A data hiding technique in binary image by using distance transform without look-up tables

Chyuan-Huei Thomas Yang Ken-Ching Lee
Department of Computer Science, Husan-Chuang University
48, Hsuan Chuang Road, Hsinchu City, Taiwan 300
e-mail：chyang@hcu.edu.tw, m94301818@umail.hcu.edu.tw

摘要

在本篇論文中我們對將資料隱藏在二元影像有興趣。被隱藏的資料可以很容易的嵌入一張二元影像，也可以不需要原始影像而很容易的將之萃取出來。在隱匿學領域裡很多學者將資料隱藏在影像中物件的邊緣上以避免影像被破壞。他們以一個複雜的對照表來決定將所藏的位置上的像素為翻轉或是僅註記而不動。我們在此提出一個簡單而且不容易察覺到的資料隱藏法，利用數學型態學上的腐蝕作用來找出棋盤距離而不用對照一個複雜的表格。我們將掩護影像分成 3 乘 3 不重疊的小區塊。我們在這些區塊中檢驗其棋盤距離不大於二的個數，若小於或等於 1 到大於或等於 8 之間，則視欲嵌入的位元為 0 或 1，對此小區塊的中心點，採取反轉或註記。而要隱藏的資料則先轉為二元流，利用擬亂數產生器將之打亂，然後以反轉或註記藏在先前利用距離轉換所找到的位置。萃取時則將含有隱藏資料的影像，計算其棋盤距離，找出可能藏的區塊，取其中點然後將之串連起來，再用原來嵌入時用的擬亂數產生器找回隱藏的資料原始的二元流順序。由實驗結果顯示，我們的方法簡單而有不錯的效果。

關鍵詞：資料隱藏、隱匿學、二元影像、距離轉換、察覺不出的。

Abstract

In this paper we are interesting in hiding data in a binary image. The hidden data can be easily to be embedding in a binary image and to be extracting without using the original image. In steganography area many authors usually embed the binary data into the boundary of the objects in an image to avoid the distortion. They flip or mark a pixel that is an embedded pixel with a complicate set of look-up tables. We proposed a simple and imperceptible data hiding method that uses the chessboard distance transformation by erosion operation of mathematical morphology to find the embedding location without using a complicate set of look-up tables. We divide the cover image into small non-overlapping 3 by 3 blocks images. We just flip or mark the embedding location, the center of the block, which has at least one and no more than eight smallest distances to the outside boundary. The hidden data are first transferred into a bit stream, shuffled them by a pseudo-random generator then embedded in the location that set up by the distance transformation. We simply read the center of each non-overlapping 3 by 3 embedding candidate block to extract the embedded data. Then we use the same pseudo-random generator to shuffle back the order to find the original secret data. The experiment results show our method is simple and quite imperceptible.

Keywords: data hiding, steganography, binary image, distance transform, imperceptible

1. Introduction

Data hiding or watermarking techniques are widely discussed and applied in digital media area. Data hiding or called steganography is embedding the secret data into the cover objects. In this paper we are interested in the cover object is a binary image. It is more difficult to hide data into a binary image since the representation of a binary image is only one bit, i.e. 0 or 1. Most previous works on data hiding are using the color and grayscale images in which the pixels have a large range of values. Since we focus on using the binary image as a cover image, here we only mention several methods have been proposed for hiding data in binary image in literature [1],[7],[10]-[15]. Some authors deal with the watermark embedded in binary [4]-[6]. The technique of imperceptible watermark is similar with data hiding. Many authors use the properties

of the halftones to handle the perceptible problem during processing the embedding secret data into a binary image. Halftone depends on the human visual system like a low-pass filter. It is expressing an impression of shades of grayscale by the appropriate dot density and dot size [2],[3],[9]. We proposed a simple and imperceptible hiding method that uses the chessboard distance transformation by recursive erosion operation of mathematical morphology to find the embedding location without using a complicate set of look-up tables. In paper [8] they propose two new Euclidean distance transformations to reduce the computation of traditional methods. They apply the two-scan recursive morphological erosions to perform the chessboard distance transformation to find the distance from the boundary of an object. Although this paper is concerning the Euclidean distance transformation, it also offers us a good chessboard distance transformation. It would be easy to find the embedded locations by using this distance transformation.

