A New Secure Image Transmission Technique via Secret-fragment-visible Mosaic Images by Nearly-reversible Color Transformations*

Ya-Lin Lee, *Student Member, IEEE* and Wen-Hsiang Tsai, *Senior Member, IEEE*

***Abstract*—A new secure image transmission technique is proposed, which transforms automatically a given large-volume secret image into a so-called secret-fragment-visible mosaic image of the same size. The mosaic image, which looks similar to an arbitrary-selected target image and may be used as a camouflage of the secret image, is yielded by dividing the secret image into fragments and transforming their color characteristics to be those of the corresponding blocks of the target image. Skillful techniques are designed to conduct the color transformation process so that the secret image may be recovered nearly losslessly. A scheme of handling the overflows/underflows in the converted pixels' color values by recording the color differences in the un-transformed color space is also proposed. The information required for recovering the secret image is embedded into the created mosaic image by a lossless data hiding scheme using a key. Good experimental results show the feasibility of the proposed method.**

***Index Terms*—Secure image transmission, mosaic image, color transformation, image encryption, data hiding.**

I. Introduction

Nowadays, images from various sources are frequently utilized and transmitted through the Internet for various applications, such as online personal photograph albums, confidential enterprise archives, document storage systems, medical imaging systems, military image databases, etc. These images usually contain private or confidential information so that they should be protected from leakages during transmissions. Recently, many methods have been proposed for the purpose of secure image transmission, for which two common approaches are image encryption and data hiding.

Image encryption is a technique that obscures the natural property of an image, such as high redundancy and strong spatial correlation, to get an encrypted image based on Shannon's confusion and diffusion properties [1]-[7]. The encrypted image is a noise image so that no one can obtain the secret image from it unless he/she has the correct key. However, the encrypted image is a *meaningless* file, which cannot provide additional information before decryption and may arouse an attacker's attention during transmission due to its randomness in form. An alternative to avoid this problem is data hiding [8]-[18] that hides a secret message into a cover image so that no one can realize the existence of the secret data, where the data type of the secret message investigated in this study is an image. Existing data hiding methods mainly utilize the techniques of LSB substitution [8], histogram shifting [9], difference expansion [10]-[11], prediction-error expansion [12]-[13], recursive histogram modification [14], discrete cosine/wavelet transformations [15]-[18], etc.

However, in order to reduce the distortion of the resulting image, an upper bound for the distortion value is usually set on the payload of the cover image. A discussion on this rate-distortion issue can be found in [19]. Thus, a main issue of the methods for hiding data in images is the difficulty to embed a large amount of message data into a single image. Specifically, if one wants to hide a secret image into a cover image with the same size, the secret image must be highly compressed in advance. For example, for a data hiding method with an embedding rate of 0.5 bits per pixel, a secret image with 8 bits per pixel must be compressed at a rate of at least 93.75% beforehand in order to be hidden into a cover image. But, for many applications, such as keeping or transmitting medical pictures, military images, legal documents, etc., that are valuable with no allowance of serious distortions, such data compression operations are usually impractical. Moreover, most image compression methods, such as JPEG compression, are not suitable for line drawings and textual graphics, where sharp contrasts between adjacent pixels are often destructed to become noticeable artifacts [20].

In this study, a new technique for secure image transmission is proposed, which transforms a secret image into a *meaningful mosaic image* with the same size and looking like a *pre-selected target image*. The transformation process is controlled by a secret key, and only with the key can a person recover the secret image *nearly losslessly* from the mosaic image. The proposed method is inspired by Lai and Tsai [21],

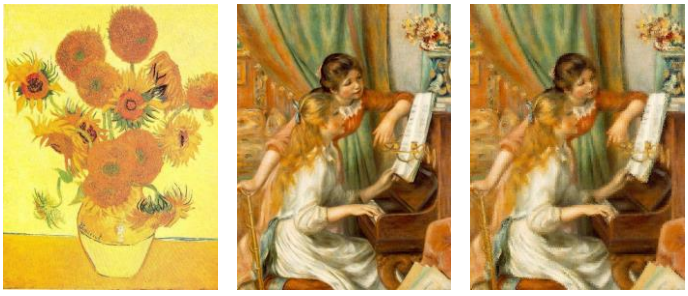
*This research was supported in part by the NSC, Taiwan under Grant No. 101-3113-P-009-006 and Grant No. 102-2218-E-009-003, and in part by the Ministry of Education, Taiwan under the 5-year Project of "Aiming for the Top University" from 2011 through 2015.

Y. L. Lee is with the Institute of Computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan 30010 (e-mail: yllee.cs98g@g2.nctu.edu.tw).

W. H. Tsai is with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan 30010. He is also with the Department of Information Communication, Asia University, Taichung, Taiwan 41354 (email: whtsai@cis.nctu.edu.tw).

where a new type of computer art image, called secret-fragment-visible mosaic image, was proposed. The mosaic image is the result of rearrangement of the fragments of a secret image in disguise of another image called the *target image* pre-selected from a database. But an obvious weakness of Lai and Tsai [21] is the requirement of a large image database so that the generated mosaic image can be sufficiently similar to the selected target image. Using their method, the user is *not* allowed to select freely his/her favorite image for use as the target image. It is therefore desired in this study to remove this weakness of the method while keeping its merit, that is, it is aimed to design a new method which can transform a secret image into a secret-fragment-visible mosaic image of the same size that has the visual appearance of *any freely-selected* target image *without the need of a database*.

As an illustration, Fig. 1 shows a result yielded by the proposed method. Specifically, after a target image is selected arbitrarily, the given secret image is first divided into rectangular fragments called *tile images*, which then are fit into similar blocks in the target image, called *target blocks*, according to a similarity criterion based on color variations. Next, the color characteristic of each tile image is transformed to be that of the corresponding target block in the target image, resulting in a mosaic image which looks like the target image. Relevant schemes are also proposed to conduct *nearly lossless* recovery of the original secret image from the resulting mosaic image. The proposed method is new in that a *meaningful* mosaic image is created, in contrast with the image encryption method that only creates *meaningless* noise images. Also, the proposed method can transform a secret image into a disguising mosaic image without compression, while a data hiding method must hide a highly compressed version of the secret image into a cover image when the secret image and the cover image have the same data volume.



