Data Compression Adapted Based Binary Image Hiding Method

Chuin-Mu Wang[1]*, Yung-Kuan Chan[2],Yu-An Ho[3], Ching-Lin Wang[4]

ABSTRACT

Due to the prevalence of the Internet and multimedia, one can transmit a huge amount of data via the Internet. Image hiding techniques is often used to prevent data transmitted on a computer network from being hacked by unauthorized users. Generally, in a binary image, only the proximity of edge can be used to carry secret data. Since a little space in a binary image can be used to carry secret data, it is much more difficult to conceal secret data in a binary image than in a gray-level or a color image. This paper provides a hiding method to embed secret data in a binary cover image. The method hides secret data in object boundaries. The experimental results show that the hiding method can give a good capacity rate and cover image quality. In addition, the proposed method can be combined with image compression methods.

Keyword: Huffman encoding, arithmetic encoding, image compression, data hiding

I. INTRODUCTION

Since the prevalence of the Internet and multimedia has increased exponentially in recent years, people can exchange and transmit a large amount of data via computer networks. However, the security mechanisms of an ordinary computer network, especially a wireless one, are not so sufficient that data transmitted on it may be easily intercepted. Thus, how to prevent data transmitted on a computer network from being hacked by unauthorized users has become an interesting issue to many researchers. Most of the technologies used currently [17, 22, 37] embed confidential data in a certain digital medium, and then transmit the data to a receiver via this medium. Even if the data are intercepted halfway, people who intercept the data can merely see the outer appearance of the medium, rather than the initially embedded data.

Image hiding is to embed secret data in an image. In general, the image that carries the secret data is called a cover image. After secret data are embedded, the cover image becomes a stego image. This process of hiding is called embedding. Therefore, the purpose of image hiding is to conceal secret data in a cover image by imposing imperceptible changes on the stego image [2].

So far, numerous data hiding methods [2, 8] have been proposed, but most of them can embed the secret data only in a grey-level or a color image. In a binary image, a pixel is generally described by only one bit. If one bit in an all-black or an all-white region is changed, human eyes can easily detect the change. Hence, only the proximity of edge can be used to carry secret data. Since not much space in a binary image can be used to conceal secret data, it is much more difficult to hide secret data in a binary image than in a gray-level or a color image.

Some methods have been proposed to hide secret data in a binary image. Matsui and Tanaka [4] embedded secret data in a dithered image by changing the dithering patterns and in fax images by changing the run-lengths. However, this method cannot be applied to general binary images. Maxemchuk and Low [5, 10] proposed three different methods for embedding secret data in text documents: line shift coding, word-shift coding, and feature coding. Although these methods can theoretically be defeated, they require interactive human intervention and are not cost-effective in practice. Wu and Liu [10] hid a moderate amount of data in a general binary image. Their method manipulates "flippable" pixels to enforce specific block based relationship to embed a significant amount of data without causing noticeable artifacts. The method can hide annotation labels or other side information and verify whether a binary document has been tampered with or not. This technique determines the hidden capacity according to the size of blocks.

This paper presents an object boundary (OB) based hiding method to embed secret data in a binary cover image during encoding the cover image. The OB-based hiding method cannot only provide a high hiding capacity but also create a stego image quite similar to the cover image. The method also can be combined with image compression methods.

II. OB-BASED HIDING METHOD

The OB-based hiding method comprises two stages: Exclusive-OR and data hiding. The Exclusive-OR stage is to make black pixels appear only around the object boundaries in a cover binary image with all the rest being white. The data hiding stage is to embed secret data in the object boundaries in the cover binary image.

1. Exclusive-OR Stage

According to Shannon [6], there is a fundamental limit on lossless data compression [1]. This limit, called "entropy", is terminologically referred to as ***H***. The exact value of ***H***

Paper submitted 08/15/11; revised 11/15/11; accepted 12/30/11.

** Corresponding Author: Yung-Kuan Chan (E-mail: ykchan@nchu.edu.tw)*

[1] Department of computer science and information engineering, National Chin-Yi University of Technology, Taiwan.

