PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 37 (2004) 1377–1385

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Sharing and hiding secret images with size constraint☆

Yu-Shan Wu, Chih-Ching Thien, Ja-Chen Lin*

Department of Computer and Information Science, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan, ROC

Received 13 January 2003; received in revised form 8 January 2004; accepted 8 January 2004

Abstract

This paper presents a method for sharing and hiding secret images. The method is modified from the (t, n) threshold scheme. (Comput.Graph. 26(5)(2002)765) The given secret image is shared and n shadow images are thus generated. Each shadow image is hidden in an ordinary image so as not to attract an attacker's attention. Any t of the n hidden shadows can be used to recover the secret image. The size of each stego image (in which a shadow image is hidden) is about $1/t$ of that of the secret image, avoiding the need for much storage space and transmission time (in the sense that the total size of t stego images is about the size of the secret image). Experimental results indicate that the qualities of both the recovered secret image and the stego images that contain the hidden shadows are acceptable. The photographers who work in enemy areas can use this system to transmit photographs.

Keywords: Sharing secret images; Quantization; Image hiding; Shadow; Size limitation

1. Introduction

Thien and Lin [1] developed a sharing method (a (t, n) threshold scheme, $t \leqslant n$) for sharing a secret image among n participants, such that any t participants could cooperate to reconstruct the secret image, while $t-1$ or fewer participants could not . In Ref. [1], after the secret image was shared, each shadow image contained partial information about the secret image, and the size of each shadow was $1/t$ of that of the secret image. However, the shadow images looked like random noise images rather than ordinary images. Therefore, some data-hiding methods [2–10] must be utilized to transform the shadow images to stego images (by hiding the shadow images in some ordinary images) so as not to attract any attacker's attention. However, each stego image was usually two or four times larger than each shadow image. Accordingly, the size of each stego image was $2/t$ or $4/t$ of that of the secret image. To solve the problem of size expansion, we present in this work a new method in which the size of the stego image (which contains the hidden shadow) is still about $1/t$ of that of the secret image. This requirement is met by shrinking the range of shadow values (which are the output values of the sharing phase in Ref. [1]); hence, the input values (which are the gray values of the secret image) must also be quantized. Therefore, a pre-processing quantization procedure is developed for narrowing the range of gray values of the secret image. The pre-processing procedure firstly quantizes the secret image using two types of blocks, producing a record of block types, namely, an S–E table. The S–E table is then embedded in the quantized image to prevent size expansion. After it has been pre-processed, the image is shared among n participants. Finally, a simple hiding procedure is proposed for hiding each shadow image in an ordinary image. The rest of this paper is organized as follows. Section 2 describes the proposed method. Section 3 presents the experimental results and compares them with those obtained by reported methods. Finally, Section 4 draws conclusions and discusses practical applications.

☆ This work was supported by National Science Council, Taiwan, ROC under Grant NSC 91-2213-E009-097.

* Corresponding author. Tel.: +886-03-5715900; fax: +886-03-5721490.

E-mail address: jclin@cis.nctu.edu.tw (J.-C. Lin).

0031-3203/$30.00
doi:10.1016/j.patcog.2004.01.002

Nomenclature

t	threshold number, such that any t shadows can be used to recover the secret image, while $t-1$ shadows recover nothing
n	number of shadows generated from the secret image, $(n \geqslant t)$
p_{ij}	value of the jth pixel in the ith block of the secret image
S	the block is a smooth block
E	the block is an edge block
d_{ij}	value of the jth pixel in the ith differential block
q_{lj}	value of the jth pixel in the lth quantized block (either quantized or quantized-embedded)
g_i	value of the ith pixel in the shadow image
H_i	value of the ith pixel in the host image
R_i	value of the ith pixel in the stego image

2. Proposed method

As indicated in Fig. 1, the proposed method has three major parts: (a) quantization, (b) sharing, and (c) hiding shadows. These three procedures are introduced in Sections 2.1–2.3, respectively.

The quantization procedure quantizes the original secret image, and yield a quantized-embedded image. The quantization procedure is complicated and so is divided into two sub-procedures (Sections 2.1.1–2.1.2) to facilitate explanation. The quantized-embedded image is then shared using the sharing procedure (Section 2.2), which generates n shadow images. Finally, the hiding shadows procedure (described in Section 2.3) hides the n shadow images in n cover images (also called host images) to yield n stego images. The secret image can be later retrieved from t of the n stego images by reverse operations.

