International Journal of Applied Science and Engineering
2009. 7, 1: 53-60

A Modified Side Match Scheme for Image Steganography

Po-Yueh Chen*, Wei-En Wu

Department of Computer Science and Information Engineering,
National Changhua University of Education No. 2 Shi-Da Road, Changhua City, Taiwan 500

Abstract: In this paper, we proposed an image stegangraphy scheme to further improve the side match method proposed by Chang et al. in 2004. By hiding more information in the edge portions, image quality is improved while maintaining the same embedding capacity. This merit results from the fact that human eyes can rarely percept trivial differences in the edge regions. The embedding capacity is adjustable according to various demands of individual users. In addition to the improvement on image quality, the proposed approach provides respectable security as well.

Keywords: Stegangraphy; Data hiding; Side match; Security

1. Introduction

In a world teeming with digital equipments, the Internet serves as an essential medium for data transmission and sharing. However, since it is world-wide and public, some confidential data might be stolen, copied, modified, or destroyed by an unintended observer. Therefore, security problems become an essential issue for the Internet. Encryption is a well-known procedure for secure data transmission. The commonly used encryption schemes include RSA [1] and DES (Data Encryption Standard) [2]. Although these two methods provide remarkable security effects, they scramble the secret messages into some unreadable and unnatural forms as well. These unreadable and unnatural messages usually attract attention of some unintended observers. Hence, a new scheme, called “steganography” [3], arises to camouflage the secret messages with some ordinary media.

For example, we may select an image as the medium to hide the secret messages without perceptibly destroying the original image. The selected image is called the “cover-image”. The embedded secret data can be of any types, such as images, texts, voices, or just binary codes. The image with the secret messages embedded is called the “stego-image”. Since the stego-image looks almost the same as the original cover image, only the intended receivers are aware of the existence of hided messages. Hence, malicious hackers are kept away from the embedded messages. For higher security requirements, we can encrypt the message data before embedding them in a cover-image to provide further protection.

Applying images as the cover, data hiding schemes can be categorized into two types, the spatial domain methods [4] and the frequency domain ones [5]. Applying spatial domain approaches, the secret messages are embedded in the image pixels directly. They reserve the simplicity and promptness but less resulted security. Applying a frequency domain approach, the sender transforms the cover-image into some frequency domain

* Corresponding author; e-mail: pychen@cc.ncue.edu.tw

Accepted for Publication: February 13, 2009

coefficients before embedding secret messages in it. These schemes are hence more complicated and secure with a longer processing time compared to the spatial domain ones.

The simplest steganography is the Least Significant Bit (LSB) approach [6] which embeds secret messages in the LSBs of the original cover image pixels. It reserves the image quality and requires no complex operation. However, employing LSB method for steganography makes it convenient to extract the secrete messages for any user. In [7], a new steganography scheme, named “side match”, was proposed. It evaluates the correlation between neighboring pixels to determine whether a pixel is located in an edge area or a smooth area, and how many bits should be embedded in that pixel accordingly. Based on the same embedding capacity, the side match scheme improves both security and image quality. The reason is that the side match scheme embeds more bits in the edge regions where the human eyes are relatively not sensitive. However, the original side match method is conservative in embedding. For example, it discards the pixels where the stego-pixel value is out of range and hence limits the embedding capacity. In this paper, we propose some modifications to further improve this novel side match scheme.

The rest of the paper is organized as follows. In section 2, the modified side match method is described in details step by step. A numerical example is provided to further illustrate the steps. Section 3 demonstrates the experiment results for the proposed scheme. Analyses and discussions are depicted in details as well. Finally, concluding remarks are provided in section 4.

2. Proposed modifications

The overview of the proposed approach is illustrated in Figure 1.

