Optical Engineering 47(4), 047006 (April 2008)

# Progressive sharing of an image

**Kuo-Hsien Hung**
**Yu-Jie Chang**
**Ja-Chen Lin**
Department of Computer Science
National Chiao Tung University
Hsinchu, 300, Taiwan
E-mail: yjchang@cis.nctu.edu.tw

**Abstract.** We propose a sharing method to progressively reveal a given important image in the recovery phase. In the encoding phase, the distributor utilizes the three frequency bands (low, middle, and high) of the given image to generate shadows according to three prespecified thresholds. In the recovery phase, the secret image cannot be revealed if the number of shadows a team collects is less than the lowest threshold. However, when the number of collected shadows reaches the prespecified low (or middle- or high) threshold, the team can reconstruct a low- (or middle- or high-) quality version of the secret image. In other words, the quality of the reconstructed image depends only on the number of shadows being received, rather than on which of the generated shadows are received. Each noise-like shadow is so small that it can be hidden in an ordinary image that is still several times smaller than the original image. © *2008 Society of Photo-Optical Instrumentation Engineers.* [DOI: 10.1117/1.2911719]

Subject terms: discrete cosine transform (DCT); hiding; sharing; progressive reconstruction; small-size stego images.

Paper 070625R received Jul. 28, 2007; revised manuscript received Dec. 11, 2007; accepted for publication Dec. 31, 2007; published online Apr. 29, 2008.

## 1 Motivation and Goals

Confidential or sensitive images often exist in industrial, commercial, medical, and military applications. Security about the transmission and storage of these images can be done using cryptography techniques.[1–5] The data encryption standard (DES)[1] and the RSA method (Rivest, Shamir, and Adleman)[2] are two famous key-based methods. The secret image is first encrypted using a predetermined key; the resulting image is called a cipherimage. Trying to decode the cipherimage would be extremely hard for unauthorized people unless they steal the encryption key. Therefore, the concern then becomes how to protect the secret key. Shamir[3] and Blakley[4] presented the idea of secret sharing, which could also be utilized to increase the safety level of key safeguarding. Their sharing system is an $(r,n)$ threshold scheme, where $r \leq n$, that divides (not duplicates) the secret key into $n$ shadows.The $(r,n)$ threshold scheme has a criterion: Using $r$ or more shadows can recover the secret key, while using $r-1$ or fewer shadows cannot. After sharing the secret key and generating shadows, these shadows are distributed to $n$ locations for safekeeping. The scheme will ensure the safety of key, even if $r-1$ shadows are stolen by an identical hacker. Therefore, a simple way to protect a confidential image might be to encrypt the image using a key and then share that key and store the key's shadows in different places. This kind of protection still has a weakness: The loss or damage of the cipherimage itself (the encrypted version of the confidential image) means that the original image is gone forever, even if we have all shadows of the key. Thien and Lin[6] thus share the secret image itself using an $(r,n)$ threshold scheme for an image.

In Ref. 6, the secret image is shared among $n$ participants, and each participant holds a (noise-like) shadow image. Any $r$ participants can cooperate to reconstruct the secret image, while $r-1$ or fewer participants cannot. The size of each shadow image is $r$ times smaller than that of the secret image; therefore, if most of the communication channels are in good condition, the communication time needed to transmit $r$ shadow images from $r$ distributed sites to an assigned destination for recovery of the secret image will not be too long (as compared with the time needed to transmit the original big-size secret image to the destination).

Notably, in the secret image-sharing method,[6] the recovery result cannot be viewed progressively. However, this was okay because the images discussed in Ref. 6 are all top-secret images, and hence the inverse-sharing output is either completely recovered or nothing but noise. However, in the real world, not every important thing is top-secret. There are some images that are a little sensitive but still need to be processed every day. For example, the owner of a company may not want any employees to sell good-quality, sensitive pictures or blueprints on the black market, yet the owner still wants the employees to cooperate on an everyday basis in order to improve the design shown in the blueprints, or to safeguard the people shown in the pictures. With our new design here, the boss can keep 3 of the 6 generated shadows (using Figs. 4 and 5 as an example), and each of his 3 employees can have one of the remaining $6-3=3$ shadows. Each day, the three employees can cooperate. If the employees want to take a closer view, then they have to ask the boss to give them support. The boss can lend them more shadows to increase the picture's clarity progressively. If, for some reason, one of the employees is absent or runs away with his shadow, then the remaining employees can still ask the boss for help. The boss can lend the incomplete team either one more shadow to show the picture blurrily or two or three more shadows to provide a much better look.

Another example is that, in real life, the channels connecting the shadows' distributed storage sites and the common meeting place for image recovery is sometimes unstable due to connection delay, error, complete failure of a channel or a storage site, or a long transmission over narrow-bandwidth network channels. When an authorized person or group tries to recover an image, the shadows collecting from several locations on the Internet may not arrive at the same time. As a result, the decoding post may not receive many shadows instantly, while the authorized person or group is eager to know what the image looks like. Again, progressive reconstruction through the available shadows is useful here (which is somehow different from the commonly seen progressive image transmission methods[7–9] because any of our generated shadows could be the missing shadow). We therefore wish to propose in this paper a progressive-viewing method to share the images. Notably, since each shadow image looks noisy, an attack from hackers is likely. Therefore, our $n$ generated shadows are hidden in $n$ cover images to form $n$ stego images, which look ordinary instead of being noisy, to avoid attracting the hacker's attention. We also wish that the size of each ordinary-looking stego image (containing a noise-like shadow image hidden inside) be still much smaller than that of the original secret image. This cannot only keep the storage space and transmission time economic, but can also increase the chance that the receiver can view the secret image (when each communication channel can only be stable for a very short time).

Therefore, our goal is to design a progressive sharing method whose stego images are small. The rest of this paper is organized as follows. Sec. 2 reviews the secret-sharing method briefly. Sec. 3 describes the encoding, while Sec. 4 introduces the decoding (the reconstruction phase). Experimental results and security analysis of shadows are detailed in Sec. 5. A discussion appears in Sec. 6, while conclusions are given in Sec. 7.

## 2 A Review of Secret-Sharing Methods

The concept of secret sharing was introduced independently by Shamir[3] and Blakley.[4] Their $(r, n)$ threshold scheme divides a secret numerical value into $n$ shares, and any $r$ shares can recover the secret numerical value. Several secret-sharing methods based on their $(r, n)$ threshold scheme have been proposed.[6,10–17] Among them, Thien and Lin[6] proposed an $(r, n)$ sharing scheme particularly for secret images. The secret image was shared among $n$ participants, and each participant held a generated shadow image whose size was only $1/r$ that of the secret image. The smaller size of their shadow images ($r$ times smaller than the shadow images created by ordinary sharing methods) is an advantage in the transmission and storage. They further developed a method in Ref. 12 that made the shadow images look like portraits of the original secret image, and thus provided a user-friendly interface to facilitate the management of the shadow images. Extensions of Shamir's masterpiece[3] to combine with visual cryptography[5] can be found in Refs. 15 and 16. Wang and Shyu[17] also proposed a scalable secret image-sharing scheme to increase the applications of the secret image-sharing scheme.[6] But the method is still not a progressive one. As for the method in the frequency domain, Lin and Tsai[13] mapped the secret image into the frequency domain and then utilized a sequence of random numbers to record the lower-frequency coefficients except the most important one (the DC value). The DC value of each block is regarded as the secret key and shared among the $n$ participants by applying the $(r, n)$ threshold scheme. Though it is a frequency domain method, Ref. 13 cannot progressively display the image; moreover, the sequence of random numbers, which records the AC lower-frequency coefficients of the secret image, must be stored elsewhere carefully.