(a) (b) (c)

Fig. 1. A result yielded by the proposed method. (a) Secret image. (b) Target image. (c) Secret-fragment-visible mosaic image created from (a) and (b) by the proposed method.

In the remainder of this paper, the idea of the proposed method is described in Sections 2 and 3. Detailed algorithms for mosaic image creation and secret image recovery are given in Section 4. In Section 5, experimental results are presented to show the feasibility of the proposed method, and in Section 6, the security issue of the proposed method is discussed, followed by conclusions in Section 7.

II. Ideas of Proposed Method

The proposed method includes two main phases as shown by the flow diagram of Fig. 2: 1) mosaic image creation; and 2) secret image recovery.

In the first phase, a mosaic image is yielded, which consists of the fragments of an input secret image with color corrections according to a similarity criterion based on color variations. The phase includes four stages: 1) fitting the tile images of the secret image into the target blocks of a pre-selected target image; 2) transforming the color characteristic of each tile image in the secret image to become that of the corresponding target block in the target image; 3) rotating each tile image into a direction with the minimum RMSE value with respect to its corresponding target block; and 4) embedding relevant information into the created mosaic image for future recovery of the secret image. And in the second phase, the embedded information is extracted to recover nearly losslessly the secret image from the generated mosaic image. The phase includes two stages: 1) extracting the embedded information for secret image recovery from the mosaic image; and 2) recovering the secret image using the extracted information.

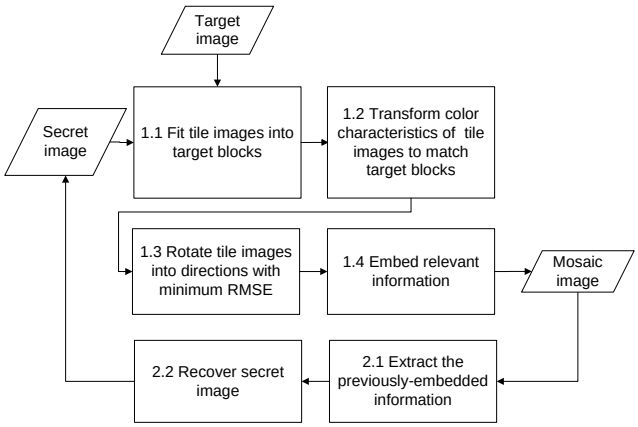


Fig. 2. Flow diagram of the proposed method.

III. Ideas of Mosaic Image Generation

Problems encountered in generating mosaic images are discussed in this section with solutions to them proposed.

A. Color Transformations between Blocks

In the first phase of the proposed method, each tile image T in the given secret image is fit into a target block B in a pre-selected target image. Since the color characteristics of T and B are different from each other, how to change their color distributions to make them look alike is the main issue here. Reinhard *et al.* [22] proposed a color transfer scheme in this aspect, which converts the color characteristic of an image to be that of another in the lαβ color space. This idea is an answer to the issue and is adopted in this study, except that the RGB color space instead of the lαβ one is used to reduce the volume of the required information for recovery of the original secret image.

More specifically, let T and B be described as two pixel sets

$\{p_1, p_2, \ldots, p_n\}$ and $\{p_1', p_2', \ldots, p_n'\}$, respectively. Let the color of each p_i be denoted by (r_i, g_i, b_i) and that of each p_i' by (r_i', g_i', b_i'). At first, we compute the means and standard deviations of T and B, respectively, in each of the three color channels R, G, and B by the following formulas:

$$\mu_c = \frac{1}{n}\sum_{i=1}^{n} c_i, \quad \mu_c' = \frac{1}{n}\sum_{i=1}^{n} c_i' ; \tag{1}$$

$$\sigma_c = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_i - \mu_c)^2}, \quad \sigma_c' = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_i' - \mu_c')^2} \tag{2}$$

where c_i and c_i' denote the C-channel values of pixels p_i and p_i', respectively, with $c = r$, g, or b and C = R, G, or B. Next, we compute new color values (r_i'', g_i'', b_i'') for each p_i in T by:

$$c_i'' = q_c(c_i - \mu_c) + \mu_c' , \tag{3}$$

where $q_c = \sigma_c'/\sigma_c$ is the *standard deviation quotient* and $c = r$, g, or b. It can be verified easily that the new color mean and variance of the resulting tile image T' are equal to those of B, respectively. To compute the original color values (r_i, g_i, b_i) of p_i from the new ones (r_i'', g_i'', b_i''), we use the following formula which is the inverse of (3):

$$c_i = (1/q_c)(c_i'' - \mu_c') + \mu_c . \tag{4}$$

Furthermore, we have to embed into the created mosaic image sufficient information about the new tile image T' for use in the later stage of recovering the original secret image. For this, theoretically we can use (4) to compute the original pixel value of p_i. However, the involved mean and standard deviation values in the formula are all real numbers, and it is impractical to embed real numbers, each with many digits, in the generated mosaic image. Therefore, we limit the numbers of bits used to represent relevant parameter values in (3) and (4). Specifically, for each color channel we allow each of the means of T and B to have 8 bits with its value in the range of 0 to 255, and the standard deviation quotient q_c in (3) to have 7 bits with its value in the range of 0.1 to 12.8. That is, each mean is changed to the closest value in the range of 0 to 255, and each q_c is changed to the closest value in the range of 0.1 to 12.8. We do *not* allow q_c to be 0 because otherwise the original pixel value cannot be recovered back by (4) for the reason that $1/q_c$ in (4) is not defined when $q_c = 0$.

