Lossless Data Hiding in Images Based on Histogram Modification

1T.Venkata Satya Vivek, 2V.Lakshma Reddy, 3M Rao Batchnaboyina, 4G.Anil, 5M.Srinivasarao, 6Ch.Koteswararao
1,3,5,6 Computer Science & Engineering Department, PACE Institute of Technology & Sciences Ongole, India
2,4 PACE Institute of Technology & Sciences Ongole, India
Email: 1tvs vivek1990@gmail.com, 2lakshma.vuyuru@gmail.com, 3mallik.mit@gmail.com,
4anilganta16@gmail.com, 5mr.srinu13@gmail.com 6chkoteswararao@hotmail.com

Abstract—This paper proposes an improved reversible data hiding method which uses a contemporary algorithm based on histogram shifting. Using the size of desired payload of an image, an embedding order based on the embedding level and the scanning technique is determined. Since the neighbour pixel values are same most of the differences between pairs of adjacent pixels are equal or close to zero. In the data embedding stage, a multilevel histogram modification mechanism is employed where the data is going to be embed in the zero points. The distortion that takes place due to the embedding can be completely removed during the data extraction. A persistent recovery strategy is exploited for each pixel is reconstructed with the aid of its previously recovered neighbour. Experimental results show that the improved method performs better than the existing methods.

Key Words:- Data embedding, Reversible Data Hiding, Histogram Modification, Histogram Shifting, Persistent Recovery, Lossless Data Hiding.

I. INTRODUCTION

Data hiding also known as Information Hiding plays a pro-vital role in the information security. The main aim of data hiding is to embed the imperceptible confidential information into the cover media such as still images, audio’s, video’s etc..It can be classified into several branches such as Watermarking, Steganography and visual Cryptography. This paper mainly concentrates on the data hiding scheme that can be classified into the category of steganography. Steganography is the art of concealing a message, image, or file within the another message, image, or file. The word Steganography combines the words as “steganos”-means “Protected or concealed” and “grapher”-means “Writing”. Electronic communications may include steganographic coding in digital steganography which takes place inside the transport layer, such as a document file, image file, program or protocol. In contrast, the digital watermarking refers to the process of embedding watermark bits into a host media signal like audio clips, images or video clips without modifying of distorting the host signal. In the applications like Authentication, Secret Communications, and Content Protection we use different types of embedding techniques like Transform Domain Spread Spectrum or Equalization techniques. With the existing of many straight-forward implementations of data hiding techniques, such as Least Significant Bit based hiding-these methods do not allow for the recovery of the original image [1]. A drawback of these methods is that the cover media will be injured and cannot be recovered after secret data extraction. To solve this problem many researchers have proposed various Reversible Data Hiding Algorithms.

The aspect of Reversible Data Hiding Algorithms is desirable for the fields that require high precision of an image where the data is embedded into the image in a manner that allows for the retrieval of the embedded data as well as the exact original image. The main provocation of Reversible data hiding algorithm is distortion-free data embedding [2]. For this reason, reversible (lossless) data hiding is widely used on sensitive imagery, such as military data, medical diagnosis, and space exploration. In such cases, every bit of information is important. Even a slight change will affect the intelligence of the image, and the access to the original, raw data is also required. There exists many emerging applications within the sensitive imagery are in the health domain. In other applications such as remote sensing and high-energy particle physical experimental investigation, it is also desired that he original cover media can be recovered because of the required high-precision nature. The different marking techniques that are satisfying this requirement are referred to as the reversible, lossless, distortion-free, or invertible data hiding techniques. Reversible data hiding algorithms facilitates the immense possibility of applications to link the two sets of data in such a way that the cover media can be lossless recovered after the hidden data been extracted out. The Performance of the Reversible Data Hiding Algorithms is measured in the following aspects: 1.Embedding Capacity, 2.Visual Quality, 3.Computational Complexity, and 4. Robustness.
II. RELATED WORK

To the best of our knowledge, the researchers have been proposed many lossless data hiding methods. In this section, we are going to concentrate on various data hiding techniques in Steganography to facilitate secure data transmission over the underlying communication network.