Below we review Ref. 6 in particular, for our method utilizes its sharing polynomials. In Ref. 6, a secret image $O$ containing $m$ pixels is shared by $n$ participants using a polynomial of module base 251. The details are as follows. The image $O$ is first permuted to a noisy image $Q$. Then, $Q$ is divided into $m/r$ nonoverlapping sections so that each section contains $r$ pixels. Let $q(x)$ be the $x$th shadow image and $q_j(x)$ be the $j$th pixel in $q(x)$, where $1 \le x \le n$ and $1 \le j \le m/r$. For each section $j$, define its sharing polynomial

$$q_j(x) = a_0 + a_1 x + \cdots + a_{r-1} x^{r-1} \pmod{251}, \tag{1}$$

whose $r$ coefficients $a_0, a_1, \ldots, a_{r-1}$ are the gray values of the $r$ pixels in section $j$. The $x$th shadow image $q(x)$ is the collection $\{q_j(x) | j = 1, 2, \ldots, m/r\}$. Since each section $j$, which has $r$ pixels, contributes only one pixel $q_j(x)$ to the $x$th shadow image, the size of each generated shadow image is only $1/r$ that of the secret image $O$. This property holds for every shadow image, i.e., for every $x \in \{1, 2, 3, \ldots, n\}$. Any $r$ of the $n$ shadow images can be utilized to reconstruct $Q$; for the inverse process to find the value of the $r$ coefficients $a_0, a_1, \ldots, a_{r-1}$ used in Eq. (1) only needs $r$ of the $n$ values $\{q_j(1), q_j(2), \ldots, q_j(n)\}$. This is a numerical interpolation problem, and the solution can be found using a linear combination of Lagrangian polynomials (see Refs. 6 or 10 or any numerical methods textbook). For example, if $r=3$ and the three received values are $\{q_j(2), q_j(3), q_j(5)\}$, then

$$q_j(x) = \left[ q_j(2) \frac{(x-3)(x-5)}{(2-3)(2-5)} + q_j(3) \frac{(x-2)(x-5)}{(3-2)(3-5)} + q_j(5) \frac{(x-2)(x-3)}{(5-2)(5-3)} \right]_{\text{mod } 251}.$$

All arithmetic operations in this equation, including division, are in the modulus sense, i.e., $1/y$ is the *integer* $z$ satisfying that $1 = (yz)_{\text{mod } 251}$. For example, $1/6 = 42$ because $(6 \times 42)_{\text{mod } 251} = 252_{\text{mod } 251} = 1$. Notably, module base is 251 in Ref. 6, for 251 is a prime very close to 256, and 256 is the number of gray levels in an image.

## 3 Encoding

As indicated in Fig. 1, the proposed *progressive image-sharing* (PIS) method consists of (1) quantization after the discrete cosine transform (DCT), (2) base-17 transform, (3) band partition, (4) sharing, (5) combining shares, and (6) data hiding. Before introducing the detail of these procedures in the subsections of this section, we first quickly glance at the algorithm.

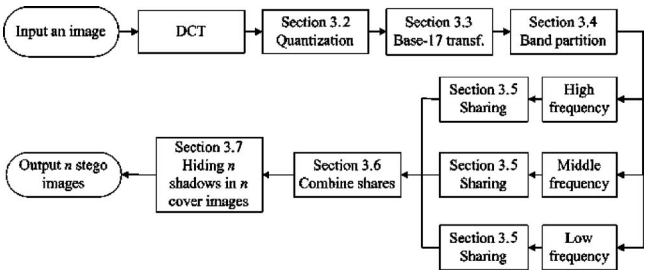



**Fig. 1** The encoding flowchart.

### 3.1 *Encoding Algorithm*

Input: the given gray-value image.

Parameter settings: Let $r_L < r_M < r_H \leq n$ be four given integers. Set the prime number $p$ to the value $p=17$ (rather than the $p=251$ used in Ref. 6).

Output: $n$ stego images.

Steps:

1. Divide the image into nonoverlapping $8 \times 8$ blocks.
2. Then for each $8 \times 8$ block, do the following:

- As in JPEG, subtract 128 from each gray value of the block; then compute the discrete cosine transform (DCT) values of the block; then quantize the DCT values by the quantization table on page 484 of Ref. 18 (see Sec. 3.2); then arrange the $8 \times 8$ quantized values in zigzag order.[18]
- According to Fig. 2, transform (re-quantize) the frequency values to numerical base-17 values (Sec. 3.3).
- According to Fig. 2, divide the frequency values into 3 frequency bands, i.e., low, middle, and high (Sec. 3.4).
- Share each band according its own threshold value. The products are called *shares* (Sec. 3.5).
- Combine three shares (low-, middle-, and high-frequency) into a "shadow" (Sec. 3.6).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 3 | 3 | 2 | 2 | 2 | 2 | 1 | 1 |
| 3 | 2 | 2 | 2 | 2 | 1 | 1 | 0 |
| 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Fig. 2** The band-partition table ("3" means that quantized coefficient is re-expressed as a 3-digit number in the base-17 system, etc.). The darkest/gray/white region is the low-/middle-/high-frequency region, respectively.

- At the corresponding block position, hide the $n$ shadows in the $n$ cover images. This creates a block, at the corresponding block position, in each of the $n$ stego images (Sec. 3.7).

3. Store the $n$ stego images in $n$ distinct places; or transmit the $n$ stego images by $n$ distinct channels.

### 3.2 *Quantization*

In the frequency domain, there are 64 DCT coefficients for each $8 \times 8$ block. In order to reduce the shadow size, it is essential to quantize the frequency values to reduce the amount of data. The standard quantization table described on page 484 of Ref. 18 is used for quantization.

### 3.3 *Base-17 Transform*

Because our sharing scheme utilizes a mod-17 operation, each digit must be in the 0–16 range. For this reason, it is necessary to transform the numeric base of the frequency values. In other words, each (quantized) frequency value must be transformed to a base-17 number so that each digit is in the 0–16 range and thus becomes more suitable for our mod-17 sharing scheme. Below is the algorithm for the base-17 transform of each coefficient.

#### 3.3.1 *Algorithm for the Base-17 transform*

Input: an integer frequency value $f_v$.

Output: a new integer in which each digit is in the range 0–16.

Steps:

1. According to Fig. 2, obtain the integer $n_u$ (the number of digits needed) at the corresponding coefficient position. For example, $n_u=3$ if $f_v$ is the DC coefficient.
2. Compute

$$s = \lfloor 17^{n_u}/2 \rfloor, \tag{2}$$

which is called the shift level.

3. Transform the shifted value $s+f_v$ into an $n_u$-digit number whose radix (numeric base) is 17.

In the algorithm, the role of $s$ is to adjust the frequency value to its positive version so that subsequent operations can be easier.

### 3.4 *Band Partition*

There are three threshold values $\{r_L, r_M, \text{and } r_H\}$. The smallest threshold value is $r_L$ (the low-frequency threshold), and the largest threshold value is $r_H$ (the high-frequency threshold). Note that low frequency represents the rough sketch of the image; hence, just a small number of shadows ($r_L$ shadows) should be eligible to reconstruct a blurred view of the image. This explains why $r_L$ should be the smallest of the 3 thresholds. An analogous reason explains why the largest is $r_H$.

As will be explained later in Sec. 3.5, due to the "threshold-times smaller" shrinkage property discussed in Ref. 6, each low-frequency share will be $r_L$ times smaller than the data size of the low-frequency data before sharing. Similar arguments hold for the middle-frequency and high-frequency shares, with $r_L$ replaced by $r_M$ and $r_H$, respec-

**Table 1** The information distribution after the frequency-band partition.

| Band | The Thresholds for Sharing | Zigzag Position in the 8×8 DCT Coefficients | Number of Base-17 Digits Used in the Frequency Band | PSNR of Reconstructed Lena Image | PSNR of Reconstructed Boat Image |
|---|---|---|---|---|---|
| Low frequency | $t_L=3$ | 0–2 | $9=3\times3$ | 28.00 | 28.18 |
| Middle frequency | $t_M=4$ | 3–10 | $16=2\times8$ | 32.64 | 32.83 |
| High frequency | $t_H=5$ | 11–29and 30–63 | $25=2\times6+1\times13+0\times34$ | 37.04 | 37.64 |

tively. This "threshold-times smaller" shrinkage property will be used in deciding how to partition the 64 frequency coefficients into 3 bands, as explained below.

