Constructive Steganography by Tangles

Zhenxing Qian, Lin Pan, Nannan Huang, and Xinpeng Zhang*
Shanghai Institute for Advanced Communication and Data Science, School of Communication and Information Engineering, Shanghai University
Shanghai, 200444, China
[e-mail: xzhang@shu.edu.cn]
*Corresponding author: Xinpeng Zhang

Received May 3, 2017; revised September 8, 2017; revised January 16, 2018; accepted February 13, 2018; published August 31, 2018

Abstract

This paper proposes a novel steganography method to hide secret data during the generation of tangle patterns. Different from the traditional steganography based on modifying natural images, we propose to construct stego images according to the secret messages. We first create a model to group a selected image contour, and define some basic operations to generate various pattern cells. During data hiding, we create a cell library to establish the relationships between cells and secret data. By painting the cell inside the image contour, we create a dense tangle pattern to carry secret data. With the proposed method, a recipient can extract the secret data correctly. Experimental results show that the proposed method has a flexible embedding capacity. The constructed stego tangle image has good visual effects, and is secure against adversaries. Meanwhile, the stego tangle pattern is also robust to JPEG compression.

Keywords: Steganography, Tangle, Information hiding

This work was supported by Natural Science Foundation of China (Grant 61572308, Grant U1536108, Grant U1736213, Grant U1636206, Grant 61525203, Grant 61472235), the Shanghai Dawn Scholar Plan (Grant 14SG36), and the Shanghai Excellent Academic Leader Plan (Grant 16XD1401200)

1. Introduction

Steganography is an important branch of information security. In order to achieve a hidden transmission of information, steganography uses human's insensitive sensory organs and the existence of multimedia digital signal redundancy to hide secret information into a carrier. Traditional steganography algorithms like LSB and JSteg [1-2] use redundant parts in images to represent the secret information. The common drawback is that the modified steganographic carrier is vulnerable against steganalysis. More secure steganographic methods have been proposed, such as wet paper code (WPC) [3], F5 algorithm [4], ZZW [5], and STC [6]. In WPC, the data hider only uses dry regions to hide secret message, and ignore the wet regions. As a result, the adversary cannot detect which region is used to carry secret bits. In F5 algorithm, matrix embedding is used to hide information in the non-zero AC coefficient, wich increases the embedding efficiency. Zhang et al. proposed the ZZW algorithm which combines hamming code and WPC. The embedding efficiency is much higher than the original methods. STC framework is state-of-the-art algorithm proposed to minimize the additive distortion during data hiding.

Since there have a lot of steganalysis methods designed to combat the high-payload stegos, the traditional steganography methods are not that safe as before. As a results, some novel steganogarhy methods emerge. Instead of embedding secret information by modifying the digital carrier, the data hider constructs the stego image according to the secret bits. The representative scheme is based on texture synthesis, which was first proposed by Otrori and Kuriyama [7-8]. Since this scheme has the drawback of low capacity and error extraction, a new solution was proposed in [9], which can achieve a large capacity for information hiding. Reversible texture synthesis [9] employs a rule of generating an index table and pasting source patches to construct a stego image. This steganography method was analyzed by Zhou et al. and a combating algorithm was proposed in [10], which can detect the steganography behavior and even extract the hidden messages. Qian et al. [11] propose a robust steganography which hides secret messages during the process of synthesizing a texture image. The proposed method provides an approach robustness to JPEG compression. We name these steganography methods as constructive steganography, inspired by the work in [14]. In some other works, it is also named coverless steganography [13].

Among various types of textures, tangle is a new form of abstract textures. Tangle is a unique style of painting which is constructed by repeating recursion, using simple graphic elements (such as lines, dots, polygons). The drawing of tangle is easy to learn, requiring no skills of painting. It is not just a kind of artistic creation, a important role of tangle is to encourage people to enjoy pleasure and relaxation. People can even let the brain overcome the anxiety and fatigue through the process of painting tangle. Many people also use it to relieve stress, which is also known as "mind yoga". As the proces of tangle painting is full of intersting, people can design various patterns relying on their imagination. Besides, the tangle images can be used as a decorative painting. Christian et al [12] proposed a method to procedurally generate tangle patterns by modelling group grammars and combining geometric operators. **Fig. 1** shows an example of tangle which are generated by these grammars.

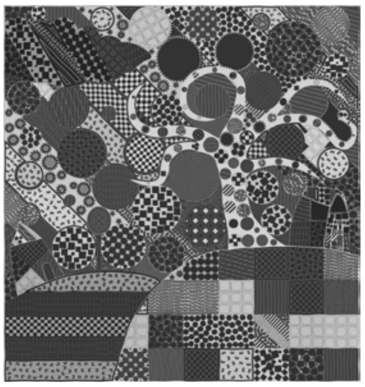


Fig. 1. An example of the tangle image

Inspired by the design of tangle patterns using a set of basic operations, this paper designs a constructive steganography through drawing tangle patterns. Driven by the secret bits, we construct the patten image by choosing different cells. As the graphic structure of tangle is generated by imagination, the generated tangle image has good visual effects and high embedding payload. During the information extraction process, the recipient can extract the secret data without any error. Since the textures of tangle images are different from natural images, it is difficult for tranditional steganalysis tools to find the steganography behavior. Besides, the proposed method is robust against JPEG compression.The rest of the paper is organized as follows. In Section 2, we propose a system of our steganography algorithm including data hiding and data extraction. Experimental results and analysis of algorithm performance are elaborated in Section 3, and conclusions are described in Section 4.

