Open access • Journal Article • DOI:10.1109/TCSVT.2003.819176

An image-sharing method with user-friendly shadow images — Source link

Chih-Ching Thien, Ja-Chen Lin

Institutions: National Chiao Tung University

Published on: 01 Dec 2003 - IEEE Transactions on Circuits and Systems for Video Technology (IEEE)

Topics: Shadow volume, Shadow, Image processing, Image restoration and Image segmentation

An Image-Sharing Method With User-Friendly Shadow Images

Chih-Ching Thien and Ja-Chen Lin

***Abstract*—This study presents a user-friendly image-sharing method for easier management of the shadow images. The sharing of images among several branches (distributed disks) using the proposed method has several characteristics: 1) fast transmission among branches; 2) fault tolerance; 3) a secure storage system; 4) reduced chance of pirating of high-quality images (as explained in Section V); and 5) most importantly, the provision to each branch manager an easy-to-manage environment (because each shadow image looks like a shrunken version of the original image). The current approach still has the small-size and channel-independent properties of our previous work, , namely, the size of each shadow image is only $1/r$ of that of the original image, and any r shadow images can be used for restoration (the restored image is independent of which r shadow images are used).**

***Index Terms*—Fault tolerance, image sharing, shadow images, user-friendly.**

I. Introduction

THE (r, m) image-sharing method proposed in [1], where $r \leqq m$, divides each input image into m shadow images (also known as shadows). The method exhibits two properties: 1) any r shadow images can be used to restore the input image [2]–[6] and 2) the size of each shadow image is only $1/r$ of that of the input image. In this study, the above properties will be used to develop an image-sharing scheme. The structure and operation of the scheme are described below. The proposed scheme consists of several distributed storage branches, each of which stores one of the m shadow images. When someone needs the original image, each branch can transmit its own shadow image to the receiver, and all branches may transmit simultaneously in parallel. The size of each shadow image is $1/r$ times that of the original image, so the transmission time also decreases. Besides, as stated earlier, a property of sharing is such that only r received shadow images are required to restore the original image. The scheme is inherently fault-tolerant (because $m-r$ channels are allowed to be out of order). This fault tolerance makes the scheme more robust and useful in network transmission. In fact, it handles the crashing of not only some network channels but also storage disks of some branches.

Although the sharing method in [1] can be applied directly to establish the scheme, managing the generated scheme is not easy because the shadow images thus generated [1] look noisy. (For example, see Fig. 1 in which any two of the four shadow images in (a)–(d) can be used to reconstruct the original image (e).) From the viewpoint of a local manager (the manager of a branch), the noise-like shadow images are difficult to identify and manage; that is, they are not user-friendly. The aim of this study is to develop another image-sharing method such that the shadow images look like portraits of the original image. Hence, identifying and managing the shadow images will no longer be difficult for the local manager. The rest of this paper is organized as follows. Section II presents an overview of the proposed method. Sections III and IV then introduce sharing and recovery phases of the proposed image-sharing method, respectively. Section V shows experimental results. Conclusions are drawn in Section VI.

II. Review, Overview, and Notation

First, [1] is reviewed. The method in [1] first divides the original image into several nonoverlapping blocks, each of which have r pixels. For block j, a corresponding shadow pixel value $S_k^{(j)}$ is generated for the jth pixel of each shadow image $k (1 \leq k \leq m)$ where

$$S_k^{(j)} = (P_0 + P_1 k + P_2 k^2 + \ldots + P_{r-1} k^{r-1}) \bmod 251. \quad (1)$$

Here, $P_0 \sim P_{r-1}$ are the gray values of the r pixels of block j. (Notably, the prime number used in [1] for the mod function is 251, which is the prime number closest to and lower than 256 since the gray values are between 0 and 255). Now, consider any specified shadow, say, Shadow 1; clearly, each r-pixel block of the original image only generates one pixel for Shadow 1. Therefore, each shadow is only $1/r$ times the size of the original image.

Now, consider the visual behavior of the shadow images of [1]. Without loss of generality, $r = 2$ and $m = 4$ are considered as an example. A two-pixel block whose gray values are (110, 112) will generate four shadow values—$\mathrm{S}(1) = 222$, $\mathrm{S}(2) = 83$, $\mathrm{S}(3) = 195$, and $\mathrm{S}(4) = 56$ where $S(k) = (110 + 112 \cdot \mathrm{k}) \bmod 251$ is the polynomial used in [1] to generate shadows, as mentioned in the preceding paragraph. The difference between any one of the r input pixel values and any one of the m shadow pixel values can fall anywhere in the range 0∼255. Hence, the gray values of the shadow images (for example, 222, 83, 195 and 56 above) may differ considerably from the input gray values (for example, 110 and 112 above). The shadow images therefore do not resemble the original image at all. (For example, 56 is not close to any of the two gray values 110 and 112 that generate 56). In the current study, an attempt is made to make each shadow image look like the original image

Manuscript received October 17, 2001; revised July 31, 2003. This was supported by the National Science Council, R.O.C., under Grant NSC91-2213-E-009-097. This paper was recommended by Associate Editor H. Zhang.

The authors are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: tgh@cis.nctu.edu.tw; jclin@cis.nctu.edu.tw).

