

Hybrid Pixel-Based Data Hiding and Block-Based Watermarking for Error-Diffused Halftone Images

Soo-Chang Pei, *Fellow, IEEE*, and Jing-Ming Guo

Abstract—A low computational complexity noise-balanced error diffusion (NBEDF) technique is proposed for embedding watermark into error-diffused images. The visual decoding pattern can be perceived when two or more similar NBEDF images are overlaid each other, even in a high activity region. Furthermore, with the modified improved version of NBEDF, the two halftone images can be made from two total different gray-tone images and still provide a clear and sharp visual decoding pattern. With the self-decoding techniques, we can also decode the pattern by only one NBEDF image. However, the NBEDF method is not so robust to the damage due to printing or other distortions. Thus, a kernels-alternated error diffusion (KAEDF) technique is proposed. We find that the two well-known kernels proposed by Jarvis *et al.* and Stucki are very compatible by alternately using them in halftone process. In the decoder, because the spectral distribution of Jarvis and Stucki kernels are different in the 2-D fast Fourier transform domain, we use the cumulative squared Euclidean distance criterion to determine each cell in a watermarked halftone image either belonging to Jarvis or Stucki, and then decode the watermark. Furthermore, because of the detailed textures of Jarvis and Stucki patterns are somewhat different in spatial domain, the lookup table (LUT) technique is also used for fast decoding. From the simulation results, the correct decoding rates using both techniques are high and extremely robust, even after printing and scanning processes. Finally, we extend the hybrid NBEDF and KAEDF algorithms to two color EDF halftone images, where eight independent KAEDF watermarks and 16 NBEDF watermarks can be inserted and still achieve a high-quality result.

Index Terms—Digital watermark, dot diffusion, error diffusion, halftone, least-squares, ordered dither.

I. INTRODUCTION

DIGITAL halftoning [1] is a technique for changing gray-level images into two-tone binary images. These halftone images can resemble the original images when viewing from a distance by the low-pass filtering in the human visual system. Halftoning is commonly used in printing books, newspapers, and magazines etc, because these printing processes can only generate two tones, black and white (with and without ink). There are several kinds of halftoning methods, including ordered dithering, error diffusion [2]–[4], least squares [5]–[9],

and dot diffusion [10], [11]. Among these, error diffusion offers good visual quality and reasonable computational complexity.

Digital watermarks have many usages, including protecting ownership of an image, preventing the illegal use without permission, and authenticating an image to examine that it has not been altered. Currently, numerous methods using halftones to embed watermarks have been studied, and they can be divided into two categories. These techniques can be used for printing security documents such as an ID card or currency, as well as confidential documents, and prevent from illegal duplication and forgery by further scanning these documents to digital forms.

Techniques in the first category embed invisible digital data into halftone images, which can be retrieved by scanning and applying some extraction algorithms. These methods include using a number of different dither cells to create a threshold pattern in the halftoning process [12], using vector quantization (VQ) to embed watermarks into the most significant bit or least significant bit (MSB/LSB) of error diffusion images [13], [14], using modified data-hiding error diffusion (MDHED) to embed data into error diffusion images [15], embedding a message into dithering images by using a pair of conjugate halftone screens [16], using smart pair-toggling (DHSPT) to embed data into error diffusion images [17], and adopting intensity and connection selection concepts to put the embedded data in a suitable location [18], coordinating the BCH error-correcting code with data-hiding techniques [19], and authentication scheme based on halftoning and coordinate projection [20].

Methods in the second category embed hidden visual patterns into two or more halftone images such that it can be perceived directly when the halftone images are overlaid each other. These techniques include using stochastic screen patterns [21], conjugate halftone screens [22], and stochastic error diffusion [23]. In [23], making a different phase version of stochastic EDF image is adopted to achieve the data-hiding technique. However, poor contrast of the hidden pattern in the high texture image region was found. For this, we propose an algorithm that is similar in concept to threshold modulation [24] to solve this problem, and the computational complexity is kept relatively low. However, not robust to the printing process is the drawback of the second category.

