A Symmetric, Fast Image Encryption and Decryption Scheme via Secret-Fragment-Visible Mosaic Images

Ithepalli Yamini
PG Scholar,
Department of ECE,
S.O.E.T, Sri Padmavathi Mahila Viswa Vidyalayam,
Tirupati.

K.Geethika
Assistant Professor,
Department of ECE,
S.O.E.T, Sri Padmavathi Mahila Viswa Vidyalayam,
Tirupati.

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

As in extinction the same paper will be transfer the image through the video as well as the signal noise ratio of the secret–fragment-visible mosaic image also reduced. Good experimental results show the feasibility of the proposed.

Index Terms: Color transformation, data hiding, image encryption, mosaic image, and secure image transmission.

1. INTRODUCTION
Currently, images from various sources are frequently utilized and transmitted through the internet for various applications, such as online personal photograph albums, confidential enterprise archives, document storage systems, medical imaging systems, and military image databases. These images usually contain private or confidential information so that they should be protected from leakages during transmissions. Recently, many methods have been proposed for securing image transmission, for which two common approaches are image encryption and data hiding. Image encryption is a technique that makes use of the natural property of an image, such as high redundancy and strong spatial correlation, to get an encrypted image based on Shannon’s confusion and diffusion properties. The encrypted image is a noise image so that no one can obtain the secret image from it unless he/she has the correct key.

However, the encrypted image is a meaningless file, which cannot provide additional information before decryption and may arouse an attacker’s attention during transmission due to its randomness in form. An alternative to avoid this problem is data hiding that hides a secret message into a cover image so that no one can realize the existence of the secret data, in which the data type of the secret message investigated in this paper is an image. Existing data hiding methods mainly utilize the techniques of LSB sub situation, histogram shifting, difference expansion, prediction-error expansion, recursive histogram modification, and discrete cosine/wavelet transformations. However, in order to reduce the distortion of the resulting image, an upper bound for the distortion value is usually set on the payload of the cover image. A discussion on this rate distortion issue can be found in . Thus, a main issue of the methods for hiding data in images is the difficulty to embed a large amount of message data.
into a single image. Specifically, if one wants to hide a secret image into a cover image with the same size, the secret image must be highly compressed in advance. For example, for a data hiding method with an embedding rate of 0.5 bits per pixel, a secret image with 8 bits per pixel must be compressed at a rate of at least 93.75% beforehand in order to be hidden into a cover image. But, for many applications, such as keeping or transmitting medical pictures, military images, legal documents, etc., that are valuable with no allowance of serious distortions, such data compression operations are usually impractical.

Moreover, most image compression methods, such as JPEG compression, are not suitable for line drawings and textual graphics, in which sharp contrasts between adjacent pixels are often destructed to become noticeable artifacts. In this paper, a new technique for secure image transmission is proposed, which transforms a secret image into a meaningful mosaic image with the same size and looking like a preselected target image. The transformation process is controlled by a secret key, and only with the key can a person recover the secret image nearly losslessly from the mosaic image. The proposed method is inspired by Lai and Tsai, in which a new type of computer art image, called secret-fragment-visible mosaic image, was proposed. The mosaic image is the result of rearrangement of the fragments of a secret image in disguise of another image called the target image preselected from a database. But an obvious weakness of Lai and Tsai is the requirement of a large image database so that the generated mosaic image can be sufficiently similar to the selected target image. Using their method, the user is not allowed to select freely his/her favorite image for use as the target image. Using their method, the user is not allowed to select freely his/her favorite image for use as the target image. Therefore desired in this study to remove this weakness of the method while keeping its merit, that is, it is aimed to design a new method that can transform a secret image into a secret fragment visible mosaic image of the same size that has the visual appearance of any freely selected target image without the need of a database.

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

II. CONVENTIONAL METHODS
In the image-processing applications, the conventional methods that are most frequently used are cryptography, watermarking and steganography using Least- Significant Bit (LSB) Algorithm. Cryptography is the method in which encryption and decryption are performed based on the secret key, which is known only to the sender and receiver. The original information is embedded by following some encoding process in which the data is re-inserted based on
certain procedure. In order to decode the message at
the receiver, the reverse process is followed which is
done at the encoding process. The difference between
steganography and cryptography is that

![Fig.2 (a). Encryption Process](image)

![Fig.2 (b). Decryption Process](image)

Cryptography is preferred to keep the contents of the
message secretly whereas steganography is the method
which keeps the existence of the message secret. These
two methods protect information from the unwanted
parties and security attacks. The other technology that
is closely related to these methods is digital water-
marking, in which an image is embedded into the
original image such that it helps in signifying the
ownership for the purpose of copyright protection[5].
Water-marking technique enables the intellectual
property of the owner to identify the customers who
break their licensing agreement by supplying the
property to third parties. Fig.2(a) and 2(b) represents
the encryption and decryption processes in
cryptography. This paper describes the steganography
algorithm that is most suitable for business and in
commercial applications.

