ISSN 2348 – 7968

Reversible data hiding with a general framework through histogram shifting method

Mr. Roshan Bonde[1], Prof. Vidya Dahake[2], Prof. Shubhangi Dhengre[3]

[1]M. Tech Student, Department of Electronics Engineering (Communication)
Abha Gaikwad-Patil College of Engineering, Nagpur, India 440016
[2]Assistant Professor, Department of Electronics Engineering (Communication)
Abha Gaikwad-Patil College of Engineering, Nagpur, India 440016
[3]Assistant Professor, Department of Electronics Engineering (Communication)
Abha Gaikwad-Patil College of Engineering, Nagpur, India 440016

Abstract

Histogram Shifting has a monopoly in the list of RDH technique constituting high capacity, low distortion and better efficiency .In this paper HS- based technique has been reanalysed and represent a framework for it. In addition one novel and efficient RDH are also introduced demonstrating suitability of the framework. The Experimental results show the better Peak Signal to Noise Ratio (PSNR) with the existing methods.

Keywords—PSNR; RDH;HM.

1. Introduction

In Ancient days the secrete data communication techniques were performed using chemicals, mediators etc. The Reversible Data Hiding was even done in olden days where the Host data or Cover is the mediator and the secrete data is hidden in the mediator's body or on the mediator's body. Now in the modern world the hidden data transfer from one place to other is called Steganography or Cover writing. In steganography, the embedding data is irrelevant to the cover media data and is used in the covert communication applications. Where as in water marking embedding data is relevant to the cover media and is used authentication applications. In this paper most image data hiding methods, the host image is distorted permanently due to data embedding and it cannot be recovered back to the original image after hidden data Extraction. But in some applications such as sharing the medical image, multimedia archive management, and remote sensing any distortion due to data embedding is intolerable and the original image is in high demand available. To this end, a solution called "reversible data hiding" (RDH) is proposed, in which without any distortion and loss the host image restored completely. Many RDH methods have been proposed in recent years, roughly the methods are classified as lossless compression method, Difference Expansion (DE) method, Histogram Modification (HM) based method and integer transform etc. all these method. Explains by keeping amount of loss and distortion less the high embedding capacity (EC) is achieved. For most image data hiding methods, the host image is distorted permanently and it cannot be restored from the marked content. But in some applications such as sharing medical image [1], and image trans-coding [2], any distortion because of embedding the data is impossible and the availability of the original image is in high demand. To this end, a solution called "reversible data hiding"(RDH) is proposed, in which after embedding data the host image can be restored completely [4]. In general, HS-based RDH is implemented by modifying host image's histogram of a certain dimension. It has two major advantages. On one hand, the maximum modification to pixel values can be controlled and thus the embedding distortion can be well limited. On the other hand, the location map used to record underflow/overflow locations is usually small in size especially for low ER case.

RDH is a hybrid method which combines various techniques to ensure the reversibility and feasible enough to ensure the lossless compressibility of natural images. This method can provide an embedding rate (ER) up to 0.5 bits per pixel (BPP) and it significantly out performs previous compression-based works. 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. Tian's DE algorithm [5] is an important work of RDH. In DE algorithm, the host image is divided into pixel pairs, and the difference value of two pixels in a pair is expanded to carry one data bit. Later on, Tian's work has been improved in many aspects. In [6], Hu et al. proposed a method by constructing a payload dependent location map. Another improvement of Hu et al.'s method is the work of Zhou and Au [7] by introducing a new capacity parameters determination strategy. Besides aforementioned works [1]-[4] many HS-based RDH algorithms have also been proposed so far.