In this paper, a hybrid method of combining noise-balanced error diffusion (NBEDF) data hiding and kernels-alternated error diffusion (KAEDF) watermarking into one or more error-diffused images is proposed. The hybrid model is further extended to color images, where large amount of information is embedded and still kept good quality result. The technique not only preserves the benefits of fast visual decoding by direct

Manuscript received December 26, 2002; revised March 13, 2003. This work was supported by the National Science Council, R.O.C., under Contract NSC91-2219-E-002-044, and by the Ministry of Education under Contract 89-E-FA06-2-4.

S.-C. Pei is with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10617, R.O.C. (e-mail: pei@cc.ee.ntu.edu.tw).

J.-M. Guo is with the Department of Electronic Engineering, Jin Wen Institute of Technology, Taipei, Taiwan 231, R.O.C. (e-mail: jmguo@seed.net.tw).

Digital Object Identifier 10.1109/TCSVT.2003.815943

perceiving the overlaid halftone images, but also is robust against the printing and scanning processes by block-wise changing the diffused kernels.

The rest of this paper is organized as follows. Section II introduces the proposed pixel-based data hiding with noise-balanced error diffusion. Section III describes block-based digital watermarking technique with KAEDF. The hybrid data hiding and watermarking for error-diffused images is introduced in Section IV. Experimental results are presented in Section V, and final conclusions in Section VI.

II. PIXEL-BASED DATA HIDING WITH NOISE-BALANCED ERROR DIFFUSION

We now describe how to embed the hidden visual pattern into an error-diffused image, which we denote EDF2. The decoding scheme superimposes EDF2 onto the original EDF image, known as EDF1, then the decoded watermark can be perceived directly from the two overlaid EDF images.

The standard EDF flow chart is shown in Fig. 1. Here we define 255 as a white pixel and 0 as a black pixel. The variable $x_{i,j}$ represents the current input pixel value and $x'_{i,j}$ is the diffused error sum added up from the neighboring processed pixels. The variable $b_{i,j}$ means binary output at position (i, j) and the Jarvis' error diffusion kernel $h_{m,n}$ [2] is used here. The variable $v_{i,j}$ is the modified gray output and $e_{i,j}$ is the difference between the modified gray output $v_{i,j}$ and binary output $b_{i,j}$. The relationships of $b_{i,j}$, $v_{i,j}$, and $e_{i,j}$ are described as follows:

$$v_{i,j} = x_{i,j} + x'_{i,j}, \text{ where } x'_{i,j} = \sum_{m=0}^2 \sum_{n=-2}^2 e_{i+m,j+n} \times h_{m,n} \quad (1)$$

$$e_{i,j} = v_{i,j} - b_{i,j}, \text{ where } b_{i,j} = \begin{cases} 0, & \text{if } v_{i,j} < 128 \\ 255, & \text{if } v_{i,j} \geq 128 \end{cases} \quad (2)$$

The size of watermark could be enlarged to fit the size of the EDF image. EDF1 is achieved by processing the original gray-level image with standard error diffusion method, as in Fig. 1. However, EDF2 is somewhat different. For this, we define W_W as the set of locations of all the "white" pixels in the watermark and W_B the set of locations of all the "black" pixels. Similarly, $(EDF1)_W$, $(EDF1)_B$, $(EDF2)_W$, $(EDF2)_B$ are defined as above. If the processing position (i, j) of EDF2 satisfies the conditions: $(i, j) \in W_B$ and $(i, j) \in (EDF1)_W$, then it should be processed with the proposed NBEDF algorithm shown in Fig. 2. Equations (1) and (2) should be modified as follows.

$$v_{i,j} = x_{i,j} + x'_{i,j} - N_B \quad (3)$$

$$e_{i,j} = v_{i,j} - b_{i,j} + N_B. \quad (4)$$

If the processing position (i, j) of EDF2 satisfies the conditions: $(i, j) \in W_B$ and $(i, j) \in (EDF1)_B$, it still be processed with standard EDF, as in (1) and (2). If $(i, j) \in W_W$ and $(i, j) \in (EDF1)_B$, (3) and (4) still be applied. If $(i, j) \in W_W$ and $(i, j) \in (EDF1)_W$, the signs of N_B in (3) and (4) are reversed.