Digital Object Identifier 10.1109/TCSVT.2003.819176

1051-8215/03$17.00 © 2003 IEEE

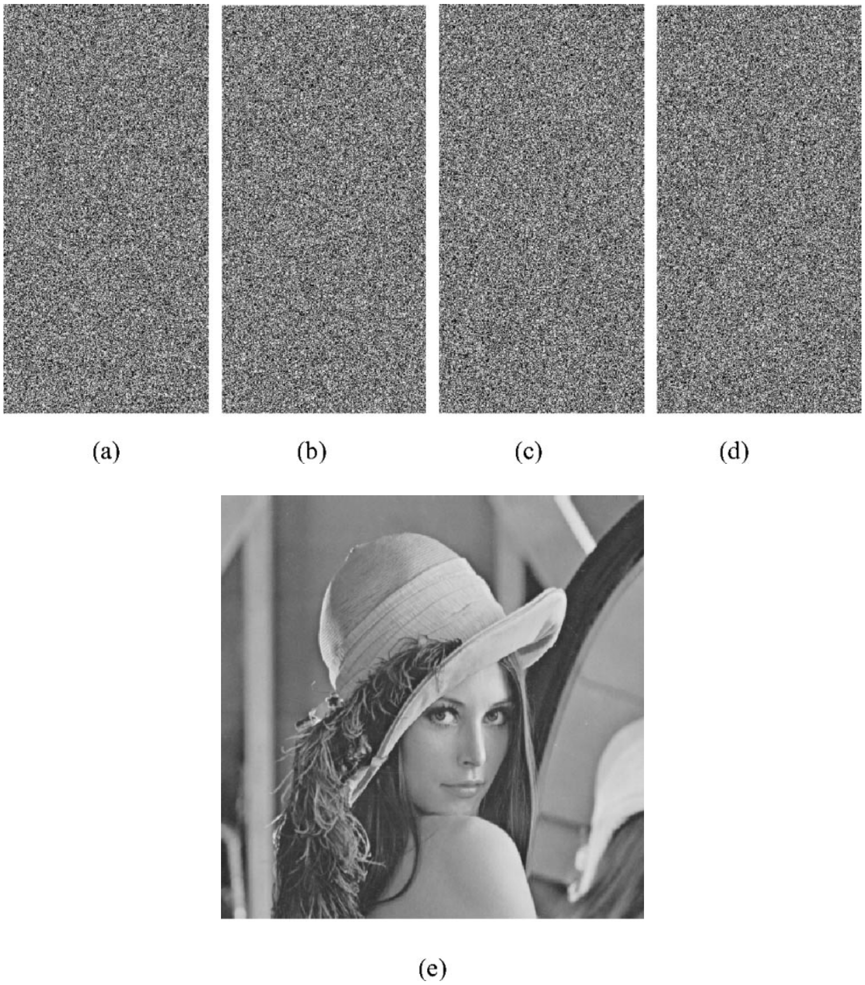

Fig. 1. Results of the method in [1]. (a)–(d) Shadow images of size 256 × 512, generated by the sharing method in [1]. Notably, all the shadow images look like noise images. (e) The 512 × 512 image recovered from any two of the images in (a)–(d).

for user-friendly management, without increasing the size of the shadow image (which remains at $1/r$ times the size of the original image).

A few methods for sharing images have been developed. Chang and Hwang [7] used the vector quantization (VQ) technique to generate a codebook useful for restoring a secret image and then shared the codebook among participants by directly applying the method introduced in [2]. The method in [7] is an elegant method that prevents hackers from seeing secret images. However, their shadows also look like noise and so are not suitable for supporting a user-friendly environment, as required here. Chen and Chang [8] used several processes to create m auxiliary images (the so-called "significant images" in [8]); they used these images as the m shadow images to restore the secret image. Their method is efficient for generating the m shadow images. However, since their method is an (m, m) scheme (in which no shadow image may be lost during restoration), it is not fault-tolerant. Visual cryptography [9] can also be viewed as a type of image-sharing method. It has the advantage that people can see the original image by the superposition of the received shadow images, without the need for complicated computation. However, each shadow image (treated as a transparent sheet in [9]) is of equal size to or larger than that of the original image, rather than of the small size ($1/r$ of the original) obtained using the proposed method.

An overview of the proposed user-friendly image-sharing method is presented below. The encoding phase can be divided into two steps, both of which are applied to each not-yet-processed block until all blocks are processed (assuming that the original image has been divided into nonoverlapping blocks and that each block contains only r pixels). These two steps, which will be detailed in the next section, are as follows.

Step 1) Classification:
The block is classified as a smooth block or a coarse block.

Step 2) Sharing:
The block is encoded according to its type (coarse versus smooth).

Notably, smooth blocks can be recovered completely without any loss, whereas coarse blocks can be recovered with partial loss, which is not severe. (The difference between the original and restored gray values is at most eight for each pixel in the coarse block, as will be seen in (22) later.) The notation used throughout the paper is now defined. The letter P is used to denote the gray values of the full-sized original image, the symbol $\hat{P}$ is used to denote the gray values of the small shadow images, and the symbol $\widetilde{P}$ is used to represent the gray values of the full-sized recovered images. In short,

$$\begin{tikzcd} \text{Original pixels}\{P\} \arrow[r, "\text{encoding}"] & \text{Shadow pixels}\{\hat{P}\} \arrow[dl, "\text{recovering.}"] \\ \text{Recovered pixels}\{\widetilde{P}\} & \end{tikzcd}$$

III. Encoding Phase of a Block

This section details the two steps of the encoding phase.

A. Step 1: Classification

In this step, the just-read-in block is classified as a smooth block or a coarse block. Suppose the gray values of the r pixels in the just-read-in block are $(P_n, P_{n+1}, \ldots, P_{n+r-1})$. Also assume that the previous block has been coded, and so can also be decoded (according to Section IV). Let the decoded gray value of the last pixel in the previous block be $\widetilde{P}_{n-1}$. (If no previous block exists, as at the very beginning of the program, let $\widetilde{p}_{n-1}$ be 0.) Now let

$$\left|P_{\max}-\widetilde{P}_{n-1}\right| = \text{Max}\left\{\left|P_{n+i}-\widetilde{P}_{n-1}\right| \,:\, i=0,\cdots,r-1\right\} \quad (2)$$

where $P_{\max} \in \{P_n, \cdots, P_{n+r-1}\}$ is the gray value of the pixel in the just-read-in block, with the largest difference from $\widetilde{P}_{n-1}$. If $|P_{\max} - \widetilde{P}_{n-1}| > 8$, then the just-read-in block is classified as a coarse block, otherwise the block is a smooth block.

B. Step 2: Sharing

The procedure for sharing the block $B_j = (P_n, P_{n+1}, \cdots, P_{n+r-1})$ depends on its block type (smooth versus coarse). The procedure for a smooth block is described as Tool 2a and that for a coarse block as Tool 2b.