A. Reversible Data Hiding Based on Histogram Modification and Shifting

Ni et al.[3] utilizes zero or minimum point of histogram. If the peak is lower than the zero or minimum point in the histogram it increases pixel values by one from higher than the peak to lower than the zero or minimum point in the histogram. While embedding, the whole image is searched. Once a peak-pixel value is encountered, if the bit to be embedded is ‘1’ the pixel is added by 1, else it is kept intact. Alternatively, if the peak is higher than the zero or minimum point in the histogram, the algorithm decreases pixel values by one from lower than the peak to higher than the zero or minimum point in the histogram, and to embed bit ‘1’ the encountered peak-pixel value is subtracted by 1. The decoding process is quiet simple and opposite of the embedding process. The algorithm essentially does not follow the general principle of lossless watermarking.

Histogram based reversible data hiding scheme was first proposed by Ni et al. [4] [5] in 2006. A new reversible data embedding technique, which can embed a large amount of data (5–80 kb for a 512×512 8 grayscale image) while keeping a very high visual quality for all natural images, specifically, the PSNR of the marked image versus the original image is guaranteed to be higher than 48 dB. It utilizes the zero or the minimum point of the histogram and slightly modifies the pixel grayscale values to embed data.

The technique [6] proposes a low-distortion transform for prediction-error expansion reversible watermarking. The transform is derived by taking a simple linear predictor and by embedding the expanded prediction error not only into the current pixel but also into its prediction context. The embedding ensures the minimization of the square error introduced by the watermarking. The proposed transform introduces less distortion than the classical prediction-error expansion for complex predictors such as the median edge detector or the gradient-adjusted predictor.

Ni et al. [7] increased the hiding capacity by extending the histogram modification technique for integer wavelet transform. This is one of the only existing semi-fragile lossless data hiding technique, which is robust against high-quality JPEG compression. This technique is mainly based on modulo-256 addition to achieve losslessness. This technique is mainly suffers from salt-and-pepper noise. So, to eliminate this noise the authors identified a novel technique which is used by identifying a robust statistical quantity based on the patchwork theory and employing it to embed data.

Tsai et al. [8] used a residue image indicating a difference between a basic pixel and each pixel in a non-overlapping block to increase the embedding capacity. In this the author used the similarity of the neighbouring pixels in the images was explored by using the prediction technique and the residual histogram of the predicted errors of the host image was used to hide the secret data in this technique. The other technique is based on the interleaving prediction to improve histogram based reversible data hiding [9]. In these interleaving methods, the predictive values are as many as the pixel values. All predictive error values are transformed into histogram to create higher peak values and to improve the embedding capacity.

In 2011, Jung et al. [10] proposed an improved histogram modification based reversible data hiding technique with a consideration of the human visual system (HVS) characteristics. In this algorithm, unlike the conventional reversible data hiding techniques, a data embedding level is adaptively adjusted for each pixel with a consideration of the Human Visual System (HVS) characteristics. To this end, an edge and the just noticeable difference (JND) values are estimated for every pixel, and the estimated values are used to determine the embedding level. In this algorithm, a local causal window is used to predict a pixel value and estimate an edge.

III. PROPOSED SCHEME

In our technique, the spiral “S” order is adopted to scan the image pixels for difference generation. The pseudo random number generates the secret data are binary sequences. In the data embedding stage, a multilevel histogram modification strategy is utilized. An integer parameter called embedding level ELevel (ELevel ≥ 0) is involved to control the hiding capacity. A larger EL indicates more secret data can be embedded. As the embedding operations for ELevel > 0 are more complicated than those of ELevel = 0, we describe them for ELevel = 0 and ELevel > 0 separately. The below are the steps for the Data Embedding:

Step 1. Spiral “S” scan I into a pixel sequence p1 , p2 , . . . , pM×N .

Step 2. Compute the differences Di and construct a histogram based on Di (2 ≤ i ≤ M × N).

