A Novel High Capacity Data Hiding Method Based on Exclusive-OR in Binary Images

Chyuan-Huei Thomas Yang#1, Ssu-Yi Liu#2

#*Department of Computer Science, Hsuan Chuang University*
48, Hsuan Chuang Rd., Hsin-Chu City, ROC
[1]chyang@hcu.edu.tw
[2]MA0966016@wmail.hcu.edu.tw

***Abstract*— In this paper we propose a high capacity data hiding method working on binary images. Hiding data in a binary image is more difficult since it has only black or white two colors. It is a trade off between capacity and imperception. We focus on the capacity in this paper. In general we shuffle the secret data by a pseudo-random number generator before hiding to support more secure. We divide the cover image, which the secret data will be concealed, into non-overlapping four by four sub-blocks. Then we repartition each four by four sub-block into four overlapping three by three sub-blocks. Due to consider embedding more data in the cover image, we only skip the all blacks or all whites in four by four sub-blocks. It avoids to observed easily and prevent perceptible simply. We embed the secret data in each four by four sub-block if it is not all the same color. We consider all four three by three sub-block to check the XOR between upper left and center, upper right and center, bottom left and center or bottom right and center, then embed one bit in each three by three sub-block. The extraction way is simply to test the XOR between the four corner pixels and centers of each three by three sub-block. If XOR equal 1, the embedding bit is 1, otherwise is 0. All embedding bits are collected and shuffled back to the original order. The secret data are extracted completely. The experimental results show that the method provides the large embedding capacity and keeps imperceptible.**

***Keywords*— data hiding, steganography, high capacity, XOR, imperceptible.**

1. INTRODUCTION

It is very necessary and important for data security on the Internet. We may use cryptography or steganography or both to prevent the secret information exposure. Data hiding and watermarking are broadly studied and applied in digital media. Many researchers work on the ownership protection, authentication, copyright, and annotation for color or greyscale images. Hiding data in a binary image is more difficulty since it has less information can be used. It should be noticeable clearly if it is not hidden around the edge pixels. We category the approach of the reference papers as pattern comparison (e.g. [1], [5], [9]), block based (e.g. [2], [6], [10], [15]), morphological method (e.g. [8], [11], [12]), distance function (e.g. [4], [12], [14]) and others (e.g. [3], [7]). Of course, some of them are in multiple categories.

