

CREATION OF A NEW TYPE OF IMAGE --- CIRCULAR-DOTTED IMAGE --- FOR DATA HIDING BY A DOT OVERLAPPING SCHEME*

Chia-Yu Hsu (許家瑜)¹ and Wen-Hsiang Tsai (蔡文祥)^{1, 2}

¹Dept. of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

²Dept. of Computer Science & Information Eng., Asia University, Taichung, Taiwan

ABSTRACT

In order to implement a concept of integrating art image creation processes with information hiding techniques, a method for automatic generation of a new type of image --- digital circular-dotted image --- as well as data hiding in it is proposed. We use the drawing order of dots in a circular-dotted image and a dot overlapping scheme for data hiding. We implement two applications of the proposed data hiding method, namely, cover communication and copyright protection. Good experimental results show the feasibility of the proposed method.

Keywords: Art image, circular-dotted image, data hiding, secret message, covert communication, watermarking, copyright protection.

1. INTRODUCTION

Many methods have been proposed in recent years to create digital art images automatically [1]-[9]. On the other hand, the purpose of information hiding is to embed information imperceptibly into a given media. Many methods for information hiding in images take advantage of the weakness of the human visual system, for example, by changing the least significant bits of the pixels of a cover image to embed information [10]. Such changes are imperceptible to human vision. The information embedded in an image can be used to protect the copyright of the image, verify the authenticity of the image, convey a secret message, or conduct secret sharing, and so on.

Researches on information hiding in images can be classified into three approaches, namely, the spatial-domain approach, the frequency-domain approach, and the combination of the two [11]. No matter what types they belong to, most of these researches are based on pixelwise or blockwise operations. In this paper, unlike the traditional methods of information hiding in

common images, we hide data in a new type of art image proposed in this study, namely, *circular-dotted image*, by deciding the drawing order of the circular dots in the digital circular-dotted image during the image creation process.

Lin and Tsai [12, 13] proposed a method to hide information in image mosaics (or photomosaics) by manipulating the four borders and the histogram of a tile image. Hung and Tsai [14] proposed two methods to hide information in art images. One is to embed data in tile mosaic images by modifying the orientations, sizes, and textures of tile mosaic images. The other is to embed data in stained glass images by creating slight glass cracking. All of these researches have implemented the concept of integrating the methods for art image creation and data hiding.

In the remainder of this paper, we first introduce a process of circular-dotted image creation in Section 2. Then, we describe the proposed method for data hiding in digital circular-dotted images in Section 3. In Section 4, we present our experimental results. And finally we make some conclusions in Section 5.

2. PROPOSED CIRCULAR-DOTTED IMAGE CREATION PROCESS

In Section 2.1, the idea of digital circular-dotted image creation is presented. And in Section 2.2, the creation method will be described in detail.

2.1. Idea of Circular-Dotted Image Creation

A circular-dotted image is composed of numerous overlapping colorful circular dots with black borders. The distance between the centers of two adjacent dots and the thickness of the black border can be modified by users. In principle, pixels may be drawn onto a digital image serially, starting from the top-left pixel, and moving horizontally in a raster scan order. However, in order to make the digital circular-dotted image look more appealing, we will not draw the dots on the image in the raster scan order. Instead, we randomize the drawing order of the dots by applying a random array creation process first. The detailed process is described as an algorithm below.

Algorithm 1: Random array creation process.

Input: A digital image I with the size $W \times H$, and the distance D between every two adjacent dots.

* This work is supported partially by the NSC Project Advanced Technologies and Applications for Next Generation Information Networks (II) with Project No. NSC93-2752-E-009-006-PAE and partially by the NSC National Digital Archives Program with Project No. NSC93-2422-H-009-001.

Output: A random array, denoted as RA .

Steps:

- Step 1 Compute the number NOD of circular dots which should be drawn on the desired circular-dotted image as $NOD = (W \times H) / (D \times D)$.
- Step 2 Create an empty array, denoted as UA , with size equal to NOD .
- Step 3 Set the values of the elements in UA as 0 through $NOD - 1$ in order.
- Step 4 Design a random number generator f with its generation range from 0 to $NOD - 1$.
- Step 5 Create two additional empty arrays, $ID0$ and $ID1$ with sizes both equal to NOD .
- Step 6 Set the value of the i th element of $ID0$ as the $(2i+1)$ -th value generated from f , and the value of the i th element of $ID1$ as the $(2i+2)$ -th value generated from f , where $i = 0$ through $NOD - 1$. Denote these two values as $ID0_i$ and $ID1_i$, respectively.
- Step 7 Swap the $ID0_i$ -th element in UA and the $ID1_i$ -th element in UA for each $i = 0$ through $NOD - 1$.
- Step 8 Take the final UA as the desired random array RA .

