REVERSIBLE IMAGE HIDING USING SHIFTING OF MEAN VALUES WITH INTERLACED GROUPING AND SIDE GROUPING

Chef Chu
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan
mr936309@cs.nthu.edu.tw

Long-Wen Chang
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan
lchang@cs.nthu.edu.tw

ABSTRACT

A new reversible data hiding is proposed. We group the pixels of block in the image into two groups with two methods called side grouping and interlaced grouping. Then, we calculate the mean values of the pixel values of the two groups, respectively. By using the relation between the difference of two mean values and the threshold value we embed secret data in the block. Because reversible data hiding may work in lossless environment, we use the multiple data hiding skill to embed secret data. By taking advantage of this skill, our algorithm could extract the secret data correctly and recover the cover image from the stego-image in a lossless environment. Also, we can extract secret data correctly in a low noisy environment due to JPEG compression.

Categories and Subject Descriptors

I.4.0 [**Image Processing and computer vision**]: **General**

General Terms:

Security

Keywords

reversible data hiding, stego-image

1. INTRODUCTION

Reversible data hiding [1] embeds a secret message into an image, and generate a stego-image. After extracting the secret message from the stego-image, it is possible to recover the original image. There are many applications in military, medical, and forensic images for reversible data hiding.

Zhicheng Ni et al. [2] proposed a lossless data hiding technique based on histogram modification. This algorithm uses the zero or minimum point, and the maximum point of the image histogram to embed secret data. The algorithm is very simple and the stego-image could get very high visual quality. Jun Tian proposed a reversible data hiding using difference expansion of pixel values [3, 4]. After doubling the difference of two pixels, the least significant bit of the difference could embed a secret bit in it. This algorithm could implement with repetition so that the capacity is quite high. Vleeschouwer proposed an interesting reversible data hiding technique which uses the "center of mass" of the circle mapped from the histogram [5].

In this paper, we propose a new method by shifting the mean values of two groups of pixels values in a block of an image. The method can do reversible data hiding for lossless environment and also extract correct secrete data in a low noisy environment.

2. THE PROPOSED ALGORITHM

An image is divided into a sequence of blocks. The block size can be 2×2, 2×4, 4×2, 4×4, etc. If a block that is suitable to embed a data bit it is called embeddable; otherwise it is not embeddable. Each block is divided into two groups. The proposed reversible data hiding technique uses the concept of the mean value of a group in an embeddable block. The number of the embeddable blocks increases as the block size decreases. It is possible that the value of a pixel in the block may become overflow or underflow if it is shifted by a certain value to embed a data bit. We adopt the histogram modification to prevent overflow and underflow. The coordinate of the modified pixels for histogram modification is the side information that is needed for the receiver to recover the image. Because the number of the pixels needed to be modified is usually quite small, we can embed the side information into the image before embedding the secret data. Like Vleeschouwer's technique [5], our technique could resist against the JPEG compression by using a multiple bit hiding strategy.

The flowchart of embedding and extracting procedure are depicted in Figure 1 and Figure 2, respectively. They both include histogram modification. In the embedding process, we need to record the coordinates of the modified pixels as side information after histogram modification, and embed this side information together with the secret data into the pre-processed image. In the extracting phase, we can use this side information and histogram modification to recover the original one.

We also need to design a data format of secret data for extraction. The data format of the secret data is shown in Figure 3. The first bit b_1 means whether the image is pre-processed by

Mobimedia 2008 July 7-9, 2008, Oulu, Finland.
 ISBN 978-963-9799-25-7/08/07
DOI 10.4108/ICST.MOBIMEDIA2008. 3918

histogram modification or not. If b_1 is 0, there is no pixel needed to be shifted, the other bits from $b_2 \sim b_N$ are secret data, where N is the number of the total bits of side information. If b_1 is 1, the image has some modified pixels. $b_2 \sim b_7$ record how many the modified pixels are, and $b_8 \sim b_{(16n+7)}$ are the coordinates of the modified pixels, where n is the decimal value of $b_2 \sim b_7$. The rest are the secret data we want to embed.