The pattern comparison approach: Ajetrao et al [1] defined a flippability score to determine the smoothness. They considered the measurement of the total number of horizontal, vertical, diagonal, and anti-diagonal four transitions in a 3x3 block respectively and used a differential operator along the corresponding directions to give the highest priority of the flipping pixel. Liang et al [5] also adopted the block data hiding method. They divided cover image into two by two sub-blocks. These sub-blocks were classified as embeddable or non-embeddable blocks according to their characteristic values. There are five different characteristic values from 0 to 4 that represent the number of white pixels. The characteristic values will be changed after embedding bit. Yang and Kot [9] used a Gold-like sequence that combined two reciprocal sequences generated from two polynomials and added two sequences chip by chip by synchronous clocking with shift operator. They defined eight denoise patterns based on five by five neighborhood. The denoise mechanism handles the boundary noise, erosion to work as hiding data. The block based approach: Chen *et al* [2] partitioned the cover image into non-overlapping four by four sub-blocks then repartitioned each sub-block into three by three overlapping sub-block. They called it block data hiding method (BDHM). The candidate of embedding position depends on the distribution of black and white or said characteristics in each four by four sub-block. Pan *et al* [6] divided cover image into four by four sub-blocks, called sup-blocks then divided each sup-block into four three by three sub-blocks. They gave the sub-block a level number (rank) according its pattern, representative the affect on perceptible by the changed center pixels in this sub-block. They embedded one bit into one sup-block. Yang and Kot [10], they employed the interlace morphological binary wavelet transform as the basic tool and preserved the connectivity of pixels in a local 3 by 3 neighborhood and dynamically improving the flippability decision. Different from the block-based approach, they considered 2 by 2 blocks inside each 3 by 3 block. Zongqing and Hongbin [15] partitioned cover binary image into two level blocks, and the embedding data are embedded into the sub-blocks based on the parity of characteristic values. The method is very efficient especially when applied to those binary images have black and white pixels that distributed around uniformly. They also use distance matrix to maintain the visual condition. The morphological method approach: Wu and Wang [8] applied the morphology, including erosion and hit-miss transform with three structuring elements. They use the hit-miss transform to interpolate the cover image to create more bits. New cover image has corners. The jaggy shape forms from corners are the candidates of the embedding positions. It also concerns with the magnified times. They used the same interlace morphological binary wavelet transform in [10] to track the shifted edges. The two processing cases that flipping the candidates of one does not affect the flippability conditions of another are employed such that a large capacity can be achieved. Since large capacity sacrifices the visual quality of the stego-image. They provided a backward-forward minimization method to minimize the visual distortion for double processing cases [11]. Yang and Lee [12] proposed 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 the complicate set of look-up tables. The distance function approach: Ho *et al* [4] built a perceptual model with the 8-directional chain code and Euclidean distance to define a curvature-weighted distance to compute the contour segment of original and watermark whether is within 0 and 1 to determine embedding. Zhang and Qiu [14] employed a secret key and a weight matrix to shield the hidden data. They claimed their scheme can embed as high as bits data at cost of two pixels flipped. Pixel flippability leads the whole scheme to control the stego-image quality. Shuffling was also implemented to improve the hiding capacity and made the scheme more secure all together. Some of other approaches: Guo [3] applied the human visual system (HVS) with least-mean-square (LMS) to improve the pair toggling data hiding method and the author claim it is better than Data Hiding Smart Pair Toggling (DHSPT) in halftone images. Tseng *et al* [7] used a weight mechanism to locate the most appropriate pixel for flipping. They used four directions horizontal, vertical, diagonal, and anti-diagonal in this weight mechanism. They accumulated these weight matrices to determine the suitable flipping pixels. Ten boundary patterns are shown to prevent the noticeable distortion.

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 we give several computer experiment results and comments. Finally, in the conclusion we summarize our major results and outline our future work.

2. PROPOSED METHOD

In this section we propose a block based data hiding method in binary images. It is hard to hide data in binary image since it has only two colors to be able to use. Here we briefly describe the proposed embedding algorithm. First, we shuffle the secret data by a pseudo-random number generator (PRNG). We divide the cover image, into non-overlapping four by four sub-blocks. In each four by four sub-block we partition it into four overlapping three by three sub-blocks. We embed the secret data in each four by four sub-block if it is not all the same color. We consider all four three by three sub-block in each four by four sub-blocks and check the XOR between upper left and center, upper right and center, bottom left and center or bottom right and center, then compare the embedding bit to decide 0 or 1 to be embedded in each three by three sub-block. The extraction algorithm just examines the XOR between the four corner pixels and centers of each three by three sub-block. The embedded bit is the same as the value of XOR. We give the embedding algorithm and extraction algorithm next two subsections.

2.1. Embedding Algorithm

Step 1: Convert the secret data *S* into a binary bit stream S then use pseudo-random number generator (PRNG) to reorder the bit stream S_r. $|S_r|$ is the length of S_r.

Step 2: Divide the cover image $C_{m \times n}$ into 4 by 4 non-overlapping blocks, denoted B_i, where i=1 to $\lfloor m/4 \rfloor \times \lfloor n/4 \rfloor$.

Step 3: Each sub-block B_i is divided into four 3 by 3 overlapping blocks, said B_{ij}, where j=1 to 4 (e.g. Fig. 1).