Without loss of generality, assume that the three thresholds are $r_L=3$, $r_M=4$, and $r_H=5$. Then, in order to reduce the total size of the joint shadow (which directly combines the low-frequency share, middle-frequency share, and high-frequency share), a general rule can be used: Before sharing, the total amount of low-frequency data should not contain too many digits (because they will be reduced by 300%), while the total amount of high-frequency data can be less strict (because they will be reduced by 500%). In other words, only a few of the $8\times8=64$ coefficients will be assigned to the low-frequency band, while most coefficients will be assigned to high-frequency band. Therefore, as shown in Fig. 2 and Table 1, there are only three low-frequency coefficients in the low-frequency band ($9=3\times3$ digits together for the low-frequency band, which contains only the three top-left coefficients, each which is of 3 digits). However, the middle-frequency band has eight middle-frequency coefficients (each is two digits, so $16=2\times8$ digits together for the middle-frequency band). Finally, the high-frequency band has all the remaining $64-3-8=53$ coefficients. In other words, the high-frequency band has $25=2\times6+1\times13+0\times34$ digits, for there are six 2-digit coefficients and, 13 1-digit coefficients, and the remaining 34 bottom-right coefficients are neglected. Notably, due to the abovementioned "threshold-times smaller" property, each shadow combining the low-, middle-, and high-frequency shares will have $(9/3)+(16/4)+(25/5)=3+4+5=12$ digits, although before sharing the total has $9+16+25=50$ digits.

In general, to partition the $8\times8=64$ frequency coefficients into 3 bands, we may try to let the low- (or middle-, or high-) frequency band have approximately, or proportional to, $r_L^2$ digits (or $r_M^2$ or $r_H^2$, respectively). By doing this, more digits will go to high-frequency band, and thus reduce the size of the 3-bands-joint-shadow because the high-frequency share shrinks most. Notably, with this kind of partition, the joint shadow contains about $(r_L^2/r_L)+(r_M^2/r_M)+(r_H^2/r_H)=r_L+r_M+r_H$ digits, or $(r_L+r_M+r_H)\log_2 17$ bits, which is usually a value at least 2 times smaller than $(8\times8)\log_2 256$ bits, if $r_H\leq20$. This will make it easy to hide the joint shadow in a cover image whose size is identical to the given sensitive image, if the hiding method has a big hiding capacity rate not worse than 1:2 (the hiding capacity is the ratio between the size of the hidden data and the size of the cover image). If $r_H\leq5$, then the cover image can even be $20/5=4=2\times2$ times smaller than the given image, as we will see in the experiment, because $(r_L+r_M+r_H)\log_2 17\leq(5+5+5)\log_2 17=61.3<64=[(4\times4)\log_2 256]/2$ indicates that the shadow of each $8\times8$ block can be hidden in a $4\times4$ block of the cover image, if the hiding method has a big hiding capacity rate not worse than 1:2.

### 3.5 *Sharing*

This section employs the format of the share-generating polynomial in Ref. 6 to share each frequency band of the given image. Assume that $n$ is the number of shadows to be created and that $r$ is one of the three thresholds $\{r_L, r_M, r_H\}$. Split the base-17 data, which are taken from the frequency band (the band corresponding to the threshold $r$) of an $8\times8$ block, into sectors of $r$ digits each. Below we show how to share the $r$ digits $\{a_0,\ldots,a_{r-1}\}$ coming from one sector. Assume that $0<r\leq n$ and $0\leq a_i<17$ for all $a_i$. Let

$$P(x)=a_0+a_1x+a_2x^2+\cdots+a_{r-1}x^{r-1}\pmod{17} \qquad (3)$$

be the share-generating polynomial. For each $k$ in the range $\{1,\ldots,n\}$, the $k$th share receives the value $P(k)$ as the share value corresponding to this sector. When all sectors and all blocks are shared, we have the $n$ shares that we want. Notably, each share receives only one value from each sector; so the number of values in a share is identical to the number of sectors contained in the data. This is why each share is $r$ times smaller than the data, for the number of sectors is $r$ times smaller than the data size.

This paper uses mod 17 (rather than the mod 251 used in Ref. 6) because if the module base being used is 17, then later, when we hide a share value in a cover image using our hiding method,[20] the gray-value distortion at each pixel is at most $\lfloor 17/2\rfloor=8$ according to Ref. 20, which is about the limit of the gray-value distortion that human vision can tolerate at each pixel.

### 3.6 *Combining Shares*

After generating the $n$ shares for each frequency band, then, for each $k=1,\ldots,n$, we directly combine the three (low-, middle-, and high-frequency) $k$th shares to form the $k$th shadow, because a single shadow is more convenient for management and safekeeping than three shares (each shadow holder only has to take and hold one shadow instead of three shares of different bands). In view of the distributed database, handling and managing one shadow is also much easier than doing so with three shares.

Notably, to increase the security level, we may use a key as the seed of a pseudo-random number generator[19] to generate a sequence that is a rearrangement of the natural numbers. Then, according to this generated sequence, we permute the blocks' location or digits' location, within each shadow. The seed of the pseudo-random number generator can depend on, rather than equal, the secret image's total sum of gray values. The seed itself can be transformed to a base-17 number, and then its base-17 digits can also be shared and inserted in some prespecified scattered locations of the $n$ shadows. The details are omitted here to save space. In any case, the seed can be recovered later when $r$ of the $n$ shadows are received.

### 3.7 *Data Hiding*

Each share, and hence, each shadow, looks noisy. The noise-like appearance often catches hacker's attention. To prevent the shadow from being eye-catching, hiding each shadow in an ordinary gray-value image is suggested.[20–24] The data-hiding procedure used here is the so-called modular hiding method,[20] which uses a modular operation to hide data in the least-significant bits of the cover pixels. The detailed algorithm is described in Ref. 20. The distortion between the cover and stego image is guaranteed to be at most $\lfloor 17/2 \rfloor = 8$ at each pixel if the module base being used is 17. The reason can also be found in Ref. 20.

## 4 Decoding (Reconstruction of the Image)

This section introduces the decoding phase that reconstructs the image when receiving enough shadows. In summary, the image can be retrieved from any $r(r \geq r_L)$ of the $n$ stego images by using reverse operations.

### 4.1 *Procedure for Reconstruction*

Input: the received $r(r \geq r_L)$ stego images.

Output: the image that was shared and hidden in the stego-images.

Steps:

1. Extract the hidden data from the $r$ collected stego images.
2. Divide directly each of the $r$ hidden data sets into their 3 corresponding frequency shares.
3. Do inverse sharing band by band: if $r \geq r_H > r_M > r_L$, then do inverse-sharing on all three bands; if $r_H > r \geq r_M > r_L$, then only on the middle and low bands; if $r_H > r_M > r \geq r_L$, then only on the low band.
4. According to the $8 \times 8$ zigzag order, distribute the numbers obtained in step 3 to the $8 \times 8$ coefficient table [according to the partition table (Fig. 2)]. For example, the first three digits are bound together and treated as a single coefficient, namely, the DC coefficient.
5. Scale back the quantized data according to the standard quantization table on page 484 of Ref. 18.
6. Do inverse DCT.
7. Add 128 to each pixel.