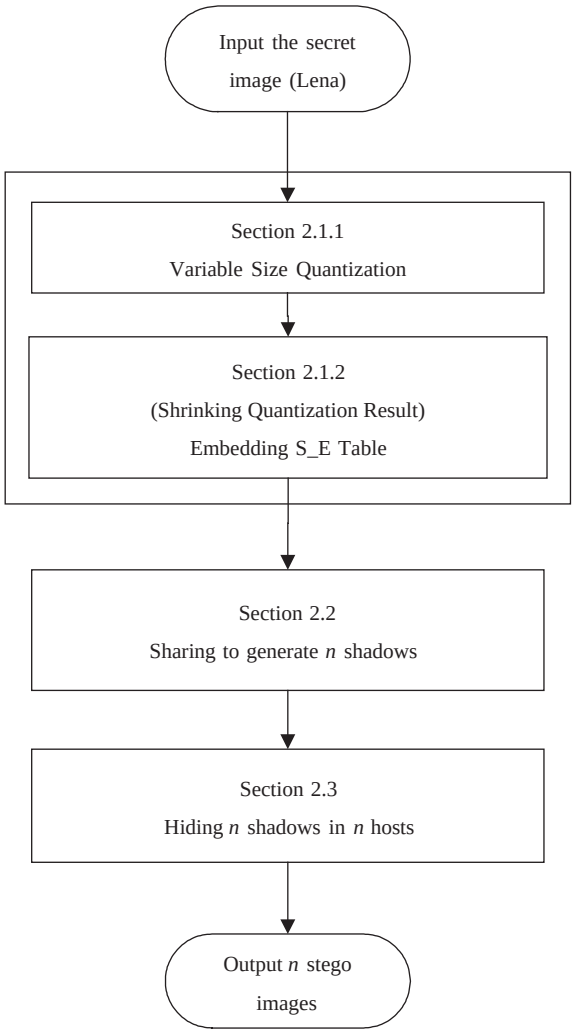


Fig. 1. Flowchart of proposed method.

2.1. Quantization (with Embedding of S–E table)

The quantization procedure consists of two stages. The first stage (Section 2.1.1) quantizes the secret image using a variable size quantization (VSQ) sub-procedure. The output of the VSQ sub-procedure includes a quantized image and an S–E table, which records the order of the smooth and edge blocks. The second stage (Section 2.1.2) embeds the S–E table in the quantized image to yield a quantized-embedded image.

2.1.1. Sub-procedure for variable size quantization

This section introduces the variable size quantization sub-procedure to quantize the original secret image and produce a quantized image and an S–E table. First, the original secret image is divided into numerous non-overlapping blocks, each of which is 1×2 and is called an unclassified block. Then, the first not-yet-processed 1×2 unclassified block i is examined to determine whether it is smooth or an edge block (according to the diversity of gray values in the block). If it is determined to be an edge block, then the edge quantization method is applied. Conversely, if it is determined to be a smooth block, then the next 1×2 unclassified block $(i+1)$ is read and determined to be smooth or an edge block. If the unclassified block $(i+1)$ is an edge block, then we go back to the unclassified block i and reset it as an edge block, then it is quantized using the edge quantization method; otherwise, the two unclassified blocks i and $i+1$ are merged to yield a 1×4 block, which is treated as a smooth block, and quantized using the smooth quantization method. (Hence, each smooth block is 1×4, while each edge block is 1×2.) The next not-yet-processed 1×2 unclassified block is processed in the above manner,

and the process repeated until all unclassified blocks have been processed. Of course, to retrieve the quantized image at a later date, an S–E table is also generated to track the smooth and edge blocks. With the S–E table, a block that is being decoded can be known to be smooth (1×4) or an edge (1×2) block, and so the appropriate method can be used to decode it.

In summary, the size of a smooth block must be four pixels, and that of an edge block must be two pixels. Note that, because the S–E table uses one bit to mark each block (where '1' indicates an edge block and '0' indicates a smooth block), the mergence of smooth blocks (turning two 1×2 blocks into one 1×4 block) helps in reducing the size of the S–E table. (In most natural images, most of the blocks are smooth, and merging them can reduce the size of the table without damaging the image quality excessively.) The complete sub-procedure for variable size quantization is presented below.

Sub-procedure for variable size quantization (VSQ)

Input: Secret image (such as "Lena", for example).

Output: Quantized image and S–E table.

Step 1: Divide the input image into numerous non-overlapping blocks, each of size 1×2. Set both counters i and l to the initial value of zero. (i is the block index for the input image, while l is the block index for the quantized image.)

Step 2: Increase the counters i and l by one. Then, read the ith block of the input image. Assume that the two pixels in the block i are (p_{i1}, p_{i2}).