2.2 Circular-Dotted Image Creation Process

Before drawing a colorful circular dot on a digital circular-dotted image, we draw a black circle as the dot border first, and then paint a color in it. The size of the circular dot and the thickness of the dot border may be decided by users. We vary the size of the circular dots and the thickness of the circle borders to achieve different art effects. The details are described as an algorithm below.

Algorithm 2: Digital circular-dotted image creation.

Input: An image I with size $W \times H$; a distance D between the centers of every two adjacent circular dots; and the thickness T of the black border circle.

Output: A digital circular-dotted image I_C .

Steps:

- Step 1 Compute the diameters $BlackD$ and $ColorD$ of the black border circle and the circular dot, respectively, from the size $W \times H$ of image I and the parameters D and T in the following way:

$$\begin{aligned} & \text{if } D \text{ is odd, then set } BlackD = D + (D+1)/2; \\ & \text{if } D \text{ is even, then set } BlackD = D + D/2; \\ & \text{set } ColorD = BlackD - T \times 2. \end{aligned}$$
- Step 2 Use the values of D and $BlackD$ to compute the image coordinates of the center of each circular dot, and so obtain an array of colors of the input image I at these coordinates, which we call as the *center color array* and denote as CCA .
- Step 3 Compute the number NOD of circular dots which will be drawn in the output circular-dotted image I_C by the input parameters W , H , and D as $NOD = (W \times H) / (D \times D)$.
- Step 4 Apply Algorithm 1 to create a random array RA

with size equal to NOD .

- Step 5 Draw circular dots according to RA in the following way.
 - 5.1 Take out elements in RA sequentially until done. Let the i th element value be v_i .
 - 5.2 Draw the i th border circle at the v_i -th location in I_C , and then paint in it a circular area of color C_i with the diameter of $ColorD$, where C_i is decided by the color specified by the i th element CCA_i of CCA .

2.3 Experimental Result

An experimental result of the above-described circular-dotted image creation process using a self-portrait image of Vincent van Gogh as the input is shown in Figure 1.

3. PROPOSED DATA HIDING IN CIRCULAR-DOTTED IMAGES BY A DOT OVERLAPPING SCHEME

In Section 3.1, the idea of data hiding in digital circular-dotted images by a dot overlapping scheme is introduced. In Sections 3.2 and 3.3, the proposed data hiding and extraction processes are described.

3.1. Idea of Data Hiding

Data hiding in digital circular-dotted images is based on the concept of *dot overlapping* following a certain dot drawing order. The data we hide in a circular-dotted image include a watermark image as well as certain secret data if necessary. Before performing dot overlapping in image creation, we combine the bits of the secret information and those of the watermark. Besides, the secret information is supposed to be extracted only by a person with a secret key. So we use the key to disorder the combined bit sequence before embedding it. The details of these works are described as an algorithm below.



(a)

- Fig.1 An experimental result. (a) An original image with size 794×960 . (b) A digital circular-dotted image created from (a) (with $D=11$ and $T=1$). (c) Another digital circular-dotted image created from (a) (with $D=17$ and $T=3$). (d) A detail of (b).