### 4.2 *Convenient Version to Do Inverse Sharing*

Below we discuss how to do step 3 above efficiently. In the image-sharing scheme (see Sec. 3.5), the $n$ shadows $P(1),P(2),\ldots,P(n)$ are generated by Eq. (3). In the decoding, after receiving $r$ shadows, the $r$ coefficients $\{a_0,a_1,\ldots,a_{r-1}\}$ of Eq. (3) can be recovered by the *matrix-vector multiplication* method rather than by the Lagrangian polynomials method used in Refs. 6 and 10, which is less convenient.

Below we introduce this more convenient method for image reconstruction. The following equation shows the relationship between the $r$-dimensional data $\vec{a}$ and the $n$-dimensional share values $\vec{S}$ in Eq. (3):

$$\begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & 2 & 2^2 & \ldots & 2^{r-1} \\ \ldots & \ldots & & \ldots & \vdots \\ 1 & n & n^2 & \ldots & n^{r-1} \end{bmatrix} \bullet \begin{bmatrix} a_0 \\ a_1 \\ \ldots \\ a_{r-1} \end{bmatrix} = \begin{bmatrix} P(1) \\ P(2) \\ \ldots \\ P(n) \end{bmatrix}. \tag{4}$$

Let the $r \times r$ generating matrix $G$ be the matrix whose rows correspond to a received participant $i$ and are of the form $1,i,i^2,\ldots,i^{r-1}$. The relationship among $G$,

$$\vec{a} = \begin{bmatrix} a_0 \\ a_1 \\ \ldots \\ a_{r-1} \end{bmatrix}, \quad \text{and} \quad \vec{s} = \begin{bmatrix} P(i_1) \\ P(i_2) \\ \ldots \\ P(i_r) \end{bmatrix}$$

is that $G\vec{a}=\vec{s}$. Here, $i_1,i_2,\ldots,i_r$ are the $r$ received shares. Therefore, $\vec{a}=G^{-1}\vec{s}$, where $G^{-1}$ can be evaluated just once, for $G^{-1}$ is identical between data sectors of $r$ digits each. Then, because $G^{-1}$ is known, the $r$-digit data $\vec{a}$ for each $r$-digit sector can be obtained quickly by $\vec{a}=G^{-1}\vec{s}$, because it is just a multiplication between a fixed $r \times r$ matrix $G^{-1}$ and an incoming $r \times 1$ vector $\vec{s}$. When we have received $r$ shares, as many $r$-dimensional vectors$\vec{s}$ keep on coming in (they are extracted sequentially from the $r$ received shares), we obtain a lot of $r$-digit data $\vec{a}$ sequentially.

For example, assume that $r=r_L=3$ and the collected shadows are Shadows 2, 3, and 5. Below we show how to reconstruct the low-frequency data. The generating matrix

$$G = G_L = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 5 & 8 \end{bmatrix}, \quad \text{where } 8 = (5^2) \bmod 17$$

$$= 25 \bmod 17, \ \vec{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \text{ and } \vec{s} = \begin{bmatrix} P(2) \\ P(3) \\ P(5) \end{bmatrix}.$$

First, evaluate and obtain

$$G^{-1} = \begin{bmatrix} 5 & 12 & 1 \\ 3 & 12 & 2 \\ 6 & 8 & 3 \end{bmatrix}.$$

Then, for $t=2,3$, and 5, grab the first digit from the low-frequency band of Share $t$, and call it $P(t)$. So we have the $\vec{s}=[P(2),P(3),P(5)]^{\text{transpose}}$ for Sector 1. Then evaluate $\vec{a}=G^{-1}\vec{s}$ to obtain the 3-digit data $\vec{a}$ for Sector 1. Then, for $t=2$, 3, and 5, grab the next not-yet-processed digit from the low-frequency band of Share $t$, and still call it $P(t)$. So we have the $\vec{s}$ for Sector 2. Then evaluate $\vec{a}=G^{-1}\vec{s}$, which is the 3-digit data for Sector 2. Repeat this process until all digits from the low-frequency band of the three received shares are processed. Note that the $3\times3$ matrix $G^{-1}$ is never changed in the whole inverse-sharing process for the low-frequency band. As a remark, for a $512\times512$ important image, there are $512\times512/(8\times8)=4{,}096$ image blocks, and each block has $3+3+3=9$ low-frequency digits according to Fig. 2. Thus, there are $4{,}096\times9/3=12{,}288$ three-digit data sectors. So, compared with the time it takes to do matrix-vector multiplication 12,288 times, the time it takes to the $3\times3$ matrix $G^{-1}$ just once can be neglected.

Later, when we obtain one more shadow, for example, Shadow 6, and assume that $r_M=4$, then we can construct a $4\times4$ matrix $G=G_M$ and obtain its inverse matrix $G^{-1}$. Then we can use an analogous inverse-sharing process to obtain middle-frequency data that were partitioned earlier as a sequence of four-digit sectors. The remaining details are omitted to save space.

## 5 Experimental Results and Security Analysis of Shadows

### 5.1 *Experimental Results*

In the experiments, the standard quantization table on page 484 of Ref. 18 is adopted, and the low-($r_L$), middle-($r_M$), and high-($r_H$) frequency thresholds are set to 3, 4, and 5, respectively. Therefore, the low-, middle-, or high-frequency threshold can be reconstructed whenever any 3, 4, or 5 shadows are received, respectively. The parameter $n$ is set to 6, i.e., there are 6 shadows or 6 stego images. The value of the variable *quality* required in JPEG compression software is set to 85 in the experiments. The frequency partition and its relative information are as in Table 1. The partition table appears in Fig. 2, where each number at each cell of the 8-by-8 grid represents the number of digits used to represent that coefficient when the numeric base is 17.

In the first experiment, the target image that the sender really wishes to send is the Lena image shown in Fig. 3(a); and Fig. 3(b1–b6) display the six cover images that will be modified slightly to cover Lena. Notably, Lena is $512\times512$, but each cover image is only $256\times256$, as explained below. According to the frequency partition information shown in Table 1, and as discussed in Sec. 3.4, the total number of generated digits in each $8\times8$ block of the joint shadow is

$$9/3 + 16/4 + 25/5 = 3 + 4 + 5 = 12$$

(each digit is an integer in the range 0–16). Since each digit in the range 0–16 can be hidden in an 8-bit gray-value pixel of the cover image, and also since $[(8\times8):12]=[5.33:1]>[4:1]$, the size of each cover image is chosen to be $4=2\times2$ times smaller than the original image.

The data-hiding method being used in Sec. 3.7 is the modular LSB method[20] with a slight modification discussed in Ref. 25 (the details are omitted because this is not the key point of this paper). Fig. 4 shows the stego images. The qualities of all stego images and reconstructed images are measured by the peak-signal-to-noise ratio (PSNR) defined as

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}}, \tag{5}$$

in which the MSE denotes the mean square error between the pixel values of the cover and of the stego images. Table 2 lists the PSNRs (the measure unit is dB) to gauge the similarity (from 0 dB to $\infty$ dB) between the cover images [Figs. 3(b1)–(b6)] and stego images [Figs. 4(b1*)–(b6*)].

The information about the low, middle, and high frequencies can be retrieved by collecting, respectively, "any" three, four, and five of the six $256\times256$ stego images shown in Fig. 4. The corresponding reconstructed versions are shown in Fig. 5. The PSNRs of the reconstructed $512\times512$ Lena image are, respectively, 28.00, 32.67, and 37.04 dB. The experimental results indicate that the reconstructed version from any $r_L=3$ collected shadows can reveal a rough sketch, while that from any $r_H=5$ collected shadows can show great details of the image. Note that the total size of any $r_L=3$ stego images is only $3\times256\times256/512\times512=75\%$ of the $512\times512$ Lena, image, and that 75% can even be reduced to

$$[(r_L\times12)/(8\times8)]\times[\log 17/\log 256] = [r_L\times12/64] \times[0.511] = r_L\times9.58\,\% = 28.74\%$$

if we did not mind transmitting noise-like shadow images rather than transmitting ordinary-looking stego images. A similar argument holds if 75% (28.74%) is replaced with 100% (38.33%) and 125% (47.91%), respectively, for the total size of the $r_M=4$ stego images [to get Fig. 5(b)] and $r_H=5$ stego images [to get Fig. 5(c)].

