A Comprehensive Survey on Various Reversible Data Hiding Methods

[a]Shruthi N, [b]Savitha C, [c]M Z Kurian
[a] M.Tech Student, Digital Electronics, Dept. of ECE, SSIT, Tumkur, India, 572105.
[b] Assistant Professor, Dept. of ECE, SSIT, Tumkur, India, 572105.
[c] HOD, Dept. of ECE, SSIT, Tumkur, India, 572105.

Abstract

A digital watermark is a kind of marker covertly embedded in a noise-tolerant signal such as image data. Watermarking is the process of hiding digital information in a carrier signal; the hidden information does not need to have a relation to the carrier signal. Watermarking is one of the promising solutions for detection and protection of hidden digital content. However, watermarking may cause damage to the sensitive information present in the cover image. Therefore, at the receiving end, the exact recovery of cover image may not be possible. Additionally, there exist certain applications that may not tolerate even small distortion in the cover image. In such applications, reversible data hiding instead of conventional watermarking is employed.

Reversible data hiding (RDH) is a technique which enables images to be authenticated and then restored to their original form by removing the digital watermark and replacing the image data that had been overwritten. In this paper, a survey on various reversible data hiding algorithms such as histogram modification, difference expansion, expansion embedding, least significant bit (LSB) modification, prediction error expansion has been discussed.

KEYWORDS— Reversible data hiding, histogram shifting, difference expansion, LSB modification, prediction error expansion.

I. Introduction

Data hiding is a process of hiding some secret information into a cover media. Reasons behind data hiding are to secure confidential data, trade secrets and to avoid misuse of data. The data hiding process links two sets of data, one is the embedded data and another is the cover media data. The relationship between these two sets of data characterizes different applications. For instance, in covert communications, the hidden data may often be irrelevant to the cover media. In authentication, however, the embedded data are closely related to the cover media. In these two types of applications, invisibility of hidden data is an important requirement. In most cases of data hiding, the cover media will experience some distortion due to data hiding and cannot be inverted back to the original media. That is, some permanent distortion has occurred to the cover media even after the hidden data have been extracted out. In some applications, such as medical diagnosis and law enforcement, it is critical to reverse the marked media back to the original cover media after the hidden data are retrieved for some legal considerations. In other applications, such as remote sensing and high-energy particle physical experimental investigation, it is also desired that the original cover media can be recovered because of the required high-precision nature. The marking techniques satisfying this requirement are referred to as reversible, lossless, distortion-free, or invertible data hiding techniques.

Reversible data hiding (RDH) facilitates immense possibility of applications to link two sets of data in such a way that the cover media can be recovered without loss after the hidden data have been extracted out, thus providing an additional avenue of handling two different sets of data. Reversible data hiding aims to embed secret message into a cover image by slightly modifying its pixel values. RDH is a special type of information hiding and its feasibility is mainly due to the lossless compressibility of natural images. The reversibility in RDH is quite desirable and helpful in some practical applications such as medical image processing.

II. Literature Review

In [1] Ni et al. proposed the first histogram shifting (HS) based RDH method. This method utilizes peak (maxima) and minimum points of the pixel-intensity-histogram to embed the data. HS adds gray values to some pixels in order to shift a range of classes of the image histogram to create a gap near the histogram maxima. Pixels which belong to the class of the histogram maxima (carrier class) are then shifted to the gap or kept unchanged to encode one bit of the message i.e, ‘0’ or ‘1’. The other pixels (non carriers) are simply shifted.

Basic idea is to shift each pixel value at most by 1, thus a good visual quality of marked image can be obtained. The PSNR of the marked image versus the original image is guaranteed to be higher than 48 dB. However, its embedding capacity is quite low and this method does not work well if the cover image has a flat histogram.

Tian [2] proposed differential expansion (DE) algorithm for a pair of pixels to devise a high-capacity and low-distortion reversible watermark. In DE algorithm, the host image is divided into pixel pairs, and the difference value of two pixels in each pair (that are not expected to cause an overflow or underflow) is expanded to carry one bit data. This method can provide an embedding capacity up to 0.5 bits per pixel (BPP) in a single pass. In particular, Tian employed a location map to record all expandable locations, and afterwards, the technique of location map is widely adopted by most RDH algorithms. The location map that indicates the modified pairs is compressed and included in the payload.

In [3] Thodi et al. introduced Expansion Embedding (EE) technique which is a generalization of Difference Expansion proposed by Tian et al. which expands the difference between two adjacent pixels by shifting to the left its binary representation, thus creating a new virtual least significant bit (LSB) that can be used for data insertion. Since then, EE has been applied in some transformed domain such as the wavelet domain or to prediction-errors. EE is usually associated with LSB substitution applied to samples that cannot be expanded due to the signal dynamic limits or in order to preserve the image quality.