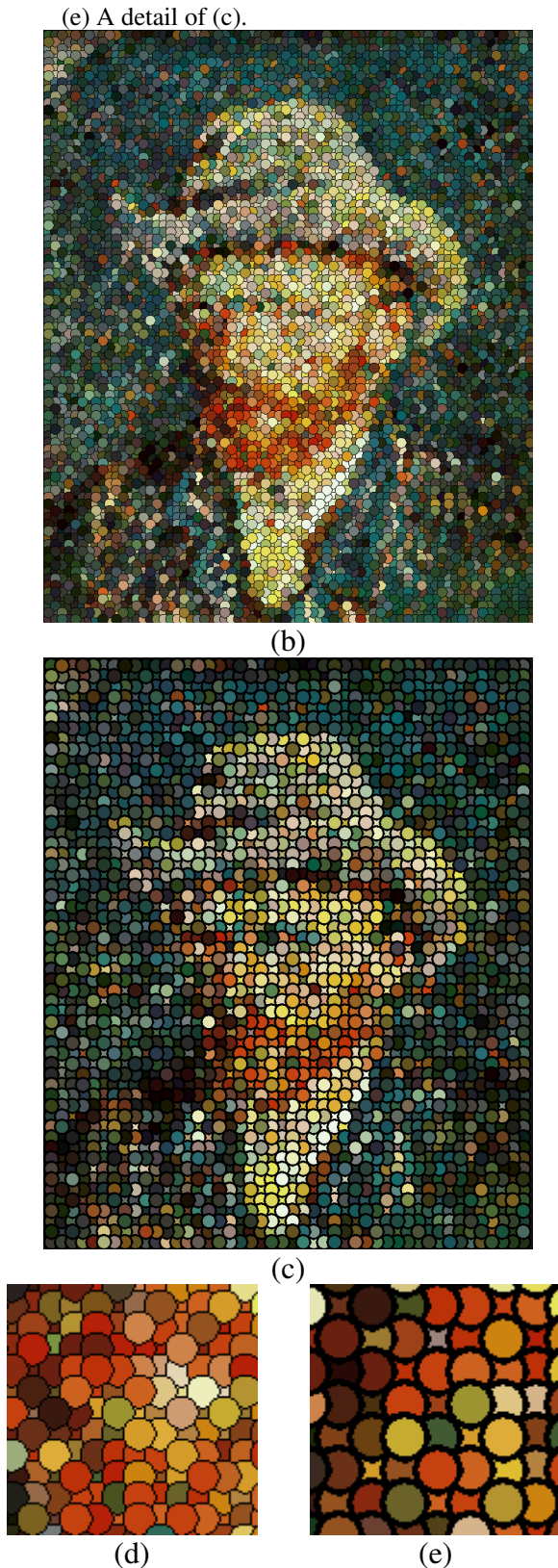


Fig.1 An experimental result. (a) An original image with size 794×960 . (b) A digital circular-dotted image created from (a) (with $D=11$ and $T=1$). (c) Another digital circular-dotted image created from (a) (with $D=17$ and $T=3$). (d) A detail of (b). (e) A detail of (c). (Continued).

Algorithm 3: Data combination and disarrangement.

Input: A bit sequence of a secret message, denoted as Mes ; a bit sequence of a watermark, denoted as $Water$; and a bit sequence of a user-specified key, denoted as $SKey$.

Output: A disordered bit sequence combining Mes and $Water$, denoted as $DData$.

Steps:

- Step 1 Create an empty array, named as $DData$, with its size equal to NOD as mentioned previously.
- Step 2 Append an *ending pattern*, denoted as Ep , of sixteen successive 0s at the end of Mes , connect $Water$ to the result, and append Ep again to the end of $Water$ to form a sequence of Mes , Ep , $Water$, Ep in order. Denote the sequence as $Data$.
- Step 3 Put the elements of $Data$ into $DData$ in order, and randomly set the value of each remaining element of $DData$ as 0 or 1.
- Step 4 Swap the elements in $DData$ by applying the random array creation process described in Algorithm 1 using the input key $Skey$ to get a randomized version of $DData$ as the output.

In Step 2, by utilizing the ending pattern Ep , we can determine where Mes and $Water$ end in a sequence of extracted bits after performing the proposed data extraction process described later. Besides, we can also find where $Water$ starts in the extracted bit sequence. More details will be described in Section 3.3.

After deriving the bit code sequence of $DData$, we perform a dot overlapping scheme, which will be discussed in Section 3.2, to deal with $DData$. Then, we use the derived drawing order to create a digital circular-dotted image by applying the image creation process described in Algorithm 2.

By applying a circular-dot detection process and a data recovering and separation process, respectively, which will be described in Section 3.3, we can extract a bit sequence $DData$ from an input data-embedded circular-dotted image, and then derive the embedded secret message Mes and watermark $Water$.