As a comparison, note that our with-hiding (without-hiding) sequence {75%(28.74%); 100% (38.33%); and 125%(47.91%)} would have become {150%(75%); 200%(100%); and 250%(125%)} if the method in Ref. 14 were used with the same settings: $r_L=3$, $r_M=4$, $r_H=5$. Therefore, compared to Ref. 14, our sizes are smaller, the transmission time can be shorter, and hence, we are more likely to succeed in an unstable/unfriendly environment. Similarly, compared to Ref. 6, each member of the current


**Fig. 3** (a) The original Lena image of size $512\times512$. (b1)–(b6) The six $256\times256$ cover images Peppers, Jet, Kiel, Lake, Baboon, and Goldhill before hiding.

without-hiding total-space sequence {28.74%, 38.33%, 47.91%} is also much shorter than the without-hiding total-space $r\times(1/r)=1=100\%$ listed in Ref. 6 when people want to view the secret image.

In the second experiment, the target Lena image is replaced with the $512\times512$ image Boat shown in Fig. 6(a). The six obtained $256\times256$ stego images, shown in Figs. 6(b1')–(b6'), are still of good quality. The recovered versions of Boat are in Fig. 7. Note that the name of the boat can be recognized after $r_M=4$ or $r_H=5$ out of the 6 generated shadow images are received [see Fig. 7(b) and 7(c)], but not if $r_L=3$ shadows are received.

## 5.2 *Security Analysis of Shadows*

An encryption scheme should be robust against attacks such as statistical attack and differential attack.[26–28] In the following paragraphs, we discuss these topics.

### 5.2.1 *Statistical analysis (histogram and correlation)*

To test the robustness of our shadows against statistical attacks, we inspect the histograms of the shadows and the correlations of two adjacent elements in the shadows.

In the histograms analysis of a shadow, we count the occurring frequency of each values. Since the proposed method uses mod 17 to obtain a value for each shadow element (i.e., shadow pixel), the value of each shadow element is between 0 and 16. Figure 8 shows an example of our experiment about histograms analysis. Figure 8(a1) is the histogram of the $512\times512$ secret image Lena. Figures 8(b1)–(b6) are the histograms of the six generated shadows for Lena. From Fig. 8, we can see that the histograms of the shadows have close to a uniform distribution. A similar observation also holds for the histograms of the six generated shadows for the secret image Boat. Although the two secret images have very distinct histograms, their shadows' histograms almost all appear to have a uniform distribution.

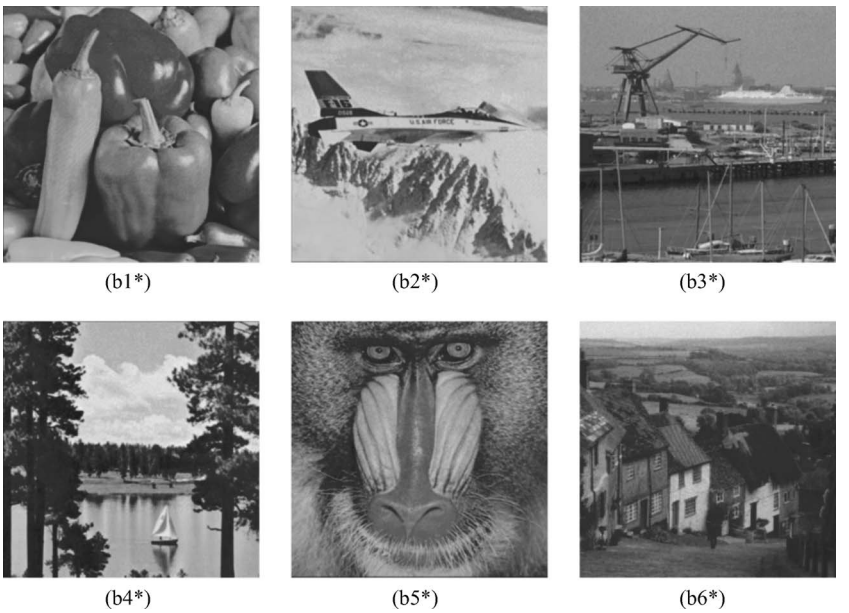



**Fig. 4** The six 256×256 stego images after hiding.

In other words, the histogram of each shadow provides no clue about the secret image used to generate this shadow.

In addition to histogram analysis, we also analyze the correlation between two vertically adjacent pixels (elements), two horizontally adjacent pixels (elements), and two diagonally adjacent pixels (elements) of the secret image (shadow, respectively). The procedure is as follows: Each time, randomly select 1,000 pairs of two adjacent pixels (or elements) from an image (or shadow). Then calculate their correlation coefficient using the following two formulas:

$$\mathrm{cov}(x,y) = E(x - E(x))(y - E(y)), \tag{6}$$

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{7}$$

where $x$ and $y$ are the gray values of two adjacent pixels in the image (or two adjacent elements in the shadow). In numerical computations, the discrete formulas being used become

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{8}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2, \tag{9}$$

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)). \tag{10}$$

**Table 2** The PSNRs of the six stego images [either Fig. 4 or Figs. 6(b1')–(b6')], as compared to the six cover images in Figs. 3(b1)–(b6).

| | Six Stego Images to Recover the Hidden Image | | | | | |
|---|---|---|---|---|---|---|
| The Hidden Image | Pepper | Jet | Kiel | Lake | Baboon | Goldhill |
| Experiment 1 (Lena) | 35.74 | 35.86 | 35.81 | 35.68 | 35.77 | 35.81 |
| Experiment 2 (Boat) | 34.59 | 34.69 | 34.53 | 34.57 | 34.59 | 34.63 |

Table 3 shows the correlation coefficient $r_{xy}$ of two adjacent pixels of the secret image, or two adjacent elements of each shadow. The correlation coefficient of secret image is $r_{xy} = 0.9875$ in the horizontal direction, which implies high correlation among these adjacent pixels of the secret image. To the contrary, the correlation coefficient $r_{xy}$ of the generated shadows is between 0.0040 and 0.0092, which shows the low correlation between adjacent elements of the generated shadows. A similar observation also holds for the vertical and diagonal directions.

### 5.2.2 *Differential analysis*

As quoted from Refs. 26 and 27, the opponent may make a slight change such as modifying only one pixel of the secret image and then observing the result in the cipher image. In this way, he may be able to find a meaningful relationship between the secret image and the cipher image. If one minor change in the secret image can cause a significant change in the cipher image, then this differential attack would become very inefficient and practically useless.

In general, to test the influence of a one-pixel change to the secret image, two commonly used measures are the

**Fig. 5** The reconstructed Lena image of (a) low, (b) low+middle, and (c) low+middle+high frequencies. The PSNRs are 28.00, 32.64, and 37.04 dB, respectively. Note that (a) is from any 3 of the 6 images in Fig. 4; (b) is from any 4; and (c) is from any 5.