The authors achieved a significant improvement by incorporating DE with HS. In addition, instead of the difference value, they suggested utilizing the prediction-error for expansion embedding since this can better exploit local correlations within neighboring pixels. The prediction-error expansion (PE) algorithm is essentially a particular form of difference expansion. Thodi et al. employed a pixel’s three-neighbor context to predict the pixel value and used the expansion of prediction-error between the original pixel value and the estimated one to embed message. The PE algorithm achieved a maximal embedding rate of 1 bit per pixel (bpp).

In [4] Sachnev et al. presents a reversible watermarking algorithm for images without using a location map in most cases. The algorithm employs prediction errors to embed data into an image. A sorting technique is used to record the prediction errors based on magnitude of its local variance. Sachnev et al. points out that RDH algorithm with no location maps, or smaller in some cases, are very desirable.

Celik et al. [5] devised a low-distortion, reversible watermark that is capable of embedding as high as 0.7 bits/pixel. The algorithm first quantizes each pixel by a quantizer of step size L, compresses the quantization noise and appends a payload to it, then adds an L-ary representation of the result to the quantized image.

Kamstra et al. [6] utilized low-pass image to predict expandable locations so that the location map can be remarkably compressed. The authors improved Tian's methods by sorting least significant bits (LSBs) or pairs of pixels to be watermarked with respect to the obtained values heuristically. The sorting improves the coding efficiency of the lossless compression, so that the overall performance is improved.

III. Reversible Data Hiding

Data hiding is a term encompassing a wide range of applications for embedding messages in content. Usually, hiding information destroys the host image even though the distortion introduced by hiding is imperceptible to the human visual system. However, there are some sensitive images for which any embedding distortion of the image is intolerable, such as medical images, military images or artwork preservation. For images like in medical field, even slight changes are unacceptable because of the potential risk of a physician misinterpreting the image. In other applications, such as remote sensing it is also desired that the original cover media can be recovered because of the required high-precision nature. In these cases a special kind of data hiding method called reversible data hiding or lossless data hiding is used. Reversible data hiding (RDH) techniques are designed to solve the problem of lossless embedding of large messages in digital images so that after the embedded message is extracted, the image can be restored completely to its original state. The block diagram of reversible data hiding is shown in figure 1.

Steps of RDH:

- Data Embedding.
- Data Extraction.

In data embedding process, the secret digital data is embedded into the cover image by using some RDH algorithm to produce a stego image that can be stored or transmitted. On the other end, the decoder or extractor receives the stego image and extracts the secret data as well as restores the cover image. In some algorithms the decoder work is only to check that data is actually embedded in the file or not. It is in the case where the hidden data are a watermark originally placed in the cover to prove ownership.

An important feature of reversible data hiding is the reversibility, that is, one can remove the embedded data from the stego image to restore it to the original image. From the viewpoint of information hiding, reversible data embedding hides some information in a digital image such a way that only an authorized party could decode the hidden information and also could restore the image to its original state. The important metrics to determine the performance of reversible data-embedding algorithm are

1) Payload capacity limit: determines the maximal amount of information that can be embedded.
2) Visual quality: determines the visual quality on the embedded image.
3) Complexity: determines the algorithm complexity.

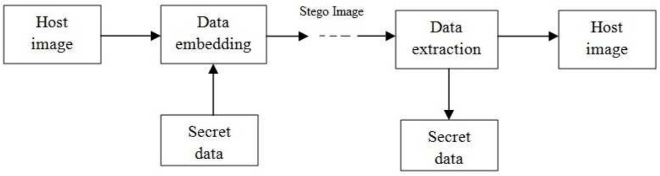

Figure 1: Reversible data hiding.

IV. RDH methods

A. Histogram Shifting

- Embedding Algorithm:

In the histogram, there are two points namely, a zero point and a peak point. A zero point corresponds to the grayscale value which has no pixel in a given image. A peak point corresponds to the grayscale value which has the maximum number of pixels. Note that zero point defined above may not exist for some image histograms. The concept of minimum point is hence more general. By minimum point, it means a grayscale value 'b', that has a minimum number of pixels, assume this value be h(b), is minimum. Accordingly, the peak point 'a' is referred to as maximum point.