[2] Department of Management Information Systems,National Chung Hsing University, Taiwan

[3] Department of Computer Science and Information Engineering, National Chung Hsing University, Taiwan

[4] Department of Information Management, National Chin-Yi University of Technology, Taiwan

depends on data — more specifically, the statistical nature of the data. It is possible to compress the data, in a lossless manner, with a compression rate close to ***H***.

Assume that the data are statistically independent. Let ***k*** be the number of different items in the data, and p_i be the occurrence probability of the ***i***-th item in the data. The entropy ***H*** of the data is defined as:

$$H = \sum_{i=1}^{k} p_i \times log_2(\frac{1}{p_i}) .$$

According to Shannon, by entropy, the number of bits required to represent a pixel is at least ***H*** bits/pixel, after removing the redundant data from the image. Therefore, one may get higher efficiency in storage space for a digital image with low entropy.

A binary image is usually composed of big all-black areas and all-white areas. To enhance the encoding efficiency, a binary image compression method generally makes the black pixels only exist near the boundaries of objects with all the other pixels being white.

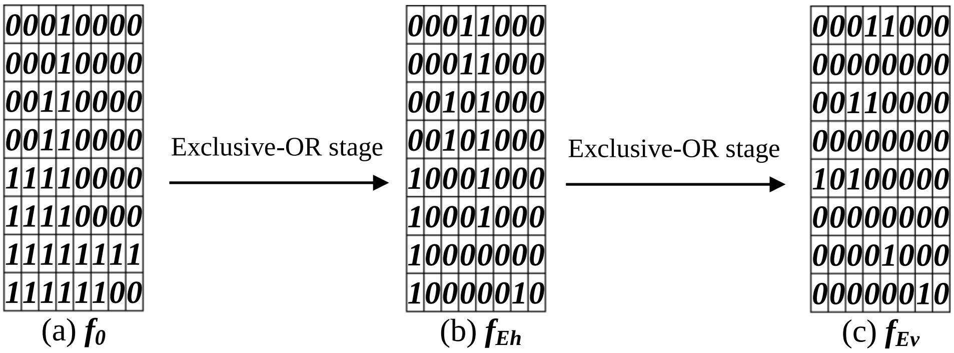
In this stage, a binary cover image f_{Eh} is created from f_c by executing an Exclusive-OR logic operation for each two adjacent pixels in f_c. The OB-based hiding method scans the whole image pixel by pixel. If the gray-level of the currently being visited pixel differs from the gray-level of its predecessor, then the method sets the gray-level of the pixel in f_{Eh}, that is in the same position as the currently visited pixel in f_0, to black; otherwise, it is set to be white. Each row of f_{Eh} can be created by the following formula, where ***'0'*** stands for a white pixel and ***'1'*** a black pixel:

$$f_{Eh}(i,j) = \begin{cases} f_0(i,j), & \textbf{if } j = 0, \\ f_0(i,j) \oplus f_0(i,j-1), & \textbf{otherwise}, \end{cases} \qquad (1)$$

where $\oplus$ is the Exclusive-OR logic operation, and $f_c(i, j)$ and $f_{Eh}(i, j)$ are the pixels located at the coordinates ***(i, j)*** on f_c and f_{Eh}.

Similarly, this method also scans each pixel in f_{Eh} pixel by pixel to create an image f_{Ev} by performing the Exclusive-OR logic operation between each two vertical neighboring pixels in f_{Eh}. f_{Ev} can be constructed by the following formula:

$$f_{Ev}(i,j) = \begin{cases} f_{Eh}(i,j), & \textbf{if } i = 0, \\ f_{Eh}(i,j) \oplus f_{Eh}(i-1,j), & \textbf{otherwise} \end{cases} \qquad (2)$$



