STEGANALYSIS OF BINARY CARTOON IMAGE USING DISTORTION MEASURE

Jun Cheng, Alex C. Kot* and Susanto Rahardja†*
*School of EEE, Nanyang Technological University, Singapore 639798
Email: cjun@pmail.ntu.edu.sg
† Institute for Infocomm Research, A-STAR, Singapore
Email: rsusanto@i2r.a-star.edu.sg

ABSTRACT

We present a steganalysis technique for data hiding in binary cartoon images. Due to the perturbation from the embedding, the contours of a stego binary cartoon image are distorted. When calculating the distortion of the image based on a distortion measure, the distortion score between a stego image and its de-noised version should be different from that between an original image and its de-noised version. Different binary image distortion measures are used to calculate the distortion scores, which are used as the features for classification. The sequential floating forward search (SFFS) method is used to search for the combination of the features that yields the best classification results.

***Index Terms* – steganalysis, binary image**

1. INTRODUCTION

In recent years, several techniques have been developed for hiding information in binary images by changing the values of individually selected pixels such as the work in [1-6]. These pixel flipping techniques hide data by flipping pixels. With the popularity in developing data hiding and watermarking techniques, steganalysis also starts receiving attentions. Several schemes [7, 8] for the steganalysis of binary text images using the correlation between similar characters or symbols within the same image are proposed. However, such correlation does not exist in binary cartoon images and these schemes cannot work. In this paper, we aim at steganalysis of data hiding in binary cartoon images.

As mentioned in [9], distortion measure can be used in steganalysis. The distortion measure based score is usually computed by comparing the distorted image with the original one. However, the original image is not available in the steganalysis. Instead, we perform de-noising process to obtain a de-noised image that is close to the original one. The rationale of the technique is that an embedding process creates distortion on the contours of the binary images and makes the contour lines less smooth after information hiding. Thus, we can de-noise the stego image by predicting and smoothing the edge contours. It is expected that the distortion in a de-noised image from its original image is different from the distortion in a de-noised image from its stego image. Based on this observation, we design a classifier to separate the stego image from original image.

In the next section, we describe the distortion control in the existing embedding algorithms. In Section 3, we propose two de-noising process that can remove edge noise in binary images. Feature extraction and experimental results are given in Section 4. The conclusions are in the last section.

2. DISTORTION CONTROL IN EMBEDDING

To start with, we give a brief review of the embedding schemes by studying how the visual distortion is minimized. Arbitrarily flipping of pixels in a binary image may cause visible distortion, especially when flipping a pixel from pure white or black region. In [1], the scheme uses a condition that only pixels along edges can be flipped. However, this may not be adequate to preserve the quality of the image. Flipping pixels along smooth contours such as straight lines may change the smoothness of the edge contour that causes large distortion in an image. In [2-6], the authors have studied the flippability of each pixel by comparing with its neighboring pixels. More strict criteria are proposed based on smoothness and connectivity, as discussed in [7]. One of the important observations is that most of these embedding schemes use the center pixel of an *L*-shape pattern (COL pixel) as shown in Fig. 1. There is a total of 16 *L*-shape patterns after taking complement, mirroring and rotation into consideration. Very often, COL pixel is chosen in most of these schemes as flippable pixel with high priority as it does not affect the smoothness and connectivity of the contours [10].

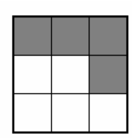


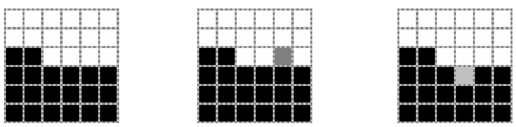
Fig. 1. An example of an *L*-shape pattern

3. DENOISING PROCESS

After studying the existing pixel flipping schemes, the following two de-noising process are applied to remove noise in different patterns due to the embedding process such that a de-noised image close to the original can be obtained.

A. Morphological smoothing

The flipping of some pixels may change the smoothness of the image. It creates holes or protrusions as shown in Fig. 2. We propose a morphological smoothing process to fill holes and remove protrusions based on the two basic morphological operations: erosion and dilation [11]. Based on the two basic operations, the opening process and the closing process [11] are defined as erosion followed by dilation and dilation followed by erosion, respectively. The opening process is to remove protrusions and the closing process is to fill holes. Thus, we apply a smoothing process as an opening process followed by a closing process to fill holes and remove protrusions in an image.