1) Tool 2a—Sharing a Block B_j When it is Smooth: Recall that the gray values of the pixels in the current block B_j are $(P_n, P_{n+1}, \ldots, P_{n+r-1})$, and the (recovered) gray value of the last pixel in the previous block is $\widetilde{P}_{n-1}$. Once the block B_j is classified as smooth, then

$$f_i = P_{n+i} - \widetilde{P}_{n-1} + 8 \quad (i = 0, \cdots, r-1) \qquad (3)$$

is evaluated. A polynomial similar to (1) (the one used in [1]) is then applied (but the prime number 251 used throughout [1] is replaced by 17) to transfer the r input integers $f_0 \sim f_{r-1}$ into m transformed numbers $S_1 \sim S_m$. This procedure can be represented as

$$(f_0, f_1, \cdots, f_{r-1}) \xrightarrow{\text{mod 17 sharing}} \{(1, S_1), (2, S_2), \cdots, (m, S_m)\} \qquad (4)$$

where $S_k = (f_0 + f_1 k + f_2 k^2 + \ldots + f_{r-1} k^{r-1})\text{mod}17$ for each $k \in \{1, \cdots, m\}$. Here, each S_k is associated with its index k. Since the prime number used in the modulus function is 17, the modulus function ensures that all the numbers $S_1 \sim S_m$ are in the range 0∼16. The output pixel value $\hat{P}_{k(j)}$ that the kth shadow image obtained from the current block B_j is defined by

$$\hat{P}_{k(j)} = \widetilde{P}_{n-1} + S_k - 8, \qquad k = 1, \cdots, m. \qquad (5)$$

$\hat{P}_{k(j)}$ is used as the pixel value for painting the current (jth) pixel of the kth shadow image. Since S_k is an integer in the range 0∼16, the difference between $\hat{P}_{k(j)}$ and $\widetilde{P}_{n-1}$ is very small, as stated in Lemma 1 [which can be proven using (5)].

Lemma 1: If the shadow gray value $\hat{P}_{k(j)}$ is derived from a smooth block B_j, then

$$|\hat{P}_{k(j)} - \widetilde{P}_{n-1}| \leq 8.$$

Moreover, the definition of a smooth block implies that the differences between $\widetilde{P}_{n-1}$ and the gray values in the smooth block B_j are also small [≤8; see the definition immediately below (2)]. Accordingly, from Lemma 1, the shadow pixel gray value $\hat{P}_{k(J)}$ resembles all the gray values in block B_j. (The difference is at most $8 + 8 = 16$.)

1) Tool 2b—Sharing a Block B_j when it is coarse: As before, assume that the gray values of the pixels in the current block B_j, which is now a coarse block, are $(P_n, P_{n+1}, \ldots, P_{n+r-1})$ and that the recovered gray value of the last pixel in the previous block is $\widetilde{P}_{n-1}$. Now, let $P_{\max}$ be as defined in (2), and then use the floor and ceiling functions to evaluate

$$P_{\text{base}} = \begin{cases} \left\lfloor \frac{P_{\max}}{17} \right\rfloor \times 17, & \text{if } P_{\max} < \widetilde{P}_{n-1} \\ \left\lceil \frac{P_{\max}}{17} \right\rceil \times 17, & \text{otherwise.} \end{cases} \qquad (6)$$

Notably, P_{base} is a multiple of 17 and is very close to the number $P_{\max}$. Also evaluate $g_0 \sim g_{r-1}$ by the definition

$$g_i = \left[\frac{P_{n+i}}{17}\right] \in N, \qquad i = 0, \cdots, r-1 \qquad (7)$$

where [ ] is the rounding function that rounds a number to its nearest integer. Now, the sharing transformation is applied to convert $g_0 \sim g_{r-1}$ to $E_1 \sim E_m$ as

$$(g_0, g_1, \cdots, g_{r-1}) \xrightarrow{\text{mod 17 sharing}} \{(1, E_1), (2, E_2), \cdots, (m, E_m)\} \qquad (8)$$

where $E_k = (g_0 + g_1 k + g_2 k^2 + \ldots + g_{r-1} k^{r-1})\text{mod } 17$ for each $k \in \{1, \cdots, m\}$. Note that each E_k is again in the range 0∼16 and each E_k is generated using index k.

Now, using the evaluated P_{base} and $E_1 \sim E_m$, the output gray value created for each shadow image can be evaluated. The output gray value $\hat{P}_{k(j)}$ for painting the current pixel (the jth pixel) of the kth shadow image is given by

$$\hat{P}_{k(j)} = \left\{ \begin{matrix} P_{\text{base}} - E_k, & \text{if } P_{\text{base}} < \widetilde{P}_{n-1} \\ P_{\text{base}} + E_k, & \text{otherwise} \end{matrix} \right\}, \qquad k = 1, \cdots, m. \qquad (9)$$

The difference between $\hat{P}_{k(j)}$ and $\widetilde{P}_{n-1}$ exceeds 8 (as proven by Lemma 2), which property is crucial in the recovery phase (Section IV). Also note that this property ($|\hat{P}_{k(j)} - \widetilde{P}_{n-1}| > 8$ for the shadow image pixel $\hat{P}_{k(j)}$) is very similar to the requirement $|P_{\max} - \widetilde{P}_{n-1}| > 8$ for the original image pixel $P_{\max}$ of the coarse block B_j.