The value of variable N_B in (3) and (4) could be tuned to control the quality of EDF2 and the decoded visual pattern. If the value of N_B is increased, the quality of EDF2 will be degraded

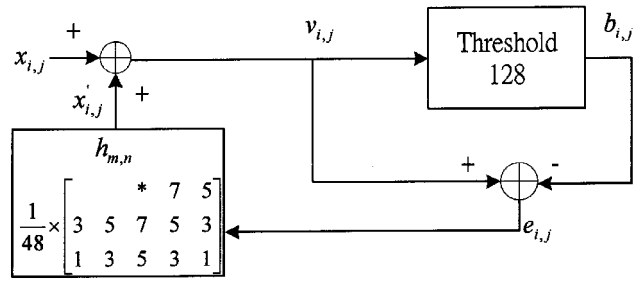


Fig. 1. Standard error diffusion scheme.

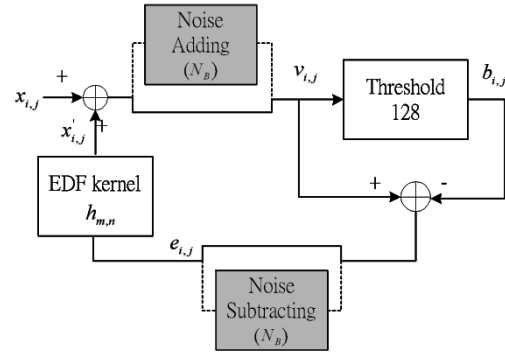


Fig. 2. Pixel-based noise-balanced error diffusion scheme.

and the decoded visual pattern will become clearer; the opposite will be true when the value of N_B is decreased. The additive noise in (3) and (4) not only forces the white pixels in EDF1 to become black in EDF2, but also forces the black pixels in EDF1 to become white in EDF2, thus balancing the visual quality in EDF2. Superimposing EDF1 and EDF2 clearly reveals the black region of the watermark. The most significant difference between the proposed NBEDF and [23] is that in [23] only a different phase version of stochastic EDF2 image is adopted to achieve the data-hiding technique. However the method used here can precisely control the pixels in EDF2 to tend to white or black by tuning the energy of the additive noise N_B even in the high texture image region. The proposed NBEDF algorithm can be further generalized to other applications as follows.

First, we can increase the number of EDF images to three or more, using an algorithm similar to the one described above. Consider encoding watermark into three EDF images as an example, if the processing position (i, j) of EDF3 satisfies the conditions: $(i, j) \in W_B$, $(i, j) \in (EDF1)_W$ and $(i, j) \in (EDF2)_W$, then it should be processed with the proposed NBEDF algorithm. If the processing position (i, j) of EDF3 satisfies the conditions: $(i, j) \in W_B$ and $((i, j) \in (EDF1)_B$ or $(i, j) \in (EDF2)_B)$, then it only needs to be processed with standard error diffusion. If $(i, j) \in W_W$ and $((i, j) \in (EDF1)_B$ or $(i, j) \in (EDF2)_B)$, (2) and (3) still be applied. If $(i, j) \in W_W$ and $((i, j) \in (EDF1)_W$ and $(i, j) \in (EDF2)_W)$, the signs of N_B in (2) and (3) are reversed. In general, when more EDF images are overlaid, the visual pattern will have better contrast. Sometimes the halftone image may suffer some distortions due to printing, dirt, and attack etc, and the visual decoded pattern obtained by overlapping only two images may not be clear enough. The

technique of superimposing more than two NBEDF images can improve this problem.