2. Proposed System

We represent a method to hide information while generating tangle patterns, which guarantees that a receiver can exactly extract the hidden message from the tangles. The proposed system of data hiding is illustrated in **Fig. 2**. The data-hider first selects a natural image and extracts its outline. The outline is then divided into a plurality of regions, each of which is separated to several groups. Subsequently, a series of graphics operator rules are builded to draw the tangle patterns. Before painting a tangle, the hider selects a certain amount of locations to paint secret graphics, using a secret key to identify the secret locations. According to some operator rules, the hider draws secret patterns at each location to represent secret messages. On other area of the outline, the hider randomly fill some patterns to construct a tangle image with a good visual effect.

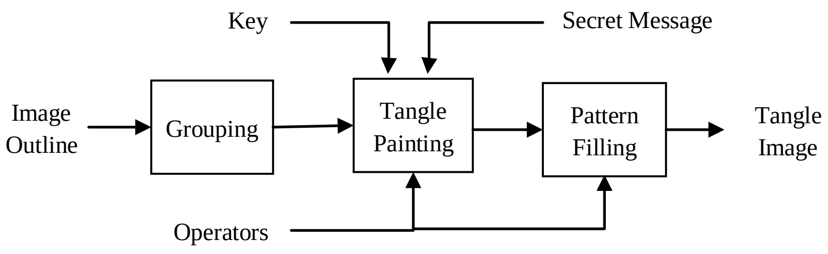


Fig. 2. Framework of the Proposed Method

2.1 Grouping grammar

Although the tangles are created by imaginations, there are some common characteristics. Generally, a tangle pattern is not randomly arranged by graphics, but made up of different kinds of meaningful graphics groups. Therefore, the generated pattern contains rich contents rather than messy artistic effects. In order to achieve rich contents in a restricted structure, we use the following grouping grammar.

We first select a nature image, and extract the outline L from it. The outline is then splitted into many regions T_i. The shape of each region is arbitrary determined by splitting algorithms, such as circular division, rectangular division and so on. The outline L can be defined as

$$\mathrm{L}=\{\mathrm{T}_i\}_{i=1}^{n} \tag{1}$$

where the parameter n is the number of regions in the outline.

To generate richer patterns with better visual effects, we further divide each region using a grouping operator. Each region T_i is subdivided into m_i groups g_j using (2).

$$\mathrm{T}_i=\{\langle \mathrm{T}_i, \mathrm{g}_j\rangle\}_{j=1}^{m_i} \tag{2}$$

Therefore, the outline L can be represented more specifically as (3)

$$\mathrm{L}=\left\{\{\langle \mathrm{T}_i, \mathrm{g}_j\rangle\}_{j=1}^{m_i}\right\}_{i=1}^{n} \tag{3}$$

where m_i represents the number of groups in the region T_i, and m_i is a positive integer.

We use an example to illustrate the splitting and grouping procedures in Fig. 3. The outline is first divided into four regions, in which each T_i (i=1~4) are marked A~D. We further divide each region into groups. For example in region A, there are three groups whose group ID are labeled with 1~3, *i.e.*, A=$\{\langle \mathrm{A}, 1\rangle, \langle \mathrm{A}, 2\rangle, \langle \mathrm{A}, 3\rangle\}$.

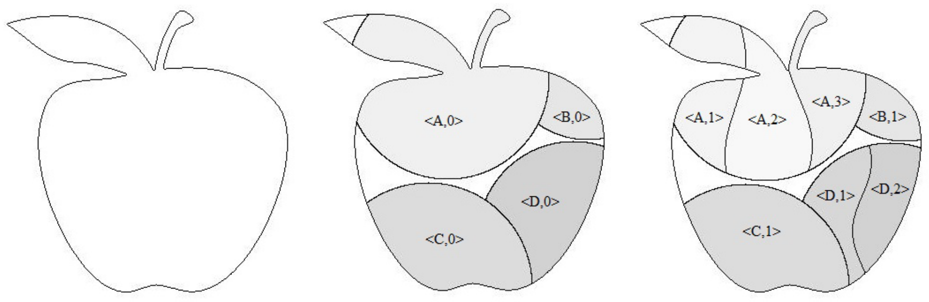


Fig. 3. An example of the outline division

In order to facilitate subsequent operations, we use a regroup operator. For example, there are m_0 groups in the region T_0, and a group is labeled as $\langle \mathrm{T}_0, \mathrm{g}_0\rangle$ with a group ID g_0. A new group ID is generated by the operator parameter k, and k is a random positive integer not greater than m_0. This new group ID can be generated by (4)

$$\mathrm{g}_0{}'=\mathrm{g}_0 \bmod k \tag{4}$$

This operation is applied in each group. In the follow-up operations, groups with the same group IDs will be applied with the same operator.

2.2 Operator Generation

We further define a series of graphic operators to create different patterns within each group. With these operators, complex patterns can also be generated by combining multiple operators. **Fig. 4** shows an example of different operators. **Fig. 4**(a)~(d) show the regular split, outline, circular split and shape placement, respectively. We note thse four operators as R_0~R_4. The meaning of each operator is depicted as follows.