We category the reference papers as three groups and introduce their proposed methods. Firstly, the cover image is binary image. In [1], Chen et al. use a block data hiding scheme to partition a cover image into 4 by 4 blocks. After that for each 4 by 4 block is repartitioned into four 3 by 3 overlapping sub-blocks to calculate the data hiding location. Although Liu et al. [7] concerning with grayscale cover image, the authors handle it as a binary image. They firstly, pixels of cover mage are grouped according to themselves luminance values. Then the occurrence of pixel-values in each group is counted and sorted in monotone increasing. Finally a bit plane-wise data hiding method is used to hide data in cover image. They use special codes to embed into the blocks of cover images and verified to achieve the authentication in Tzeng and Tsai [10]. To avoid to be detected in embedded images is done by generating the codes randomly and locate them into randomly selected locations in the image blocks. According to the pixel values replacement in embedding, it is accomplished by allowing multiple locations for embedding the codes. Tseng and Pan [11] proposed a steganography scheme to hide critical information in a cover binary image. The scheme sacrifices some hiding space to maintain high quality of cover image, but it still has good data hiding rate. They present a simple and high capacity binary steganography, Wang and Chang [12]. They divide the cover image into uniform and non-uniform blocks. Many authors develop the embedding with flip and not flip. In this paper they add marking the location to be embedding locations. These won't change any bit to extend the capacity of hiding places. Wu and Liu [14] consider the flippable pixels and use the block-based relationship to embed more data without causing visible artifacts. They shuffle the data before embedding to increase the embedding capacity from region to region. Their method doesn't need the original image to extract the hidden data. Yang and Kot [15], they use the noise of compression patterns by smoothing with a 5 by 5 neighborhood. They claim it has a good capacity and preserve the imperceptible. Secondly, the similar technique of data hiding, watermarking, is discussed in Kim and Afif [4], they propose an authentication watermarking by self toggling (to embed one bit to enforce it be either black or white). It is similar to the data hiding by self toggling. Applying a pseudo-random number generator generates a set of locations in cover image with exclusive-or to find the fingerprint of a secret image to be embedded into cover image. Lu et al. [5][6], they blur the original image first, divide into 8 by 8 blocks biased binarization, using a threshold to give 0 or 1, with error correcting code encoding to embed the watermark. They apply DCT to find the DC components with the non-uniform 8 by 8 block identification to locate the embedding places. They blur the binary cover image into a gray level image as a pre-processing and a post-processing by binarizing the image after embedding into a binary image. The third part is considering the cover image is halftone image. Fu and Au [2] propose data hiding smart pair toggling to hide data by forced complementary toggling at pseudo-random locations within a halftone image. The complementary pixels are chosen to minimize the chance of forming visually undesirable clusters. They show their method can hide a large amount of hidden data while maintaining good visual quality. The authors propose the method by using conjugate error diffusion to hide invisible patterns in two or more error diffused halftone images [3]. Those hidden patterns would become visible on the halftone images when they are overlaid. A binary visual pattern is hidden in two halftone images by applying Conjugate Error Diffusion. Sherry and Savakis [9] propose the data hiding cell parity (DHCP) works in the parity domain instead of individual pixels. They use data hiding mask toggling (DHMT) works by encoding two bits in

the 2x2 neighborhood of a pseudorandom location. Dispersed Pseudorandom Generator takes place before halftoning that achieved better visual results. Wang[13] applies the modified digital halftoning techniques to embedding information inside a binary image. Two modified ordered dithering and modified multiscale error diffusion in digital halftoning techniques are proposed.

This paper is organized as follows. In section 2 we discuss our proposed method that includes how to embed the secret data into a cover image and extract these embedded data from this marked cover image. Section 3 then gives several computer experiment results and comments. Finally, in the conclusion we summarize our major results and outline our future work.