Step 3: Assume that the two pixels of previous block, $i-1$, are $(p_{(i-1)1}, P_{(i-1)2})$. Subtract $p_{(i-1)2}$ from each of the two current pixels to yield the ith differential block (d_{i1}, d_{i2}). In other words, evaluate

$$d_{ij} = p_{ij} - p_{(i-1)2}; \quad j = 1, 2. \tag{1}$$

(Initially, let $p_{(i-1)2} = 0$ for $i = 1$.)

Step 4: (Classification)

If $\text{Max}\{|d_{i1}|, |d_{i2}|\} \leqslant 8$, then Block i is a smooth block candidate, so go to Step 5.

If $\text{Max}\{|d_{i1}|, |d_{i2}|\} > 8$, then Block i is an edge block, so go to Step 6.

Step 5: Read the next block $(i + 1)$, and evaluate the $(i + 1)$th differential block

$(d_{(i+1)1}, d_{(i+1)2})$ where

$d_{(i+1)j} = p_{(i+1)j} - p_{i2}$ with $j = 1, 2$.

If $\text{Max}\{|d_{(i+1)1}|, |d_{(i+1)2}|\} \leqslant 8$, then

$$q_{lj} = d_{ij} + 8 \quad \text{for } j = 1, 2; \tag{2}$$

$$q_{l(j+2)} = d_{(i+1)j} + 8 \quad \text{for } j = 1, 2; \tag{3}$$

a '0' (smooth) record is added to S–E table; then, increase the counter i by one and go to Step 7.

(Note that we merge two 1×2 blocks in this case, so the quantized block $(q_{l1}, q_{l2}, q_{l3}, q_{l4})$ is 1×4).

If $\text{Max}\{|d_{(i+1)1}|, |d_{(i+1)2}|\} > 8$, then cancel the "smooth" candidacy of Block i, that is, treat Block i as an edge block, and go to Step 6.

Step 6: (Quantizing an edge block). For $j = 1$, set $q_{lj} = \lceil p_{ij}/17 \rceil$ if

$$\left| p_{ij} - \left\lceil \frac{p_{ij}}{17} \right\rceil \times 17 \right| \leqslant \left| p_{ij} - \left\lfloor \frac{p_{ij}}{17} \right\rfloor \times 17 \right|. \tag{4}$$

Otherwise, set $q_{lj} = \lfloor p_{ij}/17 \rfloor$. Repeat for $j = 2$. (Notably, the retrieved value $17 \times q_{lj}$ can be viewed as a rounding of p_{ij} to its nearest multiple of 17.) Now, reset the value of p_{i2} for future reference (since it will be used in Step 3) by assigning

$$p_{i2} = q_{l2} \times 17,$$

and then insert a '1' (edge) into the S–E table to indicate that the quantized block l is a 1×2 edge block.

Step 7: Repeat 2–6 until all blocks have been processed.

2.1.2. Sub-procedure for Embedding S–E table

The VSQ sub-procedure creates an S–E table to ensure that the positions of smooth and edge blocks are traceable. If the S–E table is attached to the quantized image, then the quantized image will be expanded. Therefore, the S–E table should be embedded into the quantized image. The sub-procedure is described as follows: Note that 0 and 1 are used in the S–E table to represent smooth and edge blocks, respectively, so the S–E table is a series of zeroes and ones . Hence, an attempt is made to embed these 0s and 1s into the quantized image to yield the quantized-embedded image, such that the size of the quantized-embedded image (which contains the S–E table) is the same as that of the quantized image. First, each row of the quantized image is partitioned into several non-overlapping blocks, called quantized blocks, each with a size of 1×3. The three quantized values of a quantized block l are represented by (q_{l1}, q_{l2}, q_{l3}). Also, without loss of generality, assume that the secret image, and hence the quantized image, are both 512×512.

The embedding method reads 510 pixels (one row) of the quantized image and evaluates the $(q_{l1} \oplus q_{l2} \oplus q_{l3}) \bmod 2$ value 170(=510/3) times (with $l = 1$–170), in the hope that these 170 values are identical to the expected 170 *se_value*s (0 or 1) taken from the S–E table. If so, the 170 entries of the S–E table are already embedded into the quantized image. Of course, if some values of l exist such that $(q_{l1} \oplus q_{l2} \oplus q_{l3}) \bmod 2 \neq se\_value$, then the quantized image must be adjusted in some way. In that case, a quantized value q_{lk} is selected from (q_{l1}, q_{l2}, q_{l3}), and modified by adding or subtracting 1, such that $(q_{l1} \oplus q_{l2} \oplus q_{l3}) \bmod 2 = se\_value$. The technique by which one pixel is selected from the three (to ensure that modification error is small) is complicated, and omitted here to reduce the length of the paper.