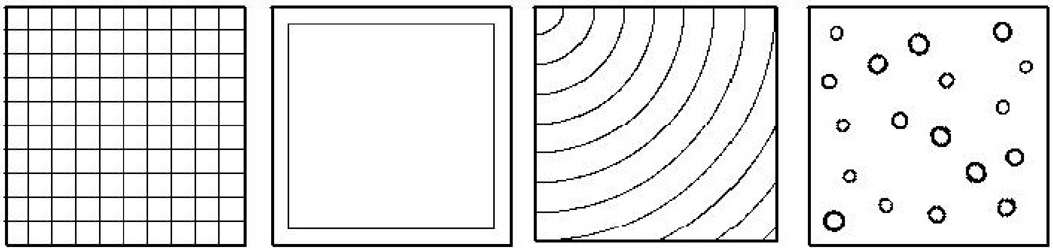


Fig. 4. An example of the operators.

R_0: *regular split*(d, r, x): The regular split operator draws straight lines along the orientation r, where the distance between each line is d. The parameter x (0 or 1) decides whether the split is dual: a dual split means that the splitting process is done an additional time, orthogonally to the orientation r.

R_1: *outline*(d): The outline operator draws a graphic whose outline is the same with the shape of the original outline. The parameter d is the distance between the drawn outline and the original outline.

R_2: *circular split*(x, r, d): Similar to the regular split, circular split operator uses a curve instead of a straight line. The parameter x is the center of the curve segmentation coordinates, and r is the radius. Each curve is distributed at a distance of d.

R_3: *shape placement*(s, d, q, z): The placement operator is used to place arbitrary shapes of graphics. The parameter s is used to decide the shape of graphics. To prevent the overlaps between graphics, the distance between the graphics are limited by a minimum spacing d. Parameters q and z are used for controlling the orientation and size, respectively.

2.3 Data Hiding by Tangle Painting and Pattern Filling

The specific steps are given in **Fig. 5** for data hiding. We create a secret cell library which is shared with the recipient for data extraction. The cell library specifies the mapping between the graphical features and the secret data. We use an example in **Table 1** to illustrate the mapping. This table gives the corresponding rules between data and graphical features. Each feature represents two bits of data. The size and direction are equivalent to the parameters z and q of the shape placement operations. For example, the size 50 means that the area of a graphic is 50 pixels, which represents the data “00”. Similarly, the orientation “0” means that the direction angle of the representation graph is zero degree, and it represents the data “01”. Each direction of the graphic also represents two bits of data.

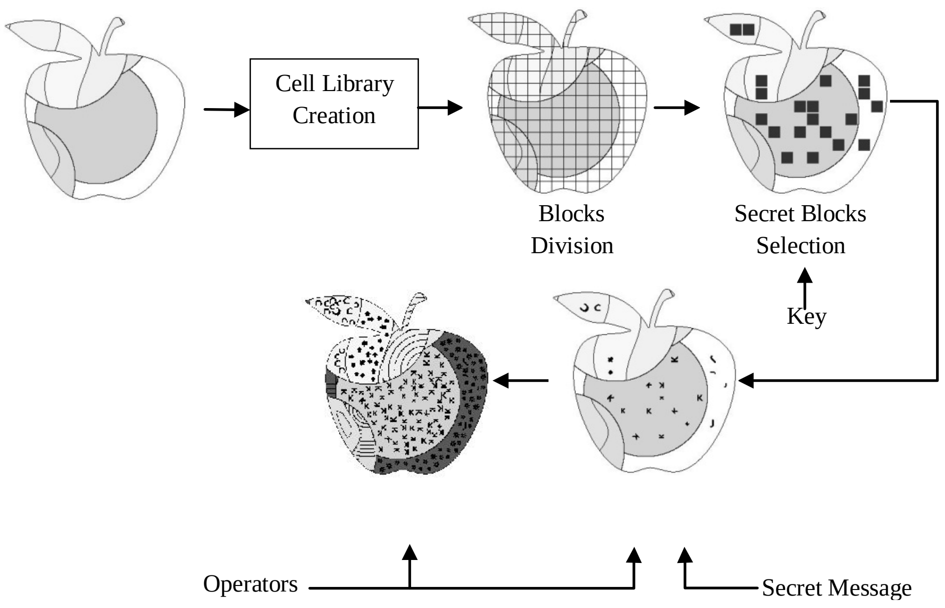


Fig. 5. Framework of Tangle Painting and Pattern Filling

Table 1. Example of the secret cell library

data	00	01	10	11
size	50	60	70	80
data	01	10	00	11
orientation	0	45	90	135

A specified example is given in **Fig. 6**. There are four triangles which are arranged according to the sizes in **Fig. 6**(a), which respectively represent the data “00”, “01”, “10”, and “11”. **Fig. 6**(b) shows the binary data represented by the triangles using different directions.

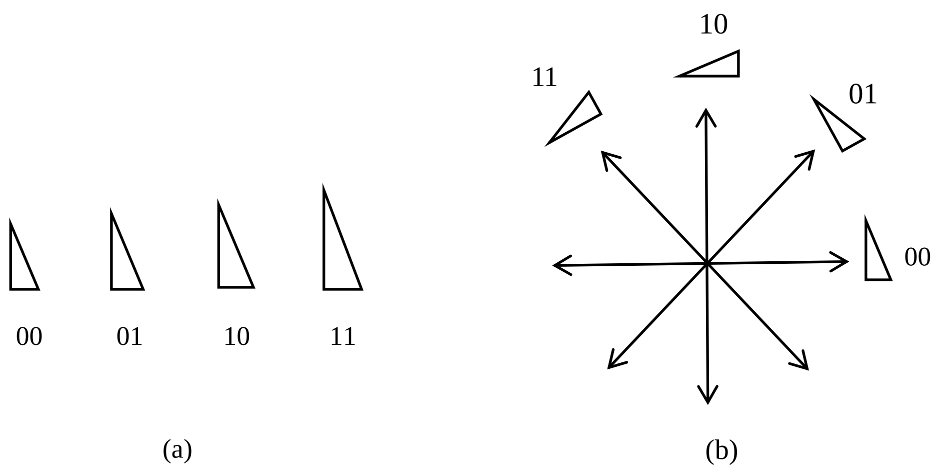


Fig. 6. A specific illustration of the mapping between graphical features and binary data.

