Secret-fragment-visible Mosaic — a New Image Art and Its Application to Information Hiding

I-Jen Lai (賴怡臻)
Institute of Computer Science and Engineering
National Chiao Tung University, Hsinchu, Taiwan
Email: nekolai.cs97g@g2.nctu.edu.tw

Wen-Hsiang Tsai (蔡文祥)
Dept. of Computer Science
National Chiao Tung University, Hsinchu, Taiwan
Email: whtsai@cs.nctu.edu.tw

***Abstract*—A new type of art image called secret-fragment-visible mosaic image is created, which is composed of rectangular-shaped fragments yielded by division of a secret image. To create this kind of mosaic image, the 3D RGB color space is transformed into a 1-dimensional *h*-colorscale based on which a new image similarity measure is proposed; and the most similar candidate image from an image database is selected accordingly as a target image. Then, a greedy algorithm is adopted to fit every tile image in the secret image into a properly-selected block in the target image, resulting in an effect of embedding the secret image fragmentally and visibly in the composed mosaic image. In addition to this type of secret image hiding, secret message bits may be embedded as well for the purpose of covert communication. Based on the fact that tile images in an identical bin of the histogram of the created mosaic image have similar colors, all the tile images in each histogram bin are reordered pairwisely and their relative positions are switched accordingly, to embed secret message bits without creating noticeable changes in the resulting mosaic image. The embedded message is protected by a secret key, and may be extracted from the stego-image using the key. Additional security measures are also discussed. Experimental results show the feasibility of the proposed methods.**

Keywords: secret-fragment-visible mosaic image, covert communication, data hiding.

I. INTRODUCTION

Mosaics are artworks created from composing small pieces of materials, such as stone, glass, tile, etc. Nowadays, they are used popularly for decorating houses and other constructions. Creation of mosaic images by computers is a new research topic in recent years. Traditional mosaic images are obtained by arranging a large number of small images, called *tile images*, in a certain manner so that each tile image represents a small piece of a source image, named *target image*. Consequently, while we see a mosaic image from a distance, as a whole it will look like its source image — an effect of a human vision property. Many methods have been proposed to create different types of mosaic images [1-8].

Haeberli [1] proposed a method for mosaic image creation using voronoi diagrams by placing the sites of blocks randomly and filling colors into the blocks based on the content of the original image. Hausner [2] created tile mosaic images by using centroidal voronoi diagrams. Dobashi et al. [3] improved the voronoi diagram to allow a user to add various effects to the mosaic image, such as simulation of stained glasses. Kim and Pellacini [4] generated jigsaw image mosaic composed of many arbitrary shapes of tiles selected from a database. Extending the concept of [4], Blasi et al. [5] presented a new mosaic image called puzzle image mosaic. Lin and Tsai [6] embedded secret data in image mosaics by adjusting regions of boundaries and altering pixels' color values. Wang and Tsai [7] hid data into image mosaics by utilizing overlapping spaces of component images. Hung and Tsai [8] embedded data into stained-glass-like mosaic images by modifying the tree structure used in the creation process. Hsu and Tsai [9] presented a new type of art image, circular-dotted image, and used the characteristics of its creation processes to hide secret messages in the generated art image. Chang and Tsai [10] proposed a new type of art image, called tetromino-based mosaic, which is composed of tetrominoes appearing in a video game. Data hiding is made possible by distinct combinations and color shifting of the tetromino elements.

A new type of art image, called *secret-fragment-visible mosaic image*, which contains small fragments of a secret source image is proposed in this study. Observing such a type of mosaic image, people can see all of the fragments of the secret image, but the fragments are so tiny in size and so random in position that people cannot figure out what the source image look like, unless they have some way to rearrange the pieces back into their original positions, using a secret key from the image owner. Therefore, the source image may be said to be *secretly* embedded in the resulting mosaic image, though the fragment pieces are all *visible* to an observer of the image. And this is just why we name the resulting image as a *secret-fragment-visible mosaic image*.

In the remainder of this paper, the proposed mosaic image creation process will be described in Section II, a covert communication method via secret-fragment-visible mosaic images will be proposed in Section III, and some experimental results will be presented in Section IV, followed by conclusions in Section V.

II. PROPOSED MOSAIC IMAGE CREATION PROCESS

The proposed mosaic image creation process is composed of two major stages. The first is the construction of a database which can be used later to select similar target images for given secret images. The quality of a constructed secret-fragment-visible mosaic image is related to the similarity between the secret image and the target image; the selected target image should be as similar to the secret image as possible. An appropriate similarity measure for this purpose is proposed in this study and described later. The other stage is the creation of a desired mosaic image using the secret image and the target image as input. In this stage, the secret image is divided into fragment pieces as tile images, which then are used to create the mosaic image. The number of tile images is *limited* by the size of the secret image and that of the tile images. Note that this is not the case in traditional mosaic image creation where available tile images for use to fit into the target image are unlimited in number. In order to solve this problem of fitting a limited number of tile images into a target image, a greedy algorithm is proposed, which is described later as well.