III. IMAGE ENCRYPTION

The information security is used from moldages, different
person using different techniques to secure their data. Following are some
techniques that use security of images from ancient
agetostill date
A. Steganography
B. WaterMarking Technique
C. Visual Cryptography
D. Without sharing Keys Techniques

A) STEGANOGRAPHY

The steganography word comes from the Greek word Stega
nos, which is used to cover secret and cryptography
used for writing in drawing. Therefore, steganography is lit-
erally, covered writing. The main idea for covering the infor-
mation or steganography is used for secure communication in a com-
pletely undetectable manner and to avoid drawings suspicious
the transmission of hidden data[4]. During the transmission pro-
cess, characteristics of these methods are to change in the struc-
ture and features so as not to be identifiable by human eye. Digital
videos, images, sound files, and other files of computer that contain
perceptually important information can be used as covers or
information carriers to hide secret messages. After embedding a mess-
ager into the cover image, asso-
called – stego image is obtained.

In[2] Security, Capacity and robustness are three different aspects which
affecting steganography and its usefulness.

Capacity is used to the amount of information that can be held
in the cover medium. Security relates to an eavesdropper’s
sinability
to detect hidden information and robustness is the amount of
modification in the stego medium. It can withstand before an ad-
versary can
destroy the hidden information.
The concept of the
mosaic images in[1] was created perfectly and it
has been widely used. Four types
of mosaic images namely crystalization mosaic, ancient mosaic,
photomosaic and puzzle imagemosaic are proposed
in[2]. In the
first two types, the source image is split into tiles and the
image is reconstructed by painting the tiles, and they are named as...
tile images. The next two types include obtaining target image
and with the help of database, cover image has been obtained. They maybe called as multi-picture mosaics.

**B) WATERMARKING TECHNIQUE**

WaterMarking is also one of the techniques used to hide the digital image. Digital watermarking is a process of embedding (hiding) marks which are typically invisible and that can be extracted only by owner’s software authentication. This is a technology that is usually embedded in [15] with the image that cannot be misused by any other unauthorized users. This technology allows any one to do without any distortion and keeping much better quality of stego-

image, also in a secure and reliable manner guaranteeing efficient retrieval of secret file. Digital watermarking finds widespread application in security, authentication, copyright protection and all walks of internet applications. There has been effective growth in developing techniques to discourage unauthorized duplication of applications and data. The watermarking technique is one which is feasible and design to protect the applications and data related. The term ‘cover’ is used to describe the original message age in which will hide our secret message, data file or image file. Invisible watermarking and visible watermarking are the two important types of the above said technology. The main objective of this package is to ensure unauthorized duplication of applications and data, providing copyright protection, security, and authentication, to all walks of inter net applications.

**C) VISUAL CRYPTOGRAPHY**

Visual Cryptography is used to hide information in images, a special encryption technique such as way that the encrypted image can be decrypted by the human eye, if the correct key image is used. The technique was proposed by Naor and Shamir in 1994 [1]. It is used in two transparent images. One image contains a secret image and the other random pixels. It is not possible to get the secret information from the images. Both layers of transparent images are encrypted to get the actual information. The easiest way to implement Visual Cryptography is to print the two layers onto a transparent sheet.

**D) WITHOUT SHARING KEY TECHNIQUES**

The author [1] is securing image transmission without sharing his encrypted key, but need to transmit for a single image transmission. In [11], the image is encrypted with private key and is sent without sharing key to the receiver. After receiving the encrypted image, receiver again, decrypted the image by its own keys, and send it to the first sender, first sender removes the first encrypted key and again sends to opponent, The opponent already had the key and with this key the image is finally decrypted. Thus different person applying different techniques for securing his information.