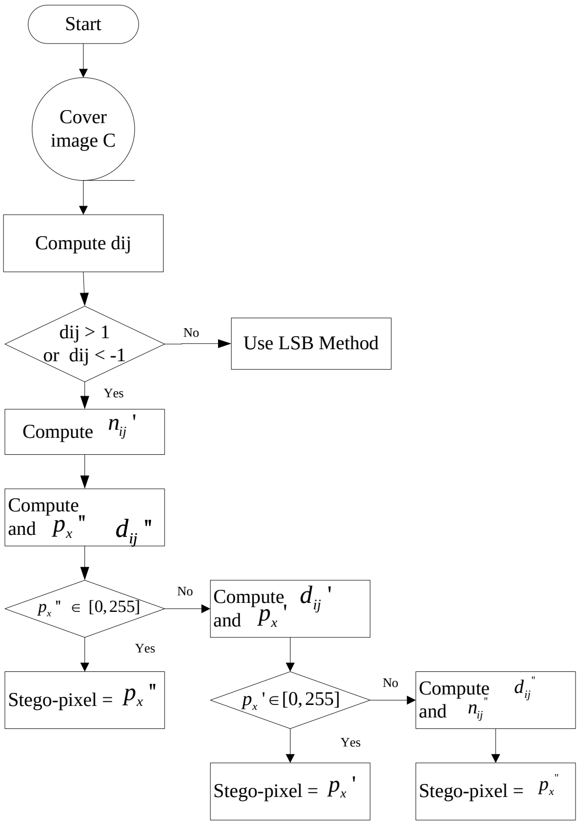


Figure 1. The proposed embedding flow.

Suppose $C=\{x_{ij} \mid 1 \le i \le M_C, 1 \le j \le N_C, x_{ij} \in \{0,1,...,255\}\}$ is the original gray-level cover-image of size $M_C \times N_C$. Like the side match approach in [7], the proposed scheme is classified into 3 cases (two-sided, three-sided, and four-sided) and explained step by step in the following sub-sections.

2.1. Embedding procedures

Figure 2 demonstrates the pixel definition in two-sided side match. The referenced pixels lie in the gray area while the embedding pixels lie in the white area.

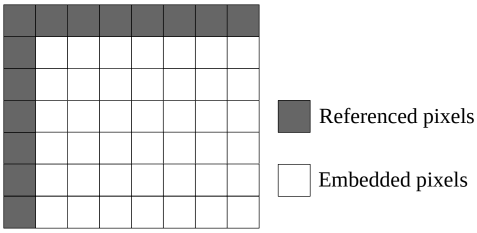


Figure 2. Two-sided side match pixels definition.

Step 1: In Figure 3, the embedding pixel is denoted as p_x while p_u and p_l represent the upper and the left pixels of it respectively.

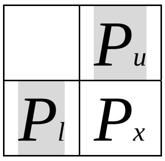


Figure 3. Pixel definition at a single position *(i,j)*.

Step 2: According to these 3 pixel values, a difference value d_{ij} can be computed by Eq. (1).

$$d_{ij} = (p_u + p_l)/2 - p_x \tag{1}$$

If d_{ij}=0, 1, or -1, we conclude that p_x is not in the edge area and just embed one bit in it using LSB method.

Step 3: If d_{ij} is not 0, 1, nor -1, we conclude that p_x is in the edge area and apply Eq. (2) to compute the number of bits to be embedded in it.

$$n_{ij} = \left\lfloor \log_2 \left| d_{ij} \right| \right\rfloor \tag{2}$$

where $|x|$ is the absolute value of x and $\lfloor\ \rfloor$ represents the floor function.

Step 4: So far the procedures are the same as that of side match method. In this step, we propose to replace n_{ij} by n_{ij}' using Eq. (3) so as to increase the embedding capacity. This modification, in general, does not degrade the image quality (sometimes the image quality is even enhanced) as will be demonstrated in subsection 2.3 and section 3.

$$\text{Case1: } n_{ij}' = n_{ij} \text{ if } 2^{n_{ij}} \le |d_{ij}| \le \left(\frac{2^{n_{ij}} + 2^{n_{ij}+1}}{2}\right) - 1$$

$$\text{Case2: } n_{ij}' = n_{ij} + 1 \text{ if } \left(\frac{2^{n_{ij}} + 2^{n_{ij}+1}}{2}\right) \le |d_{ij}| \le 2^{n_{ij}+1} \tag{3}$$

Step 5: Based on the new value n_{ij}', a new difference value d_{ij}' is computed using Eq. (4).

$$d_{ij}' = \begin{cases} 2^{n_{ij}'} + b, & \text{if } d_{ij} > 1 \\ -(2^{n_{ij}'} + b), & \text{if } d_{ij} < -1 \end{cases} \tag{4}$$

where b is the decimal value of n_{ij}' bits segment from the secret messages.