Step 3. Select an ELevel. If ELevel = 0, execute Step 4. If EL = 0, go to Step 5.

Step 4. Data embedding for ELevel = 0.

Step 4.1. Shift the right bins of b(0) rightward one level as:
Di' = \begin{cases} 
  P1 & \text{if } i=1 \\
  Di & \text{if } Di<=0, 2<=i<=M*N \\
  Di+1 & \text{if } Di>0, 2<=i<=M*N 
\end{cases}

**Step 4.2.** Examine Di = 0 (2 ≤ i ≤ M × N) one by one. Each difference equaling 0 can be used to hide one secret bit. If the current pro-cessing secret bit w = 0, it is not changed. If w = 1, it is added by 1. The operation is as:

Di'' = \begin{cases} 
  P1 & \text{if } i=1 \\
  Di+w & \text{if } Di'=0, 2<=i<=M*N \\
  Di' & \text{if } Di'=0, 2<=i<=M*N 
\end{cases}

The histogram modification strategy for ELevel = 0 is shown where the red and blue arrows indicate embedding “0” and “1”, respectively. After that, go to Step 6.

**Step 5.** Data embedding for ELevel > 0.

**Step 5.1.** Shift the right bins of b(ELevel) rightward ELevel + 1 levels, and shift the left bins of b(−ELevel) leftward ELevel as:

Di'' = \begin{cases} 
  P1 & \text{if } i=1 \\
  Di & \text{if } −ELevel<=Di<=ELevel, 2<=i<=M*N \\
  Di+ELevel+1 & \text{if } Di>ELevel, 2<=i<=M*N \\
  Di-ELevel & \text{if } Di<=−ELevel, 2<=i<=M*N 
\end{cases}

**Step 5.2.** Examine di = 0 (2 ≤ i ≤ M × N) in the range of [−EL, EL] one by one. The multilevel data embedding strategy is described as follows.

**Step 5.2.1.** Embed the secret data as:

Di'' = \begin{cases} 
  P1 & \text{if } i=1 \\
  Di' & \text{if } −ELevel<=Di'<ELevel, 2<=i<=M*N \\
  2*ELevel+w & \text{if } Di'=ELevel, 2<=i<=M*N \\
  2*ELevel-w+1 & \text{if } Di'=−ELevel, 2<=i<=M*N 
\end{cases}

**Step 5.2.2.** ELevel is decreased by 1.

**Step 5.2.3.** If ELevel = 0, execute Step 5.2.1 and Step 5.2.2 repeatedly. If ELevel = 0, then go to Step 6:

Di'' = \begin{cases} 
  P1 & \text{if } i=1 \\
  Di' & \text{if } Di'=0, 2<=i<=M*N \\
  Di' & \text{if } Di'=0, 2<=i<=M*N 
\end{cases}

The histogram modification strategy for ELevel = 2 is shown where the red and blue arrows correspond to embedding “0” and “1”, respectively.

**Step 6.** Generate the marked pixels sequence p as:

Pi' = \begin{cases} 
  P1 & \text{if } i=1 \\
  Pi-1 - Di' & \text{if } 2 <= i <= M × N 
\end{cases}

**Step 7.** Rearrange p and the marked image I is obtained.

**Data Extraction and Image Recovery:** The Data Extraction and Image Recovery is the Inverse Process of Data Embedding.

**Example 1:** The pixel values of an image are as follows:

Example for data embedding for ELevel=0.

**Example for data embedding for ELevel=2**

<table>
<thead>
<tr>
<th>P1</th>
<th>55</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>53</td>
</tr>
<tr>
<td>P1</td>
<td>52</td>
</tr>
<tr>
<td>P1</td>
<td>54</td>
</tr>
<tr>
<td>P1</td>
<td>56</td>
</tr>
<tr>
<td>P1</td>
<td>56</td>
</tr>
<tr>
<td>P1</td>
<td>54</td>
</tr>
<tr>
<td>P1</td>
<td>53</td>
</tr>
</tbody>
</table>

**Step 1:** Scan the pixel values in the Spiral S order as shown below:

55 53 52 54 56 56 54 53

**Step 2:** Calculate the difference between the pixel values; keeping the first pixel value ’55’ constant in all the steps.