**IV. IDEAS OF THE PROPOSED METHOD**

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

**A) COLOUR TRANSFORMATIONS BETWEEN BLOCKS**

In the first phase of the proposed method, each tile image T in the given secret image is fit into a target block B in a preselected target image. Since the color characteristics of T and B are different from each other, how to change their color distributions to make them look alike is the main issue here. Reinhardt et al. proposed a color transfer scheme in this aspect, which converts the color characteristic of an image to be that of another in the Iαβ color space. This idea is an answer to the issue and is adopted in this paper, except that the RGB color space instead of the Iαβ color space is used to reduce the volume of the required information for recovery of the original secret image. More specifically, let T and B be described as two pixel sets \{p1, p2, . . . , pn\} and \{p1, p2, . . . , p_m\}, respectively. Let the color of each pi be denoted by (ri, gi, bi) and that of each p_j by (r_j, g_j, b_j, i). At first, we compute the means and standard deviations of T and B, respectively, in each of the three color channels R, G, and B by the following formulas:

\[
\mu_c = \frac{1}{n} \sum_{i=1}^{n} c_i, \quad \mu_c' = \frac{1}{m} \sum_{j=1}^{m} c_j',
\]

\[
\sigma_c = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (c_i - \mu_c)^2}, \quad \sigma_c' = \sqrt{\frac{1}{m} \sum_{j=1}^{m} (c_j' - \mu_c')^2},
\]

in which \(c_i\) and \(c_j\) denote the C-channel values of pixels \(p_i\) and \(p_j\), respectively, with \(c = r, g, \text{ or } b\) and \(C = R, G, \text{ or } B\). Next, we compute new color values \((r_1, g_1, b_1, i)\) for each \(p_i\) in T by

\[
c_i'' = q_c(c_i - \mu_c) + \mu_c',
\]

in which \(q_c = \sigma_c / \sigma_c'\) is the standard deviation quotient and \(c = r, g, \text{ or } b\). It can be verified easily that the new color mean and variance of the resulting tile image \(T_i\) are equal to those of B, respectively. To compute the original color values \((r_i, g_i, b_i)\) of \(p_i\) from the new ones \((r_1, g_1, b_1, i)\), we use the following formula which is the inverse of (3):

\[
c_i = (1/q_c)(c_i'' - \mu_c') + \mu_c.
\]

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

**B) CHOOSING APPROPRIATE TARGET BLOCKS AND ROTATING BLOCKS TO FIT BETTER WITH SMALLER RMSE VALUE**

In transforming the color characteristic of a tile image T to be that of a corresponding target block B as described above, how to choose an appropriate B for each T is an issue. For this, we use the standard deviation of the colors in the block as a measure to select the most similar B for each T. Specially, we sort all the tile images to form a sequence, Stile, and all the target blocks to form another, Starget, according to the average values of the standard deviations of the three color channels. Then, we fit the first in Stile into the first in Starget, fit the second in Stile into the second in Starget, and so on. Additionally, after a target block B is chosen to fit a tile image T and after the color characteristic of T is transformed, we conduct a further improvement on the color similarity between the resulting tile image \(T_i\) and the target block B by rotating \(T_i\) into one of the four directions, 0o, 90o,
180°, and 270°, which yields a rotated version of T with the minimum root mean square error (RMSE) value with respect to B amongst the four directions for final use to fit T into B.

C) HANDLING OVERFLOWS/UNDERFLOWS IN COLOR TRANSFORMATION

After the color transformation process is conducted as described previously, some pixel values in the new tile image T might have overflows or underflows. To deal with this problem, we convert such values to be non-overflow or non-underflow ones and record the value differences as residuals for use in later recovery. Specifically, we convert all the transformed pixel values in T not smaller than 255 to be 255, and all those not larger than 0 to be 0. Next, we compute the differences between the original pixel values and the converted ones as the residuals and record them as part of the information associated with T. Accordingly, the pixel values, which are just on the bound of 255 or 0, however, cannot be distinguished from those with overflow/underflow values during later recovery since all the pixel values with overflows/underflows are converted to be 255 or 0 now. To remedy this, we define the residuals of those pixel values which are on the bound to be 0 and record them as well. However, as can be seen from (3), the ranges of possible residual values are unknown, and this causes a problem of deciding how many bits should be used to record a residual. To solve this problem, we record the residual values in the untransformed color space rather than in the transformed one. That is, by using the following two formulas, we compute first the smallest possible color value cS(with c = r, g, or b) in T that becomes larger than 255, as well as the largest possible value cL in T that becomes smaller than 0, respectively, after the color transformation process has been conducted

\[ cS = \left\lfloor (1/q_c)(255 - \mu'_c) + \mu_c \right\rfloor ; \]
\[ cL = \left\lfloor (1/q_c)(0 - \mu'_c) + \mu_c \right\rfloor . \]