B. Choosing Appropriate Target Blocks and Rotating Blocks to Fit Better with Smaller RMSE Value

In transforming the color characteristic of a tile image T to be that of a corresponding target block B as described above, how to choose an appropriate B for each T is an issue. For this, we use the standard deviation of the colors in the block as a measure to select the most similar B for each T. Specially, we sort all the tile images to form a sequence, S_{tile}, and all the target blocks to form another, S_{target}, according to the *average* values of the standard deviations of the three color channels. Then, we fit the first in S_{tile} into the first in S_{target}, fit the second in S_{tile} into the second in S_{target}, and so on.

Additionally, after a target block B is chosen to fit a tile image T and after the color characteristic of T is transformed, we conduct a further improvement on the color similarity between the resulting tile image T' and the target block B by rotating T' into one of the four directions, 0°, 90°, 180°, and 270°, which yields a rotated version of T' with the *minimum* root mean square error (RMSE) value with respect to B among the four directions for final use to fit T into B.

C. Handling Overflows/Underflows in Color Transformation

After the color transformation process is conducted as described previously, some pixel values in the new tile image T' might have *overflows* or *underflows*. To deal with this problem, we convert such values to be non-overflow or non-underflow ones and record the value differences as *residuals* for use in later recovery. Specifically, we convert all the transformed pixel values in T' not smaller than 255 to be 255, and all those not larger than 0 to be 0. Next, we compute the differences between the original pixel values and the converted ones as the residuals and record them as part of the information associated with T'. Accordingly, the pixel values which are just on the bound of 255 or 0, however, cannot be distinguished from those with overflow/underflow values during later recovery since all the pixel values with overflows/underflows are converted to be 255 or 0 now. To remedy this, we define the residuals of those pixel values which are on the bound to be "0" and record them as well.

But as can be seen from (3), the ranges of possible residual values are unknown, and this causes a problem of deciding how many bits should be used to record a residual. To solve this problem, we record the residual values in the *un-transformed* color space rather than in the transformed one. That is, by using the following two formulas we compute first the smallest possible color value c_S (with $c = r$, g, or b) in T that becomes larger than 255 as well as the largest possible value c_L in T that becomes smaller than 0, respectively, after the color transformation process has been conducted:

$$\begin{aligned} c_S &= \lceil (1/q_c)(255 - \mu_c') + \mu_c \rceil; \\ c_L &= \lfloor (1/q_c)(0 - \mu_c') + \mu_c \rfloor. \end{aligned} \tag{5}$$

Next, for an un-transformed value c_i which yields an overflow after the color transformation, we compute its residual as $|c_i - c_S|$; and for c_i which yields an underflow, we compute its residual as $|c_L - c_i|$. Then, the possible values of the residuals of c_i will all lie in the range of 0 to 255 as can be verified. Consequently, we can simply record each of them with 8 bits. And finally, because the residual values are *centralized around zero*, we use further in this study *the Huffman encoding scheme* to encode the residuals in order to reduce the numbers of required bits to represent them.

D. Embedding Information for Secret Image Recovery

In order to recover the secret image from the mosaic image, we have to embed relevant recovery information into the mosaic image. For this, we adopt a technique proposed by Coltuc and Chassery [24] and apply it to the least significant bits of the pixels in the created mosaic image to conduct data

embedding. Unlike the classical LSB replacement methods [8], [25], [26], which substitute LSBs with message bits directly, the reversible contrast mapping method [24] applies simple integer transformations to pairs of pixel values. Specifically, the method conducts forward and backward integer transformations as follows, respectively, where (x, y) are a pair of pixel values and (x', y') are the transformed ones:

$$x' = 2x - y, \quad y' = 2y - x; \tag{6}$$

$$x = \left\lceil \frac{2}{3}x' + \frac{1}{3}y' \right\rceil, \quad y = \left\lceil \frac{1}{3}x' + \frac{2}{3}y' \right\rceil. \tag{7}$$

The method yields high data embedding capacities close to the highest bit rates and has the lowest complexity reported so far.

The information required to recover a tile image T which is mapped to a target block B includes: 1) the index of B; 2) the optimal rotation angle of T; 3) the truncated means of T and B and the standard deviation quotients, of all color channels; and 4) the overflow/underflow residuals. These data items for recovering a tile image T are integrated as a five-component bit stream of the form

$$M = t_1t_2\ldots t_m r_1r_2 m_1m_2\ldots m_{48} q_1q_2\ldots q_{21} d_1d_2\ldots d_k,$$

where the bit segments $t_1t_2\ldots t_m$, r_1r_2, $m_1m_2\ldots m_{48}$, $q_1q_2\ldots q_{21}$, and $d_1d_2\ldots d_k$ represent the values of the index of B, the rotation angle of T, the means of T and B, the standard deviation quotients, and the residuals, respectively.

In more detail, the numbers of required bits for the five data items in M are discussed below: 1) the index of B needs m bits to represent, with m computed by: $m = \lceil \log[(W_S \times H_S)/N_T] \rceil$, where W_S and H_S are respectively the width and height of the secret image S, and N_T is the size of the target image T; 2) it needs two bits to represent the rotation angle of T because there are four possible rotation directions; 3) 48 bits are required to represent the means of T and B because we use eight bits to represent a mean value in each color channel; 4) it needs 21 bits to represent the quotients of T over B in the three color channels with each channel requiring 7 bits; and 5) the total number k of required bits for representing all the residuals depends on the number of overflows or underflows in T'.

Then, the above-defined bit streams of all the tile images are concatenated in order further into a *total bit stream* M_t for the entire secret image. Moreover, in order to protect M_t from being attacked, we encrypt it with a secret key to obtain an encrypted bit stream M_t', which is finally embedded into the pixel pairs in the mosaic image using the method of Coltuc and Chassery [24] described above. It may require more than one iteration in the encoding process since the length of M_t' may be larger than the number of pixel pairs available in an iteration. A plot of the statistics of the numbers of required bits for secret image recovery is shown in Fig. 8(b).