2.1 Database Construction

The database plays an important role in the secret-fragment-visible mosaic image creation process. If a target image is dissimilar to a secret image, the created image will be distinct from the target one. In order to generate a good result, the database so should be as large as possible.

Searching a database for a target image with the highest similarity to the secret image is a problem of content-based image retrieval. A technique to solve this problem is to base the similarity on 1-D color histogram transformation [12] of the color distribution of the image. The transformation maps the three color channel values into a single value. Specifically, each color channel is re-quantized first into fewer levels, yielding a new image I' with a lower resolution in color specified by (r', g', b'). Let N_r, N_g, and N_b denote the numbers of levels of the new color values r', g', and b', respectively. Then, for each pixel P' in I' with new colors (r', g', b'), the following 1-D function value f is computed:

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

However, according to our experimental experience using this 1-D color function f, it is found inappropriate for our study here where the human's visual feeling of image similarity must be emphasized, as shown by Fig. 1(a). Therefore, we propose a new function h as follows:

$$h(r', g', b') = b' + N_b \times r' + N_b \times N_r \times g' \qquad (2)$$

where the numbers of levels, N_r, N_g, and N_b, are *all set to be 8*. Differently from the case in (1), we set in (2) the largest weight $N_b \times N_r$ to the green channel value g' and the smallest weight 1 to the blue channel value b'. The reason is that the eyes of human beings are the most sensitive to the green color, and the least sensitive to the blue one. In addition, with all of N_r, N_g, and N_b set to 8 in (2), an advantage of speeding up the process of mosaic image creation can be obtained according to our experiments. Subsequently, we will say that the new color feature function h we propose above defines a 1-D *h-colorscale*. The resulting image created by our method is given in Fig. 1(b), which contrastively has less noise when compared with Fig. 1(a).


(a) (b)

Figure 1. Effects of mosaic image creation using different color similarity measures (a) Image created with similarity measure of [12]. (b) Image created with proposed similarity measure.

Furthermore, to compute the similarity measure between a *tile image* in the secret image and a *target block* in an image in a database for use in tile-image fitting in generating a mosaic image, we propose a new feature, called *h-feature*, for each block image C (either a tile image or a target block), denoted as h_C, which is computed by the following steps:

1. compute the average of the color values of all the pixels in C as (R_C, G_C, B_C);
2. re-quantize (R_C, G_C, B_C) into (r_C', g_C', b_C') using the new N_r, N_g, and N_b color levels; and
3. calculate the h-feature h_C for C by Eq. (2) above, resulting in the following equation:

$$h_C(r_C', g_C', b_C') = b_C' + N_b \times r_C' + N_b \times N_r \times g_C'. \qquad (3)$$

With Nr, N_g, and N_b all set equal to 8, the range of the computed values of the h-feature f_C above may be figured out to be from 0 to 584. The proposed algorithm for constructing a database of *candidate images* for use in generating secret-fragment-visible mosaic images is described in the following.

```
Algorithm 1: construction of candidate image database.
Input: a set $S$ of images, a pre-selected tile image size $Z_t$,
       and a pre-selected candidate image size $Z_c$.
Output: a database $DB$ of candidate images with size $Z_c$ and
        their corresponding h-colorscale histograms.
Steps:
Step 1. For each input image $I$, perform the following steps.
    1.1 Resize and crop $I$ to yield an image $D$ of size $Z_c$.
    1.2 Divide $D$ into blocks of size $Z_t$.
    1.3 For each block $C$ of $D$, calculate and round off the
        h-feature value $h_C$ described by Eq. (3).
    1.4 Generate a histogram $H$ of the h-feature values of all
        the blocks in $D$.
    1.5 Save $H$ with $D$ into the desired database $DB$.
Step 2. If the input images are not exhausted, go to Step 1;
        otherwise, exit.
```

2.2 Similarity Measure Computation

Before generating a mosaic image, we have to choose as the target image the most similar candidate image from the database based on the given secret image content. For this, we define a *difference measure* e between the 1-D histogram

H_S of the secret image S and that of a candidate image D in the database in the following way:

$$e=\sum_{m=0}^{584}\left|H_S(m)-H_D(m)\right| \tag{4}$$

where m stands for a h-feature value. The smaller the value e is, the more similar the candidate image D is to the secret image S. After calculating the errors of all the images in the database, we can select the one with the smallest error as the desired target image for use in mosaic image generation. The detail of selecting the most similar candidate image from a database is given as follows.