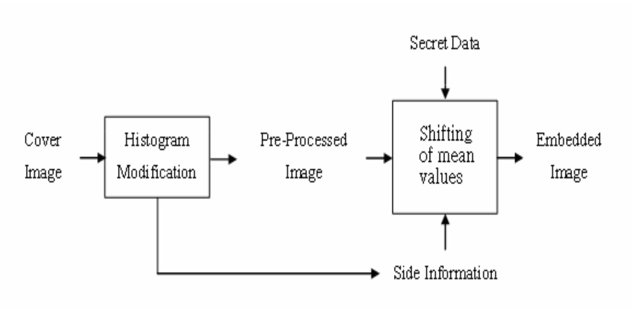


Figure 1: The embedding procedure

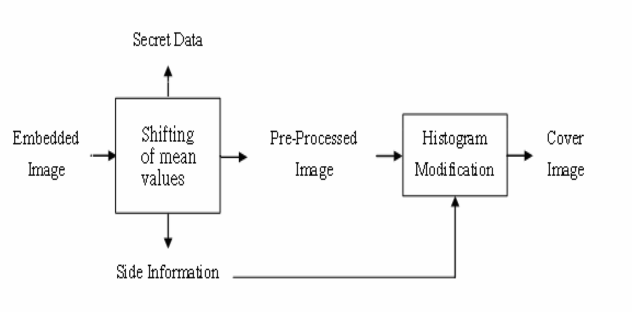


Figure 2: The extracting procedure

b_1	$b_2 \sim b_7$	$b_8 \sim b_{(16n+7)}$	$b_{(16n+8)} \sim b_N$

Figure 3: The data format of embedded data

2.1 Histogram Modification

To prevent overflow and underflow, we use histogram modification before the shifting of mean value of a group in the coding block in the embedding phase.

For the pixels whose gray values are greater than 255-*T*, the gray values of these pixels need to be decrease by *T* so that the gray values become equal to or less than 255-*T*. For the pixels whose gray values less than *T*, the gray values need to be increased by *T* so that the gray values become equal to or greater than *T*. The equation of histogram modification for a pixel f(x,y) in an image is:

$$f(x,y)=\begin{cases} f(x,y)-T & ,\text{if } f(x,y)>255-T \\ f(x,y)+T & ,\text{if } f(x,y)<T \\ f(x,y) & ,\text{otherwise} \end{cases}$$

For a given image, we embed 1 bit in each embeddable block of the image. The block size could be 2×2, 2×4, 4×2, or 4×4. The block is divided into two groups, G1 and G2, as shown in Figure 4. The grouping methods are called side grouping and interlaced grouping.

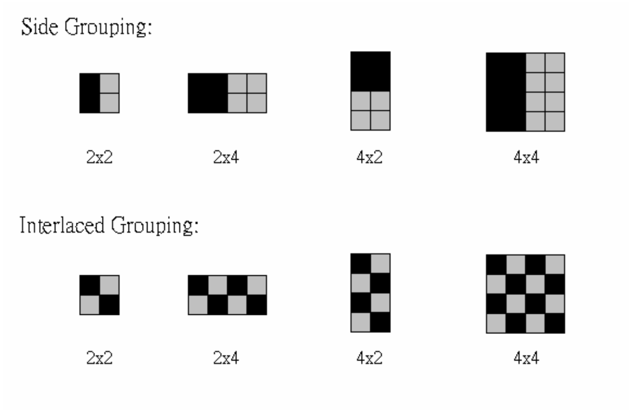


Figure 4: 2 kinds of pixel grouping for 4 kinds of blocks

To explain the proposed reversible data hiding algorithm, the following notations are used.

G_1: The set of black pixels in a block in Figure 4

G_2: The set of gray pixels in a block in Figure 4

P_1: a pixel in G_1

P_2: a pixel in G_2

$F(P_1)$ is the pixel value of P_1

$F(P_2)$ is the pixel value of P_2

M_1: The mean of pixel values in G_1

M_2: The mean of pixel values in G_2

T: A threshold value

S: A bit we want to embed or extract

D: The difference of the two mean values, M_1 and M_2,

$D = M_1 - M_2$

After histogram modification, since $0 < M_1 < (255-T)$ and $0 < M_2 < (255-T)$ we have $(T-255) < D < (255-T)$. Figure 5 shows the range of D after histogram modification. The black line means the difference value D before our watermark embedding algorithm.

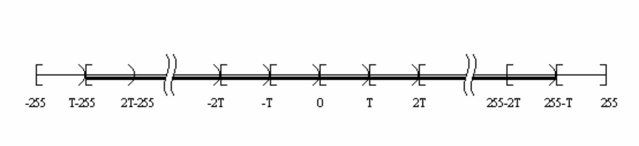


Figure 5 The range of difference value D of a block after histogram modification

When $-T < D < T$, the block is embeddable; otherwise the block is not embeddable. The algorithm that embeds a bit is written as below. The ranges of D when embedding and extracting secret bits are depicted in Figure 6 and 7, respectively.