Sub-procedure for Embedding S–E table in quantized image

Input: the quantized image and the S–E table.

Output: the quantized-embedded image (which contains the S–E table).

Step 1: Set the row index r to the initial value $r = 1$.

Step 2. Read the rth row of the quantized image. Partition the first 510 pixels into 170 non-overlapping 1×3 blocks. (The table is embedded only in the first 510 pixels of each 512-pixel row.) Set block counter l to the initial value $l = 0$.

Step 3: Increase the value of l by 1. Then read the lth quantized block (q_{l1}, q_{l2}, q_{l3}) of row r.

Step 4: Read the next record (a '0' or a '1') of the S–E table. Call this record an *se_value*.

Step 5: Embed this *se_value* in quantized block l. If $(q_{l1} \oplus q_{l2} \oplus q_{l3}) \bmod 2$ equals *se_value*, then do nothing. Otherwise, select a pixel q_{lk} from (q_{l1}, q_{l2}, q_{l3}), and then change q_{lk} to $q_{lk} + 1$ or $q_{lk} - 1$—whichever is associated with the smaller error. The detail is omitted for paper length.

Step 6: Repeat Steps 2–5 until all quantized blocks of row r are processed.

Step 7: Increase the value of r by 1 and go to Step 2.

2.2. Sharing

Section 2.1 described the quantization procedure (with table embedding) for processing the original image and outputting the quantized-embedded image. This section employs the image sharing procedure, which was proposed by the authors in Ref. [1], to share the quantized-embedded image. The procedure used in Ref. [1] was a (t, n) threshold scheme that generated n shadows, such that any $t (t \leqslant n)$ of these shadows could be used in the reconstruction phase. The (slightly modified) sharing procedure is as follows:

Procedure for sharing

Input: the quantized-embedded image.

Parameter settings: Let n and t be two given integers $(t \leqslant n)$. Set the prime number P to the value 17. (Notably, P is 17 here, rather than 251, as was used in Ref. [1].)

Output: n shadow images.

Step 1: Divide the quantized-embedded image into non-overlapping blocks, each of size $1 \times t$. Set counter l to the initial value $l = 0$.

Step 2: Increase the value of l by 1 and then read the lth block of the quantized-embedded image. Assume that the t quantized-embedded values of the block are $(q_{l1}, q_{l2}, \ldots, q_{lt})$.

Step 3: Use the polynomial

$$f(x) = (q_{l1} + q_{l2}x + \ldots + q_{lt}x^{t-1}) \bmod 17 \qquad (5)$$

to generate $n (n \leqslant 17)$ shadow-values $f(1)$–$f(n)$.

Step 4: Use the n shadow-values $f(1)$–$f(n)$ as the lth pixel value of the n shadow images F_1–F_n, respectively.

Step 5: Repeat Steps 2–4 until all blocks have been processed.

In Step 3 above, the prime number P was set to 17, the reasons for which choice are detailed below. In Ref. [1], the prime number P was 251. The shadow values in Ref. [1] were therefore in the range 0–250. This range makes each shadow image difficult to hide in an ordinary image of the same size. (The host image is commonly larger than the shadow image.) The new setting $(P = 17)$ ensures that all shadow values are in the range 0–16 (which is the range of pixel values of the quantized–embedded image), such that each shadow image can be hidden in an ordinary host image without requiring that the size of the host image exceeds the size of the shadow image. Notably, besides setting P to 17, the pixel values of the image must be quantized into the range 0–16 (as described in Section 2.1) before sharing. Without quantization, the gray values would be in the range 0–255, so $f(1)$–$f(n)$ could not be used to retrieve the gray values (when the t coefficients in Eq. (5) are replaced by gray values), because the mod 17 operation in Eq. (5) would make the set of the retrieved gray values non-unique.

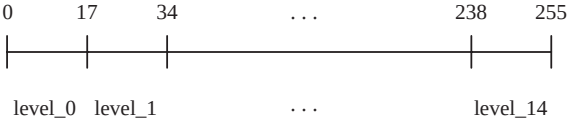


Fig. 2. Fifteen levels in the procedure for hiding shadows (Section 2.3).