00010000
00010000
00110000
00110000
11110000
11110000
11111111
11111100

00011000
00011000
00101000
00101000
10001000
10001000
10000000
10000010

00011000
00000000
00110000
00000000
10100000
00000000
00001000
00000010

Figure 1: A binary image f_c with $8×8$ pixels and its f_{Eh} and f_{Ev}

Table 1: The numbers of black and white pixels

Image	f_0			f_{Ev}		
	Black pixel	White pixel	Entropy	Black pixel	White pixel	Entropy
CCITT 1	***155,591***	***3,950,137***	***0.2326***	***51,476***	***4,054,252***	***0.0972***
CCITT 2	***184,240***	***3,921,488***	***0.2642***	***28,552***	***4,077,176***	***0.0598***
CCITT 3	***337,052***	***3,768,676***	***0.4095***	***78,392***	***4,027,336***	***0.1363***
CCITT 4	***509,635***	***3,596,093***	***0.5411***	***204,154***	***3,901,574***	***0.2852***
CCITT 5	***317,707***	***3,788,021***	***0.3929***	***90,380***	***4,015,348***	***0.1526***
CCITT 6	***207,110***	***3,898,618***	***0.2883***	***46,010***	***4,059,718***	***0.0887***
CCITT 7	***356,850***	***3,748,878***	***0.4261***	***186,544***	***3,919,184***	***0.2667***
CCITT 8	***1,766,467***	***2,339,261***	***0.9859***	***52,294***	***4,053,434***	***0.0984***

For example, Figure 1(a) shows an ***8×8*** binary image f_0. After the Exclusive-OR stage, f_0 is changed into f_{Eh} and then into f_{Ev}, as respectively shown in Figure 1(b) and 1(c). In this example, the entropy of the image in Figure 1(a) is ***0.99***, while the entropy of the image in Figure 1(c) is ***0.54***. As the lower entropy indicates, the image in Figure 1(c) is capable of offering higher compression efficiency.

Table 1 demonstrates the numbers of white and black pixels on f_0 and f_{Ev}, where **CCITT 1** to **CCITT 8** are the test images [11] shown in Figure 2. Each of the test images consists of ***2376×1728*** pixels. Table 1 also displays the entropies of each f_0 and f_{Ev}; that is, the number of white pixels in f_{Ev} is much greater than that in f_0. It also tells that the entropies of f_0 are much larger than that of f_{Ev}.

2. Data Hiding Stage

Due to the constraint on bandwidth, a user often hopes to compress the stego-image first for reducing the data size before transmitting the compressed data over the Internet. A compression method, i.e., Huffman [7], arithmetic [7], or quadtree [9] encoding methods, and so on, can be used to compress the cover image f_0. To compress f_0, one can replace encoding f_{Ev} by f_0. Before encoding f_{Ev}, the OB-based hiding method embeds secret data in the intermediate goods f_{Eh} when encoding the cover image f_0.

When a pixel in an all-white area is changed into a black one or a pixel in an all-black area is changed into a white one, the human eyes can easily detect the change. In data hiding, embedding secret data in a cover image usually distorts the cover image. The binary image hiding techniques generally hide secret data near the boundaries of objects in f_0. Hence, the OB-based hiding method also attempts to conceal secret data in the boundaries of objects in the cover image. In the Exclusive-OR stage, f_0 is transformed into f_{Eh} and then into f_{Ev}. If the gray-levels of the **(*i*-*1*)**-th and the ***i***-th pixels in f_0 are the same, the ***i***-th pixel in f_{Eh} is given a ***0***-bit; otherwise, the ***i***-th pixel in f_{Eh} is a **'*1*'**-bit. The **'*1*'**-bits in f_{Eh} are located near the boundaries of objects in f_0. Therefore, the OB-based hiding method embeds secret data in the pixels nearby the **'*1*'**-bits in f_{Eh}.