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

D) EMBEDDING INFORMATION FOR SECRET IMAGE RECOVERY

In order to recover the secret image from the mosaic image, we have to embed relevant recovery information into the mosaic image. For this, we adopt a technique proposed by Coltuc and Chassery [24] and apply it to the least significant bits of the pixels in the created mosaic image to conduct data embedding. Unlike the classical LSB replacement methods [8], [25], [26], which substitute LSBs with message bits directly, the reversible contrast mapping method [24] applies simple integer transformations to pairs of pixel values. Specifically, the method conducts forward and backward integer transformations as follows, respectively, in which (x, y) are a pair of pixel values and (x’, y’) are the transformed ones

\[ x' = 2x - y, \quad y' = 2y - x \]
\[ x = \left[ \frac{2}{3} x' + \frac{1}{3} y' \right], \quad y = \left[ \frac{1}{3} x' + \frac{2}{3} y' \right]. \]

The method yields high data embedding capacities close to the highest bit rates and has the lowest complexity reported so far. The information required to recover a tile image T which is mapped to a target block B includes: 1) the index of B; 2) the optimal rotation angle of T; 3) the truncated means of T and B and the standard deviation quotients, of all color channels; and 4) the overflow/underflow residuals. These data items for recovering a tile image T are integrated as a five-component bit stream of the form M = t1t2 . . . tnr1r2m1m2 . . . m48q1q2 . . . q21d1d2 . . . . .dk in which the bit segments t1t2 . . . tn, r1r2, m1m2 . . . m48, q1q2 . . . q21, and d1d2 . . . dk represent the values of the index of B, the rotation angle of T, the means of T and B, the standard deviation quotients,
and the residuals, respectively. In more detail, the numbers of required bits for the five data items in \( M \) are discussed below: 1) the index of \( B \) needs \( m \) bits to represent, with \( m \) computed by

\[
m = \lceil \log((W_S \times H_S) / N_T) \rceil
\]

in which \( W_S \) and \( H_S \) are respectively the width and height of the secret image \( S \), and \( N_T \) is the size of the target image \( T \); 2) it needs two bits to represent the rotation angle of \( T \) because there are four possible rotation directions; 3) 48 bits are required to represent the means of \( T \) and \( B \) because we use eight bits to represent a mean value in each color channel; 4) it needs 21 bits to represent the quotients of \( T \) over \( B \) in the three color channels with each channel requiring 7 bits; and 5) the total number \( k \) of required bits for representing all the residuals depends on the number of overflows or underflows in \( T_- \). Then, the above-defined bit streams of all the tile images are concatenated in order further into a total bit stream \( M_t \) for the entire secret image. Moreover, in order to protect \( M_t \) from being attacked, we encrypt it with a secret key to obtain an encrypted bit stream \( M_-t \), which is finally embedded into the pixel pairs in the mosaic image using the method of Coltuc and Chassery [24] described above. It may require more than one iteration in the encoding process since the length of \( M_-t \) may be larger than the number of pixel pairs available in an iteration. A plot of the statistics of the numbers of required bits for secret image recovery is shown in Fig. 8(b). Moreover, we have to embed as well some related information about the mosaic image generation process into the mosaic image for use in the secret image recovery process. Such information, described as a bit stream \( I \) like \( M_-t \) mentioned previously, includes the following data items: 1) the number of iterations conducted in the process for embedding the bit stream \( M_-t \); 2) the total number of used pixel pairs in the last iteration for embedding \( M_-t \); and 3) the Huffman table for encoding the residuals. With the bit stream \( M_-t \) embedded into the mosaic image, we can recover the secret image back as will be described later. It is noted that some loss will be incurred in the recovered secret image, or more specifically, in the color transformation process using (3), where each pixel’s color value \( c_i \) is multiplied by the standard deviation quotient \( q_c \), and the resulting real value \( c_-i \) is truncated to be an integer in the range of 0 through 255. However, because each truncated part is smaller than the value of 1, the recovered value of \( c_i \) using (4) is still precise enough to yield a color nearly identical to its original one. Even when overflows/underflows occur at some pixels in the color transformation process, we record their residual values as described previously and after using (4) to recover the pixel value \( c_i \), we add the residual values back to the computed pixel values \( c_i \) to get the original pixel data, yielding a nearly losslessly recovered secret image. According to the results of the experiments conducted in this paper, each recovered secret image has a very small RMSE value with respect to the original secret image.

