# Secret-Fragment-Visible Mosaic Image–A New Computer Art and Its Application to Information Hiding

I-Jen Lai and Wen-Hsiang Tsai, *Senior Member, IEEE*

***Abstract*—A new type of computer art image called secret-fragment-visible mosaic image is proposed, which is created automatically by composing small fragments of a given image to become a target image in a mosaic form, achieving an effect of embedding the given image visibly but secretly in the resulting mosaic image. This effect of information hiding is useful for covert communication or secure keeping of secret images. To create a mosaic image of this type from a given secret color image, the 3-D color space is transformed into a new 1-D colorscale, based on which a new image similarity measure is proposed for selecting from a database a target image that is the most similar to the given secret image. A fast greedy search algorithm is proposed to find a similar tile image in the secret image to fit into each block in the target image. The information of the tile image fitting sequence is embedded into randomly-selected pixels in the created mosaic image by a lossless LSB replacement scheme using a secret key; without the key, the secret image cannot be recovered. The proposed method, originally designed for dealing with color images, is also extended to create grayscale mosaic images which are useful for hiding text-type grayscale document images. An additional measure to enhance the embedded data security is also proposed. Good experimental results show the feasibility of the proposed method.**

***Index Terms*—Computer art, covert communication, greedy search, information hiding, secret-fragment-visible mosaic image.**

## I. Introduction

MOSAIC is a type of artwork created by composing small pieces of materials, such as stone, glass, tile, etc. Invented in ancient time, they are still used in many applications today. Creation of *mosaic images* by computer [1] is a new research direction in recent years. Many methods have been proposed to create different types of mosaic images by computer. A good survey under a unified framework can be found in Battiato *et al.* [2] in which a taxonomy of mosaic images into four types is proposed, including *crystallization mosaic*, *ancient mosaic*, *photo-mosaic*, and *puzzle image mosaic*. The first two types are obtained from decomposing a source image into tiles (with different colors, sizes, and rotations) and reconstructing the image by properly painting the tiles, and so they both may be called *tile mosaics*. The other two types of mosaics are obtained by fitting images from a database to cover an assigned source image, and both may be called *multi-picture mosaics*. Haeberli [3] proposed a method to create crystallization mosaic images using voronoi diagrams by placing blocks at random sites and filling colors into the blocks based on the content of the original image. Hausner [4] created ancient mosaic images by using centroidal voronoi diagrams. Dobashi *et al.* [5] improved the voronoi diagram to add various effects to the mosaic image, such as simulation of stained glasses. Elber and Wolberg [6] proposed a method for rendering ancient mosaics by recovering free-form feature curves from the image and laying rows of tiles along the curves. Kim and Pellacini [7] generated a kind of puzzle image mosaic, called *jigsaw image mosaic*, composed of many arbitrary shapes of tiles selected from a database. Di Blasi *et al.* [8] presented a new puzzle image mosaic as an improvement on the jigsaw image mosaic proposed in [7] in the aspect of computation time using a suitable data structure. Di Blasi and Gallo [9] created a kind of puzzle image mosaic, which reproduces the colors of the original image and emphasizes relevant boundaries by placing tiles along the edge directions. Battiato *et al.* [10], [11] generated ancient mosaic images using gradient vector flows to follow the most important edges in the original image and to maximize the covered mosaic area. Narasimhan and Satheesh [12] viewed the process of photo-mosaic generation as an optimization problem with a constraint on the repetition of a tile image and proposed a randomized iterative algorithm more efficient than the conventional genetic algorithm. By accelerating pattern searching and minimizing the memory cost, Choi *et al.* [13] presented a genetic feature selection method for optimization of an image set for producing photo-mosaics in real time. Battiato and Puglisi [14] investigated 3-D ancient mosaics recently.

A new type of art image, called *secret-fragment-visible mosaic image*, which contains small fragments of a given source image is proposed in this study. Observing such a type of mosaic image, one can see all the fragments of the source image, but the fragments are so tiny in size and so random in position that the observer cannot figure out what the source image looks like. Therefore, the source image may be said to be *secretly* embedded in the resulting mosaic image, though the fragment

Manuscript received August 26, 2010; revised February 24, 2011; accepted March 04, 2011. Date of publication April 05, 2011; date of current version August 17, 2011. This work was supported by the NSC project No. 98-2631-H-009-002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wenjun Zeng.

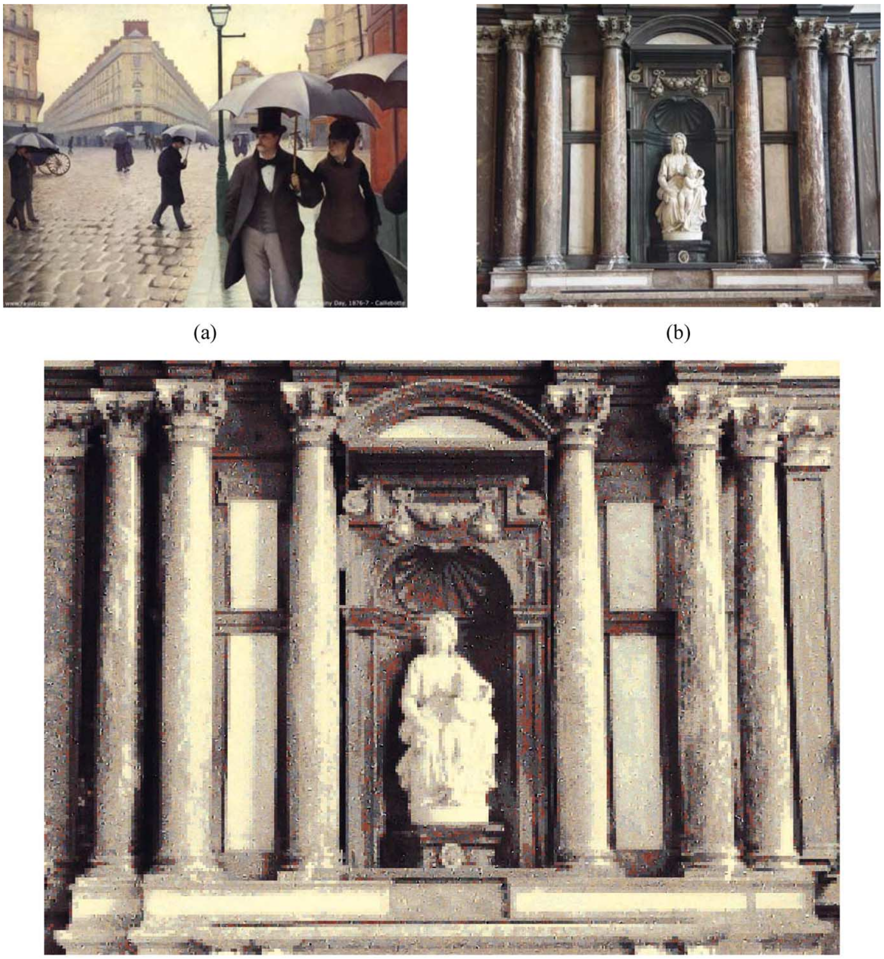
I. J. Lai is with the Institute of Computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan 30010 (e-mail: nekolai.cs97g@g2.nctu.edu.tw).