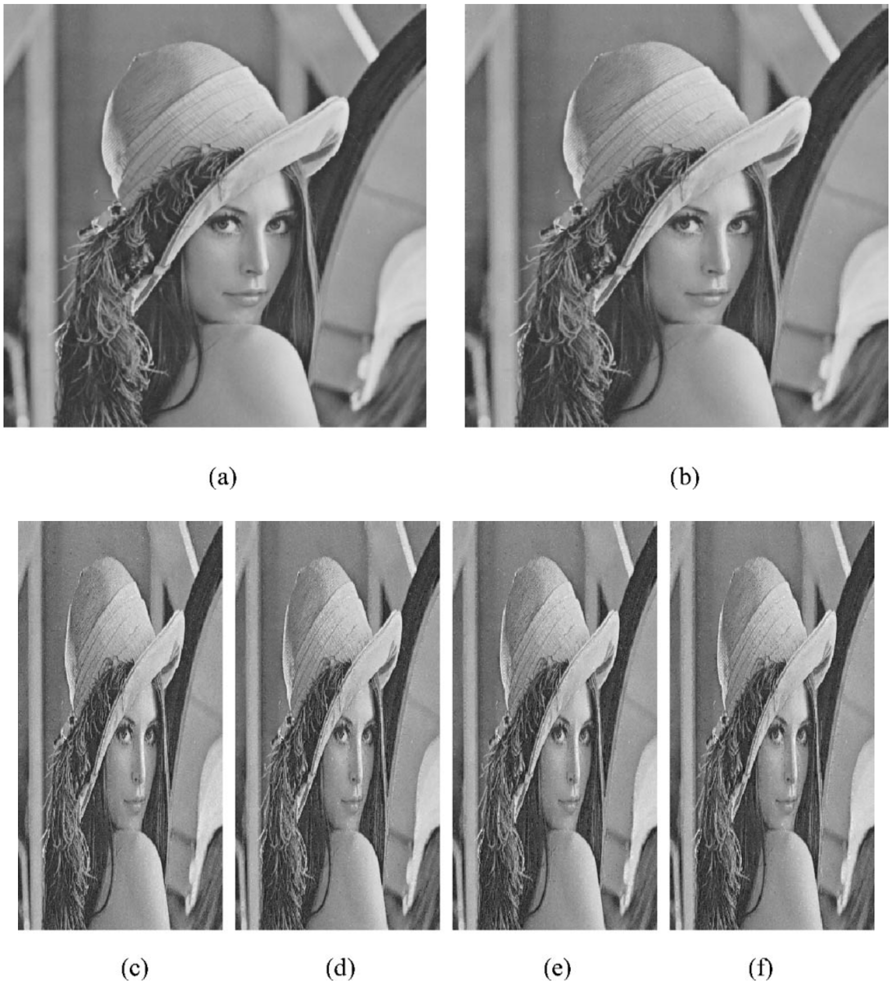

Fig. 2. Implementation of the proposed method. (a) A 512 × 512 input image. (b) Image recovered from "any" two of the images in (c)–(f). (c)–(f) Shadow images whose sizes are all 256 × 512 bytes.

Lemma 2: If the shadow gray value $\hat{P}_{k(j)}$ is derived from a coarse block B_j, then

$$\left|\hat{P}_{k(j)} - \widetilde{P}_{n-1}\right| > 8.$$

Proof: If $\hat{P}_{k(j)}$ is derived from a coarse block, then $|P_{\max} - \widetilde{P}_{n-1}| > 8$ [see (2)].

Case 1) $P_{\max} - \widetilde{P}_{n-1} < -8$ (i.e., $P_{\max} < \widetilde{P}_{n-1}$).
From (6), $0 \leq P_{base} \leq P_{\max} < \widetilde{P}_{n-1}$. Therefore,

$$\begin{aligned} \hat{P}_{k(j)} - \widetilde{P}_{n-1} &= (P_{base} - E_k) - \widetilde{P}_{n-1} && (10)\\ &\leq P_{\max} - E_k - \widetilde{P}_{n-1} && (11)\\ &\leq P_{\max} - \widetilde{P}_{n-1} && (12)\\ &< -8. \end{aligned}$$

Equation (10) follows from (9), while (11) follows from $P_{\text{base}} \leq P_{\max}$. Finally, (12) follows from $E_k \geq 0$.

Case 2) $P_{\max} - \widetilde{P}_{n-1} > 8$ (i.e., $P_{\max} > \widetilde{P}_{n-1}$).
From (6), $0 \leq \widetilde{P}_{n-1} < P_{\max} \leq P_{\text{base}}$. Therefore,

$$\begin{aligned} \hat{P}_{k(j)} - \widetilde{P}_{n-1} &= (P_{base} + E_k) - \widetilde{P}_{n-1} && (13)\\ &\geq P_{\max} + E_k - \widetilde{P}_{n-1} && (14)\\ &\geq P_{\max} - \widetilde{P}_{n-1} && (15)\\ &> 8. \end{aligned}$$

Equation (13) follows from (9), while (14) follows from $P_{\max} \leq P_{\text{base}}$. Finally, (15) follows from $E_k \geq 0$.

End of proof.

This section is summarized below as an encoding algorithm.

Encoding algorithm

```
Step 0) Divide the original image into
      nonoverlapping blocks of $r$ pixels
      each.
Step 1) (Classification)
        Read in a block. Use (2) to clas-
      sify the just-read-in block $j$ as a
      smooth or coarse block.
Step 2) (Sharing)
        If the block is smooth, then go to
      2a; else go to 2b.
  Case 2a. (Smooth block)
```