Before drawing the graphics, we divide the image into *row*×*col* blocks. Considering that the outline L has been divided, there are many dividing lines within L. We exclude the blocks that cover the division lines or overstep the outline, and use the remaining blocks. With a raster scanning order, these blocks can be written in the form of one-dimensional matrix

$$\mathrm{B}=\{b_0, b_1,\ldots,b_q\} \tag{5}$$

Assuming that the length of the secret data S is *N*, we generate a pseudo-random sequence between 0 and q whose length is *N*, and select *N* blocks to carry secret data. These *N* blocks are then ordered by a secret key. Denote the center points of these secret blocks as

$$\mathrm{P}=\{p_0, p_1,\ldots,p_N\} \tag{6}$$

We paint tangles inside the *N* secret blocks. The center points P are used as the centroid coordinates of the tangle graphics, on which the shape placement R_3 is applied to draw different graphics. The parameters q and *z* are determined according to the secret data and the cell library. Since the shape of graphics can be arbitrarily generated , the parameter *s* can be a random number. In order to prevent the overlapping between graphics, the paremeter d is determined by the maximum size of the secret graphic definded in cell library. For example, the maximum size of the secret graphic in cell library is $a_1 \times a_2$, the paremeter d can be specified as d=max{ a_1, a_2 }. To achieve a better visual effect, on the center points of secret blocks that are painted in the same IDs' groups, we paint graphics with the same shape . It means the same paremeter *s* is applied in these positions. We define the original groups inside the outline L as

$$\mathrm{G}_0=\{\mathrm{g}_1, \mathrm{g}_2,\ldots,\mathrm{g}_{n_0}\} \tag{7}$$

and the groups that have been already painted with graphics can be marked as

$$\mathrm{G}_1=\{\mathrm{g}_1, \mathrm{g}_2,\ldots,\mathrm{g}_{n_1}\} \tag{8}$$

To create more rich tangle patterns, we will continue to fill some graphics in the blank area. The shape placement operator R_3 will be continuously applied in G_1. Since these graphics do not carry secret information, the parameters of size and orientation can be selected randomly. So as to guarantee the information can be extracted without error, the parameters of shape are diffident from the graphics in the cell library. The number of operators is proportional to the blank area in a group, make sure that we can fill enough graphics in the blank area. For the remaining groups $\mathrm{G}_2=\mathrm{G}_0-\mathrm{G}_1$, as different types of patterns can be painted, we choose operations R_p other than shape placement R_3 for groups G_2.

$$R_p \rightarrow \mathrm{G}_2,\ p\epsilon\{0,1,2\} \tag{9}$$

In this data hiding scheme, the steganographic capacity is determined by the size of the selected contour image, the size of the dense graphics, and the size of the cell library. Through adjusting these parameters, any number of secret information can be hidden. Assuming that the size of the contour image is $L \times L$ pixels and the maximum size of the graphic is m×m pixels, the number of selectable positions are $\lfloor L/m \rfloor^2$. As some positions are outside the outline or over the group lines, the number of the available positions is

$$K = \lfloor L/m \rfloor^2 - Q \tag{10}$$

where Q is the number of the invalid positions. Besides, the size of the cell library contains two parts, the number of graphical features and the kinds of features. Suppose there are N features and the value of each feature is T_i=2^{t_i}, the maximum hiding capacity R is

$$R=(\lfloor L/m \rfloor^2 - Q) \times \sum_{i=1}^{N} t_i \tag{11}$$

2.4 Data Extraction

When a recipient receives the image, secret messages can be exactly extracted. The outline of the stego image is not required to be extracted. The recipient only has to divide the image into blocks and identify the features of graphic in each block. According to the cell library, the recipient can determine whether it is a secret graphic. The recipient records the location of a graphic if it's features match one graphic in the cell library. When all the locations of secret graphics have been extracted, they are sorted by a key to restore their original order. According to the order, the recipient extracts the data that are represented by the graphic features. The extraction of the secret data from the tangle image can be realized with the following steps.

1) Divide the tangle image into $row \times col$ blocks.

2) Identify the graphics in each block, and select those blocks whose graphics match the graphics of the cell library. Represent these blocks by

$$\mathrm{B}'=\{\mathrm{b}_0', \mathrm{b}_1',\ldots,\mathrm{b}_M'\} \tag{12}$$

3) With a secret key, generate a pseudo-random sequence between 0 and M to reorder these blocks. Get the center points of these blocks

$$\mathrm{P}'=\{\mathrm{p}_0', p_1',\ldots,\mathrm{p}_N'\} \tag{13}$$

4) Extract the features of graphics placed on these N positions. According to the cell library, extract the secret data represented by the features of each secret graphic.