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 (e-mail: whtsai@cis.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2011.2135853

1556-6013/$26.00 © 2011 IEEE



(a) (b) (c)

Fig. 1. Example of results yielded by proposed method. (a) An image. (b) Another image. (c) Secret-fragment-visible mosaic image created with (a) as secret source image and (b) as target image.

pieces are all *visible* to the observer. And this is the reason why the resulting mosaic image is named *secret-fragment-visible*. An example of such images created by the proposed method is shown in Fig. 1. Because of this characteristic of the new mosaic image, it may be used as a carrier of a *secret* source image in the disguise of another—a *target* image of a different content. This is a new technique of *information hiding*, not found in the literature so far. It is useful for the application of covert communication or secure keeping of secret images.

More specifically, as illustrated by Fig. 2, a secret image is first divided into rectangular-shaped fragments, called *tile images*, which are fitted next into a *target image* selected from a database to create a mosaic image. The number of usable tile images for this operation is *limited* by the size of the secret image and that of the tile images. This is not the case in traditional mosaic image creation where available tile images for use essentially are *unlimited* in number because the tile images are *not* generated from the secret image and may be used *repeatedly*. Then, the information of tile-image fitting is embedded into some blocks of the mosaic image, which are selected randomly by a secret key. Accordingly, an observer possessing the key can reconstruct the secret image by retrieving the embedded information, while a hacker without the key cannot.

In the remainder of this paper, the basic idea of the proposed method is described in Section II. Problems encountered in the mosaic image creation process are discussed in Section III. Detailed algorithms for mosaic image creation and secret image recovery are presented in Sections IV and V, respectively. Relevant experimental results are also included. An extension of the method to create grayscale mosaic images is presented in Section VI, and some discussions on security enhancement are given in Section VII. Finally, conclusions and suggestions for future studies are given in Section VIII.

## II. Basic Idea and Database Construction

### A. Basic Idea of Proposed Method

A flow diagram of the proposed method is shown in Fig. 3, which includes three phases of works:

Phase 1—construction of a color image database for use in selecting similar target images for given secret images;

Phase 2—creation of a secret-fragment-visible mosaic image using the tile images of a secret image and the selected similar target image as input;

Phase 3—recovery of the secret image from the created secret-fragment-visible mosaic image.

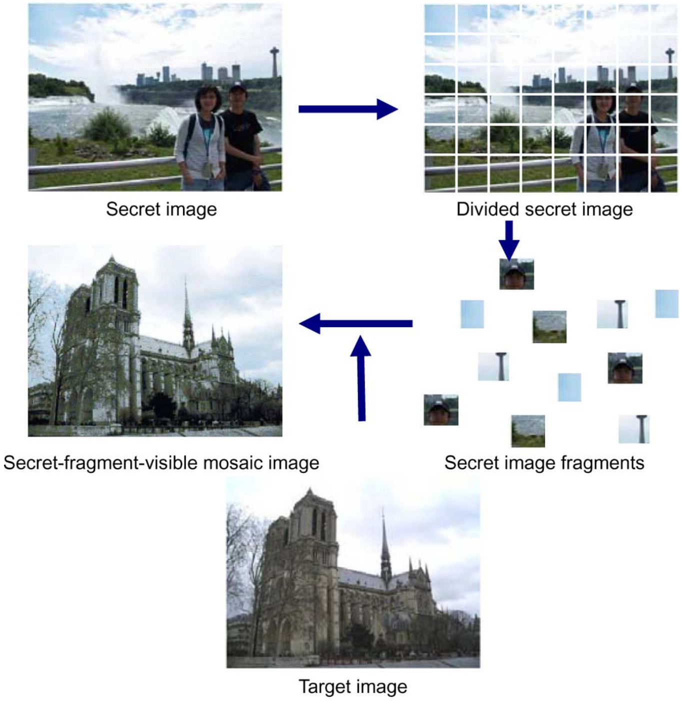


Fig. 2. Illustration of creation of secret-fragment-visible mosaic image.

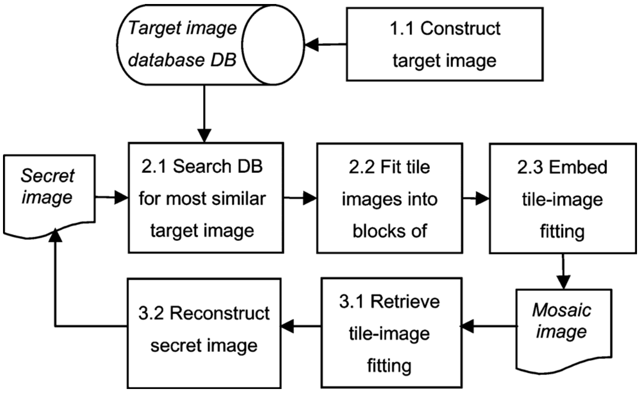

Target image database DB
1.1 Construct target image
Secret image
2.1 Search DB for most similar target image
2.2 Fit tile images into blocks of
2.3 Embed tile-image fitting
Mosaic image
3.1 Retrieve tile-image fitting
3.2 Reconstruct secret image

Fig. 3. Processes for secret-fragment-visible mosaic image creation and secret image recovery.

The first phase includes mainly the work of database construction. The second phase includes three stages of operations:

Stage 2.1—searching the database for a target image the most similar to the secret image;

Stage 2.2—fitting the tile images in the secret image into the blocks of the target image to create a mosaic image;

Stage 2.3—embedding the tile-image fitting information into the mosaic image for later secret image recovery.

And the third phase includes two stages of operations:

Stage 3.1—retrieving the previously-embedded tile-image fitting information from the mosaic image;

Stage 3.2—reconstructing the secret image from the mosaic image using the retrieved information.

In the remainder of this section, we describe how we construct the database in the first phase, and how we select a similar target image from the database for a given secret image as described in Stage 2.1 of the second phase of the proposed method. Other stages are dealt with in subsequent sections.

### B. *Database Construction*

The target image database plays an important role in the mosaic image creation process. If a selected target image from the database is dissimilar to a given secret image, the created mosaic image will be distinct from the target one, resulting in a reduction of the information hiding effect. To generate a better result, the database should be as large as possible. Searching a database for a target image with the highest similarity to a given secret image is a problem of content-based image retrieval. A state-of-art survey of studies on this problem can be found in Lew *et al.* [15]. In general, the content of an image may be described by features like shape, texture, color, etc. Due to the use of small tile images in the proposed method, which are the fragments of the secret image, it is found in this study that the most effective feature, which affects the overall visual appearance of the resulting mosaic image, is color. Therefore, we focus on extracting color distributions from images to define an appropriate image similarity measure for use in target image selection in this study.

One way for extracting the global characteristic of the color distribution of an image is the 1-D color histogram transformation technique proposed by Smith and Chang [16]. The technique requantizes first the color values $(r, g, b)$ into fewer levels, say $N_r$, $N_g$, and $N_b$ ones, respectively, resulting in the new color values $(r', g', b')$. Then, it transforms the three new values $(r', g', b')$ into a single one by

$$f(r', g', b') = r' + N_r \times g' + N_r \times N_g \times b'. \tag{1}$$

However, according to our experimental experience, the use of this 1-D color value $f$, originally proposed just for color indexing, was found *inappropriate* for our study here where the *human's visual feeling* of image similarity is emphasized. Two results yielded by the proposed method using (1) above are shown in Fig. 4, where Fig. 4(c) and (g) were created respectively with Fig. 4(a) and (e) as the input secret images and Fig. 4(b) and (f) as the selected target images. As can be seen, the resulting mosaic images in both cases are quite noisy.