2.3. Hiding shadows in host images

As mentioned in the preceding section, the quantized-embeded image is divided into n shadows. Each shadow looks like random noise that will attract an attacker's attention, increasing its probability of being destroyed. A hiding shadow procedure is proposed to solve this problem. First, n ordinary gray-value images (non-secret images) are specified as host images, each of which has the same size as the shadow image (and is therefore smaller than the secret image). Second, the n shadow images are respectively hidden in the n selected host images, in each case yielding a stego image. All n stego images still appear to be ordinary images, reducing the probability that they will be attacked. The hiding shadow procedure presented below employs simple arithmetic to hide the shadow images. For each given host image and shadow image, the procedure first divides the gray values $\{H_i\}$ of the host image into 15 possible levels according to Eq. (6). (Also see Fig. 2.) Then, the ith shadow-value g_i of the given shadow is hidden in H_i by simple mathematical operations. (See Steps 4 and 5 below.) Notably, 8 is subtracted from g_i to obtain the new value g_i' (i.e. $g_i' = g_i - 8$) before hiding (Eq. (7)). The original range of g_i is 0–16, so the range of g_i' is −8–8. Hence, the gray value R_i of the stego image will not be too different from the gray value H_i of the host image, when g_i' is added to H_i. Also, after R_{i1} and R_{i2} are evaluated according to Eqs. (8) and (9), these two values are compared to H_i, to determine which of R_{i1} and R_{i2} is closer to H_i (Eq. (10)). The closer value is used as the gray value R_i in the stego image. At a later date, g_i' can be recovered from R_i by letting $g_i' = R_i \bmod 17$ (or $g_i' = (R_i \bmod 17) - 17$, if $R_i \bmod 17 > 8$).

Procedure for hiding shadows

Input: a shadow image and a host image.

Output: a stego image.

Step 1: Set pixel counter i to the initial value $i = 0$.

Step 2: Increase the value of i by 1. Then, read the ith pixel value H_i of the host image.

Step 3: Read the ith shadow-value g_i of the shadow image.

Step 4: Assume that R_i denotes the ith pixel value in the stego image. Before hiding g_i in H_i to yield R_i, conduct the following evaluation.

$$\text{(i) } Q = \lfloor H_i/17 \rfloor. \tag{6}$$

$$\text{(ii) } g_i' = g_i - 8. \tag{7}$$

$$\text{(iii) } R_{i1} = Q \times 17 + g_i'. \tag{8}$$

$$\text{(iv) } R_{i2} = (Q+1) \times 17 + g_i'. \tag{9}$$

Step 5: (Select from R_{i1} and R_{i2} the one closer to H_i.)

$$\text{If } |H_i - R_{i1}| \leqslant |H_i - R_{i2}|, \quad \text{then } R_i = R_{i1}; \tag{10}$$

otherwise, $R_i = R_{i2}$.

Step 6: Repeat Steps 2–5 until all pixels have been processed.

A stego image can be generated based on the above technique. The procedure can be repeated n times (using n distinct host images and n shadow images) to generate the n desired stego images. Notably, the n stego images will still appear to be general images.

3. Experimental findings and comparison with results of reported sharing methods

The parameter values $n = 6$ and $t = 4$ are set in the experiments performed herein. Accordingly, six stego images are generated and the secret image can be reconstructed by collecting any four of the six stego images. Fig. 3(a) displays the secret image Lena, and Figs. 3(b1)–(b6) display the six host images. Notably, the sizes of the secret image and the host image are 512×512 and 256×256 pixels, respectively; hence, the size of each host image is $1/t = 1/4$ of the size of the secret image. Figs. 4(a1)–(a6) present the six stego images obtained by implementing the proposed method. All of them have a size of 256×256 pixels. The PSNRs of the six stego images are not identical, but all are about 34 dB. The secret image can be retrieved by collecting any four of the six stego images, and applying the mechanism for reconstructing the secret image. (The major part of the reconstruction mechanism is the transformation of each of the four shadow values [$f(1)$–$f(4)$ defined in Eq. (5), for example] back to the four quantized–embedded values q_{I1} –q_{I4}.) Fig. 4(b) shows the reconstructed secret image, with a PSNR of 37.9 dB. Notably, Fig 4(b) is independent of which four of (a1)–(a6) are employed. Any combination yields the same image with 37.9 dB.

Other 512×512 secret images are tested, and their PSNR values are listed in Table 1. Notably, the image quality (PSNR) of each recovered secret image is high (37.66–41.64 dB), and the stego images are small (256×256) and of acceptable quality (34 dB).