Moreover, we have to embed as well some related information about the mosaic image generation process into the mosaic image for use in the secret image recovery process. Such information, described as a bit stream I like M mentioned previously, includes the following data items: 1) the number of iterations conducted in the process for embedding the bit stream M_t'; 2) the total number of used pixel pairs in the last iteration for embedding M_t'; and 3) the Huffman table for encoding the residuals.

With the bit stream M_t' embedded into the mosaic image, we can recover the secret image back as will be described later. It is noted that some *loss* will be incurred in the recovered secret image, or more specifically, in the color transformation process using (3) where each pixel's color value c_i is multiplied by the standard deviation quotient q_c and the resulting real value c_i'' is truncated to be an integer in the range of 0 through 255. However, because each truncated part is smaller than the value of 1, the recovered value of c_i using (4) is still precise enough to yield a color nearly identical to its original one. Even when overflows/underflows occur at some pixels in the color transformation process, we record their residual values as described previously and after using (4) to recover the pixel value c_i, we add the residual values back to the computed pixel values c_i to get the original pixel data, yielding a nearly losslessly-recovered secret image. According to the results of the experiments conducted in this study, each recovered secret image has a very small RMSE value with respect to the original secret image, as will be shown later in Section V.

IV. Algorithms of Proposed Method

Based on the above discussions, the detailed algorithms for mosaic image creation and secret image recovery may now be described as follows.

Algorithm 1. ***Mosaic image creation.***
Input: a secret image S, a target image T, and a secret key K.
Output: a secret-fragment-visible mosaic image F.
Steps:

Stage 1 — fitting the tile images into the target blocks.

Step 1. If the size of the target image T is different from that of the secret image S, change the size of T to be identical to that of S; and divide the secret image S into n tile images $\{T_1, T_2, \ldots, T_n\}$ as well as the target image T into n target blocks $\{B_1, B_2, \ldots, B_n\}$ with each T_i or B_i being of size N_T.

Step 2. Compute the means and the standard deviations of each tile image T_i and each target block B_j for the three color channels according to (1) and (2); and compute accordingly the average standard deviations for T_i and B_j, respectively, where $i = 1$ through n and $j = 1$ through n.

Step 3. Sort the tile images in the set $S_{tile} = \{T_1, T_2, \ldots, T_n\}$ and the target blocks in the set $S_{target} = \{B_1, B_2, \ldots, B_n\}$ according to the computed average standard deviation values of the blocks; map in order the blocks in the sorted S_{tile} to those in the sorted S_{target} in a 1-to-1 manner; and reorder the mappings according to the indices of the tile images, resulting in a *mapping sequence* L of the form: $T_1 \rightarrow B_{j_1}$, $T_2 \rightarrow B_{j_2}$, …, $T_n \rightarrow B_{j_n}$.

Step 4. Create a mosaic image F by fitting the tile images into the corresponding target blocks according to L.

Stage 2 — performing color conversions between the tile images and the target blocks.

Step 5. Create a *counting table TB* with 256 entries, each with an index corresponding to a residual value, and assign an initial value of zero to each entry (note that each residual value will be in the range of 0 to 255).

Step 6. For each mapping $T_i \rightarrow B_{j_i}$ in sequence L, represent the means μ_c and μ_c' of T_i and B_{j_i}, respectively, by eight bits; and represent the standard deviation quotient q_c appearing in (3) by seven bits, according to the scheme described in Section 3(A) where $c = r$, g, or b.

Step 7. For each pixel p_i in each tile image T_i of mosaic image F with color value c_i where $c = r$, g, or b, transform c_i into a new value c_i'' by (3); if c_i'' is not smaller than 255 or if it is not larger than 0, then change c_i'' to be 255 or 0, respectively; compute a residual value R_i for pixel p_i by the way described in Section 3(C); and increment by 1 the count in the entry in the counting table *TB* whose index is identical to R_i.

Stage 3 — rotating the tile images.

Step 8. Compute the RMSE values of each color-transformed tile image T_i in F with respect to its corresponding target block B_{j_i} after rotating T_i into each of the directions $\theta = 0^o$, 90^o, 180^o and 270^o; and rotate T_i into the *optimal* direction θ_o with the smallest RMSE value.

Stage 4 — embedding the secret image recovery information.

Step 9. Construct a Huffman table *HT* using the content of the counting table *TB* to encode all the residual values computed previously.

Step 10. For each tile image T_i in mosaic image F, construct a bit stream M_i for recovering T_i in the way as described in Section 3(D), including the bit-segments which encode the data items of: 1) the index of the corresponding target block B_{j_i}; 2) the optimal rotation angle θ_o of T_i; 3) the means of T_i and B_{j_i} and the related standard deviation quotients of all three color channels; and 4) the bit sequence for overflows/underflows with residuals in T_i encoded by the Huffman table *HT* constructed in Step 9.

Step 11. Concatenate the bit streams M_i of all T_i in F in a raster-scan order to form a total bit stream M_t; use the secret key K to encrypt M_t into another bit stream M_t'; and embed M_t' into F by the reversible contrast mapping scheme proposed in [24].

Step 12. Construct a bit stream I including: 1) the number of conducted iterations N_i for embedding M_t'; 2) the number of pixel pairs N_{pair} used in the last iteration; and 3) the Huffman table *HT* constructed for the residuals; and embed the bit stream I into mosaic image F by the same scheme used in Step 11.

Algorithm 2. *Secret image recovery.*

Input: a mosaic image F with n tile images $\{T_1, T_2, \ldots, T_n\}$ and the secret key K.

Output: the secret image S.

Steps:

Stage 1 — extracting the secret image recovery information.

Step 1. Extract from F the bit stream I by a reverse version of the scheme proposed in [24] and decode them to obtain the following data items: 1) the number of iterations N_i for embedding M_t'; 2) the total number of used pixel pairs N_{pair} in the last iteration; and 3) the Huffman table *HT* for encoding the values of the residuals of the overflows or underflows.