Therefore, we propose alternatively in this study a new color transformation function $h$ as follows:

$$h(r', g', b') = b' + Nb \times r' + N_b \times N_r \times g' \tag{2}$$

where, differently from the case in (1), the numbers of levels, $N_r$, $N_g$, and $N_b$, are all set to be 8, and the largest weight, namely, the value $N_b \times N_r$, is assigned to the green channel value $g'$ and the smallest weight, the value 1, is assigned to the blue channel value $b'$. This way of weight assignment is based on the fact [17] that the human eye is the most sensitive to the green color and the least sensitive to the blue one, leading to a larger emphasis on the intensity of the resulting mosaic image. In addition, with all of $N_r$, $N_g$, and $N_b$ set to be 8 in (2), the proposed mosaic image creation process can be speeded up according to our experimental experience. Subsequently, we will say that the new color function $h$ proposed in (2) defines a 1-D *h-colorscale*. The mosaic image created by the proposed

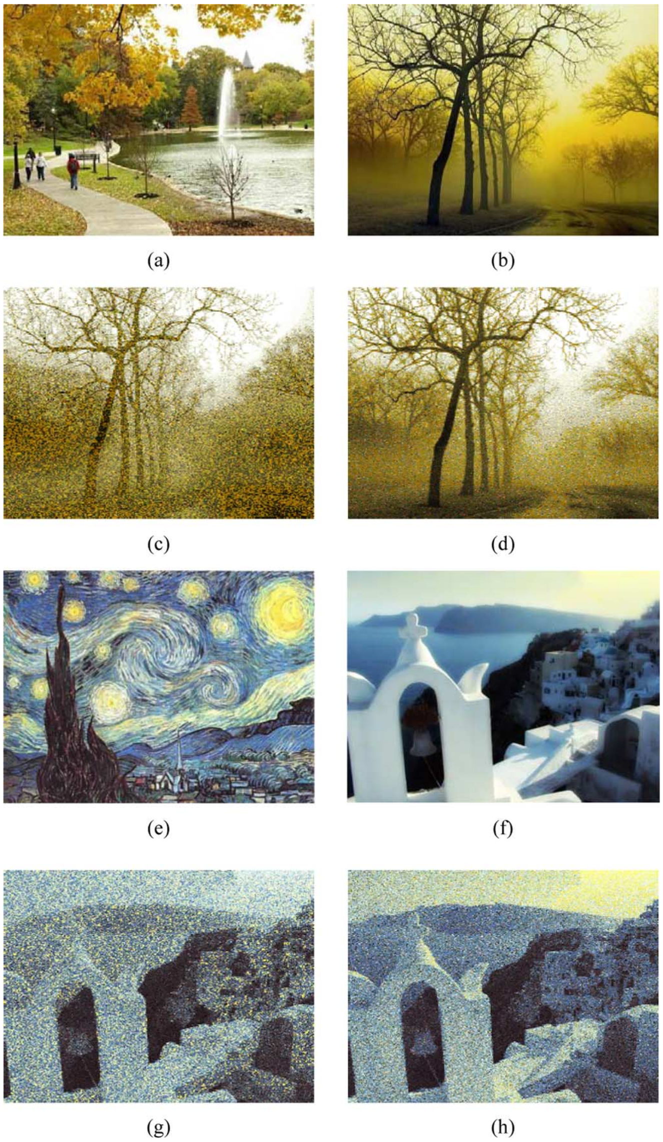


Fig. 4. Effects of mosaic image creation based on similarity measures using different 1-D color features. (a) and (e) Secret images. (b) and (f) Target images. (c) and (g) Mosaic images created with similarity measure based on (1) proposed in [16]. (d) and (h) Mosaic images created with similarity measure based on (2) of proposed method.

method using a similarity measure based on this new colorscale is given in Fig. 4(d) and (h), which contrastively have less noise when compared with Fig. 4(c) and (g), respectively.

Furthermore, to compute the similarity between a tile image in the secret image and a block in a target image (called a *target block* hereafter) for use in the tile-image fitting process during mosaic image creation, we propose a new feature for each image block $c$ (either a tile image or a target block), which is called *h-feature*, denoted as $h_c$, and computed by the following steps:

1) compute the average of the RGB color values of all the pixels in image block $c$ as $(r_c, g_c, b_c)$;
2) re-quantize the RGB color scales into $N_r$, $N_g$, and $N_b$ levels, respectively, and transform accordingly $(r_c, g_c, b_c)$ into $(r'_c, g'_c, b'_c)$ in terms of the three new color levels;
3) compute the $h$-feature value $h_c$ for $c$ by (2) above, resulting in the following equation:

$$h_c\left(r'_c, g'_c, b'_c\right) = b'_c + N_b \times r'_c + N_b \times N_r \times g'_c. \tag{3}$$

With $N_r$, $N_g$, and $N_b$ all set equal to 8, the computed values of the $h$-feature $h_c$ defined above may be figured out to be in the range of 0 to 511. The process proposed in this study for construction of a database *DB* of *candidate target images* from a set $M$ of arbitrarily-selected images all with a preselected size $Z_c$ for use in secret-fragment-visible mosaic image creation proceeds in the following way for each input image $D$ in $M$: divide $D$ into target blocks of a preselected size $Z_t$, compute the $h$-feature value defined by (3) for each target block, generate accordingly an $h$-feature histogram $H$ of $D$, and finally save all the $h$-feature values of the target blocks and the histogram $H$ of $D$ into the desired database *DB*.

### C. Image Similarity Measure and Target Image Selection

Before generating the secret-fragment-visible mosaic image for a given secret image $S$ with the preselected size $Z_c$, we have to choose from the database *DB* a target image which is the most similar to $S$. For this, first we divide $S$ into blocks of the preselected size $Z_t$, compute the $h$-feature values of all the resulting blocks by (3), and generate the $h$-feature histogram $H_S$ of $S$. Then, we define an *image similarity* value $m(S, D)$ between $S$ and each candidate target image $D$ with $h$-feature histogram $H_D$ in *DB* in the following way:

$$m(S, D) = \frac{1}{\sum_{h=0}^{511} |H_S(h) - H_D(h)|} \tag{4}$$

where $H_X(h)$ with $X = S$ or $D$ is the number of image blocks in the "bin" of feature value $h$. The larger the value $m(S, D)$ is, the more similar $D$ and $S$ are to each other. If the corresponding $h$-features in $H_S$ and $H_D$ are all identical, then $S$ and $D$ are regarded to be *totally similar in the h-feature sense*. After calculating the image similarity values of all the candidate target images in *DB* with respect to $S$, we select finally the image $D_o$ in *DB* with the largest similarity as the desired target image for $S$ for use in mosaic image creation.