Second, EDF1 and EDF2 can be made from two totally different gray-tone images. With this NBEDF algorithm, if the additive noise is large enough, the overlaid decoding pattern can still be perceived. However, the contrast of the visual pattern will be reduced, because EDF1 and EDF2 belong to two different gray-tone images. Thus, we should modify the NBEDF algorithm here. Initially, W_B should be expanded uniformly by morphological dilation, as shown in Fig. 3. In this example, the dilation mask size is 3×3 , and defined as $[X_0 X_1 X_2; X_3 X_4; X_5 X_6 X_7]$ in Matlab expression. If the pixel value of the watermark in position (i, j) is X , the new pixel value processed with the 3×3 dilation mask is given by

$$W(i, j) = X \cup P(X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7)$$

$$= \begin{cases} 0, & \text{if } X = 0 \\ 128, & \text{if } X = 255, \\ & X_0 \text{ or } X_1 \text{ or } \dots \text{ or } X_7 = 0 \\ 255, & \text{if } X = X_1 = X_2 = \dots = X_7 = 255 \end{cases} \quad (5)$$

where $P(\bullet)$ represents the union process of elements in parentheses. The dilation mask covers only a small portion of the watermark and processes with the concept of sliding window and orientated with raster scan. The gray region in Fig. 3(b) is obtained by dilation, and we define it as W_G . If the processing position (i, j) of EDF2 satisfies the conditions $(i, j) \in W_G$ and $(i, j) \in (EDF1)_B$, then it should be processed with the modified NBEDF algorithm, and (1) and (2) should be modified as follows:

$$v_{i,j} = x_{i,j} + x'_{i,j} - N_G \quad (6)$$

$$e_{i,j} = v_{i,j} - b_{i,j} + N_G. \quad (7)$$

If the processing position (i, j) of EDF2 satisfies $(i, j) \in W_B$ or $(i, j) \in W_W$, then the processing method is the same as NBEDF and the additive noise is N_B . The modified NBEDF flow chart is shown in Fig. 4. When we overlay EDF1 and EDF2, the black pixels surround the visual pattern will less than usual. With this strategy, we can obtain clearer and sharper visual decoding patterns.

Third, the NBEDF technique can be generalized to a “self-decode” mode. The embedded pattern still can be visually decoded with only one EDF image. There are two extended methods to achieve the self-decoding, as follows.

In the first self-decoding technique, referred to as “replicated self-decoding technique (RSDT)”, the morphological dilation described above is applied to the hidden pattern in the first step. Then, unlike the method used previously, a small square piece in EDF2 is processed with ordinary EDF. The small square EDF image is replicated and tiled to be as the same size as the original gray-tone image, and we take it as “EDF1”. In next step, the same modified NBEDF technique described above is applied to get “EDF2”. When the “EDF1” and EDF2 are superimposed, the visual pattern can clearly be revealed.

In the second self-decoding technique, referred to as “shifted self-decoding technique (SSDT)”, the morphological dilation described above is still applied to the hidden pattern. If (i, j) is the current process position, since the error diffusion is a causal

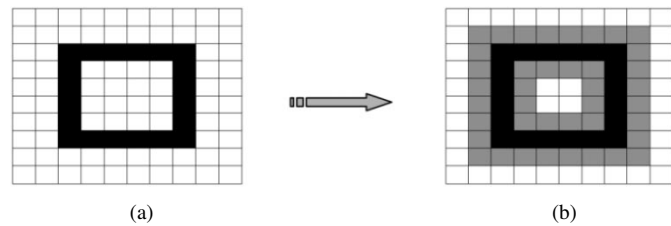


Fig. 3. Watermark processed with 3×3 morphological dilation.

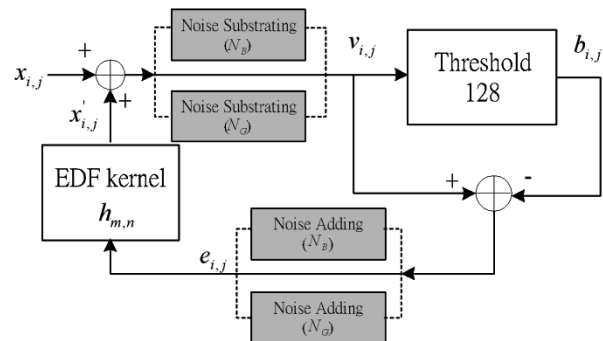


Fig. 4. Modified version of noise-balanced error diffusion scheme.