Step 6: Finally, the new value of the pixel p_x' is defined as

$$p_x' = \lfloor (p_u + p_l)/2 - d_{ij}' \rfloor \tag{5}$$

Step 7: There are two cases in this step. If p_x' is in [0,255], the final stego-pixel value is simply p_x'. Otherwise, apply $n_{ij}'' = n_{ij}' - 1$ in Eq. (4) and (5) to obtained the revised stego-pixel value p_x''.

The primary modifications are in steps 4 and 7. In some cases, one more bit can be embedded and hence improve the capacity of the original side match method. For example, if the value of stego-pixel is 267, side match method abandons this pixel without embedding any bit (i.e. it keeps this pixel unchanged). With step 7, this kind of positions can still be utilized to increase the amount of hidden massages.

For three-sided and four-sided side match schemes [7], the procedures are very similar to the steps described above. The only difference is the number of the referenced pixels. Figure 4 and 5 demonstrate the reference positions for three-sided and four-sided schemes respectively. As a result, we only need to modify Eq. (1) and (5). The interested readers are referred to [7] for the modified equations.

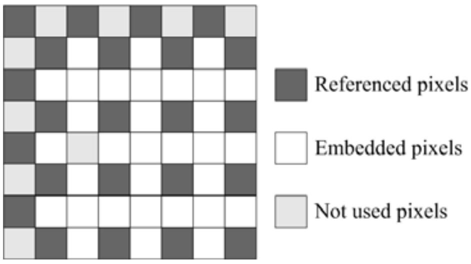


Figure 4. Pixels definition for three-sided side match.

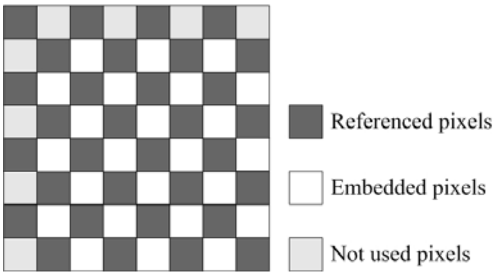


Figure 5. Pixels definition for four--sided side match.

2.2. Extracting procedures

The message extracting is explained step by step as follows. Again we adopt the two-sided case to describe the extracting steps.

Step 1: Compute the difference value $d_{ij}*$, given an input pixel value p_x^* and the upper and the left reference pixel values p_u^* and p_l^*.

$$d_{ij}^* = (p_u^* + p_l^*)/2 - p_x^* \tag{6}$$

If d_{ij}^* =0, 1, or -1, we just extract one bit from p_x^* using the LSB method.

Step 2: If d_{ij}^* is not 0, 1, nor -1, we apply Eq. (7) to compute the number of bits to be extracted.

$$n_{ij}^* = \lfloor \log_2 |d_{ij}^*| \rfloor \tag{7}$$

The n_{ij}^*-bit secret message $b(s, n_{ij}*)$ is then extracted out using the following equation.

$$b(s, n_{ij}*) = \begin{cases} d_{ij}* - 2^{n_{ij}*}, & if \quad d_{ij}* > 1 \\ -d_{ij}* - 2^{n_{ij}*}, & if \quad d_{ij}* < -1 \end{cases} \tag{8}$$

Step3: Finally, the whole secret messages S can be obtained by concatenating these $b(s, n_{ij}*)$ one by one with according order.

2.3. A numerical example

As a numerical example, assume the gray value of the given pixel p_x is 27 and the two neighboring pixel have values 40 and 44 respectively. The secret message S is assumed to be 001111. Following steps 2-7 described in sub-section 2.1, we can obtain the corresponding intermediate values as follows and then summarize the final results in Table 1 (together with the original side match results for comparison).

Step 2: Applying Eq. (1) to p_x=27, p_x=40, and p_x=44,

$$d_{ij} = (40+44)/2-27 = 42-27 = 15 .$$

Since it is not 0, 1, nor -1, p_x is regarded as a pixel in the edge area.

Step 3: Applying Eq. (2) to $d_{ij} = 15$,

$$n_{ij} = \lfloor \log_2 |d_{ij}| \rfloor = \lfloor \log_2 15 \rfloor = 3 .$$

Step 4: For this example,

$$(\frac{2^3+2^4}{2}) = 12 \le |d_{ij}| \le 16 = 2^4$$

and hence case 2 of equation (3) is satisfied. As a result, we embed one more bit at this pixel,i.e. $n_{ij}' = n_{ij}+1 = 3+1 = 4$ and the secret message to be embedded is b=0011=3.