V. SECURE IMAGE TRANSMISSIONS

The information into the original information. The word Steganography is derived from the Greek words “stegos” meaning “cover” and “graphic” which means “writing”. In most of the image processing applications, Steganography is used to hide the information in the images.

![Fig.3. Basic Steganography System Scenario](image)

Information security is the protection of the image and the systems or hardware that is used to store and transmit the images. Steganography is the most efficient method through which the existence of the message can be kept secret. This can be accomplished
through hiding the information in another image, video or audio file [1][2]. Hence the existing information is hidden secretly. Steganography supports different types of digital formats that are used for hiding the data. These files are known as carrier files. To achieve a high performance approach, both embedding ratio and image quality are considered as important issues. This paper presents the high-performance on achieving security. The steganography system scenario is shown in figure.1

VI. PROPOSED METHOD
SKIN TONE DETECTION:
For colour face images, we use the algorithm described in [1], a skin probability map is created from a special non-linear transformation that injects a zeroed R (the red component in RGB images) into its formulation.

THE EMBEDDING PROCESS
The central focus of this paper is to embed the secret message in the first-level 2D Haar DWT with the symmetric-padding mode guided by the detected skin tone areas.

Algorithms based on DWT experience some data loss since the reverse transform truncates the values if they go beyond the lower and upper boundaries (i.e., 0-255). Knowing that human skin tone resides along the middle range in the chromatic red of YCbCr colour space allows us to embed in the DWT of the Cr channel without worrying about the truncation. This would leave the perceptibility of the stego-image virtually unchanged since the changes made in the chrominance will be spread among the RGB colours when transformed. We choose wavelets over DCT (Discrete Cosine Transform) because: the wavelet transform mimics the Human Vision System (HVS) more closely than DCT does; Visual artefacts introduced by wavelets coded images are less evident compared to DCT because the wavelets transform does not decompose the image into blocks for processing. Let C and P be the cover-image and the payload respectively. The stego-image S can be obtained by the following embedding procedure:

STEP 1: Encrypt P using a user supplied key to yield

STEP 2: Generate skin tone map (skin_map) from the cover C and determine an agreed-upon orientation, if desired, for embedding using face features as described earlier (embedding angle will be treated as an additional secret key)

STEP 3: Transform C to YCbCr colour space

STEP 4: Decompose the channel Y by one level of 2D-DWT to yield four sub-images (CA, CH, CV, CD)

STEP 5: Resize skin_map to fit CA

STEP 6: Convert the integer part of coefficients of CA into the Binary Reflected Gray Code (BRGC) and store the decimal values

STEP 7: Embed (the embedding location of data is also randomized using the same encryption key) the secret bits of P’ into the BRGC code of skin area in CA guided by the skin_map

STEP 8: Convert the modified BRGC code back to coefficients, restore the decimal precision and reconstruct the image Y’

STEP 9: Convert Y’CbCr to RGB colour space and obtain the stego-image, i.e., S. (NB: the effect of embedding is spread among the three RGB channels since the colour space was transformed).
ALGORITHMS OF THE PROPOSED METHOD

Based on the above discussions, the detailed algorithms for mosaic image creation and secret image recovery may now be described respectively as Algorithms 1 and 2.

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

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

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

Step 3. Sort the tile images in the set $Stile = \{T_1, T_2, \ldots, T_n\}$ and the target blocks in the set $Starget= \{B_1, B_2, \ldots, B_n\}$ according to the computed average standard deviation values of the blocks; map in order the blocks in the sorted $Stile$ to those in the sorted $Starget$ in a 1-to-1 manner; and reorder the mappings according to the indices of the tile images, resulting in a mapping sequence $L$ of the form: $T_1 \rightarrow B_{j1}, T_2 \rightarrow B_{j2}, \ldots, T_n \rightarrow B_{jn}$.

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

Stage 2. Performing color conversions between the tile images and the target blocks.
Step 5. Create a counting table $TB$ with 256 entries, each with an index corresponding to a residual value, and assign an initial value of zero to each entry (note that each residual value will be in the range of 0 to 255).

Step 6. For each mapping $Ti \rightarrow Bi$ in sequence $L$, represent the means $\mu_c$ and $\mu_\_c$ of $Ti$ and $Bi$, respectively, by eight bits; and represent the standard deviation quotient $qc$ appearing in (3) by seven bits, according to the scheme described in Section III(A) where $c = r, g, \text{or} b$.