ISSN 2348 – 7968

2. Proposed Method

2.1 Main Idea

In our method, we first divide the host image into non overlapping blocks such that each block contains *n* pixels (e.g., $n = 1$ for Ni *et al.*'s method, and $n = 2$ for Lee *etal.*'s method).
Then, an *n*-dimensional histogram is generated by counting the frequency of the pixel-value-array sized *n* of each divided block. Finally, data embedding is implemented by modifying the resulting *n*-dimensional histogram. Notice that the pixel-value-array is an element of Zn, we then need to divide Zn into two disjointed sets, one set is used to carry hidden data by expansion embedding, and the other set is simply shifted to create vacant spaces to ensure the reversibility. We now present our new approach.
Let *S* and *T* be a partition of Zn: $S \cup T = Zn$ and $S \cap T = \emptyset$.
Suppose that three functions $g: T \rightarrow Zn$, $f0 : S \rightarrow Zn$ and $f1: S \rightarrow Zn$ satisfy the following conditions:
C1: The functions *g*, *f*0 and *f*1 are injective.
C2: The sets *g (T)*, *f*0*(S)* and *f*1*(S)* are disjointed with each other.
Here, *g* is called "shifting function" and will be used to shift pixel values, *f*0 and *f*1 are called "embedding functions" and will be used to embed data. More specifically, each block With value $\mathbf{x} \in T$ will be shifted to $g(\mathbf{x})$, and the block with Value $\mathbf{x} \in S$ will be expanded to either $f0(\mathbf{x})$ or $f1(\mathbf{x})$ to Carry one data bit. The shifting and embedding functions will give a HS-based RDH algorithm where the reversibility can be guaranteed by the conditions *C1* and *C2*.
The underflow/overflow is an inevitable problem of RDH, i.e., for gray-scale image, the shifted and expanded values should be restricted in the range of [0, 255]. To deal with this, the above defined sets *T* and *S* need be further processed.
Let
$An = \{\mathbf{x} = (x1, . . . , xn) \in Zn : 0 \leq xi \leq 255\}$ (8)
be the set of all pixel-value-arrays of length *n* of gray-scale image. We define
$Ts = An \cap g{-1}(An)$ (9)
$Se = An \cap f{-1}$
$0 (An) \cap f{-1}$
$1 (An)$ (10)
$Tu,o = An \cap T - Ts$ (11)
$Su,o = An \cap S - Se.$ (12)
The sub-indices "*s*", "*e*" and "*u, o*" mean "shift", "embed "and "underflow/overflow", respectively. Obviously, the four sets *Ts* , *Se*, *Tu,o* and *Su,o* are disjointed with each other and constitute a partition of *An*, i.e.,
$An = Ts \cup Se \cup Tu,o \cup Su,o.$ (13)
Moreover, the sets *g(Ts)*, *f*0*(Se)* and *f*1*(Se)* are contained in *An* and the condition *C2* ensures that they are also disjointed.
By definitions (9)-(12), each block with value $\mathbf{x} \in Ts$ will be shifted, each block with value $\mathbf{x} \in Se$ will be expanded to carry one data bit, and the block with value $\mathbf{x} \in Tu,o \cup Su,o$ will remain unchanged since it cannot be shifted or expanded due to underflow/overflow.

2.2 Data Embedding

First divide the image host into the non-overlapping blocks. Then divide it into three parts as I1,I2,I3. Further with the help of shifting and embedding function embedded the data intoI1 and I3.then by Lsb replacement embedded the location map in I1 and then intoI2. The I1 part is double embedded for embedding first data.
The embedding procedure can be done by following steps
Step 1: Divide the host image into k non-overlapping blocks {X1,…,Xk } such that each Xi contains n pixels. Assume the value of Xi is xi ∈ An.
Step 2: Define the location map LM: LM(i) = 0 if xi ∈Ts∪ Se, and LM(i) = 1 if xi ∈ T Su . LM is a binary sequence of length k. Denote k1 = , where is the ceiling function. Take = {i : LM(i) = 1} which is the amount of underflow/overflow blocks. Then define a binary sequence LMc of length = (+ 1)k to record all underflow/overflow locations.
Step 3: Divide the k blocks into three parts to get I1, I2and I3.
a) I1 = {X1,…, Xk1} with k1 = .
b) I2 = {Xk1+1,…, Xk1+k2} such that it contains exactly embeddable blocks
c) I3 = {Xk1+k2+1,…,Xk } is the set of rest blocks.
Step 4: Embed the hidden data into I1 and I3, i.e., for any I ∈ {1,…, k1, k1 + k2 + 1,…, k}.
a) If xi ∈Ts , replace the value of Xi by g(xi).
b) I2 = {Xk1+1,…, Xk1+k2} such that it contains exactly embeddable blocks
c) I3 = {Xk1+k2+1,…,Xk } is the set of rest blocks.
Step 4: Embed the hidden data into I1 and I3, i.e., for any I ∈ {1,…, k1, k1 + k2 + 1,…, k}.
a) If xi ∈Ts , replace the value of Xi by g(xi).
b) If xi ∈ Se, replace the value of Xi by fm(xi), where m ∈ {0, 1} is the data bit to be embedded.
c) If xi ∈ T Su, do nothing with Xi .
Step 5: Record LSBs of the first pixels of I1 to get a binary sequence SLSB .Then replace these LSBs by the sequence LMc defined in Step 2.
Step 6: Embed the sequence SLSB into I2 in the same way as in step4

2.3 Data Extraction

The data extraction procedure has following steps.
Step 1: Divide the marked image into k non-overlapping blocks {Y1,…,Yk }.Assume the value of Yi is yi ∈ An.
Step 2: Firstly, determine the amount of problematic locations *l*, by reading LSBs of the first k1 = pixels. Secondly, read LSBs of the first = (+ 1)k pixels to get the sequence LMc. Then we can get the location map LM. Finally, with k1 = ,