Algorithm 2: ***selection of the most similar candidate image as a target image.***

```
Input: a secret image $S$, a database $DB$ of candidate images,
    and the sizes $Z_t$ and $Z_c$ mentioned in Algorithm 1.
Output: the target image $T$ in $DB$ which is the most similar
    to $S$.
Steps:
Step 1. Resize $S$ to yield an image $S'$ of size $Z_c$ to become of
    the same size as the candidate images in $DB$.
Step 2. Divide $S'$ into blocks of size $Z_t$, and perform the
    following steps.
    2.1 For each block $C$ of $S'$, calculate its $h$-feature value
        $h_C$ by Eq. (3) and round off the result.
    2.2 Generate a 1-D $h$-colorscale histogram $H_{S'}$ for $S'$
        from the $h$-feature values of all the blocks in $S'$.
Step 3. For each candidate image $D$ with 1-D $h$-colorscale
    histogram $H_D$ in $DB$, perform the following steps.
    3.1 Compute the difference measure $e$ between $H_{S'}$ and
        $H_D$ according to Eq. (4) described above.
    3.2 Record the value $e$.
Step 4. If the images in $DB$ are not exhausted, go to Step 3;
    otherwise, continue.
Step 5. Select the image in $DB$ which has the minimum
    difference measure $e$ and take it as the desired target
    image $T$.
```

2.3 Algorithm for Secret-fragment-visible Mosaic Image Creation

Before presenting the algorithm for creating the proposed mosaic images, we discuss some problems which are encountered in the creation process and present the solutions we propose to solve them.

A. Problem of fitting tile images optimally and proposed solution

The first problem faced in the creation process is how to find an optimal solution for fitting a tile image of the secret image into an appropriate target block in a target image selected by Algorithm 2. For this, it *seems* that we can reduce it to a *single-source shortest path* problem. The shortest path problem is one of finding a path in a graph with the smallest sum of between-vertex edge weights. The state of fitting a tile image may be represented by a vertex of the graph. And the action of selecting the *most similar* tile image for each target block may be represented by an edge of the graph with its label taken to be that of the tile image and its weight taken to be the average Euclidean distance between the pixels' colors of the selected tile image and those of the target block. Accordingly, we can build a tree structure as the graph for this problem, as shown by Fig. 2.

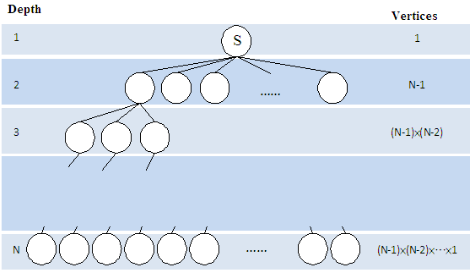

Figure 2. A tree structure of fitting tile images to target blocks.

In order to find the optimal solution, we may utilize the Dijkstra algorithm whose the running time for getting an optimal answer is $O(|V|^2)$, where V denotes for the number of vertices in the tree. Unfortunately, according to Fig. 2 the number of vertices in this problem is $\sum_{n=1}^{N-1}[(N-1)!/n!]$ where N is the number of target blocks which is larger than 40,000 for images used in this study, and so the computation time for getting an optimal solution for such a large N is obviously too high to be practical! This means that we have to find other feasible solutions to solve this problem.

The solution we propose is to use *a greedy algorithm*. We calculate the average Euclidean distance between the pixels' colors of a tile image T and those of a target block B as the similarity measure between T and B; and then use the measure as a *selection function* for the greedy algorithm to select the most similar target block for tile image fitting. However, as shown by the example of Fig. 4(a) which is the result of using such a greedy algorithm to fit the tile images of the secret image, Fig. 3(a), into the target image, Fig. 3(b), the algorithm is found *unsatisfactory*, yielding often a result with the lower part of the target image being filled with some fragment pieces of inappropriate colors. This phenomenon comes from the situation that the number of tile images obtained from the secret image, Fig. 3(a), is *limited* by the secret image's own size, so that the tile images available for choice to fit the target blocks in Fig. 3(b) become less and less near the end of the fitting process. As a result, the similarity differences between the later-fitted tile images and the chosen target blocks become bigger and bigger than the earlier-fitted ones, yielding a poorly-fitted bottom part like that shown in Fig. 4(a).

A solution to this problem found in this study is to use the previously-proposed h-feature to define the selection function for the greedy algorithm. This feature takes the *global* color distribution of an image into consideration, which helps creation of a mosaic image with its content resembling the target image more effectively, as shown by