Step 2. Extract the bit stream M_t' using the values of N_i and N_{pair} by the same scheme used in the last step.

Step 3. Decrypt the bit stream M_t' into M_t by K.

Step 4. Decompose M_t into n bit streams M_1 through M_n for the n *to-be-constructed* tile images T_1 through T_n in S, respectively.

Step 5. Decode M_i for each tile image T_i to obtain the following data items: 1) the index j_i of the block B_{j_i} in F corresponding to T_i; 2) the optimal rotation angle θ_o of T_i; 3) the means of T_i and B_{j_i} and the related standard deviation quotients of all color channels; and 4) the overflow/underflow residual values in T_i decoded by the Huffman table *HT*.

Stage 2 — recovering the secret image.

Step 6. Recover one by one in a raster-scan order the tile images T_i, $i = 1$ through n, of the desired secret image S by the following steps: 1) rotate in the reverse direction the block indexed by j_i, namely B_{j_i}, in F through the optimal angle θ_o and fit the resulting block content into T_i to form an *initial* tile image T_i; 2) use the extracted means and related standard deviation quotients to recover the original pixel values in T_i according to (4); 3) use the extracted means, standard deviation quotients, and (5) to compute the two parameters c_S and c_L; 4) scan T_i to find out pixels with values 255 or 0 which indicate that overflows or underflows, respectively, have occurred there; 5) add respectively the values c_S or c_L to the corresponding residual values of the found pixels; and 6) take the results as the final pixel values, resulting in a *final* tile image T_i.

Step 7. Compose all the final tile images to form the desired secret image S as output.

V. Experimental Results

A series of experiments have been conducted to test the proposed method using many secret and target images with sizes 1024×768 or 768×1024. To show that the created mosaic image looks like the pre-selected target image, the quality metric of root mean square error (RMSE) is utilized, which is defined as the square root of the mean square difference between the pixel values of the two images.

An example of the experimental results is shown in Fig. 3, where Fig. 3(c) shows the created mosaic image using Fig. 3(a) as the secret image and Fig. 3(b) as the target image. The tile image size is 8×8. The recovered secret image using a correct key is shown in Fig. 3(d) which looks nearly identical to the original secret image shown in Fig. 3(a) with RMSE = 0.948 with respect to the secret image. It is noted by the way that all the other experimental results shown in this paper have small

RMSE values as well, as seen in Fig. 8(c).

Fig. 3. An experimental result of mosaic image creation. (a) Secret image. (b) Target image. (c) Mosaic image created with tile image size 8×8. (d) Recovered secret image using a correct key with RMSE = 0.948 with respect to secret image (a). (e) Recovered secret image using a wrong key. (f)-(i) Mosaic images created with different tile image sizes 16×16, 24×24, 32×32, and 40×40.

Moreover, Fig. 3(e) shows the recovered secret image using a wrong key, which is a noise image. Figs. 3(f) through 3(i) show more results using different tile image sizes. It can be seen from the figures that the created mosaic image retains more details of the target image when the tile image is smaller. It can also be seen that the blockiness effect is observable when the image is magnified to be large; but if the image is observed as a whole, it still looks like a mosaic image with its appearance similar to the target image. Fig. 8(a) also shows this fact in another way — a mosaic image created with smaller tile images has a smaller RMSE value with respect to the target image. On the other hand, the number of required bits embedded for recovering the secret image will be increased when the tile image becomes smaller, as can be seen from Fig. 8(b).

Fig. 4 shows a comparison of the results yielded by the proposed method with those by Lai and Tsai [21], where Fig. 4(a) is the input secret image, Fig. 4(b) is the selected target image, Fig. 4(c) is the mosaic image created by Lai and Tsai [21], and Fig. 4(d) is that created by the proposed method. It can be seen from these results that the mosaic image yielded by the proposed method has a smaller RMSE value with respect to the target image, implying that it is more similar to the target image in appearance. The other results of our experiments also show the same conclusion. And more importantly, the proposed method allows users to select their favorite images for uses as target images.

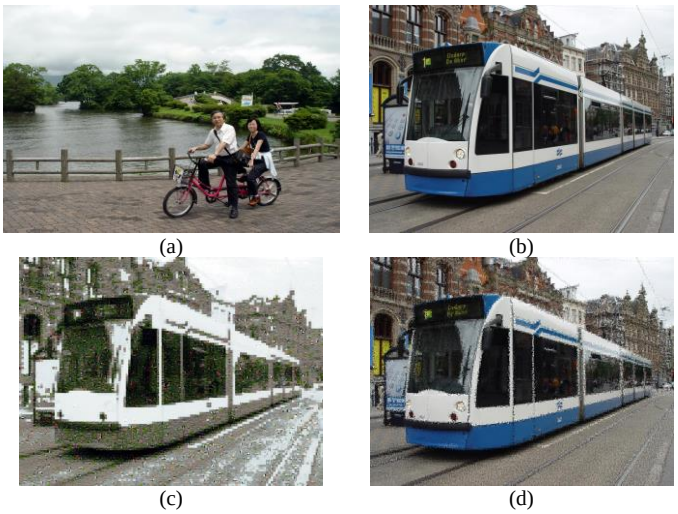
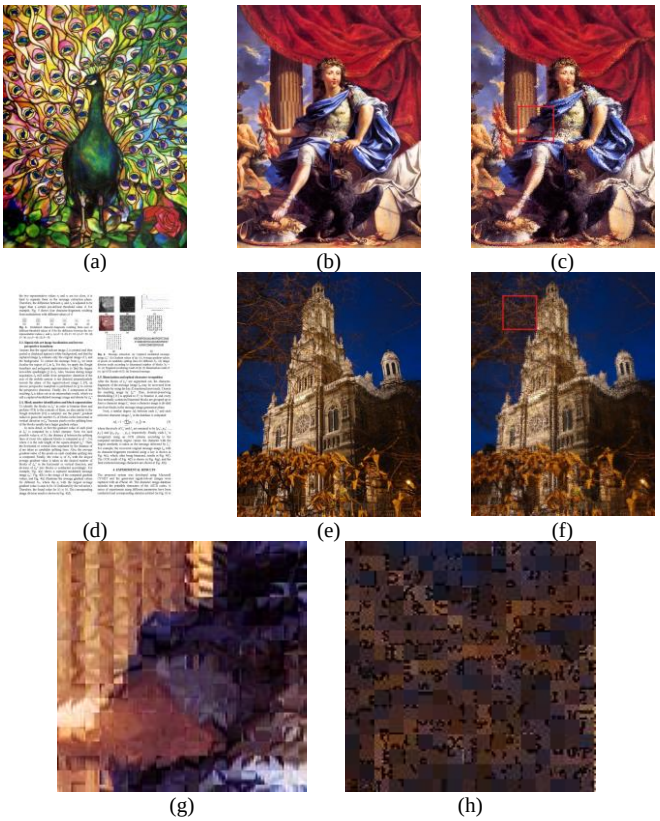