1) Embedding Algorithm with one pair of maximum and minimum points: For an $M \times N$ image, each pixel grayscale value $x \in [0, 255]$.
   a) Generate its histogram H(x).
   b) In the histogram H(x), find the maximum point h(a), $a \in [0, 255]$ and the minimum point zero h(b),
   $b \in [0, 255]$.
   c) If the minimum point $h(b) > 0$, recode the coordinate (i, j) of those pixels and the pixel grayscale value 'b' as overhead bookkeeping information (referred to as overhead information for short). Then set h(b)=0.
   d) Without loss of generality, assume $a<b$. Move the whole part of the histogram H(x) with $x \in [a, b]$ to the right by 1 unit. This means that all the pixel grayscale values (satisfying $x \in [a, b]$) are added by 1.
   e) Scan the image, once meet the pixel (whose grayscale value is 'a'), check the to-be-embedded bit. If the to-be embedded bit is '1', the pixel grayscale value is changed to 'a+1'. If the bit is '0', the pixel value remains 'a'.

2) Actual data embedding capacity (PurePayload): In this way, the actual data embedding capacity C, is calculated as follows:

$$C = h(a) - O$$

Where, O denotes the amount of data used to represent the overhead information. It is also referred to as pure payload.

Clearly, if the required payload is greater than the actual capacity, more pairs of maximum point and minimum point need to be used. The embedding algorithm with multiple pairs of maximum point and minimum point is presented below.

3) Pseudocode Embedding Algorithm with Multiple Pairs of Maximum and Minimum Points: Without loss of generality, only a pseudocode embedding algorithm for the case of three pairs of maximum and minimum points is presented below. It is straightforward to generate this code to handle the cases where any other number of multiple pairs of maximum and minimum points is used. For an M × N image with pixel gray scale value x ϵ [a, b].
   a) Generate its histogram H(x).
   b) In the histogram H(x), find three minimum point h (b1), h (b2), h (b3). Without loss of generality, assume three minimum points satisfy the following condition: 0<b1<b2<b3<255.
   c) In the intervals of (0,b1) and (b3,255), find the maximum point h(a1), h(a3), respectively, and assume a1ϵ[0, b1], a3ϵ[b3, 255]
   d) In the intervals (b1, b2) and (b2, b3), find the maximum points in each interval. Assume they are h (a12), h (a21), b1<a12<a21<b2 and h (a23), h (a32), b2<a23<a32<b3.
   e) Find a point having a larger histogram value in each of the following three maximum point pairs (h (a1), h (a2)), (h (a21), h (a23)) and (h (a32), h (a3)) respectively. Without loss of generality, assume h (a1), h (a23), h (a3) are the three selected maximum points.
   f) Then (h (a1), h (b1)), (h (a23), h (b2)), (h (a3), h (b3)) are the three pairs of maximum and minimum points. For each of these three pairs, apply the above three steps of algorithm for one pair of maximum and minimum points. That is, each of these three pairs is treated as a case of one pair of maximum and minimum points.

- Extraction Algorithm:

For the sake of brevity, only the simple case of one pair of minimum point and maximum point is described here because, as shown above, the general cases of multiple pairs of maximum and minimum points can be decomposed as a few one pair cases. That is, the multiple pair case can be treated as the multiple repetition of the data extraction for one pair case. Assume the grayscale value of the maximum point and the minimum points are ‘a’ and ‘b’, respectively. Without loss of generality, assume a<b. The marked image is of size M × N, each pixel grayscale value x ϵ [0, 255].

a) Scan the marked image in the same sequential order as that used in the embedding procedure. If a pixel with its grayscale value ‘a+1’ is encountered, a bit ‘1’ is extracted. If a pixel with its value ‘a’ is encountered, a bit ‘0’ is extracted.
b) Scan the image again, for any pixel whose grayscale value x ϵ [a, b], the pixel value ‘x’ is subtracted by 1.
c) If there is overhead bookkeeping information found in the extracted data, set the pixel grayscale value (whose coordinate (i, j) is saved in the overhead) as ‘b’.
In this way, the original image can be recovered without any distortion [1].

- The advantages of this method are – (1) it is simple, (2) it always offers a constant PSNR 48.0dB, and (3) distortions are quite invisible.

- The disadvantages are – (1) capacity is limited by the frequency of peak-pixel value in the histogram, and (2) it searches the image several times, so the algorithm is time consuming [8].

B. Difference Expansion

- Embedding algorithm
  - *a)* Differences of neighboring pixel values are calculated.
  - *b)* Changeable bits in those differences are determined.
  - *c)* Some differences are chosen to be expandable by 1-bit, so changeable bits increases.
  - *d)* Concatenated bit-stream of compressed original changeable bits, the location of expanded difference numbers (location map), and the hash of original image (payload) is embedded into the changeable bits of difference numbers in a pseudo random order.
  - *e)* Use the inverse transform to have the watermarked pixels from resultant differences.