## III. Problems Encountered in Mosaic Image Creation

### A. Problem of Fitting Tile Images Into Target Blocks

Given a secret image $S$, after the most similar target image $D_o$ is selected, we have to find a tile image in $S$ to fit into each target block in $D_o$. This problem of fitting a limited number of tile images into a target image *in an optimal way* may be reduced, as can be figured out, to be a *single-source shortest path problem*, which aims to finding a path in a graph with the *smallest sum of between-vertex edge weights*. Here, the state of fitting a tile image is represented by a vertex of the graph, and the action of fitting the tile image into a target block may be represented by an edge of the graph with its weight taken to be the similarity value between the pixels' colors of the tile image and those of the target block. Accordingly, if there are $N$ target blocks (and so the same number of tile images), the graph for this problem is just a $N$-level tree with three properties: 1) the root at the first depth is a virtually-created *single source* for the graph; 2) the $N$ nodes at the second depth specify all the $N$ tile images, each of which may be chosen to fit into the first target block, and so on; and 3) each leaf node at the deepest $N$th depth is a solution of fitting all the $N$ tile images into the $N$ target blocks.

To find the optimal solution, it seems that we may utilize the Dijkstra algorithm [19] whose running time is of the complexity of $O(V^2)$ where $V$ denotes the number of vertices in the tree. Unfortunately, the value of $V$, as can be figured out, is

$$V = 1 + N + N \times (N-1) + \cdots + N \times (N-1) \times \cdots \times 1 = \sum_{n=1}^{N} \left( \frac{N!}{n!} \right)$$

which is an enormously large number because the number $N$ of tile images in each of the secret images used in this study is larger than 40,000. Therefore, the computation time to get an optimal solution by the Dijkstra algorithm is too high to be practical, meaning that we have to find other feasible ways to solve the problem. For this, we propose to use a *greedy search* algorithm to find suboptimal solutions which, though nonoptimal, are found feasible in this study for information hiding applications.

Also, we need a *selection function* for the greedy search algorithm to select a tile image $s$ the most similar to each target block $d$. For this, it seems natural to take the function to be the measure of the *average Euclidean distance* between the pixels' colors of $s$ and those of $d$. However, as shown by the example of Fig. 5(c) which is the result of using such a selection function with Fig. 5(a) and (b) as the secret image and the target image, respectively, the performance of the greedy search algorithm was found *unsatisfactory*, yielding an unacceptable result Fig. 5(c) with the blocks of the lower image part all being filled with fragments of inappropriate colors! This phenomenon results from the situation that the number of tile images obtained from a secret image [like Fig. 5(a)] is *limited* by the size of the secret image, so that the remaining tile images available for choice to fit the target blocks [like those in Fig. 5(b)] near the end of the fitting process become less and less; and as a result, the similarity values between the later-processed target blocks and the remaining tile images become smaller and smaller than those of the earlier-processed ones, yielding a poorly-fitted lower part in the resulting mosaic image [like the lower image part of Fig. 5(c)].

A feasible solution to this problem as found in this study is to use as the selection function based on the previously-mentioned concept of $h$-feature, instead of on the concept of Euclidean distance. Specifically, we define the *block similarity value* $m(s, d)$ between a tile image $s$ with $h$-feature value $h_s$ and a target block $d$ with $h$-feature value $h_d$ by

$$m(s, d) = \frac{1}{|h_s - h_d|}. \tag{5}$$

This $h$-feature-based similarity measure takes into more consideration the relative intensity difference between the compared image blocks (the tile image and the target block), and helps creating a mosaic image with its content visually resembling the target image in a more global way, as shown by the example of Fig. 5(d) which indeed is an improvement of Fig. 5(c).

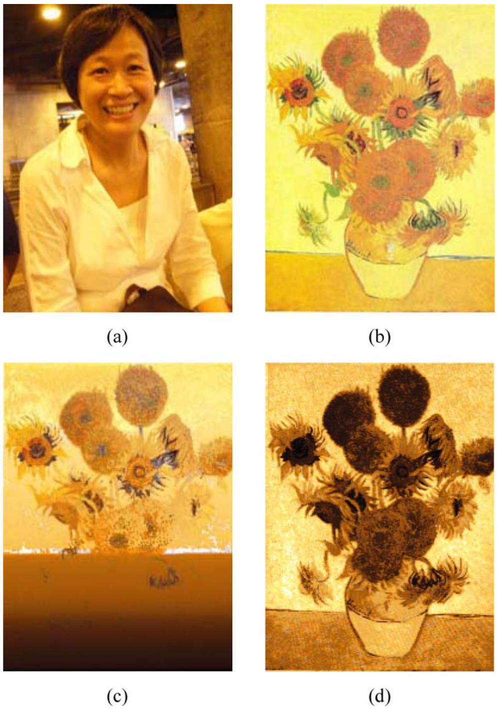

(a) (b) (c) (d)

Fig. 5. Mosaic image creation using different similarity measures. (a) Secret image. (b) Target image. (c) Mosaic image created using Euclidean distance to define select function for greedy search. (d) Mosaic image created using $h$-feature to define select function for greedy search.

### B. Issue of Recovering the Secret Image

Another issue which should be dealt with in creating the mosaic image is how to embed the information of tile-image fitting so that the original secret image can be reconstructed from the created mosaic image. Each fitting of a tile image $s$ into a target block $d$ forms a mapping from $s$ to $d$. The way we propose for dealing with the issue is to record these mappings into a sequence $L_R$, called the *secret recovery sequence*, and embed $L_R$ into randomly-selected blocks in the created mosaic image using a technique of lossless least-significant-bit (LSB) replacement proposed by Coltuc and Chassery [18].

In more detail, to get the mappings, we start from the topleftmost target block $d_1$ in the selected target image $D_o$, and find for it the most similar tile image $s_i$ in the secret image $S$ in the sense of (5), and form the first mapping $s_i \rightarrow d_1$ to be included in $L_R$. Next, in a raster-scan order, we process the target block $d_2$ to the right of $d_1$ to find the most similar tile image $s_j$ in the remaining tile images to form the second mapping $s_j \rightarrow d_2$ for $L_R$. Then, we do similarly to find the third mapping $s_k \rightarrow d_3$, and so on. We continue this *greedy search* process until the last target block at the bottom-rightmost corner in the target image is processed. The resulting $L_R$ may be regarded to include two block-index sequences, $L_1 = i, j, k, \ldots$ and $L_2 = 1, 2, 3, \ldots$ with mappings $i \rightarrow 1$, $j \rightarrow 2$, $k \rightarrow 3$, and so on. Since $L_2$ is a well-ordered sequence of $1, 2, 3, \ldots$, we can ignore it and take $L_R$ to include just $L_1$ to reduce the data volume of $L_R$ to be embedded.

Also, it is not difficult to figure out that if the width and height of a given secret image $S$ are $W_S$ and $H_S$, respectively, with $Z_t$ being the previously-mentioned size of the tile images in $S$, then the number $N$ of tile images in $S$, the number $N_X$ of bits required to specify the index of a tile image, and the number $N_R$ of bits required to represent the secret recovery sequence $L_R$, respectively, are as follows:

$$N = \frac{W_S \times H_S}{Z_t} \tag{6}$$

$$N_X = \lfloor \log_2 N \rfloor + 1 \tag{7}$$

$$N_R = N \times N_X \tag{8}$$

where $\lfloor \cdot \rfloor$ means the integer floor function. Furthermore, since each color pixel has three channels for use to embed bits and since the lossless LSB replacement scheme [18] we adopt needs two LSBs in an identical channel to embed a bit, the number $N_T$ of bits that can be embedded into a tile image is just

$$N_T = \frac{(3 \times Z_t)}{2} \tag{9}$$

because each tile image has $Z_t$ pixels. These data of $N$, $N_X$, $N_R$, and $N_T$ will be used later in describing the algorithms for mosaic image creation and secret image recovery.