Fig. 4. Comparison of results of Lai and Tsai [21] and proposed method. (a) Secret image. (b) Target image. (c) Mosaic image created from (a) and (b) by [21] with RMSE = 47.651. (d) Mosaic image created from (a) and (b) by proposed method with RMSE = 33.935.

Fig. 5 shows two other experimental results of mosaic image creation, where the utilized secret images both contain many structures (Fig. 5(a) is a stained-glass window painting and Fig. 5(d) is a document image) and Figs. 5(b) and 5(e) are the target images; Figs. 5(c) and 5(f) are the created mosaic images with image sizes 8×8; and Figs. 5(g) and 5(h) are the zoom-out images of the red square regions of Figs. 5(c) and 5(f), respectively. It can be seen from Figs. 5(c) and 5(f) that each created mosaic image still has the visual appearance of the pre-selected target image even when the secret image contains many structural elements. Especially, the secret image of Fig. 5(d) is a nearly black-and-white *document image*, which means that the proposed method can be utilized for secure transmissions of confidential document images as well. Moreover, it can be seen from Figs. 5(g) and 5(h) that each generated mosaic image has a blocky appearance which comes from the mosaic effect because the mosaic image is composed by changing the color characteristics of the fragments of the secret image and rearranging the resulting fragments. To show the flexibility of the proposed method for a user to choose *any* target image as the reference of a secret image, we selected one secret image as shown in Fig. 6(a) and two target images as shown in Figs. 5(b) and 5(e), and transformed the former to have the visual appearance of each of the latter ones. The results are shown in Figs. 6(b) and 6(c) from which we can see that the created mosaic images look similar to the respective target images even though the secret image is quite different from the target images in appearance.

However, since the mosaic image is yielded by dividing the secret image into tile images and transforming their color characteristics to be those of the corresponding target blocks,

the global color characteristics of a transformed tile image and its corresponding target block are the same but the color distributions of them may be quite different. Hence, although the mosaic image has the visual appearance of the target image, the details of each fragment in the mosaic image may have low similarity to those of its corresponding target block. To measure this *mosaic effect*, we adopt the metric of mean structural similarity (MSSIM) to compare the similarity of the created mosaic image and the target image [27]. Fig. 8(d) shows the MSSIM values of the created mosaic images with respect to the target images versus different tile image sizes, where the window size for computing the MSSIM is set to be the same as the size of the tile image. We can see from Fig. 8(d) that the MSSIM value of the created mosaic image with respect to the target image varies from 0.2 to 0.8, which shows that the similarity of the details of the created mosaic image to those of the target image is not good enough. But, this is not the main concern of the proposed method because our goal is to create a globally visually-similar mosaic image, which contains a secret image of the same size, for the purpose of secure image transmission.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5. Two other experimental results of mosaic image creation. (a) and (d) Secret images. (b) and (e) Target images. (c) and (f) Mosaic images created from (a) and (b), and (d) and (e), respectively, with tile size 8×8. (g) and (h) Zoom-out images of red square regions of (c) and (f), respectively.

A limitation of the proposed method is that the sizes of available target images should match those of possible input secret images. Specifically, if we have a very large secret image but only small target images for selections, then any selected target image should be enlarged before mosaic image creation in order to match the size of the secret image, and the created mosaic image will become *blurred*. An experimental result showing this blurring effect is presented in Fig. 7.

Furthermore, as shown in Fig. 8, we have drawn plots of the trends of various parameters versus different tile image sizes, including those for the parameters of 1) the RMSE values of the created mosaic images with respect to the target images; 2) the numbers of required bits embedded for recovering the secret images; 3) the RMSE values of the recovered secret images with respect to the original ones; and 4) the MSSIM values of created mosaic images with respect to target images.

In addition, we have conducted experiments on a set of 12 images from which a total of 12×11 = 132 secret-target image pairs are selected without repetitions, and the averages of the parameters of the 132 mosaic image creation results were also plotted in Fig. 8 as the orange curves for comparisons. Finally, it is mentioned that the images utilized in the experiments can be accessed on the internet [32].

(a) (b) (c)

Fig. 6. Created mosaic images with the same secret image. (a) Secret image. (b) Mosaic image created from (a) and Fig. 5(b) with RMSE = 26.067. (c) Mosaic image created from (a) and Fig. 5(e) with RMSE = 33.102.

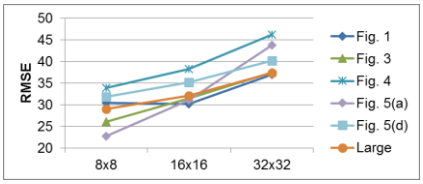
(a) (b) (c)

Fig. 7. Created mosaic images with the same secret image shown in Fig. 5(a) and small-sized target images. (a) Created image for target image shown in Fig. 5(b) with size 768×1024. (b) Created image for target image shown in Fig. 5(b) but with size reduced to (1/5)×(1/5). (c) Created image for target image shown in Fig. 5(b) but with size reduced to (1/10)×(1/10).