3.2. Data Hiding Process

Before performing the proposed process of data hiding in a circular-dotted image, we preprocess the input data Mes and $Water$ by applying the data combination and disarrangement process described in Algorithm 3 above to get a bit sequence $DData$. Besides, we replace the color of every black pixel with the value of $(0, 0, 0)$ in the input image by the color value of $(2, 2, 2)$, and set the background color of the output image as black. Because of this preprocessing, there will be no black pixel except those of the borders of the circular dots and the background of the output image. This preprocessing is necessary in order to make the data extraction process smooth, which will be described in Section 3.3. The details of the data hiding

process are described as an algorithm below. An illustration of the steps is shown in Fig. 1.

Algorithm 4: Data hiding in circular-dotted images.

Input: An input image I with size $W \times H$, a value D of the inter-dot distance, a value T of the border circle thickness, and a bit sequence $DData$ of disordered data obtained by Algorithm 3.

Output: A 2-D array TA of bit sequences, and a drawing order of the circular dots in the circular-dotted image.

Steps:

- Step 1 Create an empty 2-D array TA with size $W_T \times H_T$ where W_T and H_T are computed respectively as W/D and $(H/D)+1$.
- Step 2 Fill the array TA in the following way.
 - 2.1 Set the bit value of the first element of each row of TA as 1.
 - 2.2 Set the bit value of the last element of each row of TA as 0.
 - 2.3 Put the values of the elements of $DData$ into the empty entries of TA in order.
 - 2.4 Randomly set the value of each unfilled element of TA as 0 or 1.
- Step 3 For each row of TA , from left to right, draw circular dots using Algorithm 2 according to the following drawing order:
 - 3.1 if the values of two successive elements are 01, then draw a circular dot;
 - 3.2 if the values of two successive elements are 11, then draw a circular dot;
 - 3.3 if the values of two successive elements are 00, then draw a circular dot;
 - 3.4 if the values of two successive elements are 10, then draw a circular dot.

3.3. Data Extraction Process

The main concept behind the proposed data extraction process is to figure out whether a circular dot in an input circular-dotted image “presses on” or “is pressed by” a successive circular dot. If the circular dot presses on the successive one, we know from Algorithm 4 that the secret bit embedded between the two successive dots is 0; otherwise, the embedded bit is 1.

Because each pixel with the original color value of (0, 0, 0) in the input image was assigned the color value of (2, 2, 2) and the background color of the output image was set as black, there will be no pixel with the color value of (0, 0, 0) except those of the black border of the circular dots and the background of the output image.

On the other hand, after performing the circular dot detection process in the input image, we can derive a disordered bit sequence of the combined bit sequence of *Mes* and *Water*, denoted as $DData$. Then, we apply the data recovering and separation process described in Algorithm 5 to recover the embedded secret data *Mes* and *Water*.