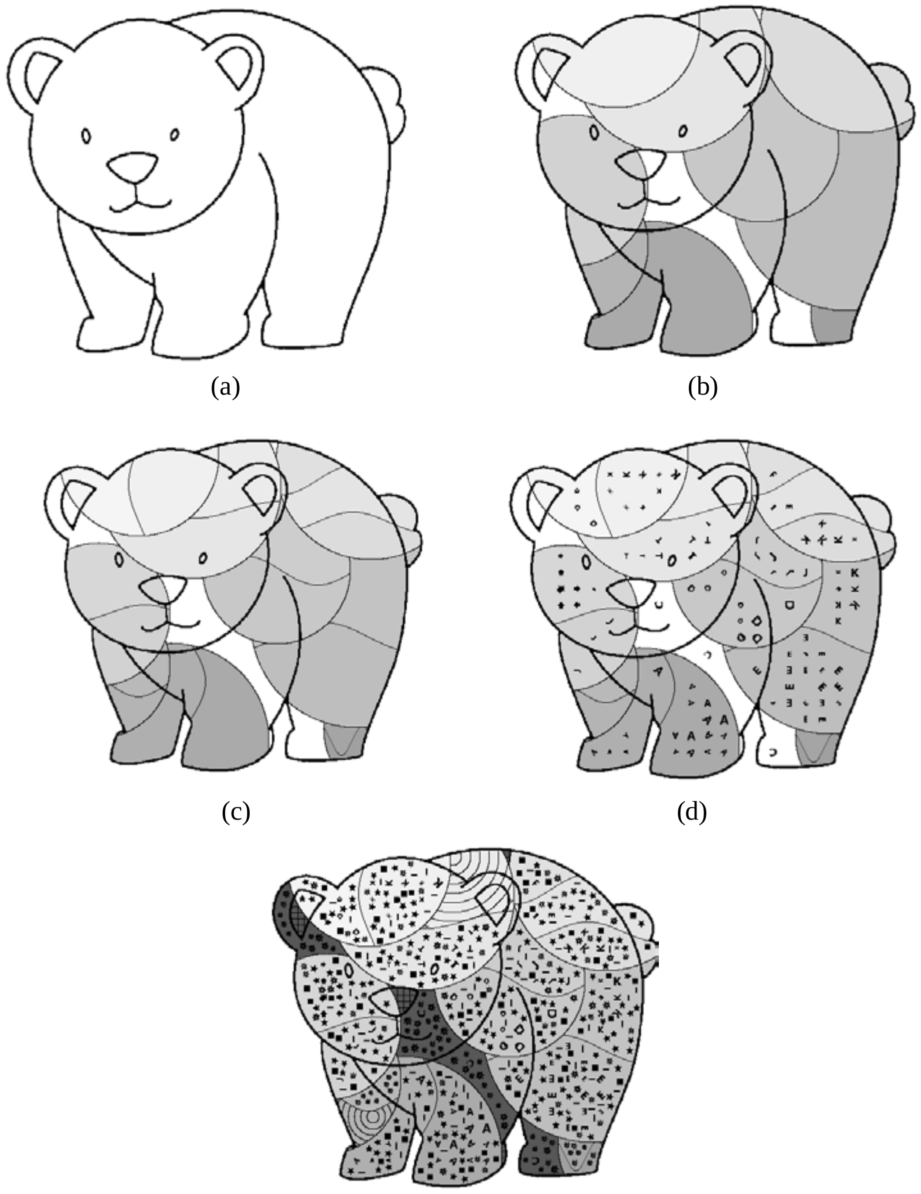
3. Experimental Results

To verify the proposed method, we first establish a cell library (**Table 2**). There are eight sizes and eight directions, the cell library defines features to represent the secret data. There are eight sizes and eight directions. As each direction and size of the graphics can represent three bits, each graphic can represent 6 bits. For example, one graphic has a size 100 representing "000", and an orientation 45 representing "001". Thus, the graphic can represent the secret data "000001".

Table 2. An example of cell library

data	000	001	010	011	100	101	110	111
size	100	120	140	160	180	200	220	240
data	000	001	010	011	100	101	110	111
orientation	0	45	90	135	180	225	270	315

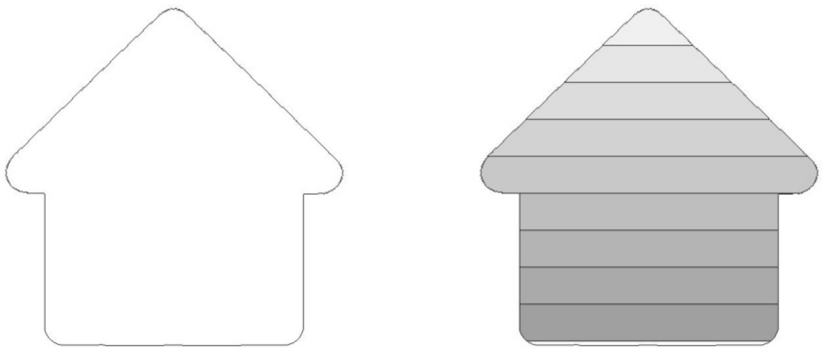
Fig. 7 shows a group of experimental results. The image contains 512×512 pixels. **Fig.7(a)** is an image outline, which is segmented into nine regions, as shown in **Fig. 7(b)**. Each region is randomly divided into two or three groups, see **Fig. 7(c)**. Different groups are filled with different pixel values. When embedding 600 secret bits, we use 100 locations to place the secret graphics. Let the maximum size of a secret graphic be 20×20. **Fig. 7(c)** is divided into 25×25 blocks. After removing invalid blocks, we select 100 blocks to draw secret graphics and record their center locations. After reordering these locations using a secret key, we apply shape placement operator in these locations. We choose the appropriate graphic features from the cell library according to the secret messages. The result is shown in **Fig.7(d)**. We further apply shape placement fill the blank areas. The final tangle pattern of hiding information has been shown in **Fig. 7(e)**.