Step 4: Convert $\lfloor m/4 \rfloor \times \lfloor n/4 \rfloor \times 4$ into binary number *L*. $|L|$ is the number of bits of *L*. We use $|L|$ bits to save $|S_r|$ then add L to the head of S_r.

Step 5: If there have no bit left in $L+S_r$ or sub-block B_i is running out, then Stop.

Step 6: If B_i has all zeros or all ones (e.g. Fig. 2), then do nothing.

Step 7: If $a_j \oplus b_j$ in B_{ij} equals current embedding bit in $L+S_r$, B_{ij} does not modified and it is embedded that bit ($= a_j \oplus b_j$).

else (note: $a_j \oplus b_j$ in B_{ij} is not equal to current embedding bit in $L+S_r$)
  if sum(B_{ij})=2~7,
    if a_j=1, b_j=~b_j, else a_j=~a_j (e.g. Fig. 3)
  else if sum(B_{ij})=0, then a_j=1,(e.g. Fig. 4)
  else if sum(B_{ij})=9, then a_j=0,(e.g. Fig. 5)
  else (sum(B_{ij})=1)
    if $a_j \oplus b_j$ =1 (note: $a_j \neq b_j$), then a_j=b_j=1
    else (note: $a_j \oplus b_j$ =0) a_j=1.

Step 8: If there still exists embedding bit in $L+S_r$, go to Step 5.

2.2. Extraction Algorithm

Step 1: Divide the Stego-image $C'_{m \times n}$ into 4 by 4 non-overlapping blocks, denoted B_i', where i=1 to $\lfloor m/4 \rfloor \times \lfloor n/4 \rfloor$.

Step 2: Each sub-block B_i' is divided in to four 3 by 3 overlapping blocks, said B_{ij}', where j=1 to 4.

Step 3: Convert $\lfloor m/4 \rfloor \times \lfloor n/4 \rfloor \times 4$ into binary number *L*. $|L|$ is the length of *L*.

Step 4: For each sub-block B_{ij}',
  If B_i has all zeros or all ones, then skip to next sub-block,
  else we collect first $|L|$ bits, where the bit equals $a_j \oplus b_j$ in B_{ij}'.

Step 5: Convert the first $|L|$ bits into a decimal number. It should equal to $|S_r|$, the length of embedding bit stream.

Step 6: For each sub-block B_{ij}', (next 3 by 3 sub-blocks of the end of Step 4)
  If B_i has all zeros or all ones, then skip to next sub-block,
  else we collect $|S_r|$ bit into a bit stream S_r', where the bit equals $a_j \oplus b_j$ in B_{ij}'.

Step 7: Use the same PRNG in embedding algorithm to transfer S_r' back to the original permutation of the embedding bit stream S_r.

Step 8: Convert S_r into secret data.

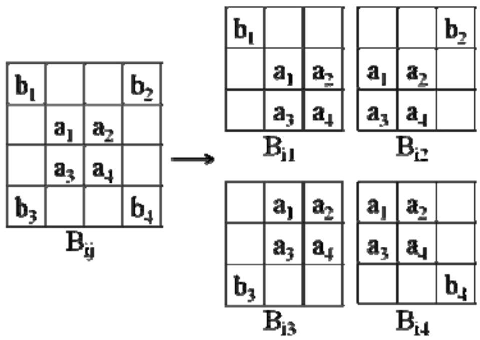


Fig. 1 Partition the cover image into 4 by 4 sub-blocks, then repartition into four overlapping 3 by 3 sub-blocks

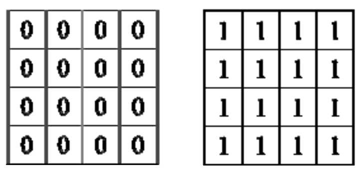
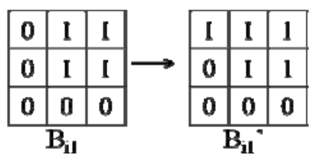
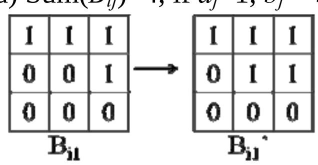