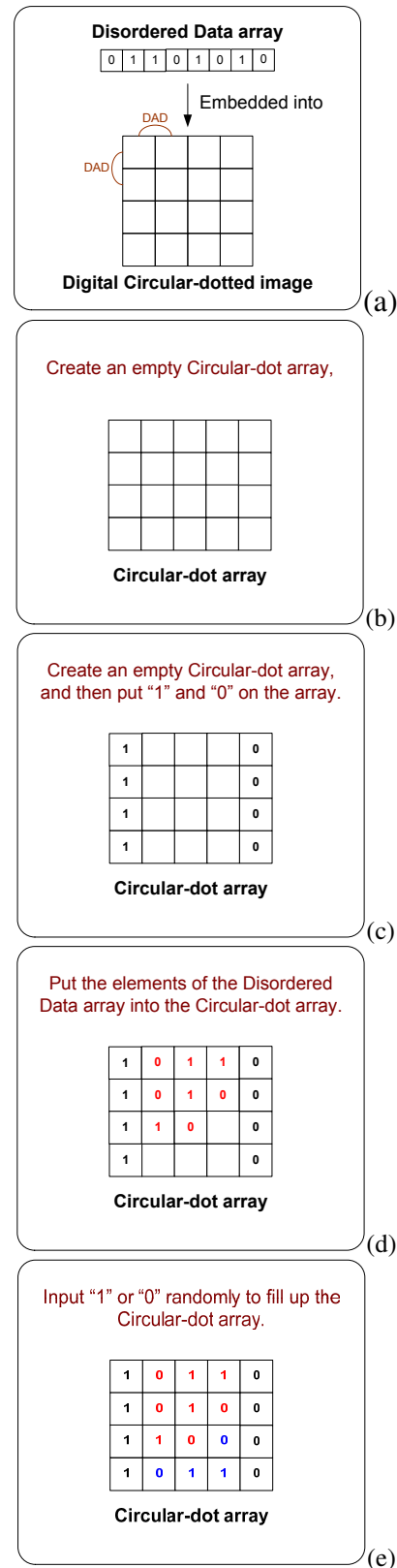


Fig. 1 An example of data hiding in circular-dotted images by dot overlapping. The colors of the circular dots are obtained from the CCA of the original input image.

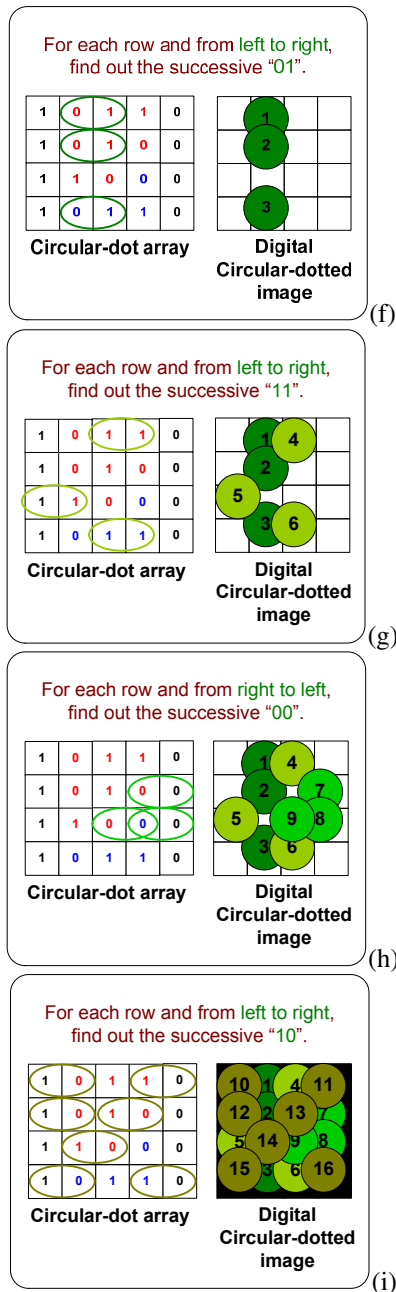


Fig. 2 An example of data hiding in circular-dotted images by dot overlapping. The colors of the circular dots are obtained from the CCA of the original input image. (Continued).

Algorithm 5: Circular dot detection.

Input: A data-embedded circular-dotted image I' , and a bit sequence of a secret key, denoted as $SKey$.

Output: A bit sequence of disordered data, denoted as $DData$.

Steps:

Step 1 Find the coordinates of the center, denoted as CC_1 , of the first circular dot.

1.1 Perform a raster scan of image I' row by row and perform the following checks:

if a black pixel is detected and the successive pixel is a non-black pixel, then denote the position of the non-black pixel as CC_{1A} , and continue the scan;

if a non-black pixel is detected, and the successive pixel is a black pixel, then denote the position of the non-black pixel as CC_{1B} .

1.2 Derive the x -coordinate x_1 and the y -coordinate y_1 of CC_1 in the following way:

$$x_1 = CC_{1A} \cdot x + \frac{CC_{1B} \cdot x - CC_{1A} \cdot x}{2};$$

$$y_1 = x_1.$$

Step 2 Find out the x -coordinate of the center, denoted as CC_2 , of the second circular dot in a similar way.

Step 3 Compute the distance D between the centers of the two adjacent circular dots CC_1 and CC_2 by

$$D = CC_{2,x} - CC_{1,x}.$$

Step 4 Derive the centers of the other circular dots by using CC_1 and D .

Step 5 Compute the thickness T of the black border circle by the following way.