Table 2: The replaced 4-bit units in '0'-group and '1'-group

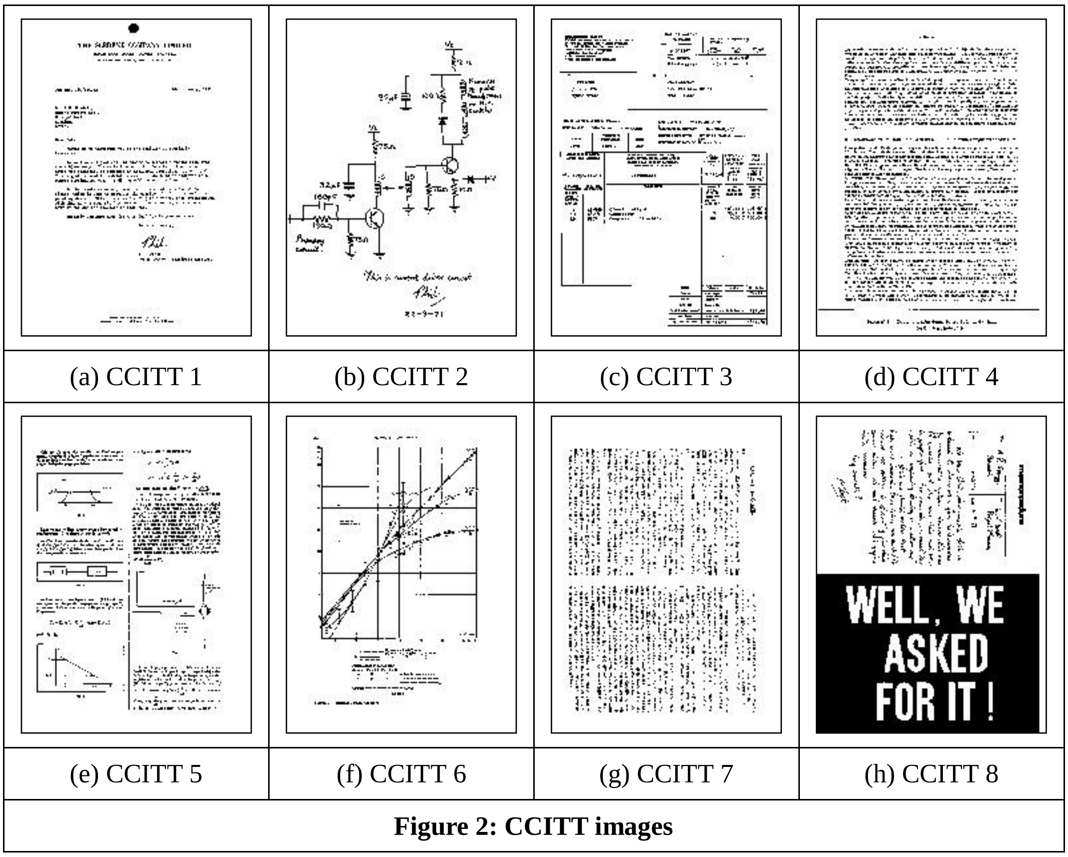
U	0-group	1-group
1111	**0011**	**1111**
0011	**0011**	**0101**
1100	**1010**	**1100**
1001	**1001**	**0101**
0101	**0011**	**0101**
1010	**1010**	**1100**
0110	**1010**	**0110**
0111	**0111**	**0001**
1101	**1101**	**0001**
1110	**1000**	**1110**
1011	**1000**	**1011**
0001	**0010**	**0001**
0010	**0010**	**0001**
1000	**1000**	**0100**
0100	**1000**	**0100**

Converting the intensity of the ***i***-th pixel in f_{Eh} will make the colors of all the pixels be altered after the ***i***-th pixel in the decompressed f_0; while the intensities of the ***i***-th and the ***j***-th pixels in f_{Eh} are changed, only the intensities of all the pixels between the ***i***-th pixel and the ***j***-th pixel in f_0 decompressed from f_{Eh} are changed. Based on this property, the OB-based hiding method considers each row in f_{Eh} to be a big binary string and partitions the big binary string into 4-bit units. Each unit containing at least one '***1***'-bit will be used to carry one secret data bit since the OB-based hiding method engages in embedding the secret data near the boundary of objects in the cover image. We call the unit with at least one '***1***'-bit an embedding unit.

The OB-based hiding method categorizes all the possible 4-bit embedding units into two groups: '***0***'-group and '***1***'-group. Each secret bit b corresponds to one embedding unit ***U***. If ***b*****='*0*'** (resp. ***b*****='*1*'**), one embedding unit U' in ***0***-group (resp. '***1***'-group) is used to replace ***U***. If the two different bits between ***U*** and U' are located more closely to each other, the distortion of the stego image compared to the cover image can be reduced more. Table 2 shows the replaced 4-bit embedding units, where there are exactly ***0*** or ***2*** different bits between ***U*** and U', and the different bits are more closely located. After that, the compression method continues to encode the f_{Eh} which carries the secret data.