(a) Original (b) protrusion (c) hole

Fig. 2 Patterns with hole or protrusion

Although the smoothing process can fill holes and remove protrusions due to the embedded data, it may also remove the one-pixel-width strokes in the original image. To preserve the one-pixel-width strokes, any two 8-connected neighboring pixels that are flipped simultaneously by the above smoothing process need to be flipped back. By doing so, we can preserve most of the one-pixel-width strokes in the original image. The trade off is that the embedding noise along the one-pixel-width strokes cannot be removed by the morphological smoothing process. The limitation of the morphological smoothing is that it cannot recover some flipped pixels such as COL pixels as the flipping does not create holes or protrusions. Smoothing scheme does not work well for such embedding techniques using COL pixels.

B. De-noising the stair case patterns

Removing noise pixels by not changing the smoothness and connectivity is difficult. In order to find a way to possibly recover the flipped pixels among these pixels such as COL pixels, we look into how the flipping of these pixels affects the edge contours. Very often, we find stair case patterns in cartoon images. For example, the pattern shown in Fig. 3 is a stair case pattern to represent an analog straight line. COL pixels appear in the pattern. The locations of the corners, i.e., the locations where the edge lines change their directions, are important in the stair case patterns. Here the edge line refers to the line between two neighboring pixels where the pixel values for the two pixels are different [12]. There are eight different types of corners including four black and four white corners to make up all possible corners in a 2×2 pixel block as indicated in Fig. 4. The four pixels that form the corner are the related pixels of the corner. Obviously, the flipping of a COL pixel shifts two corners by one pixel.

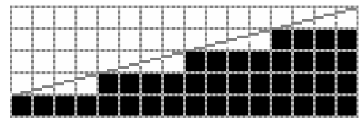


Fig. 3 Digital representation of analog line

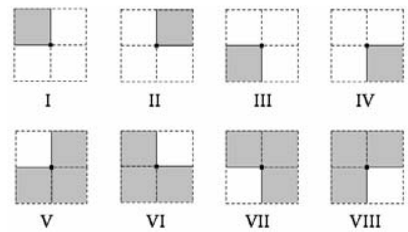


Fig. 4 Eight types of corners with black and white corners in the 1^{st} and 2^{nd} row respectively

As the consecutive corners of same type along one contour are correlated very often when they are close, we can predict the location of a corner based on its nearby preceding and subsequent corners. For example, the type IV corners in Fig. 3 can be predicted well by a linear equation. On the other hand, the corners are shifted due to the flipping in the embedding. Suppose the prediction error of the corner location before flipping is very small, then the resultant corner after flipping tends to stay further to the predicted location. Based on this observation, we propose a de-noising technique by shifting the corner back closer to the predicted location. The shifting is done by flipping a pixel along stair case pattern. Since the locations of black and white corners are dependent with each other, we use black corners only. Without lost of generality, we first study the case that the corners are shifted horizontally.

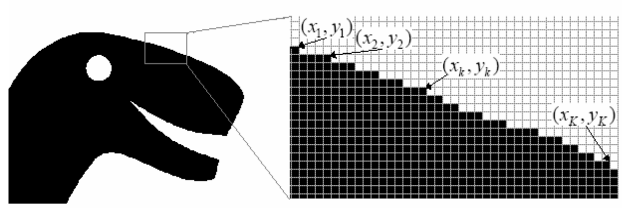


Fig. 5 Consecutive same type corners

We first obtain the locations of the corners in the sequence determined by the contour tracing. Next, the corners are divided into groups with each group formed by same type black corners consecutively. Suppose a group of

K corners is formed. The coordinates of these corners are denoted as (x_k, y_k), $k = 1,...,K$ (see Fig. 5). Since the corners are shifted horizontally, we consider y_k as a function of x_k. Assume that the coordinates of the corner (x_k, y_k) together with its neighboring corners can be approximated by a polynomial curve $f(x)$ with highest degree m. The function can be written as:

$$f(x) = \sum_{i=0}^{m} a_i x^i \tag{1}$$

To estimate the column coordinate y_k for the k^{th} corner, we need to calculate the coefficients a_i, for $i = 0,1,...,m$, based on $m+1$ sampling points from $f(x)$. Obviously, one of the choices is to use the $\left[\frac{m+1}{2}\right]$ preceding and $\left[\frac{m}{2}\right]$ subsequent corners as the $m+1$ sampling points, where $[\cdot]$ is the 'round' operation to an integer. We have:

$$\begin{cases} y_{k-\left[\frac{m+1}{2}\right]} = a_0 + a_1 x_{k-\left[\frac{m+1}{2}\right]} + a_2 x_{k-\left[\frac{m+1}{2}\right]}^{\;2} ... + a_m x_{k-\left[\frac{m+1}{2}\right]}^{\;m} \\ y_{k-\left[\frac{m+1}{2}\right]+1} = a_0 + a_1 x_{k-\left[\frac{m+1}{2}\right]+1} + a_2 x_{k-\left[\frac{m+1}{2}\right]+1}^{\;2} ... + a_m x_{k-\left[\frac{m+1}{2}\right]+1}^{\;m} \\ \vdots \\ y_{k-1} = a_0 + a_1 x_{k-1} + a_2 x_{k-1}^{\;2} ... + a_m x_{k-1}^{\;m} \\ y_{k+1} = a_0 + a_1 x_{k+1} + a_2 x_{k+1}^{\;2} ... + a_m x_{k+1}^{\;m} \\ y_{k+2} = a_0 + a_1 x_{k+2} + a_2 x_{k+2}^{\;2} ... + a_m x_{k+2}^{\;m} \\ \vdots \\ y_{k+\left[\frac{m}{2}\right]} = a_0 + a_1 x_{k+\left[\frac{m}{2}\right]} + a_2 x_{k+\left[\frac{m}{2}\right]}^{\;2} ... + a_m x_{k+\left[\frac{m}{2}\right]}^{\;m} \end{cases} \tag{2}$$

Solve the $m+1$ equations to obtain the coefficients a_i and to estimate y_k for the given row x_k. Denote the estimated y_k based on corners from both sides as y_k^B.

$$y_k^B = \sum_{i=0}^{m} a_i x_k^i \tag{3}$$

The prediction error is given by:

$$e_k^B = \left| y_k^B - y_k \right| \tag{4}$$

Besides the above estimation, we may also estimate y_k based on the $m+1$ preceding corners or the $m+1$ subsequent corners if exist. These two estimations are important because of the existence of singular points. Around a singular point, the edge contour may be well predicted based on corners from one side but not from both sides. Correspondingly, we can obtain two estimations y_k^L and y_k^R based on the $m+1$ preceding corners and the $m+1$ subsequent corners, respectively. The prediction errors denoted as e_k^L and e_k^R can be computed similar to (4).

Suppose we shift the black corner at (x_k, y_k) to $(x_k, y_k \pm 1)$ by flipping a pixel along the stair case pattern. After the shifting, we can compute the three new prediction errors $\widehat{e}_k^L$, $\widehat{e}_k^B$ and $\widehat{e}_k^R$. We shift all the black corners along the stair case pattern if the shifting satisfies the following conditions: After shifting, $\widehat{e}_k^L < e_k^L$, $\widehat{e}_k^B < e_k^B$ and $\widehat{e}_k^R < e_k^R$. It should be noticed that some of the three estimations may not be applicable in some cases and the corresponding errors cannot be computed. In that case, we ignore the inapplicable errors. When only one of the three estimations is applicable, we have an additional requirement that the new error should be zero. For example, we only have $\widehat{e}_k^L$ and e_k^L for the last corner in each group, thus we require $\widehat{e}_k^L = 0$. This additional condition is to reduce the chance of flipping original pixels at the end point.

When flipping a pixel shifts the corner vertically, we consider x_k as a function of y_k instead. The other steps are similar.

4. FEATURE EXTRACTION AND EXPERIMENTAL RESULTS

We apply the morphological smoothing first and then de-noise the stair case patterns. Once we obtained the de-noised image, we calculate the distortion scores using various distortion measures: Distance reciprocal distortion measure (DRDM) [13], Change of smoothness and connectivity measure (CSCM) [10], Edge line segment similarity measure (ELSSM) [12] and Mean Square Error (MSE). For DRDM and CSCM, two different block sizes 3×3 and 5×5 are used, respectively. Thus, for each image, we have six distortion measure based features. To achieve better results, we use sequential floating forward search methods (SFFS) [14] to search for the combination of features that yields the best classification results. We use support vector machines as the classifier and the implementation from LIBSVM [15] are used. We try different m values and find that $m=1$ gives the best classification results as the polynomial using $m=1$ gives better estimations than the other m values.

To evaluate the proposed steganalysis technique, we have conducted experiments on a large binary image database of more than 5000 images from Laboratory for Engineering Man/Machine Systems (LEMS) [16]. The following six different embedding schemes are used in our experiments: Tseng's scheme [1], Wu's scheme [3], Yang's scheme [6], Kim's scheme [5], Pan's scheme [4] and Mei's scheme [2]. In particular, for Wu's scheme and Yang's scheme, three different block sizes are tested respectively. Random messages are generated and embedded by different schemes to generate the stego images.

We randomly select half of the original images and the corresponding stego images for training. The rest are used for testing using the trained SVM. From SFFS, we find that for Tseng's scheme, we only need ELSS based feature to get the best result. The two features based on CSCM are not useful for Mei's scheme. For the rest, the best combination of features includes all the six features. The results based on the combination of features that yields the best results are given in Table I. From the results, we can see that we can classify all the stego images created by Tseng's method. It implies the limitation of the flipping that doesn't take smoothness into consideration. For other schemes which mainly use COL pixels, the capacity is the major reason that affects the results. For Mei's scheme, the performance is the worst as the capacity is the lowest.