Fig. 2 All zeros (black) or ones (white)



(a) Sum(B_{ij})=4, if a_j=1, b_j=~b_j



(b) Sum(B_{ij})=4, if a_j=0, a_j=~a_j

Fig. 3 A 3 by 3 block has four ones

$$\begin{array}{c}\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}\\ B_{il}\end{array} \longrightarrow \begin{array}{c}\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}\\ B_{il}'\end{array}$$

Fig. 4 A 3 by 3 block has all zeros

$$\begin{array}{c}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\\ B_{il}\end{array} \longrightarrow \begin{array}{c}\begin{bmatrix}1&1&1\\1&0&1\\1&1&1\end{bmatrix}\\ B_{il}'\end{array}$$

Fig. 5 A 3 by 3 block has all ones

3. EXPERIMENTAL RESULTS

We examine our proposed method with several popular images that many authors usually test their proposed methods. Table 1 demonstrates those cover images sizes, embedding capacities, the mean square error, and the Capacity/(size of Cover Image). The mean square error is

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

The testing secret data is a logo of Hsuan Chuang University which size is 100 by 100 in Fig. 6. This logo consists of 10000 bits.

Fig. 6 The logo of Hsuan Chuang University

According to the computer experiments we find the "Peppers" has the least capacity. It should have many large areas of all whites or all blacks. Actually we may understand the maximum capacity should be 1/4, due to for each 4 by 4 block we place at most four bits in it. Fig. 7~15 are Peppers, Baboon, English, Chinese Newspaper, Calligraphy, Mouse, Boat, Mountain, and Barbara.

TABLE 1
THE MSE AND CAPACITY OF COVER IMAGES, WHERE C/CI MEANS CAPACITY/(SIZE OF COVER IMAGE)

Cover Image	Capacity(bits)	MSE	C/CI
Peppers 512 * 512	Max：10540	70.0643	0.040207
	Logo：10000	69.4622	0.038147
Baboon 512 * 512	Max：40240	148.0000	0.153503
	Logo：10000	70.1071	0.038147
English 570 * 500	Max：16692	80.8764	0.058568
	Logo：10000	69.6419	0.035088
Chinese Newspaper 640 * 960	Max：61704	163.7101	0.100430
	Logo：10000	70.5195	0.016276
Calligraphy 655 * 735	Max：30944	96.3431	0.064276
	Logo：10000	69.7065	0.020772
Mouse 512 * 704	Max：16180	66.7233	0.044889
	Logo：10000	69.0435	0.027743
Boat 512 * 512	Max：14100	81.1788	0.053787
	Logo：10000	70.0571	0.038147
Mountain 640 * 480	Max：37568	119.6787	0.122292
	Logo：10000	70.0000	0.032552
Barbara 512 * 512	Max：20932	105.5936	0.079849
	Logo：10000	69.9071	0.038147

The ratios of C/CI with "Baboon" are 0.014587 and 0.018749 in [2] and [15], respectively. In our results we have 0.153503 (maximum capacity) and 0.038147 (HCU logo embedded). Both are better than theirs.

4. CONCLUSIONS

A novel and imperceptible block based data hiding method in binary image is proposed. We divide the cover image, which the secret data will be hidden into non-overlapping four by four sub-blocks. We repartition each four by four sub-block into four overlapping three by three sub-blocks. Due to consider embedding more data in the cover image, we only skip the all blacks or all whites in four by four sub-blocks. It avoids to observed easily and avoid perceptible simply. The theoretical result show the cover image can embed data up to 1/4 of its size. 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 4 by 4 non-overlapping blocks. We may also reverse embedding 0 by 1 to reduce the MSE. We may compare these modified methods with the smallest MSE to get a better stego-image with the smallest distortion.