## IV. Secret-Fragment-Visible Mosaic Image Creation

Based on the above discussions, a complete algorithm implementing the proposed idea for creating mosaic images (i.e., the phase-2 work described in Section II-A) is described in the following, followed by some experimental results.

### A. Mosaic Image Creation Algorithm

**Algorithm 1**: ***secret-fragment-visible mosaic image creation***.

**Input**: a secret image $S$ with a preselected size $Z_c$; a preselected size $Z_t$ of tile images; a database *DB* of candidate target images with size $Z_c$; and a random number generator $g$ and a secret key $K$.

**Output**: a secret-fragment-visible mosaic image $U$ for $S$.

**Steps**.

***Stage 1—selecting the most similar target image***.

Step 1. Divide $S$ into tile images of size $Z_t$, record the width $W_S$ and height $H_S$ of $S$, and compute the number $N$ of tile images in $S$ by **(6)**.

Step 2. Select from *DB* the target image $D_o$ that is the most similar to $S$ in the sense of **(4)** (see **Section II-C** for the detail).

***Stage 2—fitting tile images into target blocks***.

Step 3. Calculate the $h$-feature values of all the tile images in $S$ and take out the $h$-feature values of all the target blocks of $D_o$ from *DB*.

Step 4. In a raster-scan order of the target blocks in $D_o$, perform the greedy search process to find the most similar tile images $s_i, s_j, s_k, \ldots$ in $S$ corresponding to the $N$ target blocks $d_1, d_2, d_3, \ldots$ in $D_o$, respectively, to construct the secret recovery sequence $L_R = i, j, k, \ldots$ using the $h$-feature values obtained in the last step.

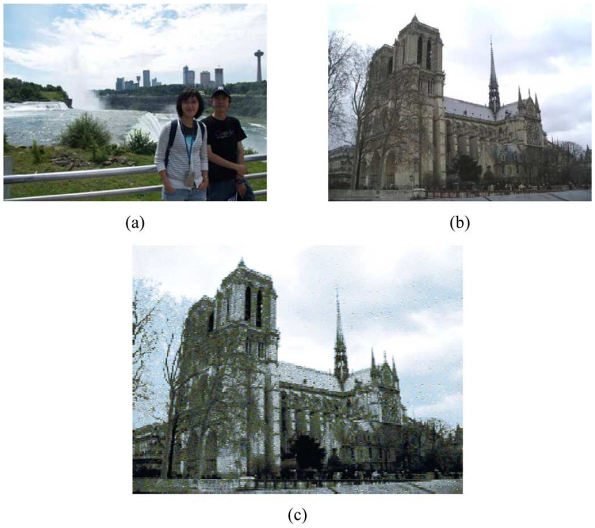


Fig. 6. Experimental result of mosaic image creation using Algorithm 1. (a) Secret image. (b) Target image. (c) Created mosaic image.

Step 5. Fit the tile images $s_i, s_j, s_k, \ldots$ into the corresponding target blocks $d_1, d_2, d_3, \ldots$, respectively, to generate a *preliminary* secret-fragment-visible mosaic image $U$.

***Stage 3—embedding tile-image fitting information***.

Step 6. Concatenate the data of the width $W_S$ and height $H_S$ of $S$ as well as the size $Z_t$, transform the concatenation result into a binary string, and embed it into the first ten pixels of the first block of image $U$ in a raster-scan order by the lossless LSB replacement scheme proposed in **[18]**.

Step 7. Transform $L_R$ into a binary string with its length $N_R$ computed by **(6)–(8)**.

Step 8. Repetitively select randomly a block $s$ in $U$ *unselected so far* other than the first block of $U$ using the random number generator $g$ with the secret key $K$ as the seed, and embed $N_T$ bits of $L_R$ into all the $Z_t$ pixels of $s$ by the lossless LSB replacement scheme proposed in **[18]**, until all the $N_R$ bits in $L_R$ are exhausted, where $N_T$ is computed by **(9)**.

Step 9. Take the final $U$ with $L_R$ embedded as the desired secret-fragment-visible mosaic image for the input secret image $S$ and exit.

### B. Experimental Results of Mosaic Image Creation

Two examples of secret-fragment-visible mosaic images generated by Algorithm 1 are shown in Figs. 6 and 7. In either figure, the secret image of (a) was embedded into the target image of (b) to yield the mosaic image of (c). The database used in the algorithm includes 841 candidate images.

We have also conducted some experiments on varying the scale of the secret image to see the effect on the visual quality of the yielded mosaic image. Two results of such experiments for the secret image of Fig. 7(a) are shown in Fig. 7(d) and (e).

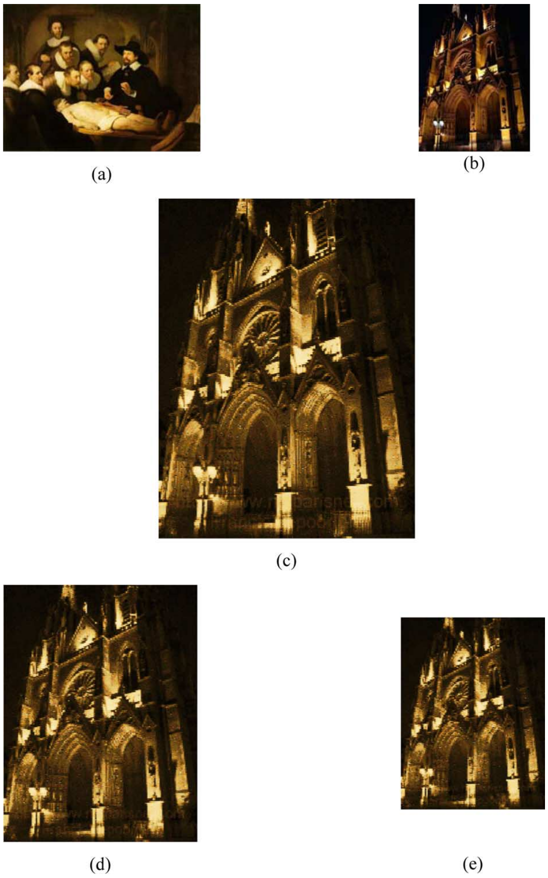


Fig. 7. Another mosaic image creation result. (a) Secret image. (b) Target image. (c) Mosaic image created from a 1024 × 768 secret image of (a) with $\mathrm{RMSE} = 32.78$. (d) Mosaic image created from a 768 × 576 secret image of (a) with $\mathrm{RMSE} = 33.82$. (e) Mosaic image created from a 576 × 432 secret image of (a) with $\mathrm{RMSE} = 33.96$.

Specifically, the sizes of the secret images used to yield Fig. 7(c)–(e) are $1024 \times 768$, $768 \times 576$, and $576 \times 432$, respectively. The size of the tile images is kept unchanged to be $4 \times 4$. It is observed from the figures that the qualities of the resulting mosaic images are visually equally good. This fact is also confirmed by the roughly equal root-mean-square-error (RMSE) values (shown in the figure captions) of the three yielded mosaic images with respect to respective-sized secret images of Fig. 7(a).