*number of pixels change rate* (*NPCR*) and the *unified average changing intensity (UACI)*. Let $C_1$ and $C_2$ be two cipher images whose corresponding original images have only a one-pixel difference. Let $C_1(i,j)$ and $C_2(i,j)$ be the gray values of the pixels at position $(i,j)$ in $C_1$ and $C_2$, respectively. Define a binary array $D$ with the same size as images $C_1$ and $C_2$. Then, $D(i,j)$ is determined by

$$D(i,j) = \begin{cases} 1 & \text{if } C_1(i,j) = C_2(i,j), \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

The *NPCR* is defined as

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\,\%\,, \tag{12}$$

where $W$ and $H$ are the width and height of $C_1$ or $C_2$. The *NPCR* measures the percentage of "unchanged" pixels between these two images. The *UACI* is defined as

$$\text{UACI} = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\,\%\,, \tag{13}$$

which measures the average intensity differences between the two images $C_1$ and $C_2$. Notably, in our method, since the shadow values are in the range 0–16 rather than 0–255, the denominator 255 should be replaced by 16, i.e.,

$$\text{UACI}' = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{16}\right] \times 100\,\%\,. \tag{14}$$

Our *NPCR* values are between 18.23% and 18.78%, with the average *NPCR* being 18.52%. It means that, on average, about $100\% - 18.52\% = 81.48\%$ of the elements are changed per shadow, although the secret image only changes by one pixel. Our *UACI′* values are between 38.11% and 38.87%, with the average *UACI′* being 38.48%. It means that, on average, each shadow value (whose range is 0–16) changes about $16 \times 38.48\% = 6$ in magnitude. With the obtained results for *NPCR* and *UACI′*, we can see that our shadows are very sensitive to small changes to the secret image (secret image changes of only

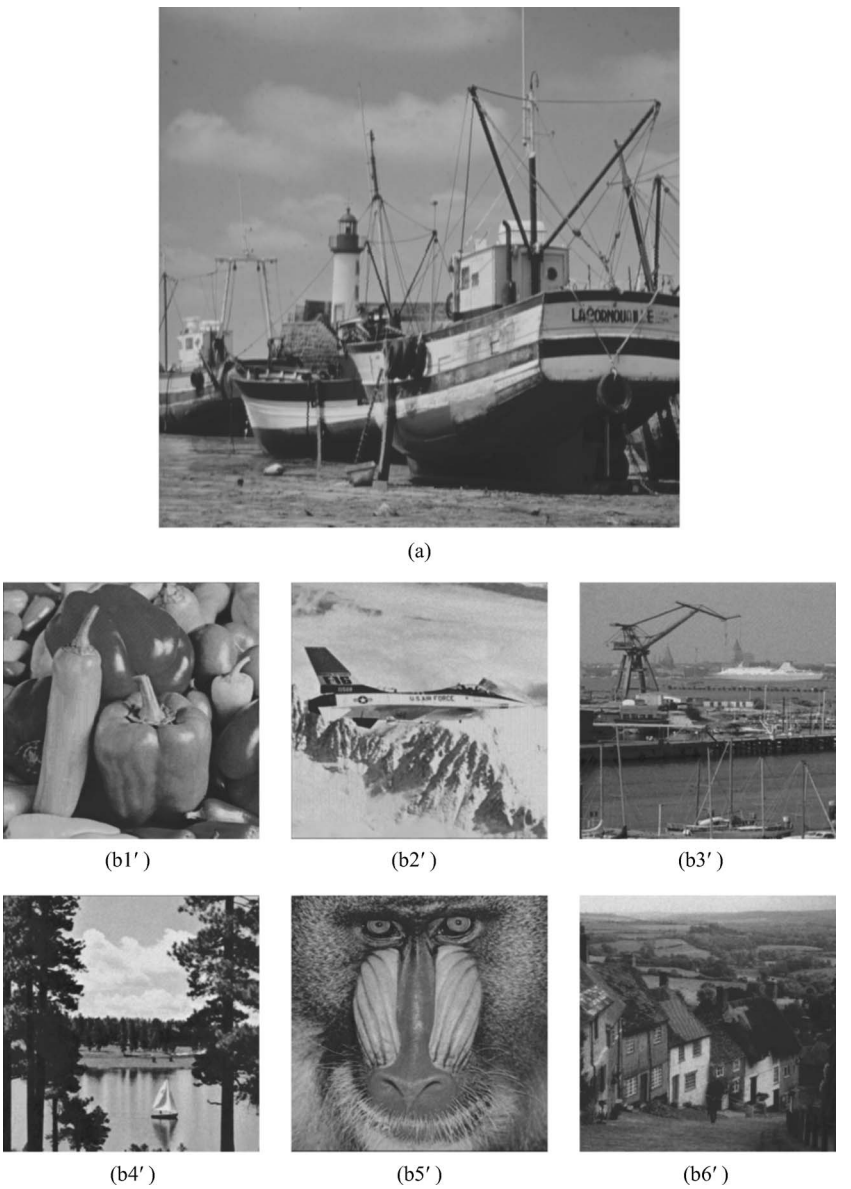



**Fig. 6** (a) The original Boat image of size 512×512. (b1′)–(b6′) The six corresponding 256×256 stego images after hiding.

one pixel). Therefore, our shadows are also robust against the differential attack.

## 6 Comparisons with Ref. 6

The procedure detailed in Ref. 6 has no progressive ability. In fact, the two papers are for two distinct groups of customers. Ref. 6 is for people transmitting top-secret images, while the current paper is for people who need to transmit an image from unstable channels, or for a company whose images are a bit confidential (but not top-secret) and need to be processed on a daily basis by employees of different ranks at different security levels. As a result, there are at least the following two *progressive* applications of the current paper that cannot be done by Ref. 6: Application 1, used in a company's daily meeting (see paragraph 3 of Section 1); and Application 2, to transmit an image in an unstable, long-delay environment (see paragraph 4 of Section 1).

Shadows in the current paper have a more economic size and, hence, more chances to survive in the recovery meeting if communication channels are stable for only a short period of time. In fact, as discussed in the fifth paragraph of Section 5, to view the secret image, the total size of the noisy shadows is 28.74%, 38.33%, or 47.91% here for the $r_L=3$, $r_M=4$, or $r_H=5$ example, respectively, which is much smaller than the without-hiding total space $r\times(1/r)=1=100\%$ needed in Ref. 6.

Note that the current paper is not just a simple extension of Ref. 6. All we used from Ref. 6 is the share-generating polynomial [Eq. (3), modified from Eq. (1)]; besides this equation, Ref. 6 has no DCT, no mod-17 transform, no