(a) (b) (c) (d) (e)

Fig. 7. A group of experimental results

Another group of expeimental results are shown in **Fig. 8**, in which different visual effects are provided. In an outline **(a)** containing 512×512 pixels, we divide it into nine groups by regular grouping method, as shown in **(b)**. We also divide the image into 25×25 blocks and select 100 blocks to place the secret graphics. With the cell library, we also also use eight sizes and eight directions for each graphic. Secret data hidden in the image are 600 bits. The final tangle image is shown in **Fig. 8(c)**.


(a) Outline (b) Grouping

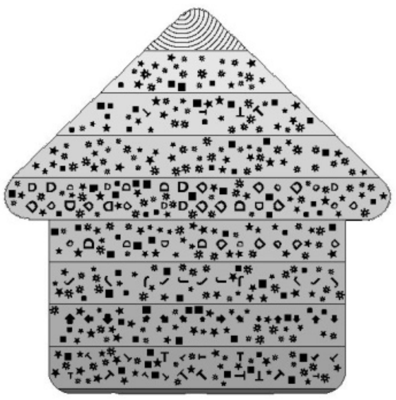
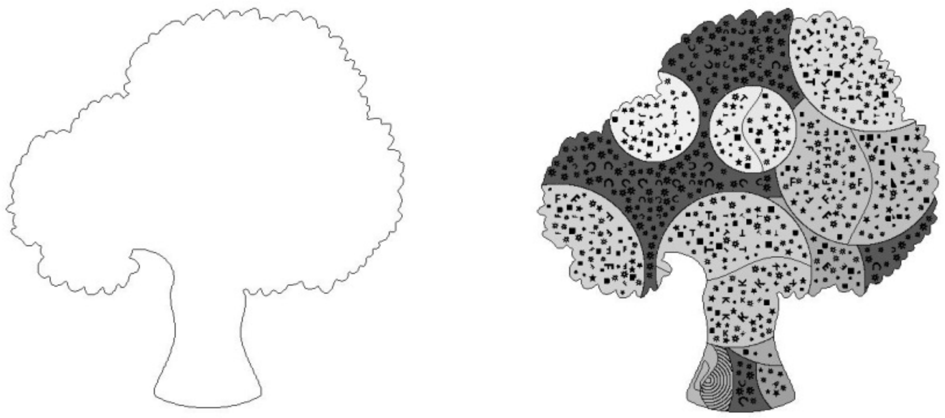

(c) Tangle image

Fig. 8. Another group of experimental results

We also conduct a group of experimental results to show that more payload can be hidden inside a tangle. There are three ways to achieve the goal. The first way is to select a larger outline, in which there are more places to paint graphics. The secode way is to increase the secret locations to represent more bits. The third way is to build a more complex cell library. In **Fig. 9(a)**, we select an outline containing 512×512pixels. The grouping method is the same as the the method used in **Fig. 7**. We divide the image into 25×25 blocks and select 200 blocks to hide information in the outline. In the cell libraty we define eight sizes and eight directions of every graphic. Therefore, 1200 bits data can been hidden in the tangle shown in **Fig. 9(b)**.



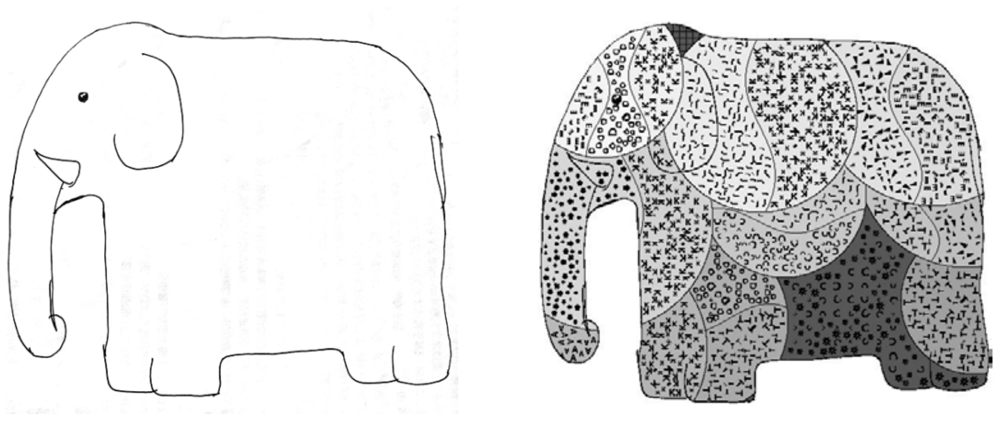
(a) Outline (b) Tangle

Fig. 9. An experiment with larger hiding capacity

We can combine a variety of methods to achieve larger information steganography. In **Fig.10(a)**, we select an outline containing 512×512 pixels. The image is divided into 38×38 blocks so that there are more locations to paint graphics. In order to prevent the overlaps between graphics , we need another cell library to redefine the size of secret graphics. As show in **Table 3**, the maximal size is 180 pixels, there are eight sizes and eight directions of every graphic. We select 900 blocks to hide 5400 bits data in the tangle as shown in **Fig.10(b)**.