The visual quality of the retrieved images is discussed below. To get high quality retrieval of a given secret image, some readers may, for example, expect just to hide the secret image in eight 512×512 host images using the 1-bit LSB (Least Significant Bit) hiding method [6], such that they can have the error-free recovery of the given 512×512 secret image at a later date. However, the hiding method alone (without image sharing) is useless, because the (t, n) fault-tolerant property does not then apply (meaning that the reconstruction system cannot tolerate the absence of any stego image), and each individual stego image may reveal a small part of the secret image after decryption. Therefore, this study focuses on methods (both the newly proposed one and others) that use sharing. In the proposed approach, the distortion in the gray value at each smooth pixel of the secret image cannot exceed one, while at each non-smooth pixel, it usually does not exceed eight; the visual quality is therefore acceptable. If the readers are very strict about the visual quality of the retrieved secret image, they may quantize all edge blocks using a finer scale value such as 8 instead of 17 (in other words, replace all those 17s written in Step 6 of Section 2.1.1 by 8s), and slightly increase the size of each host image (to $1.1/t$ for Lena [to $1.07/t$ for Jet], which is still very close to $1/t$), because, for each edge pixel, an extra bit is required to indicate the more precise quantization value, due to the use of a finer quantization scale. (This extra bit is processed separately such that each pixel [each number] that is shared in the sharing procedure is still in the range 0–16). The retrieved secret images Lena and Jet will then have PSNRs of 44.36 and 45.79 dB, respectively. Similarly, if the quantization scale value 17 is replaced by a much finer scale value 4, then the retrieved secret images Lena and Jet will have PSNRs of 49.21 and 50.47 dB, respectively. The size of each host image is $1.2/t$ (of that of the secret image) for Lena, and $1.14/t$ for Jet. The PSNR of each stego image is still around 34 dB. Although this 34 dB value may be improved by using other image hiding methods of smaller hiding capacity, the size of each host image will be further increased. Finally, if absolutely no distortion of the retrieved secret image is allowed, then certain well-known error-free compression methods, such as JPEG, S+P transform, or SCAN-based Compression [10], should be applied to compress the secret image. After the compressed file is transformed into a base-17 file (a numeric file in which each digit is in the range 0–16), the proposed sharing and hiding procedures are applied to that base-17 file. The size of each stego image will be greater than $1/t$ (but not $2/t$) of that of the secret image.

The quality of the image retrieved by the proposed method is compared to that of the images retrieved by reported image-sharing methods. In relation to image sharing,

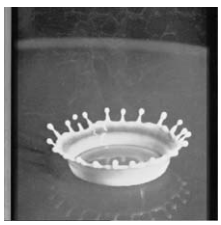
(a)

(b1)

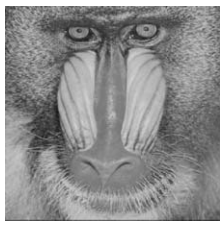
(b2)

(b3)

(b4)



(b5)



(b6)

Fig. 3. Inputs: (a) 512 × 512 secret image, Lena; (b) 256 × 256 host images.

readers may examine the visual cryptography approach (introduced by Naor and Shamir [11] in 1994, and extended by several other authors such as in Refs. [12] and [13]) and the vector quantization (VQ) approach introduced by Chang [14,15]. The advantage of the visual cryptography approach is its simplicity. It is frequently implemented with black-and-white secret images (1 bit per pixel) rather than gray-value secret images (8 bits per pixel), since the idea that underlies the approach is that it operates as a direct extension of the stacking of several transparent sheets, on each of which are scattered black dots in a special pattern. Verheul and van Tilberg [12], developed an extended technique for treating gray-value images, but the decoded image obtained by their method is extremely large. Lin and Tsai [13], employed dithering [16] to convert the gray-value secret image into a binary image, and then applied a visual cryptographic technique to the binary version to alleviate the waste of storage space due to image size. However, their shadow images were still too large (although much smaller than those obtained in Ref. [12]), and their retrieved secret images were binary images that exhibited a dithering effect; hence the quality of their images was poorer than of the images obtained herein. (See their Fig. 14 in Ref. [13].) With regard to the vector quantization approach, the earlier work of Chang and Hwang [14], in which the PSNR of the retrieved secret images was 31.26 dB for Lena and 30.12 dB for Jet, was improved later in Ref. [15] by authors in the same research group. In Ref. [15], the reconstructed secret image Jet had a PSNR of 36.96 dB, when all six stego images (each of which was one quarter

Fig. 4. Outputs: (a) 256 × 256 stego images (all with PSNRs of about 34 dB), and (b) recovered 512 × 512 secret image (with PSNR of 37.9 dB) obtained using any four of the six stego images.