## V. Secret Image Recovery

Secret image recovery is basically a reverse of the mosaic image creation process. The detail is described as an algorithm in the following, followed by an experimental result.

### A. Secret Image Recovery Algorithm

**Algorithm 2**: ***secret image recovery***.

**Input**: a secret-fragment-visible mosaic image $U$; and the random number generator $g$ and the secret key $K$ used by Algorithm 1.

**Output**: the secret image $S$ from which $U$ was created.

**Steps**.

***Stage 1—retrieving tile-image fitting information***.

Step 1. Retrieve the width $W_S$ and height $H_S$ of $S$ as well as the size $Z_t$ of the tile images from the first ten pixels in the first block of image $U$ in a raster-scan order using a reverse version of the lossless LSB replacement scheme proposed in **[18]**.

Step 2. Compute the length $N_R$ of the binary secret recovery sequence $L_R$ to be extracted using the data of $W_S$, $H_S$, and $Z_t$ according to **(6)–(8)**.

Step 3. Repetitively select randomly an *unselected block* $s$ other than the first block from $U$ using the random number generator $g$ with the secret key $K$ as the seed, extract $N_T$ bits from all the $Z_t$ pixels of $s$ using a reverse version of the lossless LSB replacement scheme proposed in **[18]**, and concatenate them sequentially, until all the $N_R$ bits of $L_R$ are extracted, where $N_T$ is computed by **(9)**.

Step 4. Transform every $N_X$ bits of $L_R$ into an integer which specifies the index of a tile image in the original secret image $S$ (to be composed), resulting in the secret recovery sequence $L_R = i_1 i_2 \ldots i_N$ where $N$ is as specified by **(6)**.

***Stage 2—reconstructing the secret image***.

Step 5. Construct the mappings of the indices of the tile images of the original secret image $S$ (to be composed next) to those of the corresponding target blocks of $U$ as $i_1 \rightarrow 1, i_2 \rightarrow 2, \ldots, i_N \rightarrow N$.

Step 6. Compose the tile images of the desired secret image $S$ in a raster-scan order according to the $N$ mappings by taking block 1 of $U$ to be tile image $i_1$ in $S$, block 2 of $U$ to be tile image $i_2$ in $S$, and so on, until all $N$ blocks of $U$ are fitted into $S$.

### B. An Experimental Result

An example of the experimental results of applying Algorithm 2 is shown in Fig. 8, where Fig. 8(c) shows the created mosaic image using Fig. 8(a) and (b) as the input secret image and target image, respectively; Fig. 8(d) shows the extracted secret image from Fig. 8(c) using Algorithm 2 with a correct secret key; and Fig. 8(e) shows the extracted one with a wrong key, which is a noise image. Note that Fig. 8(d) is an exact copy of the original secret image, and this point can be figured out from the details of Algorithms 1 and 2. In particular, the lossless LSB replacement scheme of [18] is used for parameter embedding and the tile images are fitted into the target blocks with no change. Therefore, we may say that the proposed method is a lossless secret image hiding method.

## VI. Extension to Creation of Grayscale Mosaic Images

### A. Grayscale Features of Blocks and Mosaic Image Creation and Recovery

It is often encountered that the secret image is a grayscale one. This could happen when the image is obtained, through various ways like scanning, from paper documents mainly with

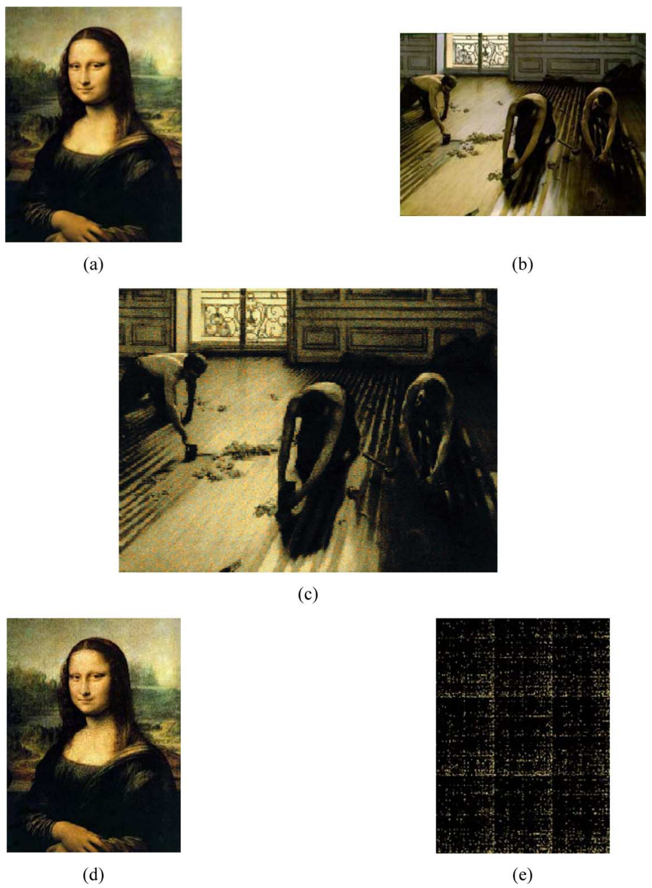


Fig. 8. Example of secret image recovery results. (a) Secret image. (b) Target image. (c) Mosaic image created from a $1024 \times 768$ secret image of (a) with $\mathrm{RMSE} = 30.86$. (d) Extracted secret image using a correct key. (e) Extracted secret image using a wrong key.

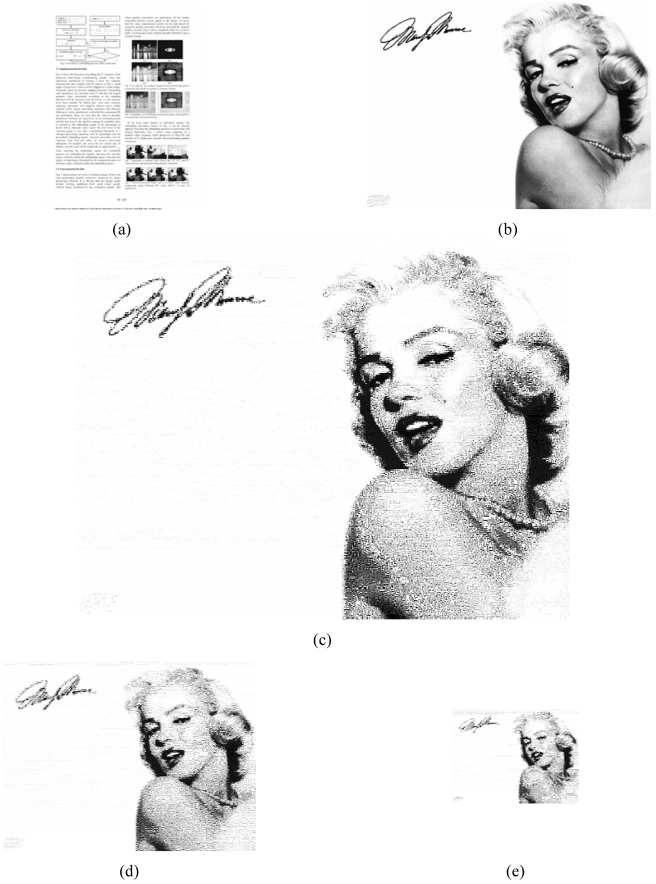


Fig. 9. Experimental result of grayscale mosaic image creation. (a) Secret image. (b) Target image. (c) Mosaic image created from a $1024 \times 768$ secret image of (a) with $\mathrm{RMSE} = 30.86$. (d) Mosaic image created from a $768 \times 576$ secret image of (a) with $\mathrm{RMSE} = 31.04$. (e) Mosaic image created from a $576 \times 432$ secret image of (a) with $\mathrm{RMSE} = 30.7$.

text contents. In this case, the selected target image obviously should be of the same type, namely, a grayscale image; and the generated mosaic image is also a grayscale one. Most parts of the previously-presented algorithms are applicable to the case here after some minor modifications, as discussed next.