REFERENCES

[1] Hema Ajetrao, Dr.P.J.Kulkarni, and Navanath Gaikwad, "A Novel Scheme of Data Hiding in Binary Images," *International Conference on Computational Intelligence and Multimedia Applications*

(ICCIMA 2007), Dec. 2007 Page(s):70-77.

[2] Jeanne Chen, Tung-Shou Chen, and Meng-Wen Cheng, " A new data hiding method in binary image," *International Symposium on Multimedia Software Engineering (ISMSE'03)*, 2003 Page(s):88-93.

[3] Jing-Ming Guo, "Improved Pair Toggling Data Hiding by Cooperating Human Visual System in Halftone Images," *International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, April 2007 Page(s):II285-II288.

[4] A.T.S. Ho, N.B. Puhan, A. Makur, P. Marziliano, and Y.L. Guan, "Imperceptible data embedding in sharply-contrasted binary images," *International Conference on Control, Automation, Robotics and Vision Conference,* China, Dec. 2004 Page(s):958-963 Vol. 2.

[5] Hua-qing Liang, Wen-bo Ran, and Xin-xin Niu, "A secure and high capacity data hiding scheme for binary images," *International Conference on Wavelet Analysis and Pattern Recognition,* Beijing, China, Nov. 2007 Page(s):224-229.

[6] Gang Pan, Yijun Wu, and Zhaohui Wu, "A Novel Data Hiding Method for Two-Color Images", *ICICS 2001*, Page(s):261-270.

[7] Hsien-Wen Tseng, Feng-Rong Wu, and Chi-Pin Hsieh, "Data Hiding for Binary Images Using Weight Mechanism," *Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2007)*, Nov. 2007 Page(s):307-310.

[8] Yajuan Wu, Minghui Wang, and Xiaofeng Liu, "Magnifying Binary Image based on Morphology," *International Conference on Image and Graphics (ICIG 2007)*, Aug. 2007 Page(s):26-31.

[9] Huijuan Yang and Alex C. Kot, "Data hiding for bi-level documents using smoothing technique," *International Symposium on Circuits and Systems (ISCAS '04)*, May 2004 Page(s):V692-V695 Vol.5.

[10] Huijuan Yang and Alex C. Kot, "Data Hiding For Binary Images Authentication By Considering A Larger Neighborhood," *International Symposium on Circuits and Systems (ISCAS 2007),* May 2007 Page(s):1269-1272

[11] Huijuan Yang and Alex C. Kot, "Backward-Forward Distortion Minimization for Binary Images Data Hiding," *IEEE International Symposium on Circuits and Systems (ISCAS 2008)*, May 2008 Page(s):404-407.

[12] Chyuan-Huei Thomas Yang and Ken-Ching Lee, "A data hiding technique in binary image by using distance transform without look-up tables," *2007 International Conference on Advanced Information Technologies (AIT)*, Taichung , April 2007.

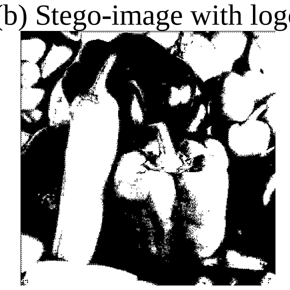
[13] Saif Zahir and Mehmood Naqvi, "A Near Minimum Sparse Pattern Coding Based Scheme for Binary Image Compression," *IEEE-ICIP 2005,* Genoa, Italy, Sept. 2005 Page(s):II - 289-92.

[14] Chun-E Zhang and Zheng-Ding Qiu, "Fragile watermarking with quality control for binary images," *International Conference on Machine Learning and Cybernetics,* Guangzhou, Aug. 2005 Page(s):4952-4956 Vol. 8.

[15] Li Zongqing and Zhang Hongbin, "A New Data Hiding Method in Binary Images," *International Conference on Innovative Computing, Information and Control (ICICIC'06)*, Aug. 2006 Page(s):66-69.