of the size of the secret image) were received. The secret image Jet, reconstructed according to the method proposed herein, had a PSNR of 39.3 dB, when any four of the n stego images (each of which was also one quarter of the size of the secret image) were received. In fact, using the finer-quantization technique introduced in preceding paragraph (replacing the quantization scale value 17 by a much finer scale value 4), the PSNR of the reconstructed secret image Jet can be increased to 50.47 dB, at the price of requiring the size of each stego image be 1.14 quarter of that of the Jet image. The total size of the stego images needed to reconstruct the 50.47 dB Jet by the proposed approach is therefore $1.14/4 \times 4 = 1.14$ times the size of Jet, while the total size of the stego images required to reconstruct the 36.96 dB Jet in Ref. [15] was $1/4 \times 6 = 1.5$ times the size of Jet. Although our reconstructed secret image is of higher quality than that in Ref. [15], the stego images therein are of better quality than those herein, because the amount of shared data was smaller in Ref. [15], and so could be better hidden in host images using the single bit LSB method. Finally, the reconstruction of the secret image in Ref. [15] relies on all n stego images, whereas $n - t$ stego images can be lost in our reconstruction. (Therefore, in our approach, $n - t$ channels may be disconnected during wartime.) In summary, a comparison of the proposed approach with that in Ref. [15] indicates that both provide own advantages.

4. Concluding remarks

This work proposed a method for generating n stego images (each of which contains one shadow image of limited size such that the size of each stego image is only [or nearly] $1/t$ of that of the secret image). Any t of the n stego images can be used to recover the secret image with good quality. The proposed method first applies the quantization

Table 1
PSNRs of the recovered secret images in some other experiments (when the 512 × 512 Lena in Fig. 3(a) is replaced by other 512 × 512 images listed in the table, and all six host images are still 256 × 256)

Recovered secret image	House	Jet	Peppers	Milk	Tiff	Woman
PSNR(dB)	41.64	39.36	37.66	40.74	38.79	40.52

procedure to pre-process the secret image. (The quantization procedure not only quantizes the secret image but also embeds the generated S–E table.) The secret sharing procedure was then employed to generate shadow images that appeared noisy (not displayed here). Finally, the hiding shadow procedure was used to hide each shadow image in an ordinary host image.

Experimental findings indicate that the image quality of the retrieved secret images is appropriate (37.66–41.64 dB), and the visual quality is satisfactory for human vision (Fig. 4(b)). The stego images (which contain hidden shadow images) are also of acceptable quality and so are unlikely to attract the attention of an attacker. Most importantly, the size of each stego image is just $1/t$ of that of the secret image (or nearly $1/t$ of that of the secret image if the PSNR of the recovered secret image is increased to 50 dB), while the size of the stego image size obtained by the method in Ref. [1] is much larger (at least $2/t$).

The proposed method is effective for individuals who must send secret images from a restricted area. For example, news photographers or spies who work in sensitive areas may send collected images to their companies or governments. Using the proposed method, they can transform secret images into several smaller stego images. They can then send "small" and "ordinary" stego images (for example, of a natural scene or landscape) in different ways (including e-mail, ftp and even through public web-sites on which people may post photographs) without being suspected by enemies or agents of other companies. The receiver can recover the secret image by gathering sufficient stego images via various channels. Notably, not all stego images need to be collected. Even if some stego images are lost or damaged, the receiver can still recover the secret image. The system is therefore fault-tolerant, increasing the chance of the successful transmission of secret images. (This characteristic is of utmost importance to governments or news companies that are waiting for secret images.)

Government's concern (how to receive secret images) is addressed in the preceding paragraph. Privacy issues are discussed below with reference to preventing the enemy (or an agent of another company) from obtaining the secret images. Even if the enemy intercepts some channels and suspects that some stego images contain hidden information, such that he can extract some shadow images from the stego images, the enemy nevertheless cannot determine the secret image from these extracted shadow images unless he has extracted t shadow images from various channels. The level of security can be further increased by utilizing some security keys to encrypt the secret image, or by changing the order of pixels in the secret image before applying the quantization procedure introduced in Section 2.1. (See SCAN-based encryption [10], introduced by Bourbakis and Dollas, which can use 10^{75000} keys with confusion functions.) The photographers can also apply different keys to different channels, or even apply n different keys to encrypt the n shadow images before hiding the n shadows in the n hosts. They can also replace our hiding algorithm—which does not need the original host images to recover the shadow values—by some other hiding algorithms that do need the original host images to recover the shadow values (thereby requiring the receiver to keep a copy of the host images). Evidently, many optional approaches to increasing the security level exist, making successful breakthrough by an enemy very difficult.