55 2 1 0 -2 -2 0 2 1

**Step 3:** Now, shift the histogram values by adding ‘1’, keeping the negative values same.

55 3 2 0 -2 -2 0 3 2

**Step 4:** After the shifting is completed, we need to embed the data into it. The data to be embedded is “10”. Now, in the place of “Zeros” replace it to “1”, if the to be embed data is “0”, then put the same value “0”, as shown below:

55 3 2 1 -2 -2 0 3 2

**Step 5:** The final pixel values which we get after embedding the data is as follows:

55 52 51 51 54 56 56 53 52

After rearranging the pixel values from Step-5, the pixel values are as follows:

55 52 51
53 52 51
56 56 54

**The Data Extraction and Image Recovery is as follows:**

<table>
<thead>
<tr>
<th>P1</th>
<th>55</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>52</td>
</tr>
<tr>
<td>P1</td>
<td>51</td>
</tr>
<tr>
<td>P1</td>
<td>54</td>
</tr>
<tr>
<td>P1</td>
<td>56</td>
</tr>
<tr>
<td>P1</td>
<td>56</td>
</tr>
<tr>
<td>P1</td>
<td>53</td>
</tr>
<tr>
<td>P1</td>
<td>52</td>
</tr>
</tbody>
</table>

**Step 1:** Scan the image in the Spiral S-order .

55 52 51 51 54 56 56 53 52

**Step 2:** Now, take the embedded data values .

55 3 2 1 -2 -2 0 3 2

**Step 3:** Take the Difference Computation values .

55 2 1 0 -2 -2 0 2 1

**Step 4:**

Extracted Bits: 1

Example 2:
The pixel values of an image are as follows

Example for data embedding for ELevel = 2
Step-1: Scan the pixel values in the Spiral S order as shown below:

| 80 | 76 | 73 |
| 79 | 77 | 73 |
| 80 | 78 | 77 |

Step-2: Calculate the difference between the pixel values; keeping the first pixel value ‘55’ constant in all the steps.

| 80 | 4  | 3  | 0  | -4 | -1 | -2 | 1  | 2  |

Step-3: Now, shift the histogram values by adding ‘1’, keeping the negative values same.

| 80 | 7  | 6  | 0  | -6 | -1 | -2 | 1  | 2  | 5  |

Step-4: After the shifting is completed, we need to embed the data into it. The data to be embedded is “01”. Now, in the place of “Zeros” replace it to “1”, if the to be embed data is “0”, then put the same value “0”, as shown below:

| 80 | 7  | 6  | 0  | -6 | -1 | -3 | 1  | 2  | 5  |

Step-5: Now embedding the data “10”.

| 80 | 7  | 6  | 0  | -6 | -2 | -3 | 2  | 5  |

Step 6: Now again we need to embed the data bit “1”.

| 80 | 7  | 6  | 1  | -6 | -2 | -3 | 2  | 5  |

Step 7: The final pixel values which we get after embedding the data is as follows:

| 80 | 73 | 70 | 72 | 79 | 79 | 81 | 78 | 74 |

After rearranging the pixel values from Step-7, the pixel values are as follows:

| 80 | 73 | 70 |
| 78 | 74 | 72 |
| 81 | 79 | 79 |

IV. EXPERIMENTAL RESULTS

The experimental results for the above algorithm is as follows:

CONCLUSION

An Improved Reversible Data Hiding technique which is proposed in this paper is to embed the data in an image using Multilevel Histogram Modification. A higher level of security is obtained on taking the Spiral order with one or two level histogram modification based methods. The secret data is embedded based on the differences of the adjacent pixel values in the peculiar order which cannot easily identified. The marked image quality will be improved when compared this with the other previous histogram modification based work.

REFERENCES