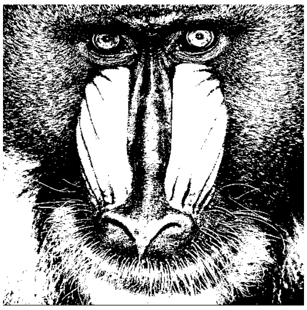
(a) Original image

(b) Stego-image with logo

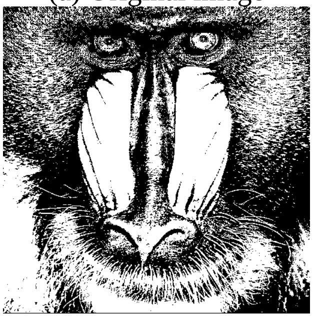


(c) Stego-image with maximum capacity

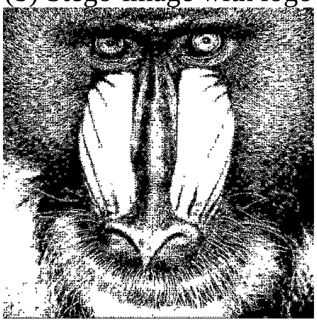
Fig. 7 Peppers



(a) Original image

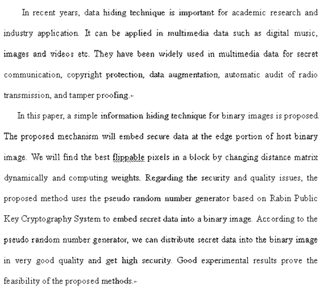


(b) Stego-image with logo



(c) Stego-image with maximum capacity

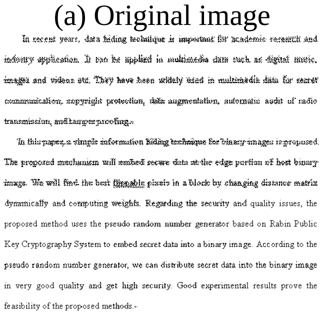
Fig. 8 Baboon



In recent years, data hiding technique is important for academic research and industry application. It can be applied in 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 secure data at the edge portion of host binary image. We will find the best flippable pixels in 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 number generator, 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.

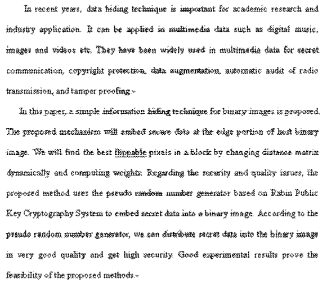
(a) Original image



In recent years, data hiding technique is important for academic research and industry application. It can be applied in 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 secure data at the edge portion of host binary image. We will find the best flippable pixels in 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 number generator, 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.

(b) Stego-image with logo



In recent years, data hiding technique is important for academic research and industry application. It can be applied in 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 secure data at the edge portion of host binary image. We will find the best flippable pixels in 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 number generator, 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.

