Information Hiding and Recovery using Reversible Embedding

A. S. Sonawane
Student of Vishwakarma Institute of Technology, Pune.

M. L. Dhore
Vishwakarma Institute of Technology, Pune, India.

S. N. Mali
Imperial College of Engineering and Research, Pune, India.

ABSTRACT

The main objective of this work is to develop data embedding technique that not only embeds secret message, in the form of binary bit stream, to the host image without any auxiliary information or location map, but also extracts that embedded secret message at decoder and restores the original content of host image, which are manipulated after the embedding at encoder. This technique carried out in two phases, embedding and extraction, using min-max approach. This technique reduces extra overhead to embed extra information other than secret message.

Keywords

Reversible data embedding, non-overlapped block division, location map free embedding, min-max approach.

1. INTRODUCTION

Data embedding technique have recently become information carrier in these multimedia days. This embedding technique not only embeds the secret message as hidden data into host image but also extracts it and restores the original content of host image. At embedding phase some distortion get introduced in embedded image which then completely removed in restoration phase at decoder. Thus to carry copyright information, authentication, ownership logo or text of any multimedia content along with that intellectual property itself is very easy and efficient.

Data embedding technique hides secret message into host image, hence in embedding phase some distortion introduce into host image. As this technique is '*reversible*' that distortions are completely removed at decoder in extraction phase. Current data hiding techniques can be classified into two groups: spatial domain and transform domain. Transform domain techniques embeds message by modulating the coefficients of transform domain, such as discrete Fourier transform (DFT), discrete cosine transform (DCT) and discrete wavelet transform (DWT) [1]. Spatial domain techniques, such as least significant bit insertion (LSB insertion) [2], vector quantization (VQ) [3], patchwork [4] and texture block, embeds message by directly modifying the pixel values of the target images. The advantage of spatial domain methods is the lower computational complexity and higher embedding capacity. Thus the reversible data embedding techniques are classified in between the cases where, some technique needed extra data as auxiliary information to embed along with secret message for restoration at decoder; and in other scenario it embeds only secret message into host image. Our technique falls into latter case. As method embeds only secret message without any location map, if embedded image undergoes through any attack such as rotation, wrapping then the extraction and recovery not possible at all. This technique is fragile.

1.1 Existing Methods

The reversible data embedding algorithms are developed from time it was suggested by its pioneers Fridrich et al., J Tian, Ni et al. First reversible data embedding scheme was proposed by Barton patent [5], in 1994; in his scheme the bits to be overlaid will be compressed and added to bit string, which will be embedded into data block. Macq [6] extends the method of Bender, et al. for reversible embedding but encounters overflow and underflow problem. Honsinger, et al. [7] mitigate the overflow and underflow problem by using modulo arithmetic operation. Fridrich, et al. [8] proposed reversible data embedding technique for authentication purpose but embedding capacity is very low. De Vleeschouwer [9] proposed the circular interpretation of bijective transformation and overcomes underflow/overflow problem. Tian [10] was proposed reversible data embedding with high embedding capacity and visual quality. Alattar [11] embeds 'n-1' bits into group of 'n' cover pixel. Kamstra and Heijman [12] also improved Tian's method in term of visual quality of low embedding capacity. Chang and Lu [13] exploit Tian's method to achieve average embedding capacity 0.92bpp and average PSNR 36.54dB, for one layer embedding. Thodi and Rodriques [14] performs prediction error method for reversible embedding purpose. Kim et al. [15] improved Tian's method by simplifying the location map. Lou et. al. [16] improved DE method by proposing multilayer data hiding scheme. He reduces the difference value of two neighbouring cover pixel to enhance visual quality. As a whole, the problems with aforementioned methods are either overflow/underflow or size of location map i.e. auxiliary information package.