- Extraction algorithm
  - *a)* Differences of neighboring pixel values are calculated.
  - *b)* Changeable bits in those differences are determined.
  - *c)* Extract the changeable bit-stream ordered by the same pseudo random order as embedding.
  - *d)* Separate the compressed original changeable bit-stream, the compressed bit-stream of locations of expanded difference numbers (location map), and the hash of original image (payload) from extracted bit-stream.
  - *e)* Decompress the compressed separated bit-streams and reconstruct the original image replacing the changeable bits.
  - *f)* Calculate the hash of reconstructed image and compare with extracted hash.

- The advantages are: (1) no loss of data due to compression-decompression, (2) also applicable to audio and video data, and (3) encryption of compressed location map and changeable bit-stream of different numbers increase the security.

- The disadvantages are: (1) there may be some round off errors, though very little, (2) largely depends on the smoothness of natural image; so cannot be applied to textured image where the capacity will be zero or very low, and (3) there is significant degradation of visual quality due to bit-replacements of gray scale pixels [8].

C. LSB Modification

One of the earliest data-embedding methods is the LSB modification. In this well-known method, the LSB of each signal sample is replaced (over written) by a payload data bit embedding one bit of data per input sample. If additional capacity is required, two or more LSBs may be over written allowing for corresponding bits per sample.

During extraction, these bits are read in the same scanning order, and payload data is reconstructed. LSB modification is a simple, non-robust embedding technique with a high-embedding capacity and small bounded embedding distortion. A generalization of the LSB-embedding method, namely G-LSB, is employed here. If the host signal is represented by a vector, the G-LSB embedding and extraction processes can be represented as

$$s_w = Q_L(s) + w$$
$$w = s_w - Q_L(s_w) = s_w - Q_L(s)$$

Where, s_w represents the signal containing the embedded information, w represents the embedded payload vector of L-ary symbols, i.e., $w_i \in \{0, 1, …., L-1\}$, and $Q_L(x) = L\,[x/L]$ is an L-level scalar quantization.

In the embedding phase, the lowest L levels of the signal samples are replaced (over-written) by the watermark payload using a quantization step followed by an addition.

During extraction, the watermark payload is extracted by obtaining the quantization error or simply reading lowest L levels of the watermarked signal.

The classical LSB modification, which embeds a binary symbol (bit) by overwriting the least significant bit of a signal sample, is a special case where L = 2. G-LSB embedding enables embedding of non-integer number of bits in each signal sample and, thus, introduces new operating points along the capacity-distortion curve [5].

- The advantages are: (1) chances for degradation of the original image are less, and (2) more data or information can be stored in an image.

- The disadvantages are: (1) it is less robust as the hidden data can be lost with image manipulation, and (2) the hidden data can be destroyed easily by simple attacks [7].

V. Conclusion

In this paper, we got into the reversible data hiding principles for digital images. Some of proposed algorithms in the field of RHD were investigated. They were: Reversible Data Hiding by Histogram Shifting, Difference Expansion and LSB Modification methods. Each of these methods with its advantages and disadvantages were discussed. The focus of all methods is on high payload with less degradation of data. The performance can be evaluated by determining the visual quality of the image and by determining the complexity of an algorithm.

References

[1] Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, and Wei Su, “Reversible data hiding”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 3, March 2006.

[2] Jun Tian, “Reversible Data Embedding Using a Difference Expansion”, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 13, No. 8, August 2003.

[3] Diljith M. Thodi and Jeffrey J. Rodríguez, “Expansion Embedding Techniques for Reversible Watermarking”, IEEE Transactions on Image Processing, Vol. 16, No. 3, March 2007.

[4] Vasiliy Sachnev, Hyoung Joong Kim, Jeho Nam, Sundaram Suresh, and Yun Qing Shi, “Reversible Watermarking Algorithm Using Sorting and Prediction”, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 19, No. 7, July 2009.

[5] Mehmet Utku Celik, Gaurav Sharma, Ahmet Murat Tekalp and Eli Saber, “Lossless Generalized-LSB Data Embedding”, IEEE Transactions on Image Processing, Vol. 14, No. 2, February 2005.

[6] Lute Kamstra and Henk J. A. M. Heijmans, “Reversible Data Embedding Into Images Using Wavelet Techniques and Sorting”, IEEE Transactions On Image Processing, Vol. 14, No. 12, December 2005.

[7] Mehdi Hussain and Mureed Hussain, “A Survey of Image Steganography Techniques”, International Journal of Advanced Science and Technology, Vol. 54, May 2013.

[8] Masoud Nosrati, Ronak Karimi and Mehdi Hariri, “Reversible Data Hiding: Principles, Techniques, and Recent Studies”, World Applied Programming, Vol (2), Issue (5), May 2012, 349-353.