First, the color image database should be converted into a grayscale version. For this, the color values $(r, g, b)$ of every pixel in each image in the database is transformed in this study into a 1-D grayscale value $y$ by the equation $y = 0.177 \times r + 0.813 \times g + 0.011 \times b$ where the weights for $r$, $g$, and $b$ are taken to be the coefficients of the luminance (the $Y$ component) used in the transformation from the RGB model to the YUV one. The reason for adopting such weights instead of the conventional value of 1/3 for each color channel is based again on the previously-mentioned human eye's higher sensitivity to the green color.

Then, the average of the grayscale values of all the pixels in each image block $c$ is computed as a feature, called the *$y$-feature*, of $c$ and denoted as $y_c$. This feature is used further as a measure like that of (3) described previously in the database construction process to compose the *$y$-feature histogram* $H_D$ of each candidate target image $D$ in the database. A similar grayscale histogram is also constructed for the input grayscale secret image $S$. The two histograms then are used to define an *image similarity* value, like that described by (4), between $S$ and $D$ in the following form:

$$m(S, D) = \frac{1}{\sum_{y=0}^{255} |H_S(y) - H_D(y)|}. \tag{4'}$$

Finally, this measure is used for selecting the most similar grayscale target image $D_o$ for each input grayscale image $S$. Furthermore, the $y$-feature is also used to define a new *block similarity value* between a tile image $s$ with $y$-feature value $y_s$ and a target block $d$ with $y$-feature value $y_d$ as

$$m(s, d) = \frac{1}{|y_s - y_d|} \tag{5'}$$

for use in Algorithm 1.

Now, the selected target image $D_o$ together with the secret image $S$ may be used as input to Algorithm 1 to generate a grayscale secret-fragment-visible mosaic image $U$ using the similarity measures defined by Eqs. $(4')$ and $(5')$. As to the process for recovering the secret image from a grayscale mosaic image, Algorithm 2 basically is applicable using the new similarity measures.

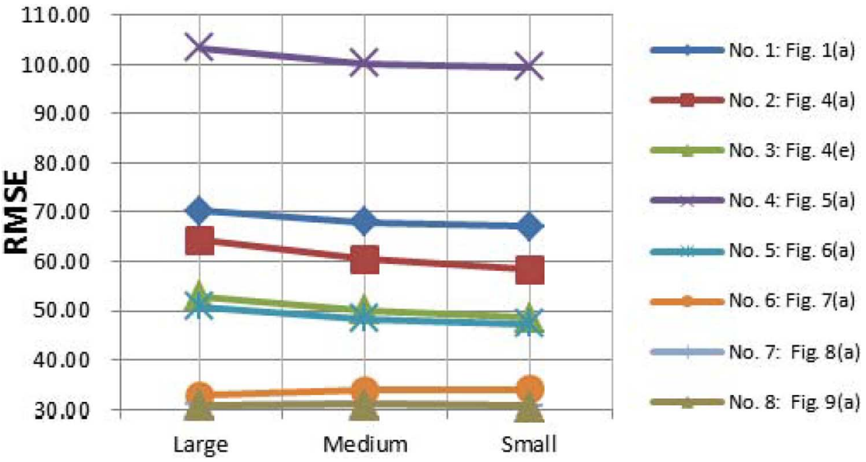


Fig. 10. Plot of RMSE values of created mosaic images with respect to three image sizes (large, medium, and small) with all secret images shown previously (indicated by the numbers shown to right of above plot).

### *B. Experimental Results*

An example of our experimental results of successful mosaic image creation with grayscale secret images as input is shown in Fig. 9, which indicates that the proposed method with the above-described alternative similarity measures is feasible for creating grayscale secret-fragment-visible mosaic images from text-type grayscale document images, proving again the usefulness of the proposed method for covert communication or secure keeping of grayscale secret images. Furthermore, similarly to what we did for Fig. 7, we have conducted experiments on varying the scale of the grayscale secret image to see the effect on the visual quality of the yielded grayscale mosaic images. Two experimental results for the secret image of Fig. 9(a) are shown in Fig. 9(d) and (e). The sizes of the secret images used to yield Fig. 9(c)–(e) again are 1024 × 768, 768 × 576, and 576 × 432, respectively. Once again, the created mosaic image quality is not seen to degrade with the decrease of the secret image size, as also proved by the RMSE values included in the captions of the figures. Actually, this trend of irrelevancy of the quality of the created mosaic image with respect to the image size is observed in the results of all the images tested in this study. As a visual proof, we draw in Fig. 10 a plot of this trend for the three-sized (large, medium, and small) mosaic image creation results of all the eight secret images mentioned previously [i.e., of Figs. 1(a), 4(a) and (e), and 5(a)–9(a)].

## VII. Security Consideration and Enhancement

Each color pixel has three channels for embedding bits, and the lossless LSB replacement scheme [18] we adopted needs two pixels to embed a bit by using an identical color channel. So, the number $N_Q$ of pixels required to embed the $N_R$ bits of the secret recovery sequence $L_R$ is equal to

$$N_Q = \left\lceil \frac{2 \times N_R}{3} \right\rceil \tag{10}$$

and the number $N_E$ of tile images required for embedding $L_R$ is

$$N_E = \left\lceil \frac{N_Q}{Z_t} \right\rceil = \left\lceil \frac{(2 \times N_R)}{(3 \times Z_t)} \right\rceil \tag{11}$$

because each tile image has $Z_t$ pixels. And in the mosaic image creation process, we use a secret key to select *randomly* $N_E$ tile images fitted in the mosaic image for embedding the $N_R$ bits of $L_R$. Therefore, if the number of tile images in a secret image is $N$, then the number of possible ways to choose $N_E$ tile images randomly, as conducted in Step 8 of Algorithm 1, is the number of permutations, $P(N, N_E)$, which equals $N!/N_E!$; and the probability for a hacker to extract $L_R$ correctly by guessing and recover accordingly the secret image successfully is just $p = 1/P(N, N_E) = N_E!/N!$. In this study, we divide a secret image into numerous 4 × 4 tile images to compose a mosaic image and the typical value of $N$ is $(1024 \times 768)/(4 \times 4) = 49{,}152$. Therefore, the value of $N_E$ may be computed to be equal to 32,768 using previously-derived equalities of (6)–(11), and so the probability $p$ for a hacker to recover the entire secret image correctly without the secret key is $N_E!/N! = 1/[N \times (N-1) \times (N-2) \times \cdots \times (N - N_E + 1)] = 1/(49152 \times 49151 \times \cdots \times 6385)$ which is very close to zero!