1.2 Proposed Method

The method which requires extra data such as auxiliary information or location map needed to embed into host image along with hidden data, [10,11]-[14,15] requires extra overhead to embed that extra bits and also focuses on to make size of the extra data must be as small as possible. To overcome the problem, reversible data embedding method based on spatial domain that uses min-max algorithm to embed secrete message into original image is developed. Though it uses only one parameter, binary bit stream of secrete message along with length of secret message, at decoder it recover not only secrete message but also restores original pixel values.

The proposed min-max algorithm is described in Section 2. Section 3 presents simulation results. The conclusion is given in Section 4.

2. SYSTEM DESIGN AND IMPLEMENTATION

A N×M size Gray-scale image whose each pixel is represented by 8 bits is used as host image and text file carries

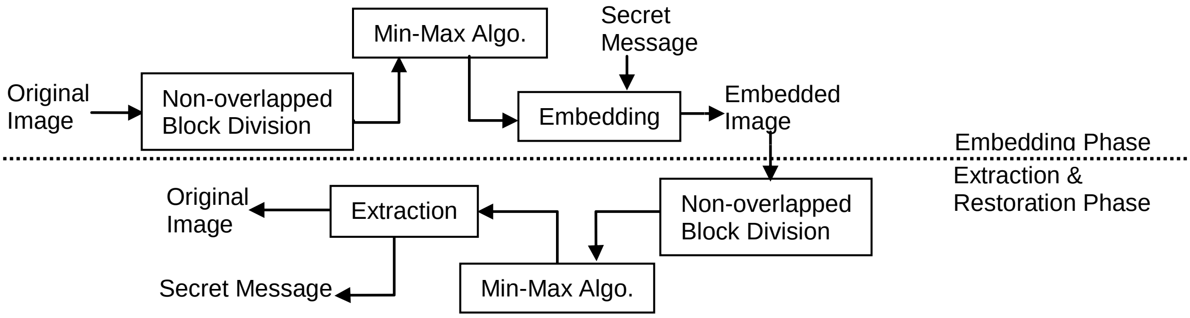


Fig 1: Block diagram of proposed method

secret information is used as secret message. The proposed method converts text file into continuous binary bit stream and then embeds it into non-overlapped, identical blocks of a host image using min-max approach.

The design centers on min-max approach. The design of reversible embedding process carried out in two phases Embedding and Extracting. Embedding phase carried out by encoder while extracting phase carried out by decoder.

The block diagram of the proposed method is shown in Fig 1. First host image divided into identical, non-overlapped blocks, these blocks may be different size like 4×4, 2×2, or 1×2. Min and max element of each block is obtained using min-max algorithm. According to that calculated min and max element secret message is embedded into host image.

At Extractor, using reverse process not only secret message is extracted but also host image is restored from embedded image.

2.1 Min-max Approach

This approach applied on equal size and non overlapping blocks of original image. Min-max approach mainly concern to find out minimum and maximum element among the block. According to calculated minimum and maximum element the current block is falls into three categories. These three categories are mentioned below:

1. Case 1:- $min \le 127$ and $max \le 127$
2. Case 2:- $min \le 127$ and $max \ge 127$
3. Case 3:- $min \ge 128$ and $max \ge 128$

According to number of blocks counted by these cases the image either falls into minimum case or maximum case. The embedding and extracting phase carried out in each case is described below:

2.1.1 *Minimum Case*

During embedding the blocks which falls under case 1 and case 2 are used. Secret message binary bits are sequentially embedded in those blocks, using embedding algorithm. During extracting, the case, min is less than equal to 127 and max is less than equal to 127 or max is greater than equal to 128, detect the blocks where secret message is embedded and performed extracting algorithm for extraction of original image and secret message on those blocks.

2.1.2 *Maximum Case*

During embedding the blocks which falls under case 2 and case 3 are used. Secret message binary bits are sequentially embedded in those blocks, using embedding algorithm. During extracting, the case, max is greater than equal to 128 and min is less than 128 or min is greater than equal to 128, detect the blocks where secret message is embedded and performed extracting algorithm, for extraction of original image and secret message, on those blocks.