LM, and by identifying embeddable blocks, we can obtain the same partition as defined in Step 3 of data embedding.
Step 3: Extract data from I2 and recover original pixel values of I2, i.e., for any i ∈ { k1 + 1,…, k1 + k2}.
a) If LM(i) = 0 and yi∈ g(Ts), the original pixel value is g−1(yi) and there is no embedded data.
b) If LM(i) = 0 and yi ∈ fm(Se) holds for a certain m ∈ {0, 1}, the original pixel value is fm −1 (yi) and the embedded data bit is m.
c) If LM(i) = 1, the original pixel value is yi itself and there is no embedded data.
The sequence SLSB defined in Step 5 of data embedding is extracted in this step.
Step 4: Replace LSBs of the first pixels of I1 by SLSB.
Step 5: Extract the embedded hidden data and recover original pixel values in I1 ∪ I3 in the same way as Step 3.
Finally, the hidden data is extracted and the original image is restored.

3. Result

Fig1: LENA IMAGE

Fig2: FISHING BOAT IMAGE

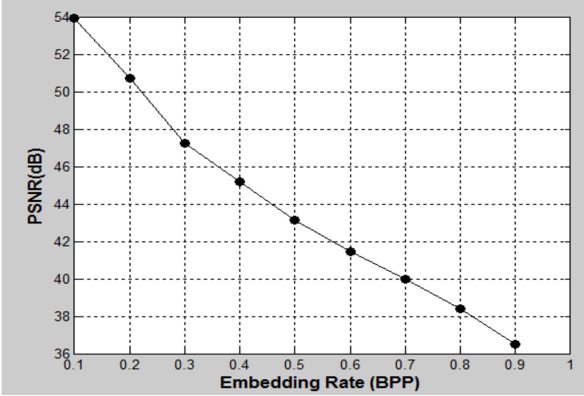

FIG 3: Performance char. For LENA image

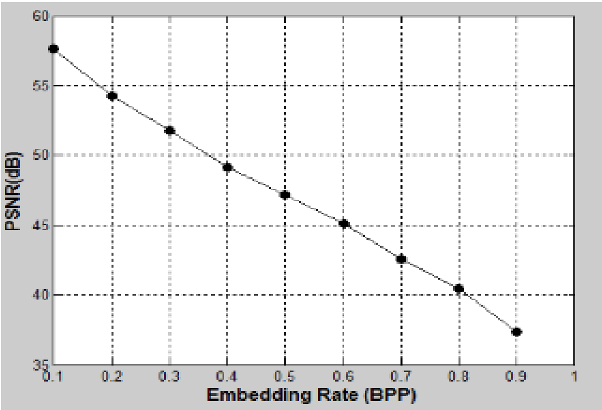

FIG4**:** Performance char. For FISHINGBOAT image

4. Comparison Table

Comparison of PSNR (IN DB) between proposed method and existing method For an ER= 0.5BPP

Images	HU et al. [7]	Li et al. [6]	Proposed method
Lena	40.73	42.37	43.1227
Fishing boat	36.45	37.82	38.5783

From the above table the proposed method for an embedding rate of 0.5BPP, improves the state-of-the art works by at least 1 DB in average.

ISSN 2348 – 7968

5. Conclusion

In this paper by analysing existing algorithm a general framework to construct histogram shifting based reversible data hiding (RDH) is proposed. Accordingly to obtain a reversible data hiding (RDH) algorithm. The shifting and embedding function are just to be defined.in advanced by integrating our work with PEE and pixel selection technique a novel RDH algorithm is introduced. Our work is facilitated by designing RDH.

References

[1] A.G. Coatrieux, C. L. Guillou, J. M. Cauvin, and C. Roux, Reversible watermarking for knowledge digest embedding and reliability control in medical images, IEEE Trans. Inf. Technol. Biomed., vol. 13, no. 2, pp. 158–165, Mar. 2009.

[2] B.H.-T. Wu and J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, Signal Process., vol. 92,no. 12, pp. 3000–3009, Dec. 2012.

[3] C.J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896,Aug.2003

[4] D.J. Zhou and O. C. Au, "Determining the capacity parameters in PEE based reversible image watermarking, IEEE Signal Process. Lett.,vol. 19, no. 5, pp. 287–290, May 2012.

[5] E.R. Li, O. C. Au, C. K. M. Yuk, S. Yip, and T. Chan, Enhanced image trans-coding using reversible data hiding, in Proc. IEEE Int. Symp.Circuits Syst., May 2007, pp. 1273–1276.

[6] F.Xiaolong Li, Bin Li, Bin Yang, and TieyongZeng General framework to histogram-shifting-based reversible data hiding.IEEE transaction on image processing vol 22 no 6 june 2013

[7] G.Y. Hu, H. K. Lee, and J. Li, DE-based reversible data hiding with improved overflow location map, IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 2, pp. 250–260, Feb. 2009