2. Proposed Method

The procedure of embedding data in a binary cover image is describing as follows. First we transfer the secret data into a bit stream. Then the bit stream is shuffled by using a pseudo-random generator (PRNG). We use distance transform to find the chessboard distance map of cover image and divide the cover image into 3 by 3 non-overlapping blocks then decide the embedding location with above two conditions. We place the 0's or 1's in the center of the 3 by 3 blocks according to the shuffled secret data bit stream. After embedding all bit stream data we obtain the marked cover image, or so called stego-image. This is done for the embedding procedure. The extraction procedure is simply to find the 3 by 3 non-overlapping blocks of the stego-image with the same condition in embedding step. We collect the 0's or 1's of the center of these possible 3 by 3 non-overlapping blocks with embedded data to get a bit stream. Then we apply the same PRNG to shuffled back the bit stream to get the original secret data. Next we are going to describe the distance transformation, embedding and extracting algorithm in detail。

2.1 Distance Transformation

A Distance Transformation (DT) converts a binary image to another image in which pixel value is the distance from this pixel to the nearest nonzero pixel of the binary image. The distance transform of an image sets each pixel in the image to a new value. This new value is set to be the minimum distance of that pixel to an edge in the original image. In [8] Shih and Yang apply the chessboard distance (d8) transformation by erosion operation of mathematical morphology. They use a 3 by 3 template $\boldsymbol{g}$ to process the erosion with the cover image $\boldsymbol{C}$. Fig. 1 shows the template g, and Fig. 2 show the Distance Transformation (DT) algorithm of the iterative erosion of the image $\boldsymbol{C}$ with template $\boldsymbol{g}$.

$$\boldsymbol{g} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Fig. 1. the erosion of template $\boldsymbol{g}$

DT Algorithm

C：the binary cover image where the pixels of object are 1's and the pixels of background are 0's.

g：the template as in Fig. 1

D8_old, ***D8_new*** ： the chessboard distance, ***D8_old*** is the chessboard distance of one iteration before ***D8_new.***

(1) Initial：$\boldsymbol{D8\_old} = \boldsymbol{C} * \infty$

$\boldsymbol{D8\_new} = 0$

(2) Do

$D8\_new = D8\_old \ominus g$

Until ($D8\_new = D8\_old$)

where the $\ominus$ means the erosion operation

(3) Output $D8\_new$

Fig. 2 the DT algorithm

We show an simple example of chessboard distance in Fig. 3.

0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

(a) a binary image

0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0
0	1	2	2	1	1	1	1	0
0	1	2	2	2	2	2	1	0
0	1	2	3	3	3	2	1	0
0	1	2	2	2	3	2	1	0
0	1	1	1	2	2	2	1	0

0	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

(b) the chessboard distance of (a)

Fig. 3 an example of chessboard distance

2.2 Embedding Algorithm

The main idea of the proposed embedding algorithm is to maintain the perceptual of cover image. This algorithm generally embeds the secret data in the boundaries of objects. These changes are less distortion for the cover image. Thus we use the **DT** algorithm to find the chessboard distance of the cover image. Since we will embed the 0's or 1's and we don't know whether the center pixel of the 3 by 3 non-overlapping block in the qualification embedding blocks sequence is 0 or 1. If we want to embed a 0, and the center pixel of the current candidate embedding block is 0, we do nothing. If the center pixel is 1, we flip it as 0. That means whether the center pixel is 0 or 1, we just place the bit is 0 or 1 that we want to embed.

A qualification embedding 3 by 3 block has at least one 1's of chessboard distance and no more than eight 1's. We consider the chessboard distance of cover image will be changed after embedding the secret data. If the pixel is 1 and embedding a 0, this will slightly change the chessboard distance of stego-image. Thus we look the chessboard distance 2 as 1. Please see Fig. 4. If the 3 by 3 block has one 1 or eight 1's, we remove the isolated pixel and fill the hole with 1 for extraction problem. Fig. 5 shows two excluding cases. This would be easily to distinguish between the changed or unchanged during the 3 by 3 block has two or seven 1's.

0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	3	3	3	1	1	0
0	1	1	1	1	3	1	1	0
0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

The modified d8 of Fig. 3 (b)

Fig. 4 an example of modified D8

0	0	0		1	1	1
0	1	0		1	0	1
0	0	0		1	1	1

(a) one 1 (b) eight 1's

Fig. 5 Two special conditions of d8

The overall embedding algorithm is shown in Fig. 6 as follows.