Step 5: Exploiting Eq. (4).

$d_{ij}' = 2^4+3 = 19$ since $d_{ij} = 15 > 1$.

Step 6: Finally, using Eq. (5), the new value of the stego-pixel p_x' is computed as

$$p_x' = \lfloor (40+44)/2-19 \rfloor = \lfloor 42-19 \rfloor = 23$$

Step 7: Because p_x' is a valid pixel value, no need to decrement the bit number of embedding.

Table 1. Results of a numerical example: p_x=27.

	Side match	Modified side match
Capacity(bit)	3 bits	4 bits
p_x'	p_x'=33	p_x'=23

In this specific example, it is obvious that the hidden quantity is one bit more than the original side match. Furthermore, the value of stego-pixel p_x' is 23, which is closer to original pixel (27), and hence a better image quality is maintained. In addition to this simple example for a single pixel, the complete comparison between the proposed method and [7] is demonstrated in the next section.

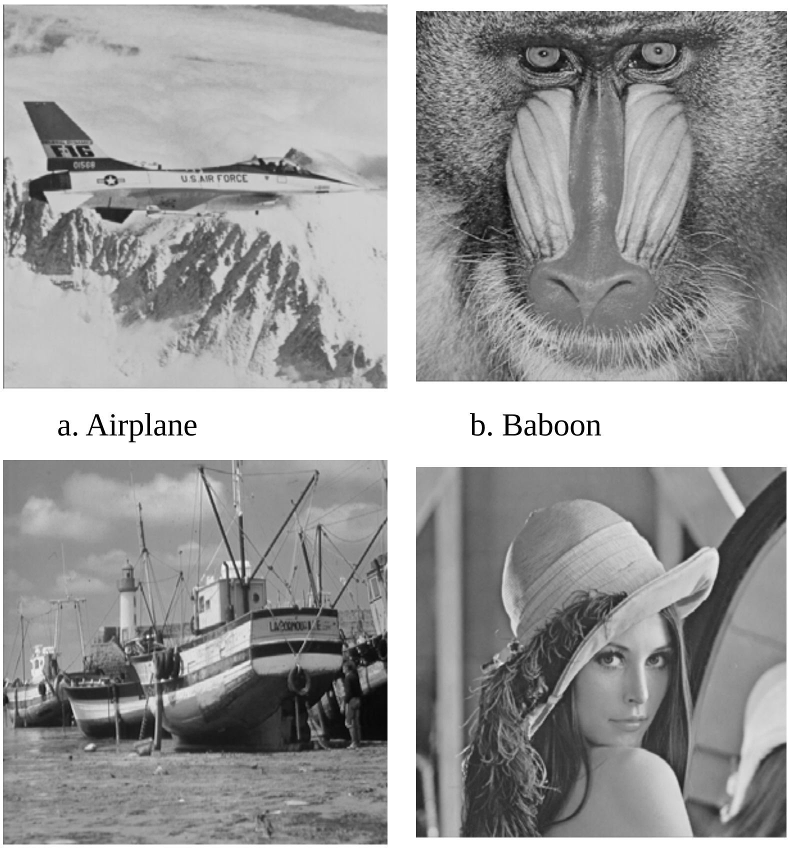
3. Simulation results

Four selected cover-images, ”Airplane（254Kbyte）, ”Baboon”（258Kbyte）, ”Boat”（256Kbyte）and ”Lenna” （258Kbyte）shown in Figure 6 are adopted for simulations. All of them are 512×512, 8bit gray-level, TIFF formatted. Hence, in all cases of simulations, $M_C = N_C = 512$. We employ a pseudo-random number generator to produce the secret messages bits. Peak signal to noise ratio (PSNR) is used for image quality evaluation. The PSNR of an 8-bit gray-level images is defined as：

$$PSNR = 10\log_{10}\frac{255^2}{MSE},$$

Wher $MSE = (\frac{1}{M_C \times N_C})\sum_{i=1}^{M_C}\sum_{j=1}^{N_C}(x_{ij}-y_{ij})^2$ and x_{ij}, y_{ij} represent the pixel values of the original cover-image and the stego-image respectively.

Table 2, 3, and 4 ensure that the embedding capacity is significantly increased by the Modified Side Match (MSM) method while the image quality is almost the same. In step 4 of sub-section 2.1, an extra bit is embedded in some pixels compared with [7]. Furthermore, if some stego-pixels are not located in [0,255], [7] just discards them while MSM still embeds extra bits by step 7.



a. Airplane

b. Baboon

c. Boat

d. Lenna

Figure 6. Four cover-images for simulations.