Algorithm that embed a bit:

If ($0 \leqq D < T$)

  If ($S = 0$)

    $F(P_1) - T$, for all pixel P_1 in G_1

  Else ($S = 1$)

    $F(P_1) + T$, for all pixel P_1 in G_1

If ($-T \leqq D < 0$)

  If ($S = 0$)

    $F(P_1) - T$, for all pixel P_1 in G_1

  Else ($S = 1$)

    $F(P_1) + T$, for all pixel P_1 in G_1

If ($D \geqq T$)

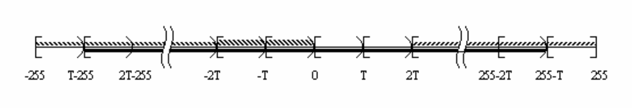
  S is not embedded in the block,, and

  $F(P_1) + T$, for all pixel P_1 in G_1

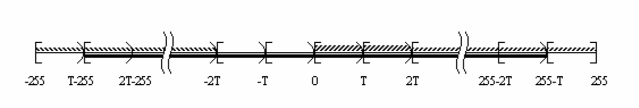
If ($D < -T$)

  S is not embedded in the block,, and

  $F(P_1) - T$, for all pixel P_1 in G_1



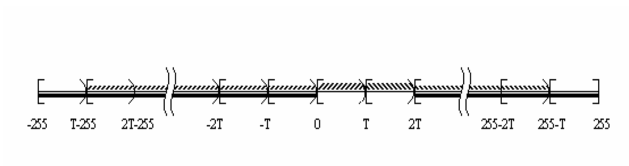
(a)



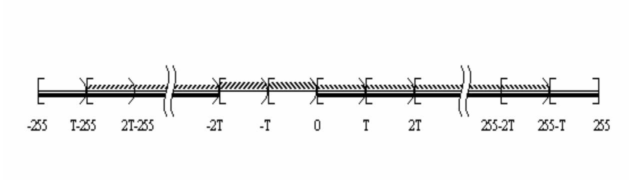
(b)

Figure 6 Shifting of D when (a) 0 is embedded (b) 1 is embedded

Figures 7(a) and 7(b) illustrate embedding 0 and 1, respectively. The black line means the difference value D before our algorithm, the slash '/' means the difference value after the mean value is added by T and the range of the backslash '\' means the difference D after the mean value is subtracted by T. Please also note that there are long slash area, short lash area, long backslash area and short backslash area. For example, in Figure 7(b), the long slash area is [0, T), in Figure 7(a), the short slash area is [2T, 255], the long backslash area is [-T, 0) and the short backslash area is [-255,-2T). The short slash area and the short backslash area mean that the pixel value whose mean value is in this area is shifted by T but is not embedded a secret bit, and the long slash area and long backslash area mean that the pixel whose mean value is in this area is shifted by T and is embedded a secret bit.



(a)



(b)

Figure 7 Shifting of D when the secret bit (a) 0 is extracted (b) 1 is extracted.

3. SIMULATION RESULTS

Two standard test images, Lena and Baboon, are used to be experimented by our proposed technique. We show our experimental images of 256×256 Lena in Figure 8 using the side grouping method with 2×2 block size and different thresholds. Secondly, we show the experimental results of 256×256 Baboon in Figure 9 using the interlaced grouping method with 2×2 block size and different thresholds. Table 1 shows PSNRs for various thresholds. Table 2 shows capacity and bitrate of Lena using interlaced grouping method with different thresholds and block sizes. Table 3 shows capacity and bitrate of Lena using the side grouping methods with different thresholds and block sizes. From these three tables, when the threshold becomes big, the embedding capacity becomes large. Table 4 shows the correct bits extracted from JPEG images with different quality factors using side grouping and 2×4, 4×4, and 4×8 block sizes. It shows that the correct bits extracted from JPEG images with different quality factors using side grouping and 2×4, 4×4, and 4×8 block sizes