Embedding Algorithm

C : the binary cover image where the pixels of object are 1's and the pixels of background are 0's. and its size is m by n

ST : the binary cover image where the secret data have been embedded

Data : the secret data(image)

$D8$: the chessboard distance of C

$MD8$: the modified chessboard distance of $D8$

W_{ij} : the 3 by 3 sliding window with upper left corner is (i, j) coordinate in ***MD8***

(1) ***ST*** = ***C***; ***D8*** = **DT**(***C***),

(2) Modified ***D8*** to ***MD8***

(3) Transfer the ***Data*** as a bit stream and use a PRNG to shuffle ***Data***

(4) For i=1 to m step 3

For j=1 to n step 3

If W_{ij} has at least one 1's of d8 and no more than eight 1's of d8 which are corresponding to ***ST***, then the center of W_{ij}, i.e. the stego-image of ***ST***(i+1, j+1) = current embedding bit of ***Data***

end

end

(5) Output the stego-image ***ST***

Fig. 6. Proposed Embedding Algorithm

We also use the simple example in Fig. 3 to demonstrate the embedding algorithm in Fig. 7. This example also is the worst case to embed, since the bit stream is 0 0 1 0 0 1 0 0. All center pixels of the embedding candidate blocks will be changed. The mean square error of this example is 0.0988. The center of the middle block in Fig.7 (a) means it is not an embedding place.

0	0	0	0	0	0	0	0	0
0	**0**	1	1	**0**	0	0	**1**	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	**0**	1	3	**3**	3	1	**0**	0
0	1	1	1	1	3	1	1	0
0	1	1	1	1	1	1	1	0
0	**1**	0	1	**0**	1	1	**0**	0
0	0	0	0	0	0	0	0	0

(a) the embedded bit in each center of 3 by 3 non-overlapping blocks

0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0

(b) the stego-image(marked image)

Fig. 7 an example of stego-image

2.3 Extraction Algorithm

The extraction algorithm is quite simple. When the sliding window moves from upper left corner to lower right corner, we just collect the center pixels of the embedding candidate blocks to get a bit stream and use PRNG to shuffle back to get the original secret data.

Extraction Algorithm

ST：a binary cover image where the secret data have been embedded

Data：the secret data(image)

D8：the chessboard distance of *ST*

MD8：the modified chessboard distance of *D8*

W_{ij}：the 3 by 3 sliding window with upper left corner is (*i*, *j*) coordinate in ***MD8***

(1) ***D8*** = **DT**(***ST***),
(2) Modified ***D8*** to ***MD8***
(3) For i=1 to m step 3
  For j=1 to n step 3
    If W_{ij} has at least one 1's of d8 and no more than eight 1's of d8 which are corresponding to ***ST***, then we collect the center of W_{ij} and save in ***Data***, i.e. we collect the stego-image of ***ST***(i+1, j+1)and save in ***Data***
  end
 end
(4) Use the PRNG which is the same as in embedding algorithm to shuffle back ***Data***
(5) Output the extraction secret data ***Data***

Fig. 8. Proposed Extraction Algorithm

We follow the example in Fig. 7 to use the extraction algorithm to extract the embedded secret data. Fig. 9 shows the results.

0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0
0	1	1	1	1	1	1	1	0

0	1	1	2	2	2	1	1	0
0	0	1	2	3	2	1	0	0
0	1	1	2	2	2	1	1	0
0	1	1	1	1	1	1	1	0
0	1	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0

(a) the d8 map of Fig. 7 (b), i.e. the d8 of the stego-image

0	0	0	0	0	0	0	0	0
0	**0**	1	1	**0**	0	0	**1**	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	**0**	1	1	3	1	1	**0**	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	**1**	0	1	**0**	1	1	**0**	0
0	0	0	0	0	0	0	0	0

(b) the sliding window only move to middle block that is not a candidate

Fig.9 an example to extract the secret data

The following figure shows flow-chart of the proposed embedding and extraction algorithms.