5.1 Perform a raster scan from the pixel at $(CC_{1,x}, 0)$ to the pixel at $(CC_{1,x}, CC_{1,y})$.

5.2 Derive T by calculating the number of black pixels between these two pixels.

Step 6 Compute a *detection threshold*, denoted as DT , by:

$$DT = \frac{D - T}{2}.$$

Step 7 Compute a *detection value*, denoted as DV_i , for each circular dot in the following way:

for the i th circular dot in I' , perform a raster scan from pixel $(CC_{i,x}, CC_{i,y})$ to pixel $(CC_{i+1,x}, CC_{i+1,y})$ and calculate as DV_i the number of pixels on the scanning line until a pixel with the color value of $(0, 0, 0)$ is detected.

Step 8 Perform the following data extraction steps:

for the i th circular dot in I' ,

if $DV_i < DT$, then set $DData_i = 1$ (the i th circular dot is pressed by the successive dot);

otherwise, set $DData_i = 0$ (the i th dot presses on the successive dot).

In Step 1, the first circular dot is located at the most-west-north position of I' , and the second circular dot is right adjacent to the first.

4. EXPERIMENTAL RESULTS

Fig. 3 shows some experimental results of secret data hiding in a digital circular-dotted image. Fig. 3(a) is a circular-dotted image with the Chinese secret message “阿祥，有一句話我一直很想告訴你，那就是我喜歡你!!” (in English, “A message I like to tell you is that I like you.”) and the watermark of Fig. 3(c) embedded. Fig. 3(b) shows the secret message extracted from Fig. 3(a) with a correct key, and the message is identical to the one we have embedded in Fig. 3(a). Fig. 3(d) is the watermark extracted from Fig. 3(a) with a correct key. Fig. 3(e) shows the extraction result from Fig. 3(a) with a wrong key. We can see that the text shown in Fig. 3(e) is disordered and meaningless. This proves that the randomization with the key we applied in the proposed data hiding process really works.

Fig. 4 shows some other experimental results. The secret data embedded in Fig. 3(a) and Fig. 4(a) are the same, but the input values of the inter-dot distances D and the black border circle thickness values T of the figures are different. The results prove that although the sizes and the borders of the circular dots in Fig. 3(a) and Fig. 4(a) are different, the data extraction process proposed in this chapter still works.

5. CONCLUSIONS

In this paper, we have proposed methods for circular-dotted image creation and data hiding. These two topics are integrated into one which is then solved by a single approach described by the proposed methods. Therefore, users can easily and simultaneously generate art images and embed data in them. We use the drawing order of the circular dots onto a circular-dotted image as the feature for data hiding. We append an ending pattern to combine a secret message and a watermark so that they can be embedded simultaneously. By applying a dot overlapping scheme to control the drawing order of circular dots, we can achieve the purpose of covert communication, and copyright protection [15].

There are still some interesting topics which are worth further studies. For example, we can add visual effects, such as burnish, texture, etc. into each circular dot region to make the circular-dotted image look more appealing. By burnishing each circular dot, the circular-dotted image may look like a stained glass image. We can use more features such as the texture of the circular dots for embedding more data into circular-dotted images.

REFERENCES

[1] A. Secord, “Weighted Voronoi Stippling,” *Proceedings of SIGGRAPH 02*, Annecy, France, 2002. pp. 37-43.
 [2] A. Secord, W. Heidrich, and L. Streit, “Fast primitive distribution for illustration,” *ACM International*