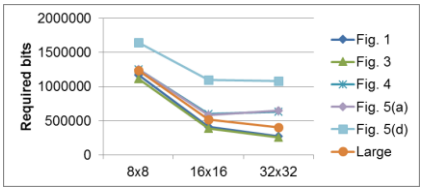
VI. Security Considerations

In order to increase the security of the proposed method, the embedded information for later recovery is encrypted with a secret key as seen in Algorithm 1. Only the receiver who has the key can decode the secret image. However, an eavesdropper who does not have the key may still try all possible permutations of the tile images in the mosaic image to get the secret image back. Fortunately, the number of all possible permutations here is $n!$, and so the probability for him/her to correctly guess the permutation is $p = 1/n!$ which is very small in value. For example, for the typical case where we divide a

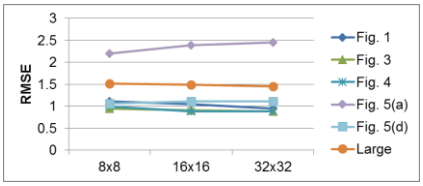
secret image of size 1024×768 into tile images with block size 8×8, the value n is (1024×768)/(8×8) = 12,288. So the probability to guess the permutation correctly without the key is $1/n! = 1/(12{,}288!)$. So breaking the system by this way of guessing is computationally infeasible.



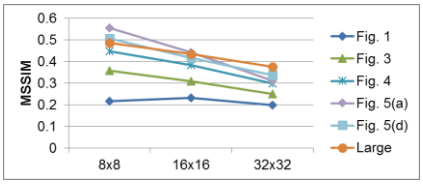
(a)



(b)



(c)



(d)

Fig. 8. Plots of trends of various parameters versus different tile image sizes (8×8, 16×16, 32×32) with input secret images shown previously and coming from a large dataset. (a) RMSE values of created mosaic images with respect to target images. (b) Numbers of required bits embedded for recovering secret images. (c) RMSE values of recovered secret images with respect to original ones. (d) MSSIM values of created mosaic images with respect to target images.

In fact, we can view the addressed problem here as a *square jigsaw puzzle problem*, which is to reconstruct a complete image from a set of unordered square puzzle parts. Recently, many methods have been proposed to try to solve this problem automatically by utilizing measures of feature-based similarity [28], dissimilarity-based compatibility [29], prediction-based compatibility [30], etc. But these state-of-art methods can only solve partially problems with limited numbers of puzzle parts automatically. Also, the jigsaw puzzle problem has been proved to be NP-complete [31], which means that we cannot solve the problem in polynomial time. In fact, the time complexity is $n! \approx \sqrt{2\pi n}(n/e)^n$ as mentioned in [31], which is too big a number as well for our case here with n = 12,288.

However, when n is much smaller, say smaller than 1000, some compatibility metrics may be utilized to solve the square jigsaw problem [30]. So, a large value of n should be used to increase the security of the proposed method. In addition, the addressed puzzle problem of the proposed method is more complicated than the conventional square jigsaw puzzle problem because the color characteristics of the puzzle parts have been changed, i.e., adjacent puzzle parts have different color appearances, meaning that a greedy search using color similarities between originally adjacent fragments for image reconstruction as done in conventional manual reconstruction techniques is infeasible, either.

Furthermore, even if one happens to guess the permutation correctly, such as the correctly guessed permutations shown in Fig. 9, he/she still does not know the correct parameters for recovering the original color appearance of the secret image because such parameter information for color recovery is encrypted as a bit stream using a secret key. Even so, it still should be assumed, in the extreme case, that he/she will observe the content of the mosaic image with a correct permutation, and try to figure useful information out of it. For example, an attacker might analyze the spatial continuity of the mosaic image in order to estimate a rough version of the secret image. To increase the security of the proposed method against this type of attack, one possible way to is to use the key to randomize the important part of a secret image, such as the positions of the pixels in the secret image, before transforming the secret image into a mosaic image by the proposed method. Consequently, only authorized users with the key can know the correct secret image while an attacker cannot.

VII. Conclusion

A new secure image transmission method has been proposed, which not only can create *meaningful* mosaic images but also can transform a secret image into a mosaic one with the same data size for use as a camouflage of the secret image. By the use of proper pixel color transformations as well as a skillful scheme for handling overflows and underflows in the converted values of the pixels' colors, secret-fragment-visible mosaic images with very high visual similarities to arbitrarily-selected target images can be created with no need of a target image database. Also, the original secret images can be recovered nearly losslessly from the created mosaic images. Good experimental results have shown the feasibility of the proposed method. Future studies may be directed to applying the proposed method to images of color models other than the RGB.

(a)

(b)

Fig. 9. Correct permutations of tile images in the mosaic image without recovering the original color characteristics. (a) The correct permutation of tile images of Fig. 1(c). (b) The correct permutation of tile images of Fig. 5(c).

Acknowledgment

The authors would like to thank the reviewers for many

useful comments and suggestions which improve the presentation of the paper.

REFERENCES

[1] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurc. Chaos*, vol. 8, pp.1259-1284, 1998.

[2] G. Chen ,Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp.749-761, 2004.

[3] L. H. Zhang, X. F. Liao and X. B. Wang, "An image encryption approach based on chaotic maps," *Chaos, Solitons & Fractals*, vol. 24, no. 3, pp. 759-765, 2005.

[4] H. S. Kwok and W. K. S. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," *Chaos, Solitons & Fractals*, vol. 32, no. 4, pp.1518-1529, 2007.

[5] S. Behnia, A. Akhshani, H. Mahmodi, and A. Akhavan, "A novel algorithm for image encryption based on mixture of chaotic maps," *Chaos, Solitons & Fractals*, vol. 35, pp. 408-419, 2008.

[6] D. Xiao, X. Liao and P. Wei, "Analysis and improvement of a chaos-based image encryption algorithm," *Chaos, Solitons & Fractals*, vol. 40 pp. 2191-2199, 2009.