2.2 Data Embedding

This phase embeds secret message binary nit stream into host image as follows:

```
1.  accept Input Image (Height×Width)
2.  store image size(N×M)
3.  Divide input image into equal size, non overlapping blocks
4.  for each block find min and max element
    if ($min \le 127$ and $max \le 127$)
        increment counter by one(c1++)
    else if ($min \le 127$ and $max \ge 127$)
        increment counter by one(c2++)
    else if ($min \ge 127$ and $max \ge 127$)
        increment counter by one(c3++)
5.  if($c1 \ge c2$)
    select first block
    subtract min element from each remaining element
    embed the binary bit sequentially to this block
    if bit=='1'
        (embedded value of pixel = current value*2+1)
    if bit=='0'
        (embedded value of pixel = current value*2)
    repeat for all blocks till last bit encounters
6.  else if($c2 \le c1$)
    select first block
    subtract each element from max element
    embedd the binary bit sequentially to this block
    if bit=='1'
        (embedded value of pixel = current value*2+1)
    if bit=='0'
        (embedded value of pixel = current value*2)
    repeat for all blocks till last bit encounters
7.  end
```

The flow chart of the encoding algorithm for the minimum and maximum case is as shown in Fig 2.

2.3 Data Extraction and Recovery

The following algorithm is applied on embedded image for extraction of hidden secret message and restoration of original host image. Extracted secret message is in binary bit string. Host image is restored and is identical to original one.

```
1.  Accept embedded Image (Height×Width)
2.  store image size(N×M)
3.  Divide input image into equal size, non overlapping blocks
4.  for each block find min and max element
5.  if ($min \le 127$ and $max \le 127$)
        increment counter by one(c1++)
    else if ($min \le 127$ and $max \ge 127$)
```

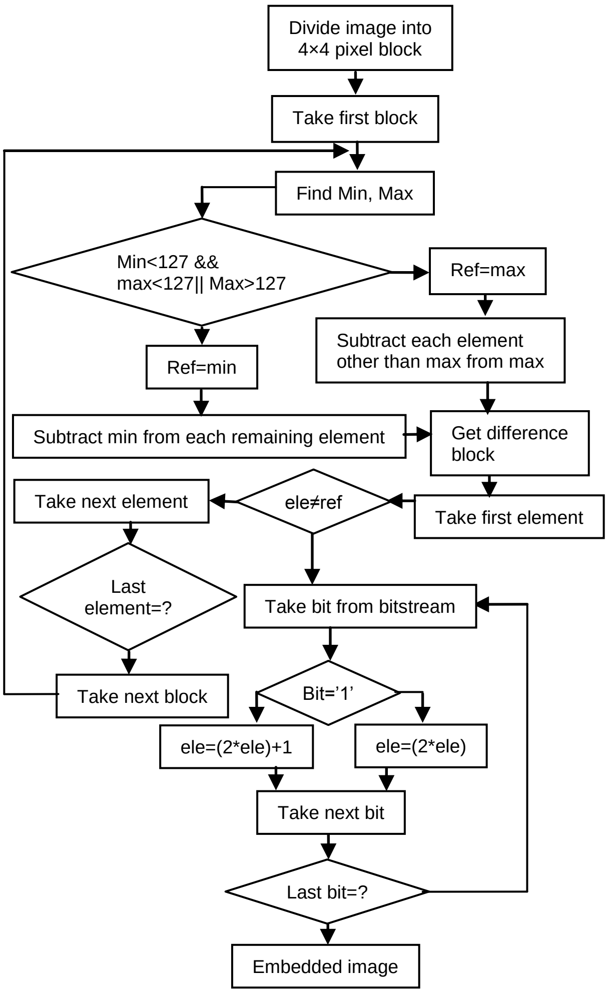


Fig 2: Flow chart for data embedding

```
          increment counter by one(c2++)
     else if (min≥127 and max≥127)
          increment counter by one(c3++)
6.   if(c1≥c2)
     select first block
     subtract min element from each remaining element
     if current pixel value is odd then bit '1' extracted
          original pixel = min+( current pixel value -1)/2;
     if current pixel value is even then bit '0' extracted
          original pixel = min+( current pixel value /2);
     repeat for all blocks till all bit extracted
7.   else if(c2≤c1)
     select first block
     subtract each element from max element
     if current pixel value is odd then bit '1' extracted
          original pixel = max-( current pixel value -1)/2;
     if element is even then bit '0' extracted
          original pixel = max-( current pixel value /2);
     repeat for all blocks till all bit extracted
8.   end
```