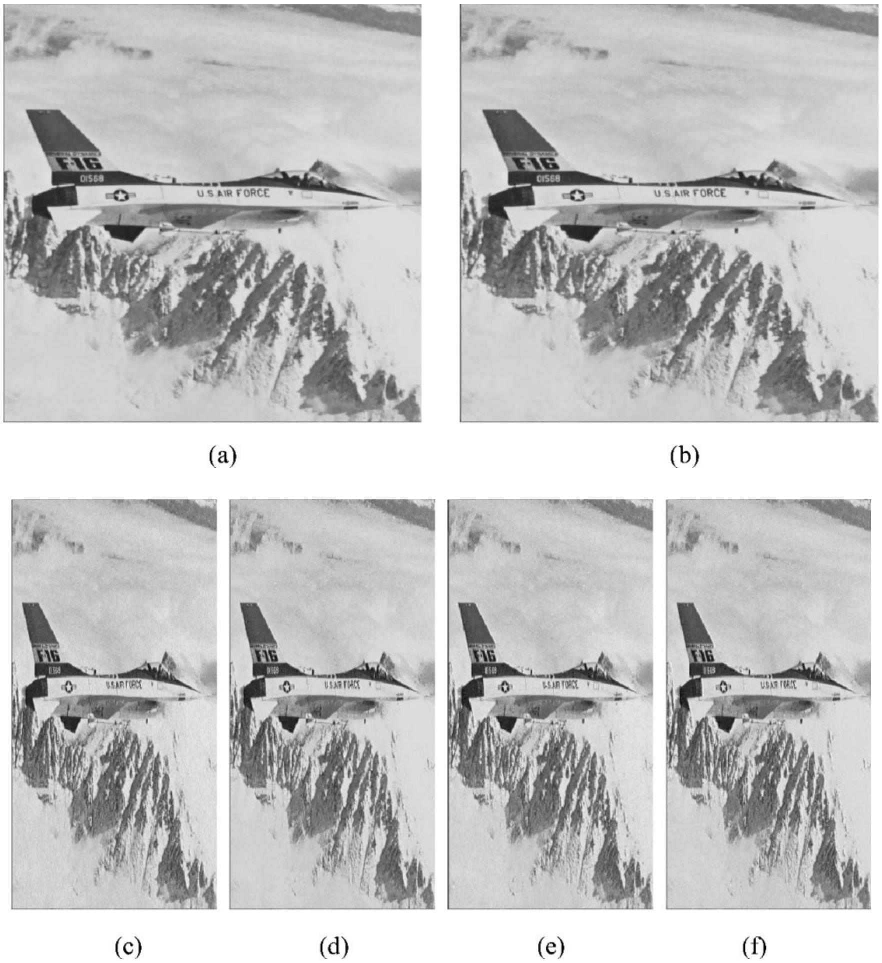

Fig. 3. Similar to Fig. 2, except that Lena is replaced by Jet. (a) A 512 × 512 input image. (b) Image recovered from "any" two of the images in (c)-(f). (c)-(f) Shadow images whose sizes are all 256 × 512 bytes.

```
          Evaluate $(f_0, f_1, \cdots, f_{r-1})$ by (3);
        then share $f_0 \sim f_{r-1}$ using (4)
        to get the transformed numbers
        $S_1 \sim S_m$. Use (5) to obtain the
        gray values $\hat{P}_{1(j)} \sim \hat{P}_{m(j)}$ to paint
        the $j$th pixel of the $m$ shadow
        images.
  Case 2b. (Coarse block)
          Evaluate $P_{\mathrm{base}}$ and $(g_0, g_1, \cdots, g_{r-1})$
        using (6) and (7), respec-
        tively; then share $(g_0, g_1, \cdots, g_{r-1})$
        using (8) to get the trans-
        formed numbers $E_1 \sim E_m$. Sub-
        stitute the obtained values of
        $P_{base}$ and $E_1 \sim E_m$ into (9) to
        get the gray values $\hat{P}_{1(j)} \sim \hat{P}_{m(j)}$
        to paint the $j$th pixel of the $m$
        shadow images.
Step 3. Repeat steps 1–2 (until all blocks
      have been processed).
```

IV. Recovery Phase of Block B_j

Assume that r out of m shadow images have been received. (More precisely, assume that the first j pixels of each of these r shadows have been received.) Let $\{I_i\}_{i=1}^{r}$ be the indices of these r images. (For example, if the r available shadow images are $\{\mathrm{Image1}, \mathrm{Image3}, \mathrm{Image4}, \cdots, \mathrm{Image}r+1\}$, then,

$$I_1 = 1, I_2 = 3, I_3 = 4, \ldots, I_r = r + 1.$$

Assume that the previous block B_{j-1} has been recovered and that the recovered gray value of the final pixel in the previous block is $\widetilde{P}_{n-1}$. (Let $\widetilde{P}_{n-1} = 0$ if no previous block exist.) Now, to recover the current block B_j, and to generate r gray values $\widetilde{P}_n \sim \widetilde{P}_{n+r-1}$ for this r-pixel block, the next not-yet-used pixel (the jth pixel) from each of the r shadow images (one pixel per shadow image) is read in. Let $\hat{P}_{i(j)}$ be the read-in gray value associated with the jth pixel of the shadow image whose index is I_i. Then, the block must be classified. According to Lemmas 1 and 2, either $|\hat{P}_{i(j)} - \widetilde{P}_{n-1}| \leq 8$ for all $i = 1 \sim r$, or, $|\hat{P}_{i(j)} - \widetilde{P}_{n-1}| > 8$ for all $i = 1 \sim r$. Hence, if $|\hat{P}_{1(j)} - \widetilde{P}_{n-1}| > 8$, then the block B_j is classified as a coarse block; otherwise, it is a smooth block. The block type then determines the procedure.

1) Case a (Smooth Block): If block B_j is a smooth block, then use the following procedure to recover the block. Let

$$S_i = \hat{P}_{i(j)} - \widetilde{P}_{n-1} + 8, \qquad i = 1, \cdots, r \qquad (16)$$