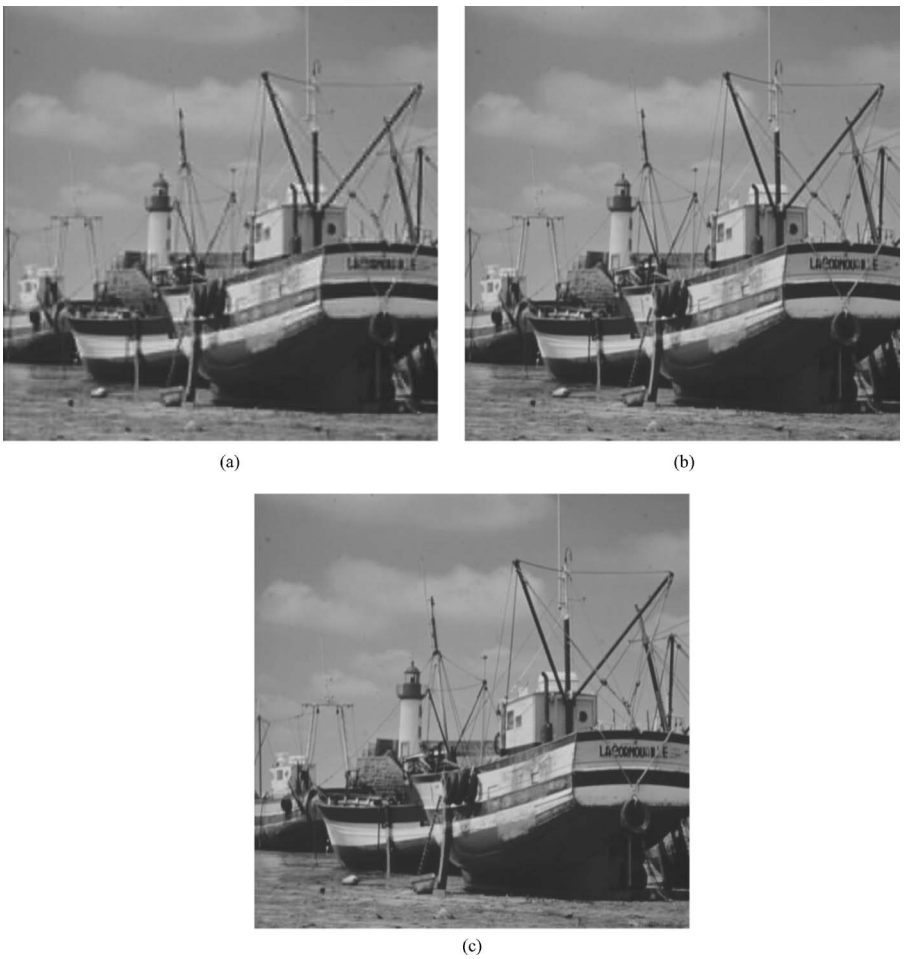


**Fig. 7** The reconstructed Boat image of (a) low, (b) low+middle, and (c) low+middle+high frequencies. The PSNRs are 28.18, 32.83, and 37.64 dB, respectively. Note that (a) is from any 3 of the 6 images in Fig. 6(b1')–(b6'); (b) is from any 4; and (c) is from any 5.

band partition table (Fig. 2 and Table 1), and no joint shares. All these components are needed to build up the current progressive method.

A more convenient version to do inverse sharing is also inserted in Sec. 4. It bypasses the Lagrangian polynomials approach used in Ref. 6, instead using a direct multiplication approach that is easier to implement.

## 7 Conclusions

We propose a sharing method to recover an image progressively, which can also be utilized to transmit an image from an unstable/unfriendly environment. The transmission of an image uses $n$ stego images that each hides a shadow. The $n$ smaller-size stego images can be transmitted or hidden using $n$ distinct channels to increase the survival rate, and the recovery only cares about the question, "How many stego images have arrived?" rather than the question, "Which stego images have arrived?"

In the encoding, the spatial domain is transformed into the frequency domain by DCT. The $8\times8=64$ frequency coefficients of each $8\times8$ block are partitioned into three bands (low, middle, and high). The three bands are shared separately, and each band generates $n$ shares. The $3n$ generated bands are merged to get $n$ shadows, and each shadow contains information from all three bands. Each of the $n$ shadows looks noisy and is hidden in an ordinary-looking cover image to reduce the chance of being attacked. For each pixel, the gray-value difference between the cover image and the stego image is at most 8; hence, the stego images have no visible artifact after hiding the shadows.

In the decoding, when the receiver receives $r_L$ of the $n$ shares, a low-resolution version of the given image can be reconstructed. When $r_M$ of the $n$ shares are received, a middle-resolution version can be reconstructed. Finally, when $r_H$ of the $n$ shares are received, a high-resolution version can be reconstructed. Note that if a user would like to use more thresholds, for example, four thresholds, then he can partition the 64 frequency coefficients into four bands, rather than the three bands used in this paper. It is not hard to do this modification from the information in Section 3.4 and Fig. 2; the details are omitted to save space.

The experiments show that the stego images (with shadows hidden in them) are of acceptable quality [see Fig. 4 and Figs. 6(b1')–(b6')], and they reduce the chance to attract attackers' attention. In the progressive display, the reconstructed version from $r_L$ collected stego images can reveal a rough sketch [see Fig. 5(a) and Fig. 7(a)]; and then

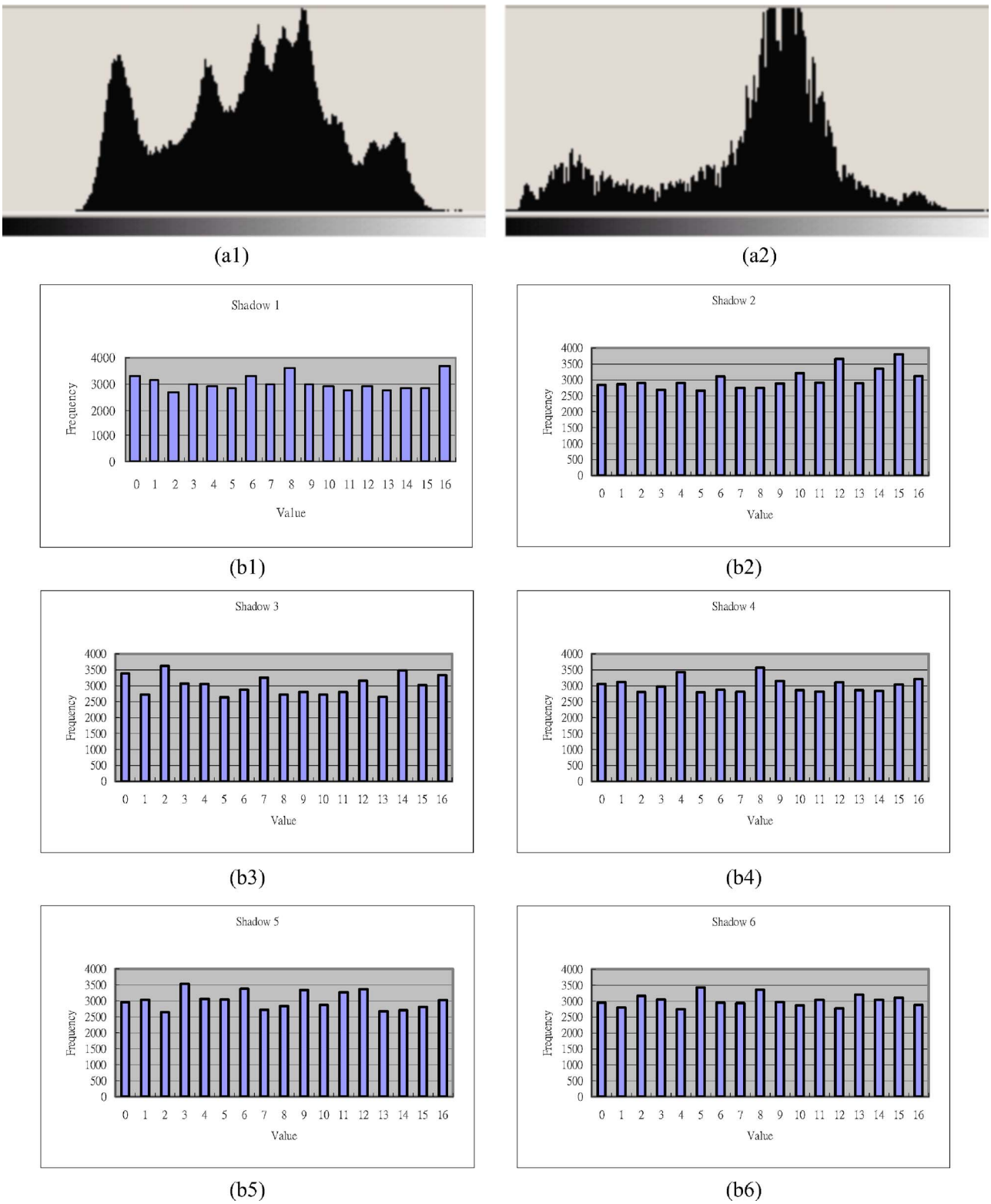



**Fig. 8** Histogram analysis. (a1)–(a2) are, respectively, the histograms of the 512×512 secret images Lena and Boat in Fig. 3(a) and Fig. 6(a). (b1)–(b6) are the histograms of Lena's six generated shadows. (The six histograms of Boat's generated shadows also look like they have a uniform distribution.)

the details appear [see Fig. 5(b) and 5(c) and Fig. 7(b) and 7(c)] as $r_M$ or $r_H$ stego images become available. Furthermore, the size of each stego image is much smaller than the given image (for example, 1:4), so the waiting time to receive a sufficient number of stego images will not be too long when the $n$ stego images are sent from $n$ distinct channels. This smaller-size property also increases the survival rate when the sending channels are in an unstable/

**Table 3** Correlation coefficients $r_{xy}$ of two adjacent pixels in secret image/shadows.

| Direction of Adjacency | Original Image's $r_{xy}$ | Each Shadow Image's $r_{xy}$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ |
| Horizontal | 0.9875 | 0.0040 | 0.0079 | 0.0057 | 0.0092 | 0.0047 | 0.0074 |
| Vertical | 0.9781 | 0.0021 | 0.0036 | 0.0025 | 0.0048 | 0.0015 | 0.0041 |
| Diagonal | 0.9616 | 0.0033 | 0.0051 | 0.0034 | 0.0063 | 0.0032 | 0.0049 |

unfriendly environment, because each channel will not be take too long. Notably, a lossless recovery version is also possible if we allow the stego images to have a larger size. This lossless adaptation, if needed, can be done through using lossless compression to replace the first part of step 2 in the encoding algorithm.