Step 7. For each pixel $pi$ in each tile image $Ti$ of mosaic image $F$ with color value $ci$ where $c = r, g, \text{or} b$, transform $ci$ into a new value $c\_\_i$ by (3); if $c\_\_i$ is not smaller than 255 or if it is not larger than 0, then change $c\_\_i$ to be 255 or 0, respectively; compute a residual value $Ri$ for pixel $pi$ by the way described in Section III(C); and increment by 1 the count in the entry in the counting table $TB$ whose index is identical to $Ri$.

Stage 3. Rotating the tile images.
Step 8. Compute the RMSE values of each color transformed tile image $Ti$ in $F$ with respect to its corresponding target block $Bi$ after rotating $Ti$ into each of the directions $\theta = 0^\circ, 90^\circ, 180^\circ$ and $270^\circ$; and rotate $Ti$ into the optimal direction $\theta_0$ with the smallest RMSE value.

Stage 4. Embedding the secret image recovery information.

Fig. 4(A,B). Hiding data in human skin tone areas, bottom shows the differences between the original and stego-images.
Step 9. Construct a Huffman table $HT$ using the content of the counting table $TB$ to encode all the residual values computed previously.

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

Step 11. Concatenate the bit streams $Mjiof all Ti in F in a raster-scan order to form a total bit stream $Mt$; use the secret key $K$ to encrypt $Mt$ into another bit stream $M_\_t$; and embed $M_\_t$ into $F$ by the reversible contrast mapping scheme proposed in [24].

Step 12. Construct a bit stream $I$ including: 1) the number of conducted iterations $Ni$ for embedding $M_\_t$; 2) the number of pixel pairs $Npair$ used in the last iteration; and 3) the Huffman table $HT$ constructed for the residuals; and embed the bit stream $I$ into mosaic image $F$ by the same scheme used in Step 11.

Algorithm 2 SECRET IMAGE RECOVERY
Input: a mosaic image $F$ with $n$ tile images $\{T1, T2, \ldots, Tn\}$ and the secret key $K$.

Output: the secret image $S$.

Stage I. Extracting the secret image recovery information.

Step 1. Extract from $F$ the bit stream $I$ by a reverse version of the scheme proposed in [24] and decode them to obtain the following data items: 1) the number of iterations $Ni$ for embedding $M_\_t$; 2) the total number of used pixel pairs $Npair$ in the last iteration; and 3) the Huffman table $HT$ for encoding the values of the residuals of the overflows or underflows.

Step 2. Extract the bit stream $M_\_t$ using the values of $Ni$ and $Npair$ by the same scheme used in the last step.

Step 3. Decrypt the bit stream $M_\_t$ into $Mt$ by $K$.

Step 4. Decompose $Mt$ into $n$ bit streams $M1$ through $Mn$ for the $n$ to-be-constructed tile images $T1$ through $Tn$ in $S$, respectively.

Step 5. Decode $Mjiof each tile image $Ti$ to obtain the following data items: 1) the index $jiof the block $Bji$; 2) the optimal rotation angle $\theta^o$ of $Ti$; 3) the means of $Ti$ and $Bji$ and the related standard deviation quotients of all color channels; and 4) the overflow/underflow residual values in $Ti$ decoded by the Huffman table $HT$.

Stage 2. Recovering the secret image.

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

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

VII. Experimental Results
A series of experiments have been conducted to test the proposed method using many secret and target images with sizes 256 x 256. To show that the created mosaic image looks like the preselected target image,
the quality metric of root mean square error (RMSE) is utilized, which is defined as the square root of the mean square difference between the pixel values of the two images. An example of the experimental results is shown; Fig. 5(a) & 5(b) shows the created mosaic image using Fig. 3 as the secret image and Fig. 4 as the target image of size 8×8 and 16×16. The tile image size is 8×8. The recovered secret image using a correct key is shown in Fig. 6 which looks nearly identical to the original secret image shown in Fig. 3 with respect to the secret image. It is noted by the way that all the other experimental results shown in this paper have target vs mosaic, as seen in Fig. 8. Moreover, Fig. 7 shows the recovered secret image using a wrong key, which is a noise image. Fig. 6(a), 6(b) show more results using different tile image sizes. It can be seen from the figures that the created mosaic image retains more details of the target image when the tile image is smaller. It can also be seen that the blockiness effect is observable when the image is magnified to be large; but if the image is observed as a whole, it still looks like a mosaic image with its appearance similar to the target image. Fig. 8 shows the graph between target image vs secret images. Fig 9 (a) RMSE for secret image vs extracted image Fig 9 (b) required bits for secret image vs extracted image.
CONCLUSION

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

REFERENCES