Table 3. Another example of cell library

data	000	001	010	011	100	101	110	111
size	30	40	60	80	100	120	140	160
data	000	001	010	011	100	101	110	111
orientation	0	45	90	135	180	225	270	315



(a) Outline (b) Tangle

Fig. 10. Another experiment with larger hiding capacity

We have also implemented some experiments to verify the steganographic capacity of our steganographic method. We use an outline sized 512×512 pixels. After splitting it into ten groups, we select two kinds of features to represent secret message. Each feature has eight kinds of values. Thus, each graph can represent six bits. **Table 4** shows the data hiding capacity when different graph sizes are used. The maximal graph size is $m \times m$ pixels, and the capacity is labeled as R (bits). The results show that the hiding capacity is larger when smaller cells are used.

Table 4. steganographic capacity with different size of graph

m	12	14	16	18	20	22	24
R	6120	3930	2700	2040	1500	930	540

As there are no such steganography algorithms using tangles, so we compare the steganographic capacity of our method with other steganography of texture synthesis. Compare with the algorithms in [9] and [11], the comparison result of data embedding capacity is shown in **Table 5**. When the size of stego image is 512×512, the maximum steganographic capacity of our method is 6120 bits, while another two steganographic methods are 1600 bits and 7290 bits. For the difference of the hiding capacity, we has done some analysis for our method and algorithm in [9]. The two algorithms use different covers for steganography. As there are more complex contents in the texture images than in the tangling images, the hiding capacity in [9] is larger.

Table 5. Comparing the steganographic capacity of our method with other algorithm

Steganography algorithm	Maximum embedding capacity(bits)	Steganography scheme
Our method	6120	Hiding data by tangle
Algorithm in [9]	7290	Texture synthesis
Algorithm in [11]	1600	Texture synthesis

The proposed method is robust against JPEG compression. There are six tangle images that carry different number of secret information as show in **Fig. 11**. There are 600 bits, 1200 bits, 1800 bits, 2400 bits, 3000 bits and 6000 bits secret data hidden in **Fig. 11(a)~(f)**. These images are applied with different degrees of JPEG compression. **Table 6** shows the error rate(%) when we extract information form these compressed images. When the amount of secret data is less than 2000 bits, we can extract data without error when the compression factor is above 50. If the amount of data reaches 3000 bits, we can extract data correctly when the compression factor is above 65. If we hide more imformation in tangles, such as the data reaches 6000 bits, a larger quality factor is needed to guarantee the correct extraction.

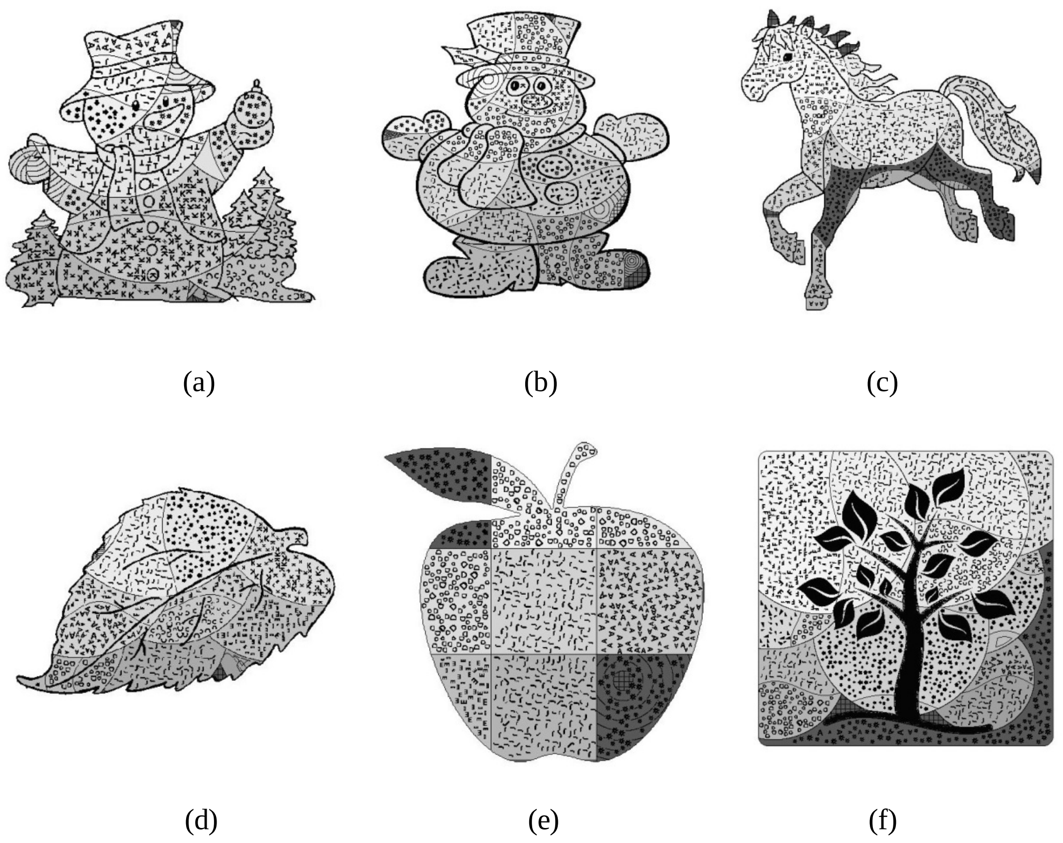


Fig. 11. Tangle images with different data hiding capacity

Table 6. The relationship of error rate and quality factors under different hiding capacities