As a final remark, the proposed method has several notable attributes: (1) It is missing-allowable (allowing up to $n-r_L$ stego images get lost); (2) the shadows are equally important, so there is no need to worry about which part is lost or transmitted first; (3) the method is secure (intercepting fewer than $r_L$ stego images cannot reveal the given image); (4) progressive viewing, is allowed; (5) it has improved inverse sharing by skipping the Lagrangian polynomials approach; (6) the shadow size is much smaller than those in Ref. 14; hence, it is more likely to succeed when being transmitted from an unstable/unfriendly environment in which none of the existing channels is reliable for a long period, and the moment when a channel becomes blocked is totally unpredictable; and (7) as discussed in Sec. 5, to view the secret image, the total size of the noisy shadows needed is 28.74%, or 38.33%, or 47.91%; which is also much shorter than the without-hiding total size $r\times(1/r)=1=100\%$ needed in Ref. 6.

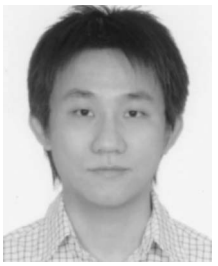
## Acknowledgments

This work was supported by the National Science Council of the Republic of China, under Grant NSC962221-E-009-039. The authors would like to thank the editor and the two reviewers for their valuable suggestions.

## References

1. Nation Bureau of Standards, "Data Encryption Standard," Federal Information Processing Standards Publication 46 U.S. Government Printing Office, Washington, DC (1977).
2. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Comm. Assoc. Comput. Mach.* **21**, 120–126 (1978).
3. A. Shamir, "How to share a secret," *Commun. ACM* **22**(11), 612–613 (1979).
4. G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. AFIPS 1979 National Computer Conference*, **48**, 313–317 (1979).
5. M. Naor and A. Shamir, "Visual cryptography," Advances in Cryptography—EUROCRYPT '94, *Lect. Notes Comput. Sci.* **950**, 1–12 (1995).
6. C. C. Thien and J. C. Lin, "Secret image sharing," *Comput. Graphics* **26**(5), 765–770 (2002).
7. A. Benazza-Benyahia and J.-C. Pesquet, "A unifying framework for lossless and progressive image coding," *Pattern Recogn.* **35**(3), 627–638 (2002).
8. K. L. Chung and S. Y. Tseng, "New progressive image transmission based on quadtree and shading approach with resolution control," *Pattern Recogn. Lett.* **22**(14), 1545–1555 (2001).
9. C. C. Chang, J. J. Jau, and T. S. Chen, "A fast reconstruction method for transmitting image progressively," *IEEE Trans. Consum. Electron.* **44**(4), 1225–1233 (1998).
10. C. C. Chang and R. J. Huang, "Sharing secret images using shadow codebooks," *Inf. Sci. (N.Y.)* **111**(1–4) 335–345 (1998).
11. J. B. Feng, H. C. Wu, C. S. Tsai, and Y. P. Chu, "A new multi-secret images sharing scheme using Lagrange's interpolation," *J. Syst. Softw.* **76**(3), 327–339 (2005).
12. C. C. Thien and J. C. Lin, "An image-sharing method with user-friendly shadow images," *IEEE Trans. Circuits Syst. Video Technol.* **13**(12), 1161–1169 (2003).
13. C. C. Lin and W. H. Tsai, "Secret image sharing with capability of share data reduction," *Opt. Eng.* **42**(8), 2340–2345 (2003).
14. S. K. Chen and J. C. Lin, "Fault-tolerant and progressive transmission of images," *Pattern Recogn.* **38**(12), 2466–2471 (2005).
15. S. J. Lin and J. C. Lin, "VCPSS: A two-in-one two-decoding-options image sharing method combining visual cryptography (VC) and polynomial-style sharing (PSS) approaches," *Pattern Recogn.* **40**(12), 3652–3666 (2007).
16. W. P. Fang and J. C. Lin, "Visual cryptography with extra ability of hiding confidential data," *J. Electron. Imaging* **15**(2), 0230201–0230207 (2006).
17. R. Z. Wang and S. J. Shyu, "Scalable secret image sharing," *Signal Process. Image Commun.* **22**(4), 363–373 (2007).
18. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd edition, Addison-Wesley, Reading, MA (2002).
19. M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," *ACM Trans. Model. Comput. Simul.* **8**(1), 3–30 (1998).
20. C. C. Thien and J. C. Lin, "A simple and high hiding capacity method for hiding digit-by-digit data in images based on modulus function," *Pattern Recogn.* **36**(12), 2875–2881 (2003).
21. C. Y. Yang and J. C. Lin, "Image hiding by base-oriented algorithm," *Opt. Eng.* **45**(11), 117001–(1–10) (2006).
22. R. Z. Wang, C. F. Lin, and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," *Pattern Recogn.* **34**(3), 671–683 (2001).
23. D. C. Wu and W. H. Tsai, "Spatial-domain image hiding using image differencing," *IEE Proc. Vision Image Signal Process.* **147**(1), 29–37 (2000).
24. Y. S. Wu, C. C. Thien, and J. C. Lin, "Sharing and hiding secret images with size constraint," *Pattern Recogn.* **37**(7), 1377–1385 (2004).
25. K. H. Hung, "Progressive image sharing," MS thesis, Department of Computer and Information Science, National Chiao Tung University, Taiwan (2003).
26. G. Chen, Y. B. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals* **21**(3), 749–761 (2004).
27. H. El-din, H. Ahmed, H. M. Kalash, and O. S. Farag Allah, "Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images," *Int. J. Comp., Inform., Syst. Sci. Eng.* **1**(1), 33–39 (2007).
28. H. El-din, H. Ahmed, H. M. Kalash, and O. S. Farag Allah, "Encryption quality analysis of the RC5 block cipher algorithm for digital images," *Opt. Eng.* **45**(10), 107003–(1–7) (2006).

**Kuo-Hsien Hung** received his MS degree in computer and information science from National Chiao Tung University, Taiwan, in 2003. His recent research interests include image processing and secret sharing. He is a member of the Phi-Tau-Phi Scholastic Honor Society.



**Yu-Jie Chang** received his MS degree in computer and information science in 2001 from National Chiao Tung University, Taiwan. He is now a PhD candidate in the Computer Science Department of National Chiao Tung University. His research interests include digital watermarking, image processing, and pattern recognition.

**Ja-Chen Lin** received his BS degree in computer science in 1977 and his MS degree in applied mathematics in 1979, both from National Chiao Tung University (NCTU), Taiwan. In 1988, he received his PhD degree in mathematics from Purdue University, Indiana. From 1981 to 1982, he was an instructor at NCTU. From 1984 to 1988, he was a graduate instructor at Purdue University. He joined the Department of Computer and Information Science at NCTU in August 1988 and became a professor there. His research interests include pattern recognition and image processing. He is a member of the Phi-Tau-Phi Scholastic Honor Society.