2.4 Illustration of proposed method using example

Fig.3. shows an example of bits being embedded using min-max algorithm. In fig. 3(1), original image divided into 4×4 size block and secret message bit stream is "110110111101101". For this particular block minimum value, 'min' is 76 and maximum value, 'max' is 90. As both are less than 127 then minimum case embedding is used. Fig. 3(2) shows the difference block and fig. 3(4) is block getting after embedding process. During extraction phase fig. 3(4) is taken as an input and fig. 3(3) is difference block.

76	80	86	90
78	81	85	88
81	82	84	87
84	83	83	87

(1) Original pixel value

76	04	10	14
02	05	09	12
05	06	08	11
08	07	07	11

(2) Difference block

110110111101101

76	85	97	104
81	87	94	101
87	89	93	98
93	91	90	99

(4) Embedded pixel value

76	09	21	28
05	11	18	25
11	13	17	22
17	15	14	23

(3)Modified diff. block

Fig 3: Illustration of proposed method

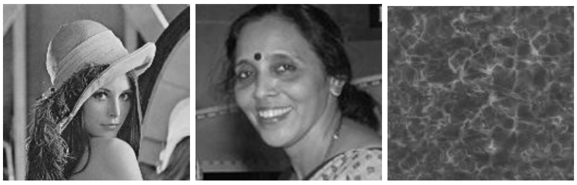
3. Experimentation and Results

All experiments are performed by embedding and extracting an actual bit stream. Binary bit stream is obtained by converting actual text file into binary string. The algorithm implemented on java platform and applies on different 8 bit grayscale images they are of size 512×512.

Implementation of three algorithms is on the basis of division of original image into different non overlapped, equal size block.

- 4×4 block
- 2×2 block
- 1×2 block

Experimentation carried out on following test images shown in fig. 4 downloaded from US-SIPI. Analyses of results are done using embedding capacity (bpp) and visual quality in term of PSNR.



(a) (b) (c)

Fig 4: (a) 'Lena' test image, (b) 'face' test image, (c) 'texture' test image.

Embedding capacity for particular image is nothing but the maximum number of bits that are able to embed into an image. Embedding capacity for different image is also different. Embedding capacity for each test image explored in graph shown in fig. 5.