(c) Stego-image with maximum capacity

Fig. 9 English

聯合筆記

阿土伯老了

華英惠

(a) Original image

聯合筆記

阿土伯老了

華英惠

(b) Stego-image with logo

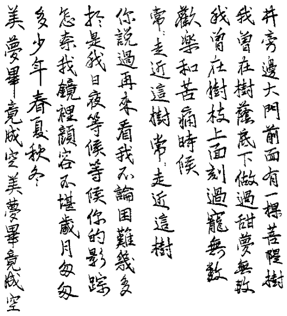
聯合筆記

阿土伯老了

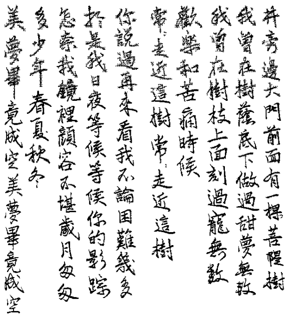
華英惠

(c) Stego-image with maximum capacity

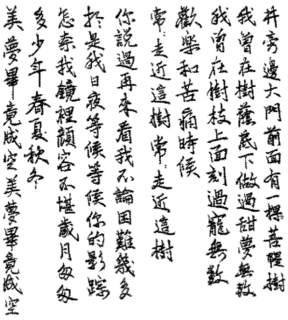
Fig. 10 Chinese Newspaper



(a) Original image

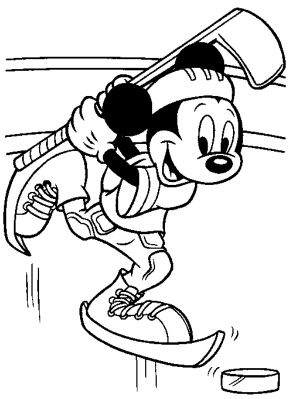


(b) Stego-image with logo

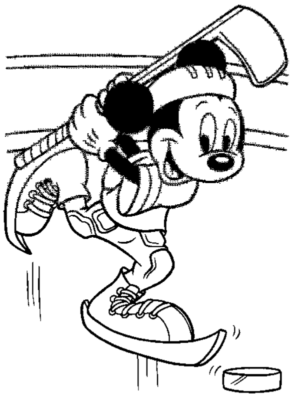


(c) Stego-image with maximum capacity

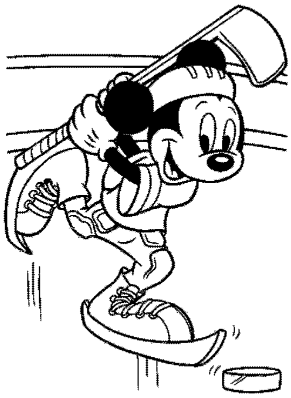
Fig. 11 Calligraphy



(a) Original image



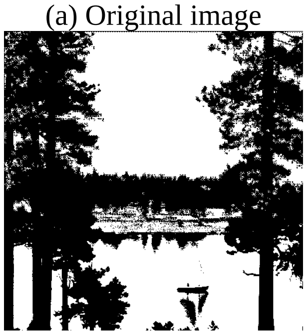
(b) Stego-image with logo



(c) Stego-image with maximum capacity

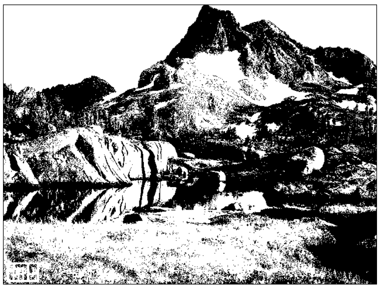
Fig. 12 Mouse

(a) Original image

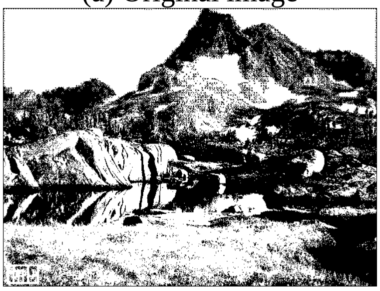


(b) Stego-image with logo

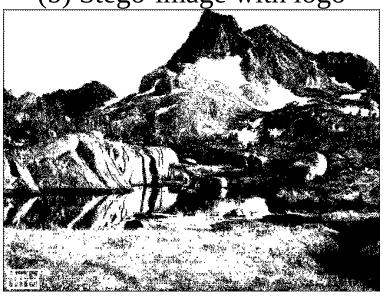
(c) Stego-image with maximum capacity
Fig. 13 Boat



(a) Original image



(b) Stego-image with logo



(c) Stego-image with maximum capacity
Fig. 14 Mountain

(a) Original image

(b) Stego-image with logo

(c) Stego-image with maximum capacity
Fig. 15 Barbara