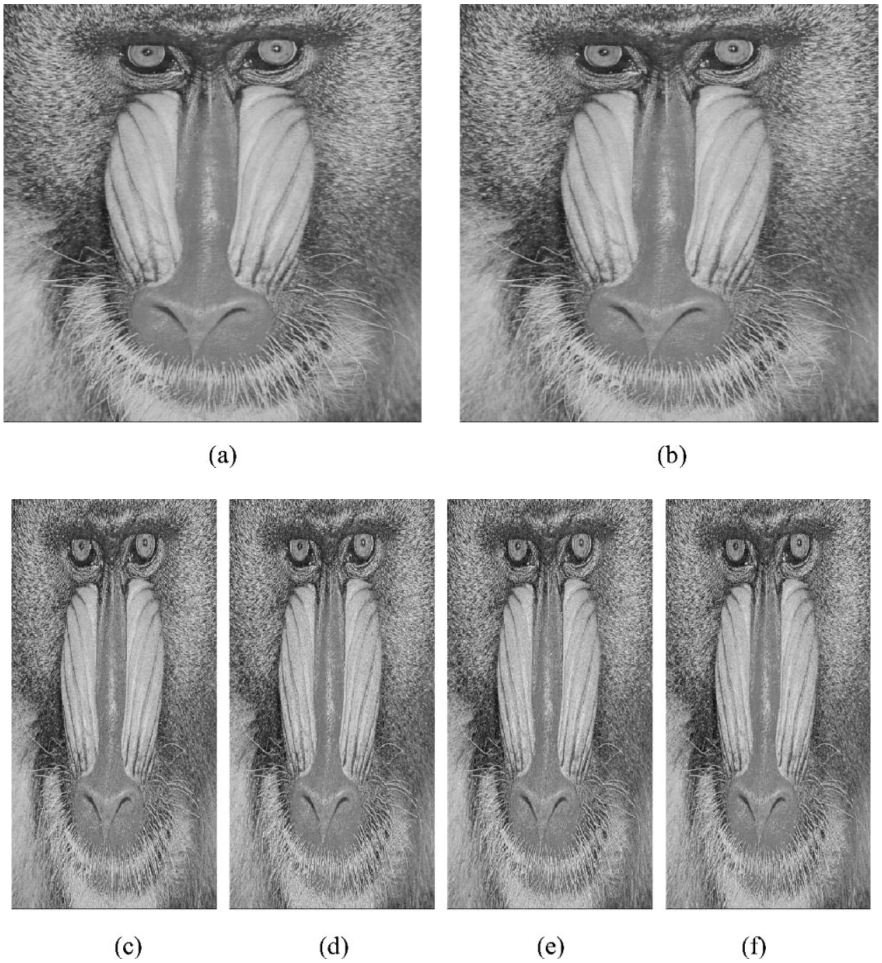

Fig. 4. Similar to Fig. 2, except that Lena is replaced by Monkey. (a) A 512 × 512 input image. (b) Image recovered from "any" two of the images in (c)–(f). (c)–(f) Shadow images whose sizes are all 256 × 512 bytes.

and then evaluate the numbers $f_0 \sim f_{r-1}$ using

$$(f_0, f_1, \cdots, f_{r-1}) \xleftarrow{\text{Inverse of mod 17 sharing}} \{(I_1, S_1), (I_2, S_2), \cdots, (I_r, S_r)\}\,. \quad (17)$$

(This is a numerical problem of finding an interpolation polynomial $q(k) = (f_0 + f_1 k + f_2 k^2 + \ldots + f_{r-1} k^{r-1}) \bmod 17$ such that $q(I_i) = S_i$ where $i = 1 \sim r$. The solution can be found using, say, Lagrange polynomials. [See [2], [7] or any book on numerical methods.]) Finally, based on (3), P_{n+i} can be recovered using

$$P_{n+i} = f_i + \tilde{P}_{n-1} - 8, \qquad i = 0, \cdots, r-1. \quad (18)$$

Notably, the gray values $(P_n, P_{n+1}, \ldots, P_{n+r-1})$ of the smooth block can be recovered without any loss. (In contrast, as explained below, a coarse block can only be recovered using a distorted value $\widetilde{P}_n \sim \widetilde{P}_{n+r-1}$, although the distortion in that case is still small.)

2) *Case b (Coarse Block):* If block B_j is coarse, then use the following procedure to recover the block. For each $i \in \{1, \cdots, r\}$ evaluate

$$E_i = \begin{cases} \left\lceil \frac{\hat{P}_{i(j)}}{17} \right\rceil \times 17 - \hat{P}_{i(j)}, & \text{if } \hat{P}_{i(j)} < \tilde{P}_{n-1} \\ \hat{P}_{i(j)} - \left\lfloor \frac{\hat{P}_{i(j)}}{17} \right\rfloor \times 17, & \text{otherwise} \end{cases} \quad (19)$$

and then evaluate the numbers $g_0 \sim g_{r-1}$ using

$$(g_0, g_1, \cdots, g_{r-1}) \xleftarrow{\text{Inverse of mod 17 sharing}} \{(I_1, E_1), (I_2, E_2), \cdots, (I_r, E_r)\}. \quad (20)$$

Finally, based on (7), compute

$$\tilde{P}_{n+i} = g_i \times 17, \quad i = 0, \cdots, r-1 \quad (21)$$

and use $(\widetilde{P}_n, \widetilde{P}_{n+1}, \cdots, \widetilde{P}_{n+r-1})$ as the recovered gray values of the block $B_j = (P_n, P_{n+1}, \cdots, P_{n+r-1})$. Notably, the difference between the recovered gray value $\widetilde{P}_{n+i}$ and the gray value P_{n+i} of the coarse block (of original image) is small. In fact,

$$|P_{n+i} - \tilde{P}_{n+i}| \le 8 = \left\lfloor \frac{17}{2} \right\rfloor \quad (22)$$