$$\begin{matrix} & * & 8/42 & 4/42 \\ 2/42 & 4/42 & 8/42 & 4/42 & 2/42 \\ 1/42 & 2/42 & 4/42 & 2/42 & 1/42 \end{matrix}$$

Fig. 5. Stucki error-diffusion kernel ($h_{m,n}$).

process, the pixel at $(i - D_x, j - D_y)$ should have been processed by NBEDF (the variables D_x and D_y stand for the spatial displacement, and both set to 10 in this paper). Now we take the displacement version $(i - D_x, j - D_y)$ as EDF1. In next step, the same modified NBEDF technique described above is applied to get “EDF2”. When the “EDF2” is superimposed with its displacement version $(i - D_x, j - D_y)$, the visual pattern can be clearly perceived.

III. BLOCK-BASED DIGITAL WATERMARKING WITH KAEDF

A. Encoding

The computational complexity of the NBEDF scheme in Section II is low and offers high visual decoding quality. However, it is not robust enough to resist harm caused by printing or other witting attack. For this, we now describe another watermark embedding technique into an EDF image, in which the computational complexity of encoding process is not increased at all. The encoding is similar to [12], where Hel-Or used a pair of dithering cells embedding watermarks into dithered images. However, the method cannot be applied to error diffusion image because the limitation of decoding rule established there. The flowchart of the proposed encoding is the same as that in Fig. 1, but the error diffusion kernel $h_{m,n}$ used here was alternately changed between two different kernels that correspond to the values in the watermark.

First, we should select two different error diffusion kernels. Some well-known kernels are Jarvis–Judice–Ninke [2], Jarvis

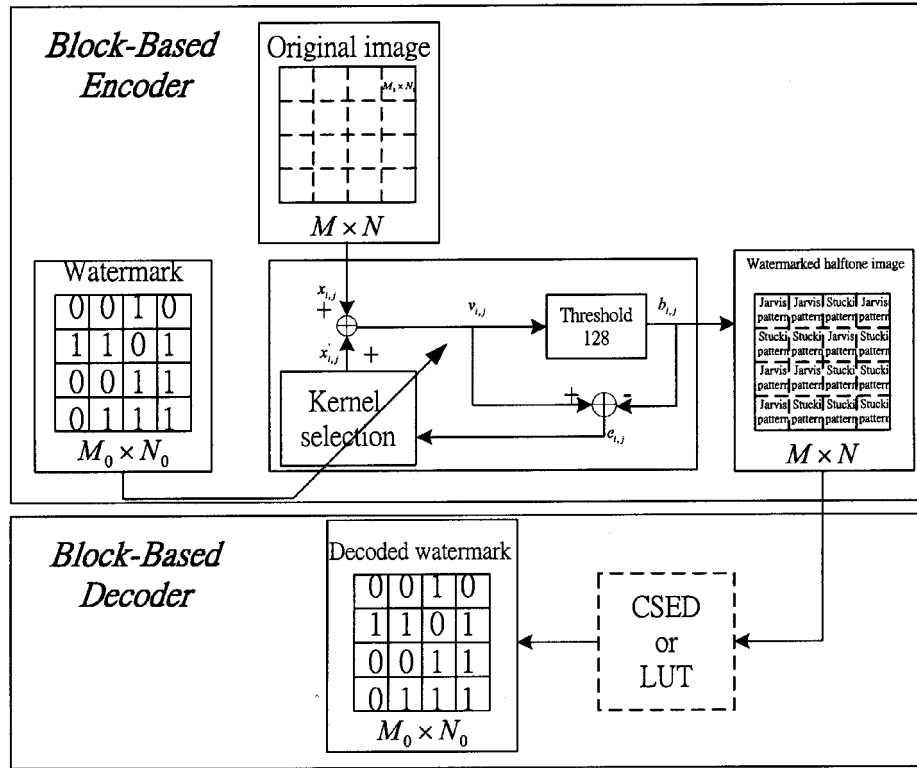


Fig. 6. Codec of the block-based KAEDF.