Quality factor / Capacity(bits)	5	20	35	50	65	80	95
600	32.8	19.2	5.7	0	0	0	0
1200	35.0	23.2	6.3	0	0	0	0
1800	36.6	25.0	6.8	0	0	0	0
2400	48.2	32.2	19.5	4.6	0	0	0
3000	51.3	33.0	19.6	4.8	0	0	0
6000	72.0	61.3	42.6	19.5	3.4	0	0

As there are no steganalysis methods for the proposed method, we use Rich-Model to do analysis, which was designed for detecting steganography behaviors in nature images. With clean tangle images and the stego tangle images, we use the tool to detect the steganography behavior. We denote the error rate of detection for both kinds of tangles as r_1 and r_2. After many experiments, we found that the rate r_1 and r_2 are both close to 0.02. This result shows that both clean tangle images and secret tangle images are considered as stego images using Rich-Model. This illustrates that the traditional steganalysis is not suitable for this kind of steganography, since it is unable to separate the clean from the stego. As there are many tangle images on internet and state-of-the-art steganalysis tools cannot identify whether they are clean or not, the proposed constructive steganography by tangle is therefore secure.

4. Conclusions

This paper proposes a constructive steganography algorithm using tangles. With an outline of a natural image and a pre-defined cell library, we can construct a tangle image to hide secret messages through a series of graphic operations. Different from traditional steganography methods based on modifying a selected image. The proposed method generates the stego according to the secret bits. With the tangle operations, different kinds of tangle images can be generated with good view. The proposed method guarantees that the hidden bits can be extracted exactly. This method also provides a capability of countering JPEG compression.

References

[1] H. Wang and S. Wang, "Cyber Warfare: Steganography vs. Steganalysis," *Communication of ACM*, 47(10), pp. 76-82, 2004. Article (CrossRef Link)

[2] J. Fridrich and M. Goljan, "Practical Steganalysis of Digital Images - State of the Art," *Security and Watermarking of Multimedia Contents IV, Proceedings of SPIE*, 4675, pp. 1-13, 2002. Article (CrossRef Link)

[3] J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal, "Writing on Wet Paper," *IEEE Transactions on Signal Processing*, vol.53, no.10, pp.3923-3935, 2005.Article (CrossRef Link)

[4] A.Westfeld, "F5—A Steganographic Algorithm High Capacity Despite Better Steganalysis," in *Proc. of The Second International Workshop on Digital-forensics and Watermarking*, Seoul, Korea, October, pp.154-167, 2003. Article (CrossRef Link)

[5] W. Zhang, X. Zhang, and S. Wang, "Maximizing Steganographic Embedding Efficiency by Combining Hamming Codes and Wet Paper Codes," in *Proc. of 10th International Workshop on Information Hiding*, Santa Barbara, CA, USA, May, pp.60-71, 2008.Article (CrossRef Link)

[6] T. Filler, J. Judas, and J. Fridrich, "Minimizing Additive Distortion in Steganography using Syndrome-Trellis Codes," *IEEE Transactions on Information Forensics and Security*, Vol.6, no.3, pp.920-935, 2011. Article (CrossRef Link)

[7] H. Otori and S. Kuriyama, "Data-embeddable texture synthesis," in *Proc. of the 8th International Symposium on Smart Graphics*, Kyoto, Japan, pp. 146-157, 2007.Article (CrossRef Link)

[8] H. Otori and S. Kuriyama, "Texture synthesis for mobile data communications," *IEEE Comput. Graph*, Appl., vol. 29, no. 6, pp. 74-81, 2009. Article (CrossRef Link)

[9] K. Wu and C. Wang, "Steganography using reversible texture synthesis," *IEEE Trans. Image Process*, vol. 24, no. 1, pp. 130–139, Jan. 2015. Article (CrossRef Link)

[10] H. Zhou, K. Chen, W. Zhang, and N. Yu, "Comments on "Steganography Using Reversible Texture Synthesis," *IEEE Trans. Image Process*, vol. 26, no. 4, 2017.Article (CrossRef Link)

[11] Z. Qian, H. Zhou, W. Zhang, and X. Zhang, "Robust steganography using texture synthesis," in *Proc. of Twelfth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IIH-MSP 2016, 25-33, Kaohsiung, Taiwan, Nov. 21-23, 2016. Article (CrossRef Link)
[12] C. Santoni, F. Pellacini, "gTangle: a Grammar for the Procedural Generation of Tangle Patterns," in *Proc. of ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2016*, Volume 35 Issue 6, 2016. Article (CrossRef Link)
[13] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, "Coverless image steganography without embedding," in *Proc. of International Conference on Cloud Computing and Security*, Springer, pp. 123-132, August, 2015. Article (CrossRef Link)
[14] J. Hayes, G. Danezis, "ste-GAN-ography: Generating Steganographic Images via Adversarial Training," 2017. Article (CrossRef Link)

Zhenxing Qian received both the B.S. and the Ph.D. degrees from University of Science and Technology of China (USTC) in 2003 and 2007, respectively. Since 2009, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. He has published over 80 peer-reviewed papers on international journals and conferences. His research interests include information hiding, image processing and multimedia security.

Lin Pan received the B.S. degree in School of Communication and Information Engineering, Shanghai University. She is currently working toward the Master degree at Shanghai University. Her research interests include steganography and image processing.

Nannan Huang received the B.S. degree in School of Communication and Information Engineering, Shanghai University. She is currently working toward the Master degree at Shanghai University. Her research interests include information hiding and image processing.

Xinpeng Zhang received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. His research interests include information hiding, image processing and digital forensics.