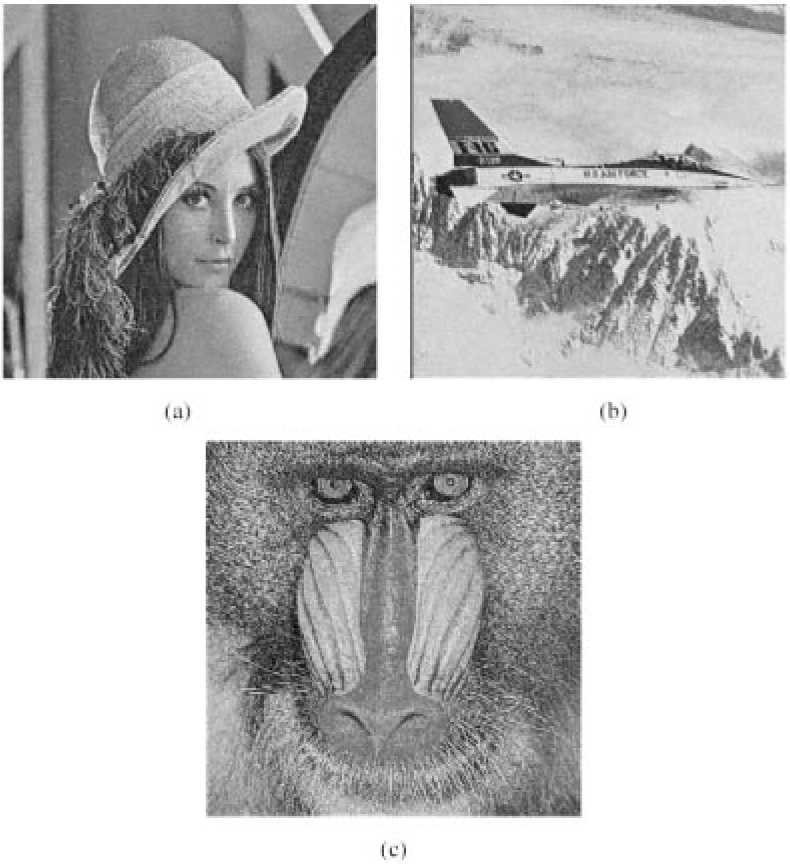

Fig. 5. Direct expansion of the shadow images. (a)–(c) Expanded from Figs. 2(d), 3(d), and 4(d), respectively. The value of r is only 2. (A greater r would make these images look even worse).

because (7) and (21) imply $\widetilde{P}_{n+i} = 17\, g_i = 17[(P_{n+i}/17)]$ is the "nearest multiple of 17" to the number P_{n+i}. (The nearest multiple of 17 to any given number is at most $\lfloor 17/2 \rfloor = 8$ units away from that number.)

This section is summarized below as a decoding algorithm.

Decoding algorithm

```
Step 0) Receive $r$ available shadow images
        to recover the original image. Let
        $j = 1$.
Step 1) (Classification)
          To recover Block $B_j$, read in the
        $j$th pixel value $\hat{P}_{i(j)}$ for each shadow
        $i = 1 \sim r$. Let $\widetilde{P}_{n-1}$ be the recovered
        gray value of the final pixel of the
        previous block $B_{j-1}$. ($\widetilde{P}_{n-1} = 0$ if no
        previous block.) If $|\hat{P}_{1(j)} - \widetilde{P}_{n-1}| > 8$,
        then $B_j$ is coarse; otherwise $B_j$ is
        smooth.
Step 2) (Inverse sharing) Go to $2a'$ if $B_j$
        is smooth; else go to $2b'$.
          Case $2a'$. (Smooth block)
            Evaluate $S_1 \sim S_r$ using (16);
          then implement the inverse
          sharing procedure ((17)) to obtain
          $(f_0, f_1, \cdots, f_{r-1})$. Then, use (18) to
          obtain the $r$ recovered pixels
          $(P_n, P_{n+1}, \ldots, P_{n+r-1})$ without loss for
          the $r$-pixel block $B_j$.
          Case $2b'$. (Coarse block)
            Evaluate $E_1 \sim E_r$ using (19);
          then implement the inverse
          sharing procedure ((20)) to
          obtain $(g_0, g_1, \cdots, g_{r-1})$. Now, directly
          multiply each $g_i$ by 17 $(i = 0, \ldots, r-1)$
          to obtain the $r$ recovered pixels
          $(\widetilde{P}_n, \widetilde{P}_{n+1}, \cdots, \widetilde{P}_{n+r-1}$ for the $r$-pixel
          block $B_j$.
Step 3) $j \leftarrow j + 1$ and go to step 1, unless
        all blocks have been recovered.
```

V. Experimental Results

The ($r = 2$, $m = 4$) image sharing is considered as an example. Fig. 2 shows the results obtained when the input image is Lena. Fig. 2(a) shows the original image whose data size is 512 × 512 bytes. Fig. 2(c)–(f) present the four shadow images, the size of each of which is 256 × 512 bytes. Notably, each shadow image looks like a portrait of the original image (a). Fig. 2(b) displays the image recovered by "any" two of the four shadow images; for example, the image in (b) obtained from the set {(c), (d)} is identical to that obtained from the set {(e), (f)}. The quality of the recovered image is evaluated by the

TABLE I
PSNR AND MSE OF THE RECOVERED IMAGES AND THE EXPANDED SHADOW IMAGES

	The recovered images (as in Fig. 2(b))		The expanded shadow images (as in Fig. 5.)	
	MSE	PSNR	MSE	PSNR
Jet (2,*m*)	6.61	39.93	225.91	24.59
Jet (4,*m*)	9.99	38.14	583.27	20.47
Lena (2,*m*)	10.36	37.98	247.00	24.20
Lena (4,*m*)	15.25	36.30	582.06	20.48
Monkey (2,*m*)	19.08	35.33	632.04	20.12
Monkey (4,*m*)	22.76	34.56	1387.60	16.71

peak signal-to-noise ratio (PSNR), which is defined as PSNR= $10\log_{10}(255^2/MSE)$. For an $n \times n$ image, the mean-square error (MSE) is defined as MSE= $(1/n^2)\sum_{i=1}^{n}\sum_{j=1}^{n}(p_{ij} - \widetilde{p}_{ij})^2$, where p_{ij} is the original pixel value while $\widetilde{p}_{ij}$ is the recovered pixel value. The PSNR of the recovered image in (b) is 37.98 dB, and the visual quality of that image is also good. The good quality of the recovery follows from that fact smooth blocks are lossless, so information loss only occurs in coarse blocks, while human vision is less sensitive to distortion in a coarser area [10], [11]. (Even in the coarse blocks, the distortion of gray values is very small, no more than eight in the gray value for each pixel, as determined by (22) in Section IV.) The experiment performed to obtain Fig. 2 was repeated to process the images Jet and Monkey, yielding Figs. 3 and 4, respectively, which support similar observations. Some readers might wonder why the (smaller) shadow images should not just be expanded into a full size replacement of the original image. Fig.5 (a)–(c) shows the images derived by performing a direct expansion to 512 × 512 of the 256 × 512 shadow images in Figs. 2(d), 3(d), and 4(d). The expanded shadow images look like the original images, but the quality is not good enough (24.20, 24.59, and 20.12 dB, respectively) for the practical applications. Table I lists the PSNR and MSE of the recovered and expanded shadow images. The recovered images have the higher PSNR (34.56∼39.93 dB), and the expanded shadow images have the lower PSNR (16.71∼24.59 dB).