It is easy to transform digital data into binary codes. In this paper, we assume that the secret data ***SD*** is a big binary string and |***SD***| is the size of ***SD***. Before hiding, the OB-based hiding method first concatenates |***SD***| and ***SD*** into a binary string SD'. To prevent the secret data from unauthorized access, this method uses a private key ***PK*** (over ***512*** bits) as the seed of a random number generator G to generate a big binary string ***K***, where |***K***|=|SD'|. The method then computes SD'' =KΘ SD' and hides SD'' in the embedding units bit by bit.

To extract the secret data from the stego image, the OB-based hiding method transforms the stego image into f_{Eh} by Formula (1). The OB-based hiding method considers each row in f_{Eh} to be a big binary string and partitions the big binary string into ***4***-bit units in order. Next, the method takes out each embedding unit U'; one **'*0*'**-bit is appended to the rear of SD'' if U' is in **'*0*'**-group, and else one **'*1*'**-bit is appended to the rear of SD''. The proposed method then uses the same private key ***PK*** as the seed of the random number generator ***G*** to generate a big binary string ***K***, where |***K***|=|SD''|. Besides, the proposed method computes SD' =***K***Θ SD''. It removes the leftmost integer of SD'; the removed integer is |***SD***|. The leftmost |***SD***| bits of ***SD*** are the embedded secret data. The secret data extracted by the OB-based hiding method is lossless. Since the secret data are embedded only near the boundaries of objects in the cover image, the stego image is quite similar to the cover image.

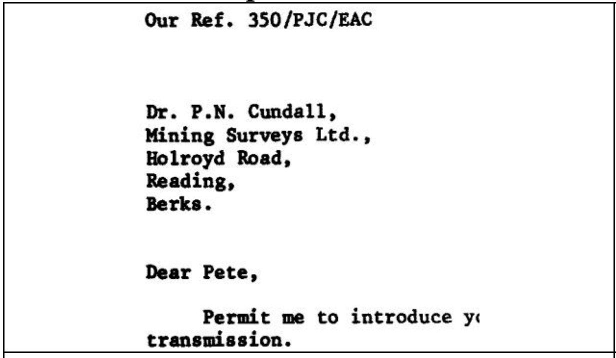


(a) CCITT 1 (b) CCITT 2 (c) CCITT 3 (d) CCITT 4

(e) CCITT 5 (f) CCITT 6 (g) CCITT 7 (h) CCITT 8

Figure 2: CCITT images

III. Experiments



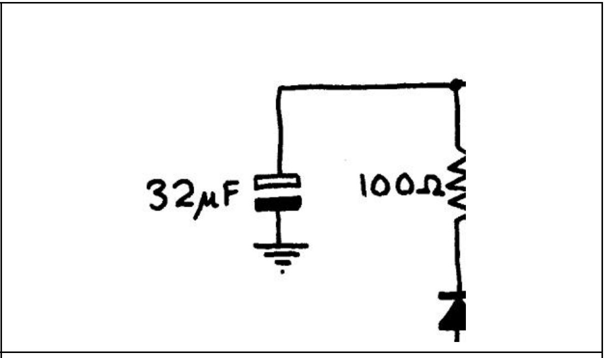
Our Ref. 350/PJC/EAC

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

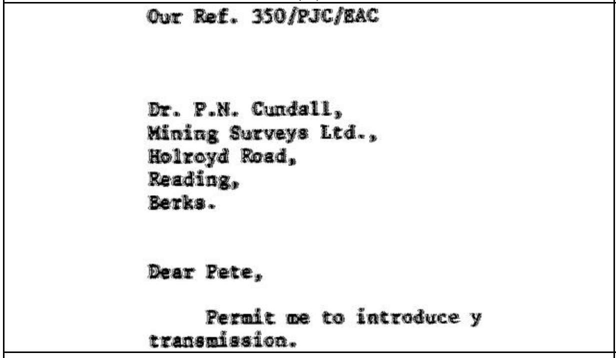
Dear Pete,

Permit me to introduce yo
transmission.

