were used.

---

**TABLE II**

**Essential Conditions for Categorizing Blocks in Extractor Side**

<table>
<thead>
<tr>
<th>Category</th>
<th>Conditions in Extractor and $LM$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>We have only two non-zero bin and $d_{max} = 0$, $LM = 0$</td>
</tr>
<tr>
<td>2</td>
<td>$d_{max} = 0$, $LM = 0$</td>
</tr>
<tr>
<td>3</td>
<td>$d_{max} = 1$, $LM = 0$</td>
</tr>
<tr>
<td>4</td>
<td>$d_{max} = 1$, $LM = 1$</td>
</tr>
<tr>
<td>5</td>
<td>$d_{max} = 1$, $d_{max} &gt; d_{max} + 1$, $LM = 0$</td>
</tr>
<tr>
<td>6</td>
<td>$d_{max} = 1$, $d_{max} &gt; d_{max} + 1$, $LM = 0$</td>
</tr>
<tr>
<td>7</td>
<td>$d_{max} = 2^T$, $LM = 1$</td>
</tr>
</tbody>
</table>
images were used and watermark was chosen to be a binary random sequence with the same size of the corresponding images embedding capacity. As it is common in literature, we adopted PSNR as imperceptibility measure to compare our scheme with other schemes.

In Fig.2,3 the embedding capacity in bpp and peak-signal-to-noise-ratio PSNR in dB using this method and for various T values was shown for standard test images.

Fig.4 demonstrates the visual imperceptibility of the scheme and Table III shows a performance comparison between the proposed method and the method proposed in [10] for single-layer and multilayer embedding respectively. In this Table we take T = 2 and we used arithmetic coding for compression of LM. Although the best embedding capacity for Ni et al.’s method is 0.04 bpp [1], our proposed method can achieve 0.1 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 this Table the proposed method achieved higher embedding capacity while maintaining the quality of the watermarked image. Besides, in Ni et al.’s method, once the peak and minimum points have been chosen, no matter how small the amount of message to be embedded, the pixels with gray value between the peak and minimum points will be changed one grayscale unit. Therefore, Ni et al.’s method suffers from undesirable distortion at low embedding rates, and lack of the capacityPSNR control. On the contrary, proposed method modifies just enough number of blocks for smaller amount of payload, and uses a parameter T to control image quality. It is worth mentioning that block-based characteristic of proposed method increase the robustness and can prevent error distribution. In other hand, occurance of error in some block can’t affect on the bits that extract from another ones. Experimental value of PSNR (namely 47 dB) proves high visual quality of the watermarked image.

<table>
<thead>
<tr>
<th></th>
<th>Proposed Scheme</th>
<th>Ni et al.’s Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LM size</td>
<td>Pure Cap</td>
</tr>
<tr>
<td>Lena</td>
<td>3013</td>
<td>10222</td>
</tr>
<tr>
<td>Boat</td>
<td>2137</td>
<td>15317</td>
</tr>
<tr>
<td>House</td>
<td>1819</td>
<td>21801</td>
</tr>
<tr>
<td>F-16</td>
<td>2663</td>
<td>22466</td>
</tr>
</tbody>
</table>
Kuo et al. [13] proposed a new method for optimization Ni et al.’s method. Instead of selecting the peak-valley pairs in a greedy way, their method presents a dynamic programming based reversible data hiding algorithm to determine the most suitable peak-valley pairs such that the embedding capacity object 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 and . In contrast, experimental results shows our method improve embedding capacity more than 50% averagely.

V. CONCLUSION

A reversible watermarking 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 introduced in [10]. 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.

REFERENCES