Generally, the recovered images [such as the one in Fig. 2(b)] can be used to replace the original images [such as the one in Fig. 2(a)], while the shadow images of lower quality (and smaller image size) are suitable for managing (easier to identify and smaller) in a branch. In Table I, the quality advantage of the images recovered by the proposed over the directly expanded shadow images becomes more obvious as the r in the (r, m) system increases. (For example, the PSNR advantage for Jet (2, m) is 39.93 versus 24.59, while the PSNR advantage for Jet (4, m) is 38.14 versus 20.47). Consequently, using a larger r might be more suitable when a local manager (the manager of one of the m branches) is strictly forbidden to sell a shadow image on the black market, because the expanded shadow image will have a poor quality and thus attract no customers.

The method in [1] is appropriate for special applications that require lossless restored images, while the proposed method facilitates management since the shadows look like portraits of the original image. Moreover, the shadows of the proposed method might be more efficiently coded if the shadow images need to be compressed further to save storage space or transmission time.

VI. Concluding Remarks

This paper presents a user-friendly image-sharing method that facilitates management of shadow images. The proposed approach has several characteristics: 1) fast transmission (the transmission time is only $1/r$ times the period needed to transmit a full-size image); 2) fault tolerance is maintained (because $m - r$ channels are allowed to be out of order); 3) a secure storage environment (because some branches are allowed to crash); 4) reduced chances of pirating of high-quality images (since r shadow images (each of lower quality) are required to reveal a high-quality image); and 5) the provision to each local manager of an easy-to-manage environment, as stated in the next paragraph.

The proposed user-friendly image-sharing method causes the shadow images to look like portraits of original images and the size of each shadow image to be only $1/r$ times that of the original images. Hence, the identification and management of the shadow images becomes easy for a local manager. Notably, a tradeoff exists between the shadow image's size and the quality of the recovered image in the (r, m) system. A smaller r corresponds to recovered images of higher quality, whereas a larger r corresponds to smaller shadow images and, hence, shortened transmission time.

The proposed method may be used by a parent company with several branches in various cities. In general, the storing of every image by the parent company is dangerous (since any damage to the parent company may destroy some images permanently) and also depends on a very large storage space. Storing the images among all branches helps to prevent nonrecoverable damage, and reduces the amount of storage space at the parent company. Meanwhile, as explained in Section V, the local manager of a branch is less likely to pirate and sell shadow images on the black market because the expanded shadow images are of poor quality, especially if a larger r is used. (For example $r = 4$ makes the Jet and Lena images have PSNR values of nearly 20.5 dB, and Monkey has a PSNR of nearly 16.7 dB).

Recently, video on demand (VOD) has become a popular research topic [12]. The direct extension of our method here is not practical for VOD use, however. Therefore, the authors will need several years to adapt the current approach so that it can eventually be extended to a VOD system. As a final remark, the proposed method can be applied not only to gray-valued images but also to color images (just apply the method to each of the three components of the color images, then combine the results). Some experiments have been performed on color images; color shadow images have been generated, and color images have been reconstructed. The shadow images still look like portraits of the input images and the reconstructed color images are also of high quality. To save space, the figures showing the experimental results of sharing color images are not included here.

Acknowledgment

The authors wish to thank the three referees whose comments have greatly improved this paper.

References

[1] C. C. Thien and J. C. Lin, "Secret image sharing," *Computers & Graphics*, vol. 26, pp. 765–770, 2002.

[2] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[3] R. Ahlswede and I. Csiszàr, "Common randomness in information theory and cryptography—part I: secret sharing," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1121–1132, Aug. 1993.

[4] D. R. Stinson, "Decomposition constructions for secret-sharing schemes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 118–124, Jan. 1994.

[5] A. Beimel and B. Chor, "Universally ideal secret-sharing schemes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 786–794, May 1994.

[6] ——, "Secret sharing with public reconstruction," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1887–1896, Sept. 1998.

[7] C. C. Chang and R. J. Hwang, "Sharing secret images using shadow codebooks," *Inform. Sciences*, vol. 111, no. 1–4, pp. 335–345, 1998.

[8] T. S. Chen and C. C. Chang, "New method of secret image sharing based upon vector quantization," *J. Electron. Imaging*, vol. 10, no. 4, pp. 988–997, 2001.

[9] D. Stinson, "Visual cryptography and threshold schemes," *IEEE Potentials*, vol. 18, pp. 13–16, Jan. 1999.

[10] J. F. Delaigle, C. De Vleeschouwer, and B. Macq, "Watermarking algorithm based on a human visual model," *Signal Processing*, vol. 66, no. 3, pp. 319–335, 1998.

[11] C. I. Podilchuk and W.Wenjun Zeng, "Image-adaptive watermarking using visual models," *IEEE J. Select. Area Commun.*, vol. 16, pp. 525–538, Apr. 1998.

[12] L. Golubchik, J. C. S. Lui, and M. Papadopouli, "A survey of approaches to fault tolerant design of VOD servers: techniques, analysis and compression," *Parallel Computing*, vol. 24, pp. 123–155, 1998.

Chih-Ching Thien was born in 1975 in Taiwan, R.O.C. He received the B.S. degree in computer and information science from National Chiao Tung University, R.O.C., in 1997. He is currently working toward the Ph.D. degree at the same university.

His recent research interests include data hiding, image compression, and information security.

Mr. Thien is a member of Phi Tau Phi.

Ja-Chen Lin was born in 1955 in Taiwan, R.O.C. He received the B.S. degree in computer science and the M.S. degree in applied mathematics, both from National Chiao Tung University, Taiwan, in 1977 and 1979, respectively, and the Ph.D. degree in mathematics from Purdue University, West Lafayette, IN, in 1988.

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, 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 Phi Tau Phi.