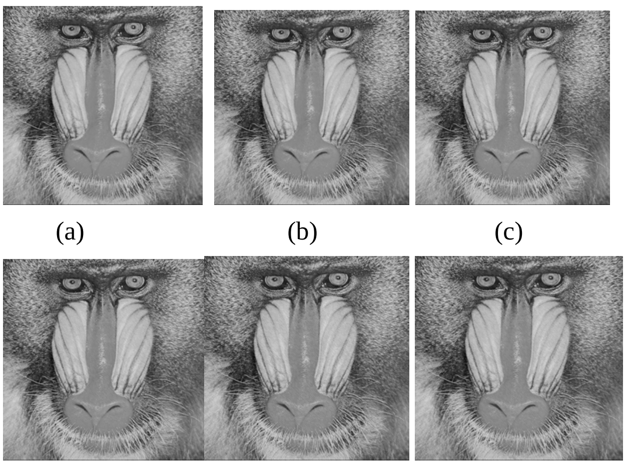
4. CONCLUSIONS

We proposed a reversible data hiding technique based on side grouping and interlaced grouping. Our technique uses the mean value other than the center of mass, in order to decrease the computational complexity. As to capacity, because the neighboring values usually are similar, the mean values could be very close with small block size. Small size increases the total number of embeddable blocks and leads to a high capacity. Our technique could get high PSNR values in the data hided images because only half of pixels in a block are changed other than total of pixels. Also, we can extract secret data correctly in a low noisy environment due to JPEG compression.

(a) (b) (c)

(d) (e) (f)

Figure 8:(a) Original 256x256 gray leve limage (b) T＝1 (c) T＝2(d) T＝3(e) T＝4(f) T=5



(a) (b) (c)

(d) (e) (f)

Figure 9: (a) Original 256x256 gray level image (b) T＝1 (c) T＝2(d) T＝3(e) T＝4(f) T＝5

Threshold	1	2	3	4	5
PSNR	51.1411	45.1205	41.5987	39.0999	37.1617

Table 1: PSNR of stego-image with different thresholds

Block Size \ Threshold		1	2	3	4	5
2×2	Capacity	4635	8172	10539	12064	12961
	Bitrate	0.0707	0.1247	0.1608	0.1841	0.1978
2×4	Capacity	3039	5021	6158	6684	6998
	Bitrate	0.0464	0.0766	0.0937	0.1020	0.1068
4×2	Capacity	3085	5051	6084	6684	7011
	Bitrate	0.0467	0.0771	0.0928	0.1020	0.0170
4×4	Capacity	1085	1959	2563	2981	3232
	Bitrate	0.0166	0.0298	0.0391	0.0455	0.0493

Table 2: Capacity and bitrate of Lena using interlaced grouping method with different thresholds and block sizes

Block Size \ Threshold		1	2	3	4	5
2×2	Capacity	3358	7051	9397	10962	12004
	Bitrate	0.0589	0.1076	0.1434	0.1673	0.1832
2×4	Capacity	1202	2267	3154	3780	4265
	Bitrate	0.0183	0.0346	0.0481	0.0577	0.0651
4×2	Capacity	1776	3180	4241	4956	5426
	Bitrate	0.0269	0.0485	0.0647	0.0756	0.0828
4×4	Capacity	345	670	1002	1280	1506
	Bitrate	0.0053	0.0102	0.0153	0.0195	0.0230

Table 3: Capacity and bitrate of Lena using side grouping methods with different thresholds and block sizes

QF / block size	90	80	70	60	50	40	30	25	20
2×4	100	100	100	97	87	81	79	67	59
4×2	100	100	100	96	95	84	62	54	46
4×4	100	100	100	100	100	98	88	83	64

Table 4: The correct bits extracted from JPEG images with different quality factors using side grouping and 2×4, 4×4, and 4×8 block sizes

5. REFERENCES

[1] Yun Q. Shi, Zhicheng Ni, Dekun Zou, Changyin Liang and Guorong Xuan, "Lossless data hiding: Fundamentals, Algorithms and Applications," Proceeding of ISCAS 2004, Vol. 2, pp.23-26.

[2] Zhicheng Ni, Yun Q. Shi, Nirwan Ansari and Wei Su, "Reversible Data Hiding," IEEE Transactions on Circuits and Systems for Video Technology, Vollume 16, March 2006, pp. 354-362.

[3] Jun Tian, "High Capacity Reversible Data Embedding and Content Authentication," Proceeding of ICASSP 2003, Vol. 3, pp. 517-520

[4] Jun Tian, "Reversible Data Embedding Using a Difference Expansion," IEEE Transactions on Circuits and Systems for video Technology, Vol.13, no.8, August 2003, pp. 890-896.

[5] Christophe De Vleeschouwer, Jean-François Delaigle, and Benoît Macq, "Circular Interpretation of Bijective Transformation in Lossless Watermarking for Media Asset Management," IEEE Transactions on Multimedia, Vol.5, no.1, March 2003, pp. 97-105.