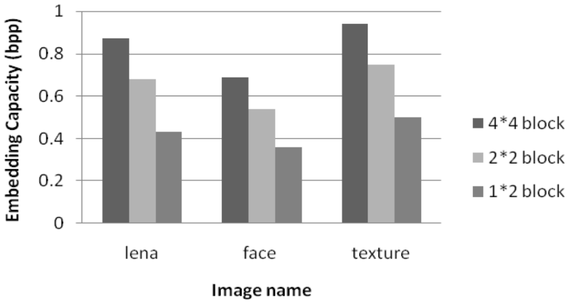


Fig 5: Embedding capacity of Lena, face and texture image

Table 1 secret message size vs. PSNR of embedded images

Image name	Payload (Byte)	16384	13107	11469	9830	8192	6553	4915	3277	2949	2622	2294	1966
	Capacity (bpp)	0.5	0.4	0.35	0.3	0.25	0.2	0.15	0.1	0.09	0.08	0.07	0.06
Lena	4×4 block	24.66	25.18	25.77	26.81	28.45	30.84	33.81	37.38	37.48	37.51	37.55	37.91
	2×2 block	29.72	30.58	30.76	31.16	32.3	34.05	37.3	41.2	43.24	44.23	44.42	44.53
	1×2 block	NA	34.9	35.12	35.57	36.04	36.72	38.33	43	44.18	45.93	47.09	49.88
Face	4×4 block	33.45	34.2	34.9	35.68	36.67	38.54	41.26	42.66	42.92	43.14	43.38	43.98
	2×2 block	39.414	40.94	41.54	42.04	43.02	44.553	46.58	49.36	49.71	50.05	50.44	50.84
	1×2 block	NA	41.46	41.5	41.53	41.98	43.73	45.67	57.96	58.53	59.1	59.77	60.52
Texture	4×4 block	31.97	33.13	33.81	34.62	35.37	36.22	37.59	39.89	40.47	40.95	41.52	42.11
	2×2 block	38.32	39.2	39.92	40.71	41.61	42.55	43.72	45.89	46.42	47.06	47.72	48.37
	1×2 block	40.16	41.05	41.53	42.11	43	44.17	45.55	47.37	47.9	48.49	49.22	50.09

The graph depicts that as block size decreases, embedding capacity also decreases. Less the Block size, less number of embeddable pixels are available, and so as less data will hide into an image.

Table 1 shows embedded secret message size (byte), its corresponding bit rate, and peak signal to noise ratio (PSNR) of embedded images are listed for different block size embedding.

The results shows that, more redundant part is present in image more numbers of bits are embedded into an image and corresponding distortion introduce in an image is less.

In the literature, they needs location map for extraction of secret message and for restoration of original image. This location map is embedded along with secret message into an image. It requires extra overhead to embed location map and to compress location map because sometimes size of location map is larger than that of actual secret message.

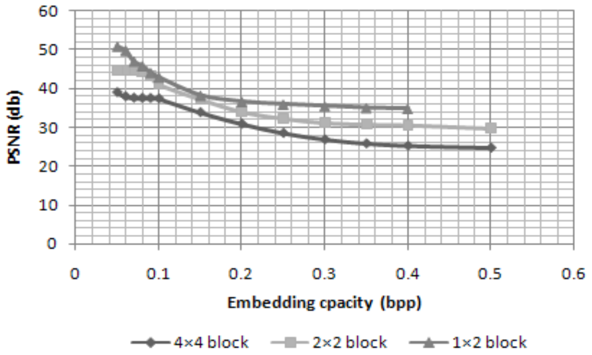


Fig 6: Embedding capacity vs. distortion for 'Lena' image.

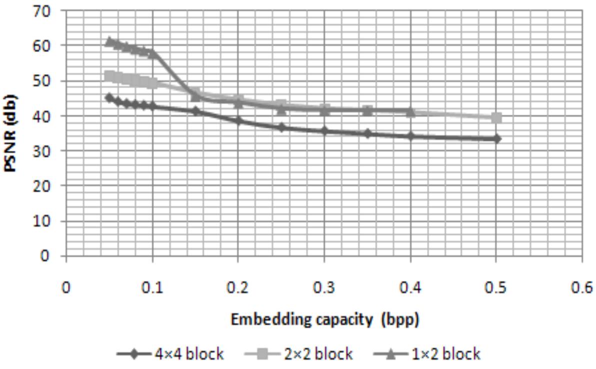


Fig 7: Embedding capacity vs. distortion for 'face' image.

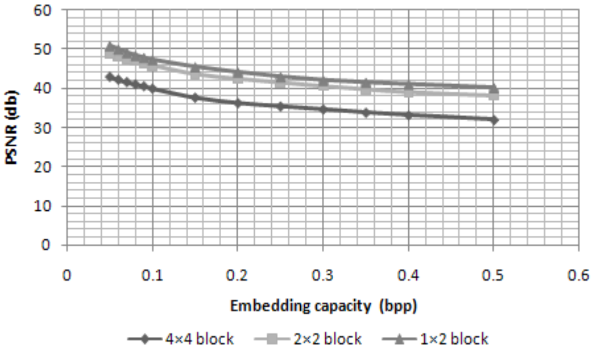


Fig 8: Embedding capacity vs. distortion for 'texture' image.