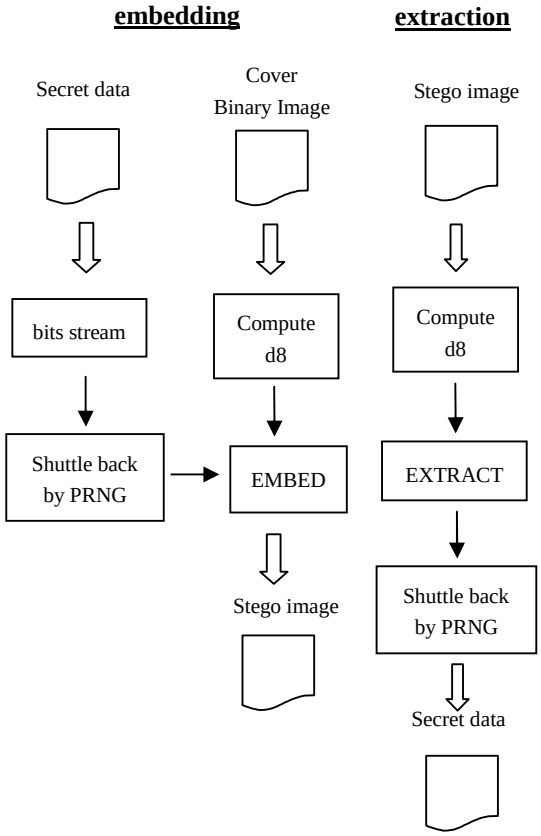


Fig. 10 flow-chart of the proposed embedding and extraction algorithms

3. Experiment Results

We test our proposed method with several popular images that many authors like to use. Table 1 demonstrates those sizes, embedding capacities, and the mean square error. The mean square error is

$$MSE = \frac{\sqrt{\sum_{ij}(stego\_image_{ij} - cover\_image_{ij})^2}}{m \times n} \quad (3.1).$$

The testing secret data is a logo of Hsuan Chuang University which size is 30 by 30 in Fig. 11. This logo consists of 3600 bits.

Fig. 11. the logo of Hsuan Chuang University

According to the computer experiments we find the "Peppers" has the least capacity. It should have to many large area of all whites or all blacks. Actually we may understand the maximum capacity should be 1/9, due to for each 3 by 3 block we place at most one bit in it.

Table 1. the MSE and capacity of cover images

Cover image	Capacity(bits)	SE	MSE
Peppers 512 × 512	**Max：3377**	**40.6079**	**0.000155**
	not enough	**40.6079**	**0.000155**
Baboon 512 × 512	**Max：14488**	**84.9647**	**0.000324**
	Logo：3600	42.3438	0.000162
English 560 × 500	**Max：6333**	**40.3980**	**0.000144**
	Logo：3600	41.9047	0.000150
Chinese Newspaper 640 × 960	**Max：22546**	**99.0858**	**0.000161**
	Logo：3600	42.5323	0.000069
Calligraphy 655 × 735	**Max：10562**	**71.0704**	**0.000147**
	Logo：3600	42.5441	0.000088
Mouse 512 × 704	**Max：5140**	**51.0294**	**0.000141**
	Logo：3600	42.5911	0.000118

(a) Original image

(b) Stego-image

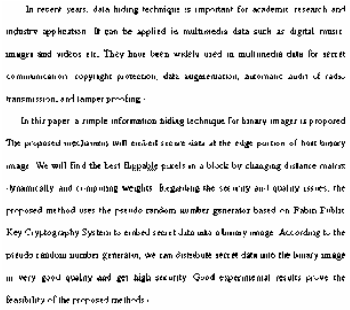
Fig. 12 Peppers

(a) Original image

(b) Stego-image with logo

(c) Stego-image with Max data

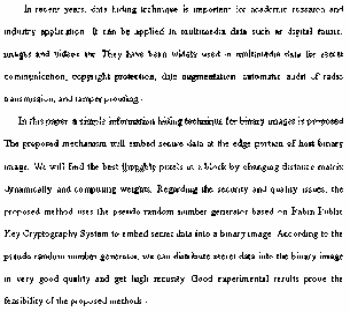
Fig. 13 Baboon



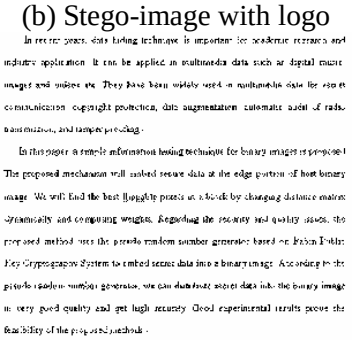
In recent years, data hiding technique is important for academic research and industry application. It can be applied to multimedia data such as digital music images and videos etc. They have been widely used in multimedia data for secret communication, copyright protection, data augmentation, automatic audit of radio transmission, and tamper proofing.