In relation to the preceding paragraph, although encryption can improve the security, encryption cannot completely replace the sharing of images. Without image sharing, an enemy can intercept a single channel and obtain the encrypted data and recover the secret image by repeatedly guessing the keys. Also, corruption of the single channel would prevent the secret image from being received. Using multiple channels and image sharing increases the probability that the friendly receiver can receive the secret image, while maintaining the difficulty to the enemy of obtaining the secret image, especially in a short period. (If each stego image is transmitted through its own channel [but with some coy images], then the enemy must intercept at least t channels; after intercepting and recognizing which t stego images are useful, the enemy must still extract the shadow images and reconstruct the secret images, which tasks will require plenty of time if the secret image or shadow images have been encrypted, as mentioned in the paragraph above.)

Acknowledgements

The authors would like to thank the referee whose comments have greatly improved this paper.

References

[1] C.C. Thien, J.C. Lin, Secret image sharing, Comput. Graph. 26 (5) (2002) 765–770.
[2] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, IBM Syst. J. 35 (3–4) (1996) 313–316.

[3] W. Bender, F.J. Paiz, W. Butera, S. Pogreb, D. Gruhl, R. Hwang, Applications for data hiding, IBM Syst. J. 39 (3–4) (2000) 547–568.
[4] E. Adelson, Digital signal encoding and decoding apparatus, U.S. Patent No. 4,939,515, 1990.
[5] L.F. Turner, Digital data security system, Patent IPN WO 89/08915, 1989.
[6] R.Z. Wang, C.F. Lin, J.C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognition 34 (3) (2001) 671–683.
[7] W.N. Lie, L.C. Chang, Data hiding in images with adaptive numbers of least significant bits based on the human visual system, International Conference on Image Processing, Kobe, Japan, Vol. 4, October 1999, pp. 286–290.
[8] Y.C. Hou, P.M. Chen, Y.F. Chiao, Stegonography: an efficient data hiding method, Proceedings of CVPRIP'98, Durham, Vol. 4, October 1998, pp. 211–214.
[9] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding—a survey, Proc. IEEE 87 (7) (1999) 1062–1078.
[10] N. Bourbakis, A. Dollas, SCAN-based compression-encryption-hiding for video on demand, IEEE Multimedia Mag. 10 (2003) 79–87.
[11] M. Naor, A. Shamir, Visual Cryptography, Advances in Cryptography-EUROCRYPT'94, Lecture Notes in Computer Science, Vol. 950, Springer, Berlin, 1994, pp. 1–12.
[12] E.R.Verheul, H.C.A. van Tilborg, Construction and properties of k out of n visual secret sharing schemes, Design Code. Crytogr. 11 (1997) 179–196.
[13] C.C. Lin, W.H. Tsai, Visual cryptography for gray-level images by dithering technique, Pattern Recogn. Lett. 24 (2003) 349–358.
[14] C.C. Chang, R.J. Hwang, Sharing secret images using shadow codebooks, Inform. Sci. 111 (1998) 335–345.
[15] T.S. Chen, C.C. Chang, New methods for secret image sharing based upon vector quantization, J. Electron. Imaging 10 (4) (2001) 988–997.
[16] Y. Zhang, Space-filling curve ordered dither, Comput. Graph. 22 (4) (1998) 559–563.

About the Author—YU-SHAN WU was born in 1978 in Taiwan, Republic of China. She received her B.S. degree in Computer Science and Information Engineering from Tamkang University in 2000. Then she received her M.S. degree in Computer and Information Science from National Chiao Tung University in 2002. Her recent research interests include pattern recognition and image processing. She is a member of the Phi-Tau-Phi Scholastic Honor Society.

About the Author—CHIH-CHING THIEN was born in 1975 in Taiwan, R.O.C. He received his B.S. and Ph.D. in 1997 and 2003, respectively, from the computer and information science department of National Chiao Tung University, Taiwan R.O.C. His recent research interests include data hiding, image compression, and information security. Dr. Thien is a member of Phi Tau Phi Scholastic Honor Society.

About the Author—JA-CHEN LIN was born in 1955 in Taiwan, Republic of China. He received his B.S. degree in computer science in 1977 and M.S. degree in applied mathematics in 1979, both from National Chiao Tung University, Taiwan. In 1988 he received his Ph.D. degree in mathematics from Purdue University, U.S.A. In 1981–1982, he was an instructor at National Chiao Tung University. From 1984 to 1988, he was a graduate instructor at Purdue University. He joined the Department of Computer and Information Science at National Chiao Tung University in August 1988, and is currently a professor there. His recent research interests include pattern recognition and image processing. Dr. Lin is a member of the Phi-Tau-Phi Scholastic Honor Society.