This proposed method hence successfully eliminates the need of location map by using min-max approach for embedding and extraction. Proposed method embeds only secret message as pure payload and retrieve it at decoder without any location map or auxiliary information package. Comparison results of 4×4 block, 2×2 block, and 1×2 block embedding algorithms implemented by ours. Fig. 6-8 shows that block size is directly proportional to distortion introduce into image after embedding.

4. CONCLUSION

An effective lossless data hiding scheme that embeds data in spatial domain was proposed. Reversible data embedding links two sets of data, one set is of secret message and another set of host image, in such way that the host image is losslessly recovered after extraction of secret message at decoder. From application point of view reversible data embedding method can be used as an information carrier. The difference between an embedded image and host image is undetectable by human eyes. The proposed method consist min-max approach for hiding secret message into host image and retrieve it again back at decoder. Experiment indicates that not only secret message was successfully extracted but also a host image losslessly restored. Moreover, the resultant perceptual quality generated by proposed method is good.

The proposed method is fragile one, it can't withstand against any attack. We further explore the relationship between various attacks and embedding algorithm so that a secret message must be recovered even from manipulated or attacked embedded image.

5. REFERENCES

[1] J. Tian, "Wavelet-based reversible watermarking for authentication", Proceedings of SPIE on Security and Watermarking of Multimedia Contents IV vol. 4675, Jan. 2002, pp. 679-690.

[2] S. Walton, "Image authentication for a slippery new age", Dr. Dobb's Journal 20 (4), April 1995, pp. 18-26.

[3] T.-S. Chen, C.-C. Chang and M.-S. Hwang, "Virtual image cryptosystem based upon vector quantization", IEEE Transactions on Image Processing 7 (10), Oct. 1998, pp. 1485-1488.

[4] J. Fridrich, M. Goljan and R. Du, "Lossless data embedding - new paradigm in digital watermarking", EURASIP Journal of Applied Signal Processing 2002 (2), Feb. 2002, pp. 185-196.

[5] J.M. Barton, "Method and Apparatus for Embedding Authentication Information Within Digital Data," U.S. Patent 5, 1997, pp. 646-997.

[6] B. Macq, "Lossless multiresolution transform for image authenticating watermarking," presented at the European Signal Processing Conf., Tampere, Finland, Sep. 2000.

[7] C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," U.S. Patent 6 278 791, Aug. 21, 2001.

[8] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in Proc. SPIE Security Watermarking Multimedia Contents, San Jose, CA, Jan.2001, pp. 197–208.

[9] C. D. Vleeschouwer, J.-F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," IEEE Trans. Multimedia, vol. 5, no. 1, pp. 97–105, Mar. 2003.

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

[11] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," IEEE Trans. Image Process., vol.13, no. 8, pp. 1147–1156, Aug. 2004.

[12] L. Kamstra and H. J. A. M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," IEEE Trans. Image Process., vol. 14, no. 12, pp. 2082–2090, Dec. 2005.

[13] C. C. Chang, W.L. Tai, and C.C. Lin, "A Reversible Data Hiding Scheme Based on Side Match Vector Quantization," IEEE Transactions on Circuits and Systems for Video Technology, Vol.16, No.10, pp.1301-1308, 2006.

[14] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," IEEE Trans. Image Process., vol. 16, no.3, pp. 721–730, Mar. 2007.

[15] HJ Kim, V Sachnev, YQ Shi, J Nam, HG Choo, "A novel difference expansion transform for reversible data hiding," IEEE Trans. Information forensicc and security., vol. 3, no.3, pp. 456–465, Jun. 2008.

[16] Lou, D.C., Hu, M.C., Liu, J.L., "Multiple layer data hiding scheme for medical images," Computer Standards and Interfaces 31 (2), 329–335, 2009.