In this paper a simple information hiding technique for binary images is proposed. The proposed mechanism will embed secret data at the edge portion of host binary image. We will find the best flippable pixels on a block by changing distance matrix dynamically and computing weights. Regarding the security and quality issues, the proposed method uses the pseudo random number generator based on Rabin Public Key Cryptography System to embed secret data into a binary image. According to the pseudo random numbers generation, we can distribute secret data into the binary image in very good quality and get high security. Good experimental results prove the feasibility of the proposed methods.

(a) Original image



(b) Stego-image with logo

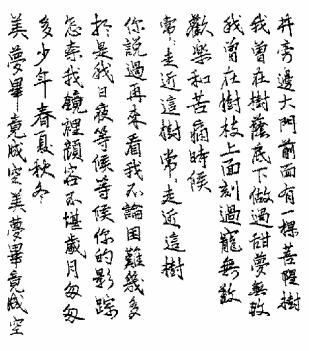


(c) Stego-image with Max data
Fig. 14 English

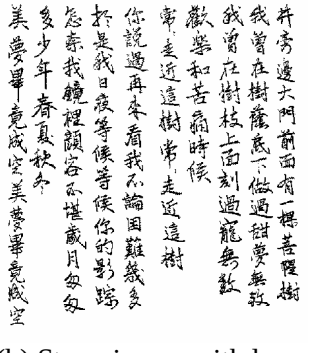
(a) Original image

(b) Stego-image with logo

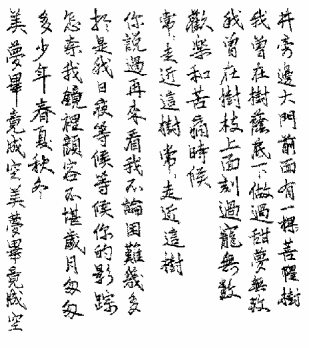
(c) Stego-image with Max data
Fig. 15 Chinese Newspaper



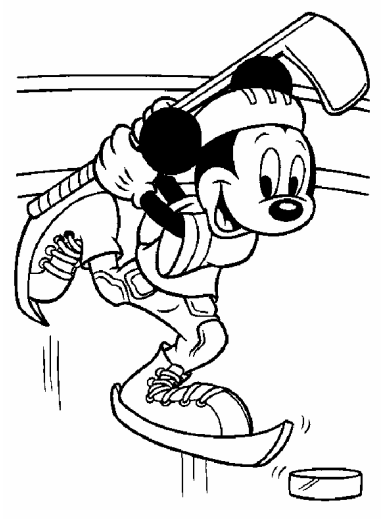
(a) Original image



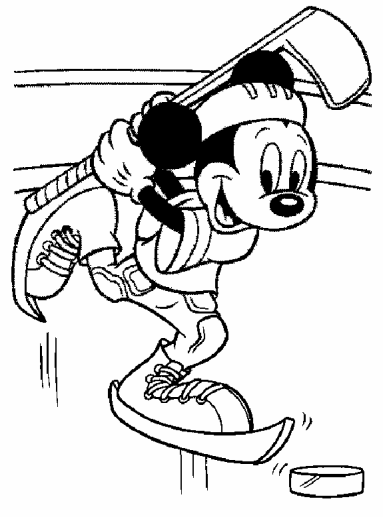
(b) Stego-image with logo



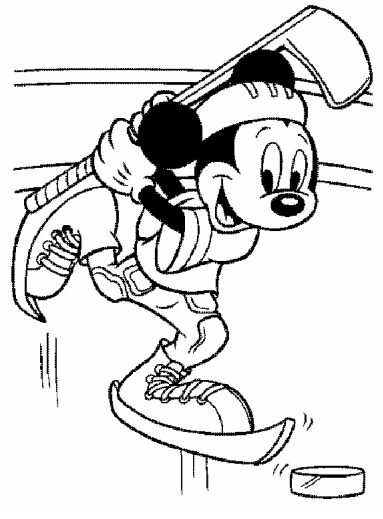
(c) Stego-image with Max data
Fig. 16 Calligraphy