Conference Proceeding Series; Vol. 28, Pisa, Italy, 2002, pp. 215-226.
 [3] M. P. Salisbury, M. T. Wong, J. F. Hughes, and D. H. Salesin, “Orientable textures for image-based pen-and-ink illustration,” *Proceedings of SIGGRAPH 97*, 1997, pp. 401-406.
 [4] P. E. Haeberli, “Paint by Numbers: Abstract Image Representations,” *Proceedings of SIGGRAPH 90*, August 1990, pp. 207-214.
 [5] A. Hertzmann, “Fast Paint Texture,” *Proceedings of SIGGRAPH 02*, Annecy, France, June 3-5, 2002. pp. 91-96, 161.
 [6] A. Hertzmann, “Painterly Rendering with Curved Brush Strokes of Multiple Sizes,” *Proceedings of SIGGRAPH 98*, Orlando, Florida. July 1998, pp. 453-460.
 [7] C. J. Curtis, S. E. Anderson, J. E. Seims, K. W. Fleischer, and D. H. Salesin, “Computer-generated watercolor,” *Proceedings of SIGGRAPH 97*, August 1997, pp.421-430.
 [8] J. Hays and I. Essa, “Image and video based painterly animation,” *Proceedings of SIGGRAPH 04*, Annecy, France, 1997, pp.113-120.
 [9] T. Kim, and M. C. Lin, “Visual simulation of ice crystal growth,” *Proceedings of SIGGRAPH 03*, San Diego, California, 2003, pp.86-97.
 [10] D. C. Wu and W. H. Tsai, “A Steganographic Method for Images by Pixel-Value Differencing,” *Pattern Recognition Letters*, Vol. 24, No. 9-10, 2003, pp. 1623-1636.
 [11] Y. C. Chiu “A study on digital watermarking and authentication of images for copyright protection and tampering detection,” *M. S. Thesis*, Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2004.
 [12] W. L. Ling “Data hiding in image mosaics,” *M. S. Thesis*, Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2004.
 [13] W. L. Lin and W. H. Tsai, “Data Hiding in Image Mosaics by Visible Boundary Regions and Its Copyright Protection Application against Print-And-Scan Attacks,” *Proceedings of International Computer Symposium 2004*, Taipei, Taiwan, Dec. 15-17, 2004.
 [14] S. C. Hung, “Data hiding in art images,” *M. S. Thesis*, Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2005.
 [15] C. Y. Hsu, “A Study on Art Image Generation and Information Hiding,” *M. S. Thesis*, Institute of Computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2006.



(a)



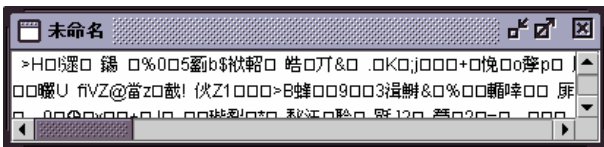
(b)



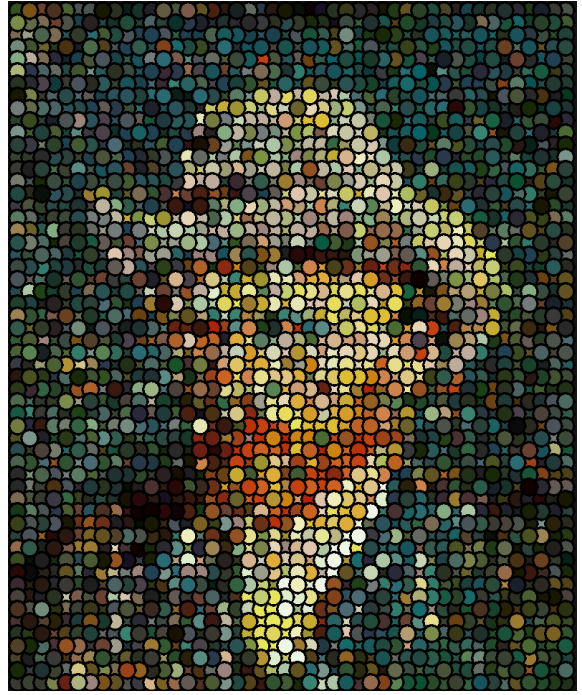
(c)



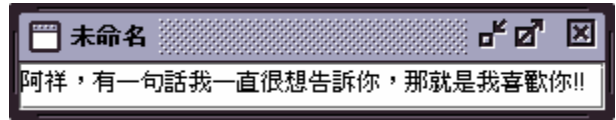
(d)



(e)



(a)



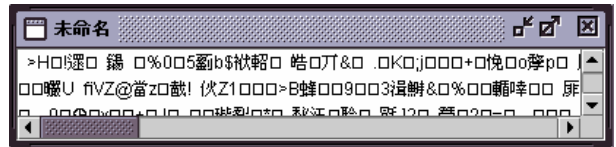
(b)



(c)



(d)



(e)

Fig. 3 Experimental results of data hiding in a circular-dotted image. (with input $D=11$, $T=1$) (a) A circular-dotted image with secret message and watermark (c) embedded. (b) The secret message extracted from (a) with a correct key. (d) The watermark extracted from (a) with a correct key.

Fig. 4 Experimental results of data hiding in a circular-dotted image. (with input $D=17$, $T=3$) (a) A circular-dotted image with secret message and watermark (c) embedded. (b) The secret message extracted from (a) with a correct key. (d) The watermark extracted from (a) with a correct key.