(a)



(b)



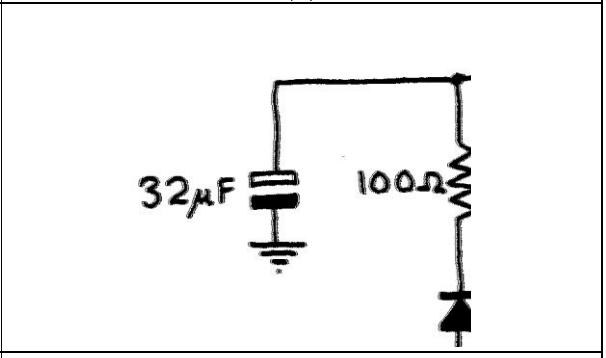
Our Ref. 350/PJC/EAC

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

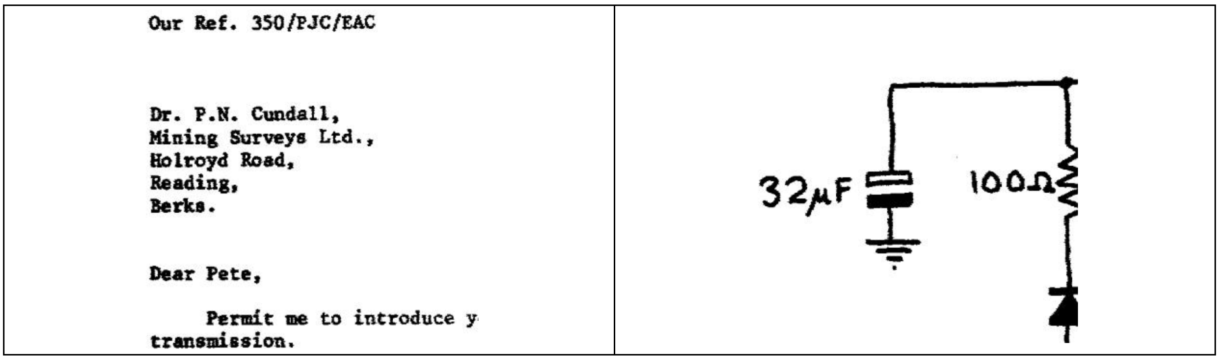
Dear Pete,

Permit me to introduce y
transmission.

(c)



(d)



Our Ref. 350/PJC/EAC

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce y
transmission.

(e) (f)

Figure 3: Two regions respectively on CCITT 1 as well as CCITT 2 and their corresponding regions on the stego images obtained by the OB-based hiding method and the PWLC hiding method

This section is to explore the performance of the OB-based hiding method and compare it to the performance of the pair-wise logical computation (PWLC) hiding method [3] by experiments. Eight document images in Figure 2 are used as the test images each with **1728×2376** pixels. There are many big all-black areas or big all-white areas in images CCTTT2 and CCTTT8, while images CCTTT4 and CCTTT7 have fewer big areas consisting of all-black pixels or all-white pixels.

Table 3: |*SD*| and *CR* obtained by PWLC hiding method and OB-based hiding method

Images	PWLC		QLS-Hiding	
	|*SD*|	***CR* (%)**	**|*SD*|**	***CR* (%)**
CCITT 1	29431	98.92	42598	99.49
CCITT 2	23916	99.12	25079	99.70
CCITT 3	51322	98.12	71188	99.13
CCITT 4	95560	96.50	165323	97.98
CCITT 5	53141	98.05	81074	99.01
CCITT 6	34544	98.73	47635	99.42
CCITT 7	77279	97.17	133074	98.37
CCITT 8	45733	98.32	49041	99.41

Table 3 demonstrates the results of this experiment. In this experiment, a random number generator ***G*** is applied to generate **|*SD*|** bits of secret data and then the OB-based hiding method is used to hide the generated **|*SD*|**-bits secret data in a ***W***×***H*** cover data, where **|*SD*|** is the maximal number of bits which can be embedded in the cover image by the OB-based hiding method. Here, the distortion degree correction rate (***CR***) is adopted to measure the distortion between the cover image and the stego image. ***CR*** can be defined as:

$$CR = \left(\frac{100}{W \times H}\right)\sum_{i=1}^{W}\sum_{j=1}^{H}\left(1-\left(b_{ij}-b'_{ij}\right)^{2}\right)$$

where b_{ij} is the color of the pixel located at (i, j) in the cover image, and b'_{ij} the color of the pixel located at (i, j) in the stego image.