We also evaluate the system for the ensemble of the different schemes. Different set of 500 original images are embedded with each embedding scheme to obtain 5000 stego images. Then half of the original images and the stego images created by each of embedding schemes are used for training and the other half are used for testing. From SFFS, the best combination of features includes all the six features. The results of false alarm and missing rate are given in Table II.

Table I. The detection results for individual scheme

Embedding Scheme	False Alarm	Missing Rate
Tseng	0.0%	0.0%
Wu's (8x8)	4.5%	2.9%
Wu's (12x12)	11.6%	9.9%
Wu's (16x16)	13.8%	12.8%
Yang's (3x3)	14.6%	13.0%
Yang's (4x4)	10.1%	8.9%
Yang's (5x5)	11.7%	8.1%
Kim	18.7%	19.1%
Pan	14.8%	16.6%
Mei	24.7%	30.7%

Table II. The detection result for ensemble schemes

False Alarm	Missing Rate
16.5 %	14.6 %

5. CONCLUSIONS

In this paper, we have proposed a steganalysis technique for binary images. The proposed detection approach is based on the hypothesis that the embedding of the message makes the image less smooth or predictable. Two de-noising techniques for binary images are proposed to remove noisy pixels added into different image patterns. Support vector machines are used to get an optimal classifier. The experimental results show that the steganalysis of hiding scheme that affect smoothness and connectivity is straight forward. The steganalysis of schemes that mostly use COL pixels is more difficult. However, for cartoon images which contain stair case patterns, we can use of the location correlation among close corners for the detection. It implies that the selection of flippable pixels should also take the location correlation among close corners into consideration. The performance could be improved with a better de-noising technique or a better distortion measure. As we assume the edge contour of an image should be smooth before flipping, this detection scheme is not suitable for scan images.

6. REFERENCES

[1] Y. Tseng; Y. Chen; H. Pan, " A secure data hiding scheme for binary images", *IEEE Transactions on Communications,* vol. 50, pp. 1227-1231, 2002.

[2] Q. Mei, E. K. Wong, and N. Memon, "Data hiding in binary text documents," in *Proc.SPIE, Security and Watermarking of Multimedia Contents III*, vol. 4314, pp. 369-375, 2001.

[3] M. Wu and B. Liu, "Data Hiding in Binary Image for Authentication and Annotation," *IEEE Trans. On Multimedia*, vol. 6, NO. 4, pp. 528-538, 2004.

[4] G. Pan, Y. J. Wu, and Z. H. Wu, " A novel data hiding method for two-color images," *Lecture Notes in Computer Science,* vol. 2229, pp. 261-270, 2001.

[5] H. Y. Kim and R. L. Queiroz, "A Public-Key Authentication Watermarking for Binary Images," in *Proc. of IEEE Int. Conf. Image Processing*, vol. 5 , pp.3459 – 3462, 2004.

[6] H. Yang and A. C. Kot, "Data Hiding for Text Document Image Authentication by Connectivity-Preserving," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* vol. II, pp. 505-508, 2005.

[7] J. Cheng, A. C. Kot, J. Liu and H. Cao, "Detection of Data hiding in Binary Text Images," in *Proc. IEEE Inf. Conf. Image Processing*, vol. III, pp.73-76, 2005.

[8] M. Jiang, X.Wu, E.K.Wong, and N. Memon, "Quantitative Steganalysis of binary images," in *Proc. IEEE Int. Conf. Image Processing,* pp. 29-32, 2004.

[9] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics ", *IEEE Trans. On Image Processing,* vol. 12, pp. 221- 229, 2003.

[10] J. Cheng and A. C. Kot, "Objective distortion measure for binary images," in *Proc. IEEE TENCON,* pp. 355-358, 2004.

[11] A. K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cli s, NJ: PrenticeHall, 1989.

[12] J. Cheng and A.C. Kot, "Objective distortion measure for binary text image based on edge line segment similarity", Submitted to *IEEE Trans. On Image Processing*.

[13] H. Lu, A. C. Kot, and Y. Q. Shi, "Distance-reciprocal distortion measure for binary document images," *IEEE Signal Processing Letters*, vol. 11, Issue: 2 , pp. 228 – 231, 2004.

[14] P. Pudil, J. Novovicova and J. Kittler, "Floating search methods in feature selection," *Patttern Recognitions Letters*, vol. 15, pp. 1119-1125, 1994.

[15] C.C. Chang and C.J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at: http://www.csie.ntu.edu.tw /~cjlin/libsvm.

[16] A large binary image database. http://www.lems.brown.edu/~dmc/