IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

Secret Image Hiding In Targert Image Of Mosaic Form

S. Jayaprakash[1], G.Sujitha[2], G.Manju[3]

[1]Assistant Professor, Idhaya Engineering College for Women, Tamilnadu.
[2,3]Final year Student/CSE, Idhaya Engineering College for Women, Tamilnadu.

***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.**

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.

A new type of art image, called *secret-fragment-visible mosaic image*, which contains small fragments of a given source image 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 pieces are all *visible* to the observer. And this is the reason why the resulting mosaic image is named *secret-fragment-visible*. Because of this characteristic of the new mosaic image, it may be used 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. 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 find image.

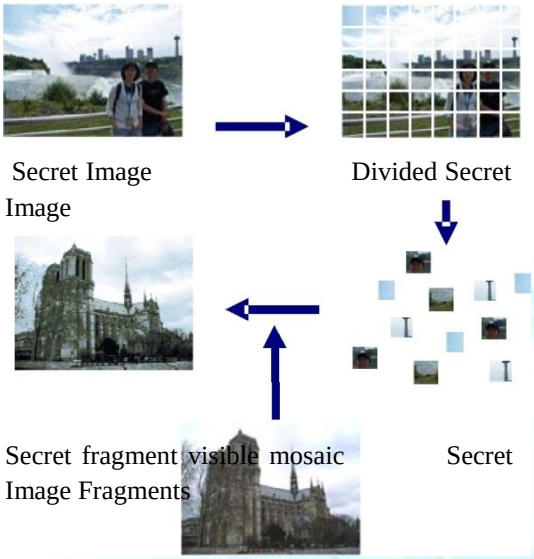


Figure 2.Illustration of creation of secret-fragment-visible mosaic image.

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.

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.

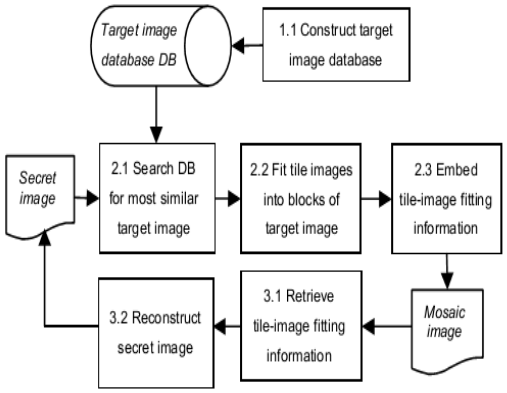


Figure 3. Processes for secret-fragment-visible mosaic image creation and secret image recovery.

Database Construction

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. 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.

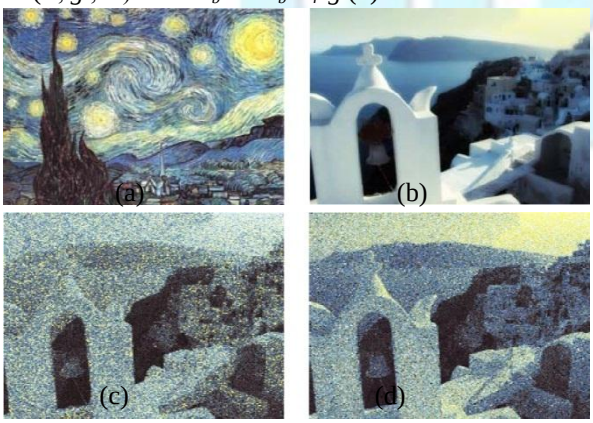
The technique re-quantizes first the color values (r, g, b) into fewer levels, say N_r, Ng, and N_b ones, respectively, resulting in the new color values (r', g', b'). Then, it transforms the three new values (r', g', b') into a single one by:

$$f(r', g', b') = r' + N_{r*}g' + N_{r*}N_{g*}b'. \quad (1)$$

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.

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

$$h(r', g', b') = b' + N_{b*}r' + N_{b*}N_{r*}g' \quad (2)$$



where, differently from the case in Eq. (1), the numbers of levels, N_r, Ng, and N_b, are all set to be 8, and the largest weight, namely, the value $N_{b*}N_r$, 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 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, Ng, and N_b set to be 8 in Eq. (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 Eq. (2) defines a 1-D *h-color scale*. The mosaic image created by the proposed method using a similarity measure based on this new color scale.

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, Ng, and N_b levels, respectively, and transform accordingly (r_c, g_c, b_c) into (r_c', g_c', b_c') in term of the three new color levels;
3. compute the h-feature value h_c for c by Eq. (2) above, resulting in the following equation:

$$h_c(r_c', g_c', b_c') = b_c' + N_{b*}r_c' + N_{b*}N_{r*}g_c'. \quad (3)$$

With Nr, Ng, 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 pre-selected 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 pre-selected size Z_t, compute the h-feature value defined by Eq. (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 pre-selected 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 pre-selected size Z_t, compute the h-feature values of all the resulting blocks by Eq. (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:

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

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

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:

$$m(s, d) = 1/|h_s - h_d|.$$

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 global way.