However, a hacker without the secret key but knowing the proposed method might still have a chance with probability $p = 1/N$ to retrieve correctly the mapping of a tile image to a target block in the step of extracting the secret recovery sequence $L_R$ (Step 3 of Algorithm 2) because $L_R$ is known to be composed sequentially of the $N$ indices of the tile images with each index having a *fixed* length of $N_X$ bits (see (7)). This means that, after a sufficiently large number of trials, it is possible for the hacker to see part of the secret image consisting of a few blocks distributed at correct positions! To prevent this to happen, it is proposed to use *an additional secret key* to generate random numbers, each with $N_X$ bits, and to randomize the bits of each index $i$ by exclusive-ORing them bit by bit with those of a generated random number before the index $i$ is included into $L_R$. In this way, even if a hacker's random trial leads to correct extraction of a tile-image index in $L_R$, the extracted index will be still in the form of a random-bit pattern; and without the help of the second key, the original bit pattern cannot be recovered. If the hacker still tries to guess the correct index value, then because in this study $N_X$ is approximately equal to $\lfloor \log_2 N \rfloor + 1 \approx \lfloor \log_2 49152 \rfloor + 1 \approx 16$, the probability for the binary index to be guessed correctly is roughly $1/2^{16}$ which is also small enough.

## VIII. Conclusion and Suggestions for Future Studies

A new type of digital art, called secret-fragment-visible mosaic image, has been proposed, which can be used for secure keeping or covert communication of secret images. This type of mosaic image is composed of small fragments of an input secret image; and though all the fragments of the secret image can be seen clearly, they are so tiny in size and so random in position that people cannot figure out what the source secret image looks like. Specifically, a new colorscale and a new grayscale have been proposed to define a new $h$-feature and a new $y$-feature, which then are used to define appropriate similarity measures for images and blocks for generating secret-fragment-visible mosaic images more effectively. A greedy search algorithm has also been proposed for searching the tile images in a secret image for the most similar ones to fit the target blocks of a selected target image more efficiently. Tile-image fitting information for secret image recovery is embedded into randomly-selected tile images in the resulting mosaic image controlled by a secret key. An additional security enhancement measure was also proposed. The method has been extended to generate grayscale mosaic images with grayscale secret images as input. Good experimental results have been shown to prove the feasibility of the proposed method.

Good mosaic image creation results are guaranteed only when the database is large in size so that the selected target image can be sufficiently similar to the input secret image. Future works may be directed to allowing users to select target images from a smaller-sized database or even freely without using a database, as well as to developing more information hiding applications using the proposed secret-fragment-visible mosaic images.

## References

[1] R. Silver and M. Hawley, *Photomosaics*. New York: Henry Holt, 1997.

[2] S. Battiato, G. Di Blasi, G. M. Farinella, and G. Gallo, “Digital mosaic framework: An overview,” *Eurograph.—Comput. Graph. Forum*, vol. 26, no. 4, pp. 794–812, Dec. 2007.

[3] P. Haeberli, “Paint by numbers: Abstract image representations,” in *Proc. SIGGRAPH*, Dallas, TX, 1990, pp. 207–214.

[4] A. Hausner, “Simulating decorative mosaics,” in *Proc. SIGGRAPH*, Los Angeles, CA, Aug. 2001, pp. 573–580.

[5] Y. Dobashi, T. Haga, H. Johan, and T. Nishita, “A method for creating mosaic image using voronoi diagrams,” in *Proc. Eurographics*, Saarbrucken, Germany, Sep. 2002, pp. 341–348.

[6] G. Elber and G. Wolberg, “Rendering traditional mosaics,” *Vis. Comput.*, vol. 19, pp. 67–78, 2003.

[7] J. Kim and F. Pellacini, “Jigsaw image mosaics,” in *Proc. SIGGRAPH*, San Antonio, TX, Jul. 2002, pp. 657–664.

[8] G. Di Blasi, G. Gallo, and M. Petralia, “Puzzle image mosaic,” in *Proc. IASTED/VIIP*, Benidorm, Spain, Sep. 2005, pp. 33–37.

[9] G. Di Blasi and G. Gallo, “Artificial mosaics,” *Vis. Comput.*, vol. 21, pp. 373–383, 2005.

[10] S. Battiato, C. Guarnera, G. Di Blasi, G. Gallo, and G. Puglisi, M. Bubak, Ed. *et al.*, “A novel artificial mosaic generation technique driven by local gradient analysis,” in *Proc. ICCS*, Crakov, Poland, Jun. 2008, vol. 5102, pp. 76–85.

[11] S. Battiato, G. Di Blasi, G. Gallo, G. C. Guarnera, and G. Puglisi, “Artificial mosaic by gradient vector flow,” in *Proc. Eurographics*, Creete, Greece, Apr. 2008, pp. 53–56.

[12] H. Narasimhan and S. Satheesh, “A randomized iterative improvement algorithm for photomosaic generation,” in *Proc. NaBIC*, Coimbatore, India, Dec. 2009, pp. 777–781.

[13] Y.-S. Choi, B.-k. Koo, and B.-R. Moon, “Optimization of an image set by genetic feature selection for real-time photomosaics,” in *Proc. GECCO*, Portland, OR, Jul. 2010, pp. 1309–1310.

[14] S. Battiato and G. Puglisi, “3D ancient mosaics,” in *Proc. ACM Int. Conf. Multimedia*, Florence, Italy, Oct. 2010, pp. 1751–1753.

[15] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, “Content-based multimedia information retrieval: State of the art and challenges,” *ACM Trans. Multimedia Comput., Commun., Appl.*, pp. 1–19, Feb. 2006.

[16] J. R. Smith and S. F. Chang, “Tools and techniques for color image retrieval,” *ProcIS & T/SPIE*, vol. 2670, pp. 2–7, Feb. 1995.

[17] M. K. Agoston, *Computer Graphics and Geometric Modeling: Implementation and Algorithms*. New York: Springer-Verlag, 2004, pp. 61–61.

[18] D. Coltuc and J. M. Chassery, “Very fast watermarking by reversible contrast mapping,” *IEEE Signal Process. Lett.*, vol. 14, no. 4, pp. 255–258, Apr. 2007.

[19] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Upper Saddle River, NJ: Prentice-Hall, 1998.

**I-Jen Lai** received the B.S. degree in computer engineering from National Central University, Taiwan, in 2008 and the M.S. degree in computer science from National Chiao Tung University, Taiwan in 2010.

She is currently a Research Associate with the Information Service Center, National Chiao Tung University. She was a Research Assistant at the Computer Vision Laboratory, Department of Computer Science, National Chiao Tung University from August 2008 to July 2010. Her current research interests include information hiding, image processing, and computer art.

**Wen-Hsiang Tsai** (S’78–M’79–SM’91) received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, in 1973, the M.S. degree in electrical engineering from Brown University, Providence, RI, in 1977, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, 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 Department 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. His current research interests include computer vision, information security, video surveillance, and autonomous vehicle applications.

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 144 journal papers and 227 conference papers 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. He is a Life Member of the Chinese Pattern Recognition and Image Processing Society, Taiwan.