(a) Original image



(b) Stego-image with logo



(c) Stego-image with Max data

Fig. 17 mouse

4. Conclusions

A simple novel and imperceptible data hiding method is proposed. We use the recursive erosion operation of mathematical morphology to find the chessboard distance of the cover image. By using the properties of chessboard distance we may check whether each 3 by 3 block is a good hiding place or not. We set up the embedding locations without using a complicate look-up tables. We also find it can embed data up to 1/9. It is an excellent capacity of a binary cover image. For the security consideration we use the PRNG to scramble the secret data before embedding. The future works we may shift the one or two rows, one or two column to be the starting pixel to divide the cover image into 3 by 3 non-overlapping blocks. We may also reverse embedding 0 by 1 to reduce the MSE. We can compare these modified methods with the smallest MSE to get a better stego-image with the smallest distortion.

Reference

[1] Jeanne Chen, Tung-Shou Chen, and Meng-Wen Cheng, "A New Data Hiding Method in Binary Image," ***Proceedings of the IEEE Fifth International Symposium on Multimedia Software Engineering*** (ISMSE'03)

[2] Ming Sun Fu, and Oscar C. Au, "Data Hiding Watermarking for Halftone Images," ***IEEE TRANSACTIONS ON IMAGE PROCESSING,*** VOL. 11, NO. 4, APRIL 2002, pp.477-484

[3] Ming Sun Fu, Oscar C. Au, "Data Hiding in Halftone Images by Conjugate Error Diffusion," **2003 IEEE**, II921-II923

[4] Hae Yong Kim, Amir Afif, "Secure Authentication Watermarking for Binary Images," ***Proceedings of the XVI Brazilian Symposium on Computer Graphics and Image Processing*** (SIBGRAPI'03)

[5] Haiping Lu, Alex C. Kot, and Rahardja Susanro, "BINARY IMAGE WATERMARKING THROUGH BIASED BINARIZATION," ***IEEE International Conference on Multimedia & Expo*** (ICME) 2003, pp.III.101 - III.104

[6] Haiping Lu, Xuxia Shi, Yun Q. Shi,

Alex C. Kot and Lihui Chen "Watermark Embedding in DC Components of DCT for Binary Images," ***2002 IEEE***

[7] Shao-Hui Liu ,Tian-Hang Chen, Hong-Xun Yao, and Wen Gao, "A VARIABLE DEPTH LSB DATA HIDING TECHNIQUE IN IMAGES," ***Proceedings of the Third International Conference on Machine Learning and Cybernetics***, Shanghai, 26-29 August 2004

[8] F. Y. Shih and C.-H. T. Yang, "On solving exact Euclidean distance transformation with invariance to object size," ***Proc. IEEE Conf. Computer Vision and Pattern Recognition***, New York City, NY, June 1993, pp. 607-608.

[9] Phil Sherry and Andreas Savakis, "IMPROVED TECHNIQUES FOR WATERMARKING HALFTONE IMAGES," ***ICASSP 2004***, pp.V1005-V1008

[10] Chih-Hsuan Tzeng and Wen-Hsiang Tsai, "A New Approach to Authentication of Binary Images for Multimedia Communication With Distortion Reduction and Security Enhancement," ***IEEE COMMUNICATIONS LETTERS***, VOL. 7, NO. 9, SEPTEMBER 2003, pp. 443-445

[11] Yu-Chee Tseng and Hsiang-Kuang Pan, "Secure and Invisible Data Hiding in 2-Color Images," ***IEEE INFOCOM*, 2001**, pp. 887-896

[12] Chun-Chieh Wang and Long-Wen Chang, "High Capacity Binary Image Steganography," ***18th Conference on Computer Vision, Graphics and Image Processing*** (CVGIP 2005), Taipei, ROC, August, 2005, pp. 1309-1316

[13] Hsi-Chun Alister Wang, "Data Hiding Techniques for Printed Binary Images," ***IEEE 2001***

[14] Min Wu, and Bede Liu, "Data hiding in binary image for authentication and annotation," ***IEEE TRANSACTIONS ON MULTIMEDIA,*** VOL. 6, NO. 4, AUGUST 2004, pp. 528-538

[15] Huijuan Yang and Alex C. Kot, "DATA HIDING FOR BI-LEVEL DOCUMENTS USING SMOOTHING TECHNIQUE," ***IEEE International Symposium on Circuits and Systems*** (ISCAS) 2004, pp. V692-V695