[3], and Floyd–Steinberg [4], etc. However, not all of these are “compatible” to each other. That is, if we randomly choose two of them to alternately process the gray-tone image, the halftoned image may decrease the quality due to an unnatural blocking effect. From experiments, we find that the Jarvis and Stucki kernels, as shown in Figs. 1 and 5, respectively, are the most compatible. The results in other combinations of these three kernels are not as good as Jarvis and Stucki, and they will be shown in Section V.

The codec flow chart is depicted in Fig. 6. Without loss of generality, the size of the original gray-tone image is $M \times N$, and the watermark is $M_0 \times N_0$. With this definition, the original gray-tone image is divided into several cells and each with size equal to $M_1 \times N_1$, where M_1 is obtained from dividing M by M_0 , and N_1 is obtained from dividing N by N_0 . If the cell corresponding to the value of watermark is 0 (black), then we process it with the Jarvis kernel. If the cell corresponding to the value of watermark is 1 (white), we process it with the Stucki kernel. Note that, to avoid the blocking effect, the amount of errors that are produced in the periphery of a cell should still be diffused to the neighboring cells.

B. Decoding

Here, we propose two decoding techniques. The first one is used in frequency domain, and the decoding criterion is cumulative squared Euclidean distance (CSED). The second one is used in spatial domain, and lookup table (LUT) is used in fast decoding without time consuming fast Fourier transforms (FFTs).

First, by experiments, the energy distributions of halftoned images for each distinct kernel in the 2-D FFT transform

domain are very different, as shown in Fig. 7. Fig. 7 is obtained by transforming the 512×512 Lena image with 2-D FFT. The center of Fig. 7 belongs to the high-frequency components of the halftoned image, and the four corners are the low-frequency components. In Fig. 7, it is clear that the energy of Stucki halftoned image is more concentrated in the center (high frequency region) than Jarvis image. This is because Stucki halftoned image inherently has finer textures. Furthermore, we also tested the other eight different 512×512 images, 8 different 128×128 images, and eight different 16×16 images. These images included the airplane, earth, lake, mandrill, milk, peppers, shuttle, and tiffany images. All of the images were halftoned with Jarvis, Stucki, and Floyd error diffusion kernels and then processed with 2-D FFT. Surprisingly, all of them still have almost the same distribution as the 512×512 Lena image. Since the two error-diffusion kernels used in this paper are Jarvis and Stucki, we concentrated on the difference between these two kernels in the FFT domain.

In the decoder, the halftoned image first divided into several cells with size equals to $M_1 \times N_1$ as defined above. Then each of the cells feeds into the 2-D FFT process. After transformation, we reserve a certain amount of pixels (ten, in this paper) with higher energy in a cell and the rest pixels with lower energy are deleted. After that, the CSED from all the high-energy entries to the center of cell is calculated. The CSED is defined as follows:

$$\text{CSED} = \sum_{i=1}^w \left[\left(x_i - \left(\frac{M_1}{2} \right) \right)^2 + \left(y_i - \left(\frac{N_1}{2} \right) \right)^2 \right] \quad (8)$$

where w is the number of reserved pixels in a cell, (x_i, y_i) represents the coordinate of the i th reserved pixel in a cell, and M_1