[7] V. Patidar, N.K. Pareek, G. Purohit, and K.K Sud, "A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption," *Opt. Commun.*, vol. 284, no. 19, pp. 4331–4339, 2011.

[8] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recog.*, vol. 37, pp. 469-474, March 2004.

[9] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible Data Hiding," *IEEE Trans. Circuits Syst. & Video Technol.*, vol. 16, no. 3, pp. 354-362, March 2006.

[10] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. & Video Technol.*, vol. 13, no. 8, pp. 890-896, Aug. 2003.

[11] Y. Hu, H.-K. Lee, K. Chen, and J. Li, "Difference expansion based reversible data hiding using two embedding directions," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1500-1512, 2008.

[12] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. & Video Technol.*, vol. 19, no. 7, pp. 989-999, 2009.

[13] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Trans on Image Processing*, vol. 20, no. 12, pp. 3524-3533, 2011.

[14] W. Zhang, X. Hu, X. Li, and N. Yu, "Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression," *IEEE Trans. Image Processing*, vol. 22, no. 7, pp. 2775-2785, 2013.

[15] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," *Proc. SPIE*, vol. 3971, pp. 197-208, 2001.

[16] C. C. Chang, C. C. Lin, C. S. Tseng, and W. L. Tai, "Reversible hiding in DCT-based compressed images," *Information Sciences*, vol. 177, pp. 2768-2786, 2007.

[17] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Trans. Information Forensics and Security*, vol. 2, no. 3, pp. 321-330, 2007.

[18] W.-H. Lin, S.-J. Horng , T.-W. Kao , P. Fan , C.-L. Lee and Y. Pan, "An efficient watermarking method based on significant difference of wavelet coefficient quantization," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp.746-757, 2008.

[19] X. Hu, W. Zhang, X. Hu, N. Yu, X. Zhao, and F. Li, "Fast estimation of optimal marked-signal distribution for reversible data hiding," *IEEE Trans. Information Forensics and Security*, vol. 8, no. 5, pp. 187-193, May 2013.

[20] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, pp. 34-38, 1993.

[21] I. J. Lai and W. H. Tsai, "Secret-fragment-visible mosaic image — a new computer art and its application to information hiding," *IEEE Trans. Information Forensics and Security*, vol. 6, no. 3, pp. 936-945, 2011.

[22] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, 2001.

[23] D. L. Ruderman, T. W. Cronin, and C. C. Chiao, "Statistics of Cone Responses to Natural Images: Implications for Visual Coding," *J. Optical Soc. of America*, vol. 15, no. 8, pp. 2036-2045, 1998.

[24] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255-258, 2007.

[25] R. Z. Wang, C. F. Lin, and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," *Pattern Recog.*, vol. 34, no. 3, pp. 671-683, 2001.

[26] C. H. Yang, "Inverted pattern approach to improve image quality of information hiding by LSB substitution," *Pattern Recog.*, vol. 41, no. 8, pp. 2674-2683, 2008.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.

[28] T. R. Nielsen, P. Drewsen, and K. Hansen, "Solving jigsaw puzzles using image features," *Pattern Recog. Letters*, vol. 29, no. 14, pp. 1924-1933, 2008.

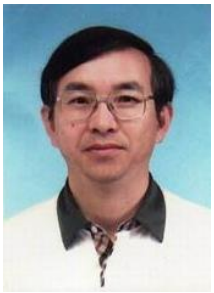
[29] T. S. Cho, S. Avidan, and W. T. Freeman, "A probabilistic image jigsaw puzzle solver," *Proc. IEEE CVPR*, San Francisco, CA, USA, pp. 183-190, 2010.

[30] D. Pomeranz, M. Shemesh, and O. Ben-Shahar, "A fully automated greedy square jigsaw puzzle solver," *Proc. IEEE CVPR*, Providence, RI, USA, pp. 9-16, 2011.

[31] E. Demaine and M. Demaine, "Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity," *Graphs and Combinatorics*, vol. 23, pp. 195-208, 2007.

[32] Related images of the experiments, [Online]. Available: http://people.cs.nctu.edu.tw/~yllee/yllee&whtsai_sfv.html.

Ya-Lin Lee received the B. S. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan in 2009. She is currently pursuing the Ph.D. degree at the College of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. Her current research interests include information hiding, image processing, pattern recognition, and artificial intelligence.



Wen-Hsiang Tsai (S'78-M'79-SM'91) received the B.S. degree in EE from National Taiwan University, Taiwan, in 1973, the M.S. degree in EE from Brown University, USA in 1977, and the Ph.D. degree in EE from Purdue University, USA in 1979. Since 1979, he has been with National Chiao Tung University (NCTU), Taiwan, where he is now a Chair Professor of Computer Science. At NCTU, he has served as the Head of the Dept. of Computer Science, the Dean of General Affairs, the Dean of Academic Affairs, and a Vice President. From 1999 to 2000, he was the Chair of the Chinese Image Processing and Pattern Recognition Society of Taiwan, and from 2004 to 2008, the Chair of the Computer Society of the IEEE Taipei Section in Taiwan. From 2004 to 2007, he was the President of Asia University, Taiwan.

Dr. Tsai has been an Editor or the Editor-in-Chief of several international journals, including *Pattern Recognition*, the *International Journal of Pattern Recognition and Artificial Intelligence*, and the *Journal of Information Science and Engineering*. He has published 157 journal papers and 246 conference papers and received many awards, including the Annual Paper Award from the Pattern Recognition Society of the USA; the Academic Award of the Ministry of Education, Taiwan; the Outstanding Research Award of the National Science Council, Taiwan; the ISI Citation Classic Award from Thomson Scientific, and more than 40 other academic paper awards from various academic societies. His current research interests include computer vision, information security, video surveillance, and autonomous vehicle applications. He is a Life Member of the Chinese Pattern Recognition and Image Processing Society, Taiwan and a Senior Member of the IEEE.