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.

In more detail, to get the mappings, we start from the top-leftmost 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, 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 an well-ordered sequence of 1, 2, 3,, 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 to embed a bit, the number N_T of bits that can be embedded into a tile image is just 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-II 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 pre-selected size Z_c; a pre-selected 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 Eq. (6).

Step 2. Select from DB the target image D_o that is the most similar to S in the sense of Eq. (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, … in S corresponding to the N target blocks d_1, d_2, d_3, … 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.

Step 5. Fit the tile images s_i, s_j, s_k, … into the corresponding target blocks d_1, d_2, d_3, …, 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.

Step 7.Transform L_R into a binary string with its length N_R

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, until all the N_R bits in L_R are exhausted

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.

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 the description of 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 Uw as 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.

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

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, and concatenate them sequentially, until all the N_R bits of L_R are extracted

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 $L_R = i_1 i_2 \ldots i_N$

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.

VI. EXTENSION TO CREATION OF GRAYSCALE MOSAIC IMAGES

A. Grayscale Features of Blocks and Mosaic Image Creation

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 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 it 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 $y = 0.177r + 0.813g + 0.011b$ 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 Eq. (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 e. 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 Eq. (4), between S and D in the following form:

$$m(S, D) = 1/\Sigma|H_S(y) - H_D(y)|.$$

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

m(s, d) = 1/|ys – y for use in algorithm

Now, the selected target image Do together with the secret image may be used as input to Algorithm 1 to generate a grayscale secret-fragment-visible mosaic image U using the similarity measures. As to the process for recovering the secret image from a grayscale

mosaic image, Algorithm 2 basically is applicable using the new similarity measures.

VII. SECURITY CONSIDERATION AND ENHANCEMENT

Each color pixel has three channels for embedding bits, and the lossless LSB replacement scheme 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

$NQ = \lceil 2 \times NR/3 \rceil$

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

$NE = \lceil NQ/Zt \rceil = \lceil (2 \times NR)/(3 \times Zt) \rceil$

because each tile image has Z t pixels. And in the mosaic image creation process, we use a secret key to select randomly NE tile images fitted in the mosaic image for embedding the NR bits of

LR. Therefore, if the number of tile images in a secret image is N, then the number of possible ways to choose NE tile images randomly, as conducted in Step 8 of Algorithm 1, is the number

of permutations, P(N, NE), which equals N!/NE!; and the probability for a hacker to extract LR correctly by guessing and recover accordingly the secret image successfully is just p= 1/P(N, NE) = NE!/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 Nis (1024×768)/(4×4) = 49,152. Therefore, the value of NE may be computed to be equal to 32,768 using previously-derived equalities of (6) through (11), and so the probability p for a hacker to recover the entire secret image correctly without the secret key is

$NE!/N! = 1/[N \times (N-1) \times (N-2) \times \ldots \times (N - NE + 1)] = 1/(49152 \times 49151 \times \ldots \times 6385)$

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 LR(Step 3 of Algorithm 2) because LR is known to be composed sequentially of the N indices of the tile images with each index having a fixed length of NX bits. 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 NX bits, and to randomize the bits of each index I by exclusive-O Ring them bit by bit with those of a generated random number before the index iis included into LR. In this way, even if a hacker's random trial leads to correct extraction of a tile-image index in LR, 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 $\lfloor \log 2N \rfloor + 1 \approx \lfloor \log 49152 \rfloor + 1 \approx 16$, the probability for the binary index to be guessed correctly is roughly 1/216which is also small enough.

VII. CONCLUSIONS 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 color scale 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 result 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, NY, USA: Henry Holt, 1997.

[2] S. Battiato, G. Di Blasi, G.M. Farinella and G. Gallo, "Digital mosaic framework: an overview," *Euro graphics – Computer Graphic Forum*, vol. 26, no. 4, pp. 794-812, Dec. 2007.

[3] P. Haeberli, "Paint by numbers: abstract image representations," *Proc. of SIGGRAPH 99*, pp.207-214, Dallas, USA, 1990.

[4] A. Hausner, "Simulating decorative mosaics," *Proc. of 2001 Int'l Conf. on Computer Graphics & Interactive Techniques (SIGGRAPH 01)*, Los Angeles, USA, August 2001, pp. 573-580.

[5] Y. Dobashi, T. Haga, H. Johan and T. Nishita, "A method for creating mosaic image using voronoi diagrams," *Proc. of 2002European Association for Computer Graphics (Eurographics 02)*,Saarbrucken, Germany, Sept. 2002, pp. 341-348.