Table 2. Comparison in two-sided case.

	Side match		MSM	
	Capacity(bits)	PSNR(dB)	Capacity(bits)	PSNR(dB)
Boat	462,449	38.77	477,501	39.21
Baboon	660,725	33.53	695,145	34.14
Lenna	389,004	41.22	397,513	41.85
Airplane	331,247	40.94	342,185	41.58

Table 1. Comparison in three-sided case.

	Side match		MSM	
	Capacity(bits)	PSNR(dB)	Capacity(bits)	PSNR(dB)
Boat	329,053	41.06	338,451	41.34

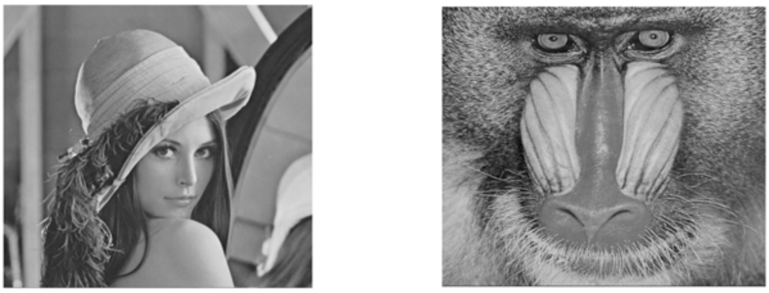
Baboon	483,758	34.93	498,167	35.24
Lenna	267,242	45.03	274,548	45.52
Airplane	232,327	44.61	239,472	45.06

Table 4. Comparison in four-sided case.

	Side match		MSM	
	Capacity(bits)	PSNR(dB)	Capacity(bits)	PSNR(dB)
Boat	201,138	44.33	207,497	44.33
Baboon	298,413	38.56	306,209	39.31
Lenna	164,538	48.18	168,289	48.64
Airplane	146,987	47.91	151,843	48.23

The experiment results indicate that we can raise the capacity by an amount of 8,509 to 34,420 bits in the two-sided case. The capacity improvement for the three-sided case ranges from 7,306 to 14,409 bits, and from 7,306 to 14,409 bits for the four-sided case.

Fig. 7 shows the resulted stego-images using MSM method (two-sided case), where image Lenna and Baboon are used as the cover-images. Since the PSNR are high (41.22dB and 33.53dB respectively), one can barely distinguish the difference from the cover images shown in Fig. 6.



a. Lenna

b. Baboon

Figure 7. The resulted stego-images of the two-sided case.

4. Conclusions

In this paper, we propose a modifying scheme to improve the side match method. The major merit of the proposed method is the increase of the embedding capacity without scarifying the image quality. The experiment results ensure the superiority of the proposed modifications. Like the original side match method, it also provides respectable security since the embedding procedures are not as straight as the LSB scheme. In the future, applying the modified side match method on the frequency domain schemes might be an interesting topic for research.

5. References

[1] R. Rivest, A. Shamir, and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communication of the ACM*: 120-126.

[2] DES Encryption Standard (DES),

National Bureau of Standard (U.S.). *Federal Information Processing Standards Publication 46*, National Technical Information Service, Springfield, VA, 1997.

[3] Pfitzmann, B. 1996. Information hiding terminology," *Proc. First Workshop of Information Hiding Proceedings*, Cambridge, U.K., Lecture Notes in Computer Science, Vol.1174: 347-350.

[4] Chan, C. K., and Cheng, L. M. 2003. Hiding data in image by simple LSB substitution. *Pattern Recognition*, Vol. 37: 469-474.

[5] Motoi IWATA, Kyosuke MIYAKE, and Akira SHIOZAKI, 2004. Digital Steganography Utilizing Features of JPEG Images. *IEICE Trans. Fundamentals*, Vol. E87-A: 929-936.

[6] Li, Zhi., Sui, Ai, Fen., and Yang, Yi, Xian. 2003. A LSB steganography detection algorithm. *IEEE Proceedings on Personal Indoor and Mobile Radio Communications*: 2780-2783.

[7] Chang, C. C., and Tseng, H. W. 2004. A Steganographic method for digital images using side match. *Pattern Recognition Letters*: 1431-1437.