Table 3 demonstrates that the OB-based hiding method can obtain better **|*SD*|** and ***CR*** than the PWLC hiding method. Figure 3(a) (resp. Figure 3(b)) is one region on CCITT 1 (resp. CCITT 2), Figure 3(c) (resp. Figure 3(d)) and Figure 3(e) (resp. Figure 3(f)) are its corresponding regions on the stego images respectively generated by the PWLC hiding method and the OB-based hiding method. Figure 3 shows that the PWLC hiding method gives a much worse quality of stego image than the OB-based hiding method, especially near the boundaries of objects.

IV. CONCLUSIONS

In this paper, the OB-based hiding method is proposed to embed secret data in a binary cover image. The OB-based hiding method adopts Exclusive-OR logic operation to indicate the boundaries of objects in a cover image and then embeds secret data in the object boundaries. The experimental results demonstrate that the OB-based hiding method can give much better **|*SD*|** and ***CR*** than the PWLC hiding method. The OB-based hiding method cannot only exactly extract the secret data from the stego image but also provides better stego image quality, especially for the regions near object boundaries than the PWLC hiding method.

REFERENCES

[1] T. Batu, S. Dasgupta, R. Kumar, R. Rubinfeld, "The Complexity of Approximating Entropy," *Proceedings of the 17th Annual IEEE Conference on Computational Complexity,* Montreal, Canada, May

2002, pp. 678-687.

[2] D. C. Huang, Y. K. Chan,_ and J.H. Wu, "An Agent-Based LSB Substitution Image Hiding Method," *International Journal of Innovative Computing, Information and Control*, Vol. 6, No. 3(A), March 2010, pp. 1023–1038.

[3] K. Matsui, and K. Tanaka, "Video-Steganography: How to Secretly Embed a Signature in a Picture," *Proceedings of IMA Intellectual Property Project*, Vol. 1, No. 1, 1994, pp. 187-206.

[4] P. Mateu-Villarroya, and J. Prades-Nebot, "Lossless Image Compression Using Ordered Binary-Decision Diagrams," *Electronics Letters*, Vol. 37, No. 3, February 2001, pp.162-163.

[5] N. F. Maxemchuk, and S. Low, "Marking Text Documents," *Proceedings of the 1997 International Conference on Image Processing* (*ICIP '97*), 3-Volume Set-Volume 3, October 1997, p 13.

[6] C. E. Shannon. "A Mathematical Theory of Communication," *The Bell System Technical Journal*, Vol. 27, No. 3, July 1948, pp. 379-423.

[7] V. Singla, R. Singla, and S. Gupta, "Data Compression Modeling: Huffman and Arithmetic," *International Journal of The Computer, the Internet and Management*, Vol. 16. No.3, September-December 2008, pp 64-68.

[8] C. C. Thien, and J. C. Lin, "A Simple and High-Hiding Capacity Method for Hiding Digit-by-Digit Data in Images Based on Modulus Function," *Pattern Recognition*, Vol. 36, No. 12, December 2003, pp. 2875-2881.

[9] C. L. Wang, S. C. Wu, Y. K. Chan, and R. F. Chang, "Quadtree and Statistical Model-Based Lossless Binary Image Compression Method," *Imaging Science Journal*, Vol. 53, No. 2, June 2005, pp. 95-103.

[10] M. Wu, and B. Liu, "Data Hiding in Binary Image for Authentication and Annotation," *IEEE Transactions on Multimedia*, Vol. 6, No. 4, August 2004, pp.528-538.

[11] CCITT Standard Fax Images at ftp://nic.funet.fi/pub/graphics/misc/test-images/