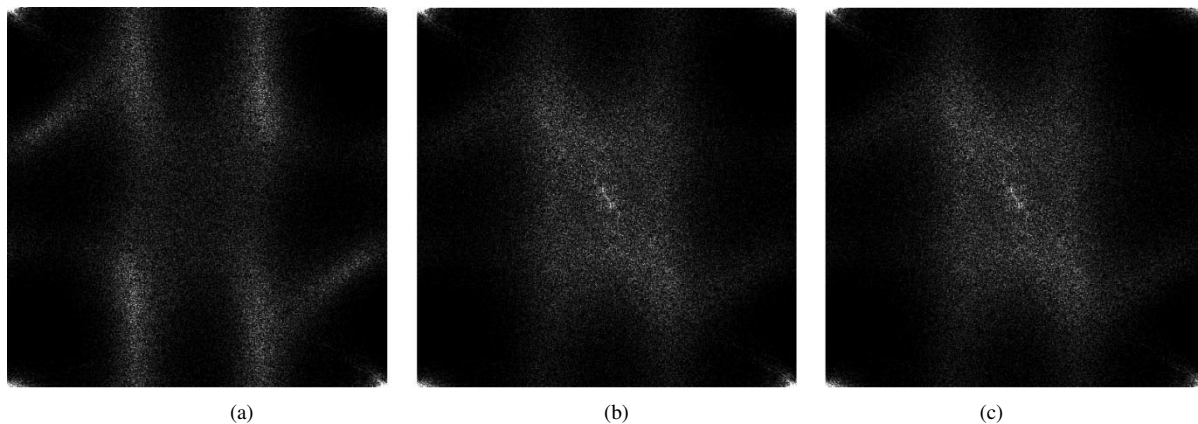
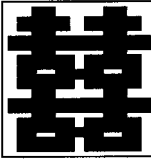
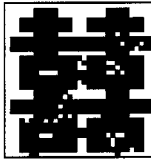


Fig. 7. Energy distributions of three 512×512 error-diffused images in 2-D FFT domain. (a) Jarvis. (b) Stucki. (c) Floyd.

TABLE I
DECODING PERFORMANCE OF THE PROPOSED LUT TECHNIQUE IN ORDERED DITHERING [12] AND KAEDF IMAGES

Matched patten size	Mask [12]	Utility rate	Pattern similarity	Decoding region size	Correct decoding rate	Decoded watermark
4×4	C_0	20.64% (13551)	42.89%	4×4	100%	
	C_1	22.49% (14744)				
(a)						
Matched pattern size	kernel	Utility rate	Pattern similarity	Decoding region size	Correct decoding rate	Decoded watermark
5×5	Jarvis	2.5% (841269)	26.06%	11×11	96.97%	
	Stucki	1.8% (606225)				
(b)						

and N_1 represent the width and length of a cell. Here, we define another threshold T_{JS} to determine whether a cell has been processed by Jarvis or Stucki. If the CSED of a cell is greater than T_{JS} , then we assume that this cell is processed by Jarvis and it represents 0 (black) in relative position of the watermark. Otherwise, we know that the cell is processed by Stucki and it represents 1 (white).

Second, because of the detail textures of Jarvis and Stucki patterns are somewhat different in spatial domain (e.g., Jarvis with coarser textures and Stucki with finer textures), the LUT technique can also be used for fast decoding the KAEDF images without FFT transforms. In [12], Hel-Or used a pair of dithering cells to embed the watermark. The decoding method used there calculating the average color in a halftone cell first, then two different dither cells are used to process the average color cell. Between the two simulated halftone results, choose the one most similar to the original halftone cell and decide which dither cell is used to halftone the particular region of the halftone image. However, the same average color cells need not possess the same “error-diffused halftone dot distribution”

because some error will diffuse from the neighboring cells, and the error amounts are not necessarily the same between two different cells. So the decoding method used in [12] cannot be applied to KAEDF images. On the contrary, the proposed LUT decoding technique is not only successfully be used in KAEDF images, but also the embedded ordered dithering images in [12] is also 100% precisely decoded. Here, we use eight training images described above to establish the LUT. One of the decoding performances of the embedded dithering and error-diffused images are shown in Table I. In Table I, the “matched pattern size” indicates the support region size of a matched pattern, where the matched patterns in a cell can be used to judge whether it is the same as the patterns in Jarvis or Stucki trained look up table. If more matched patterns are similar to the patterns in the Jarvis LUT, then we decide that the cell has been processed by Jarvis or vice versa. The “utility rate” is how many different patterns have been occurred in eight training images divided by the total possible ones (e.g., $2^{4 \times 4}$ for 4×4 pattern size); the “pattern similarity” means how many patterns added up in Jarvis patterns are the

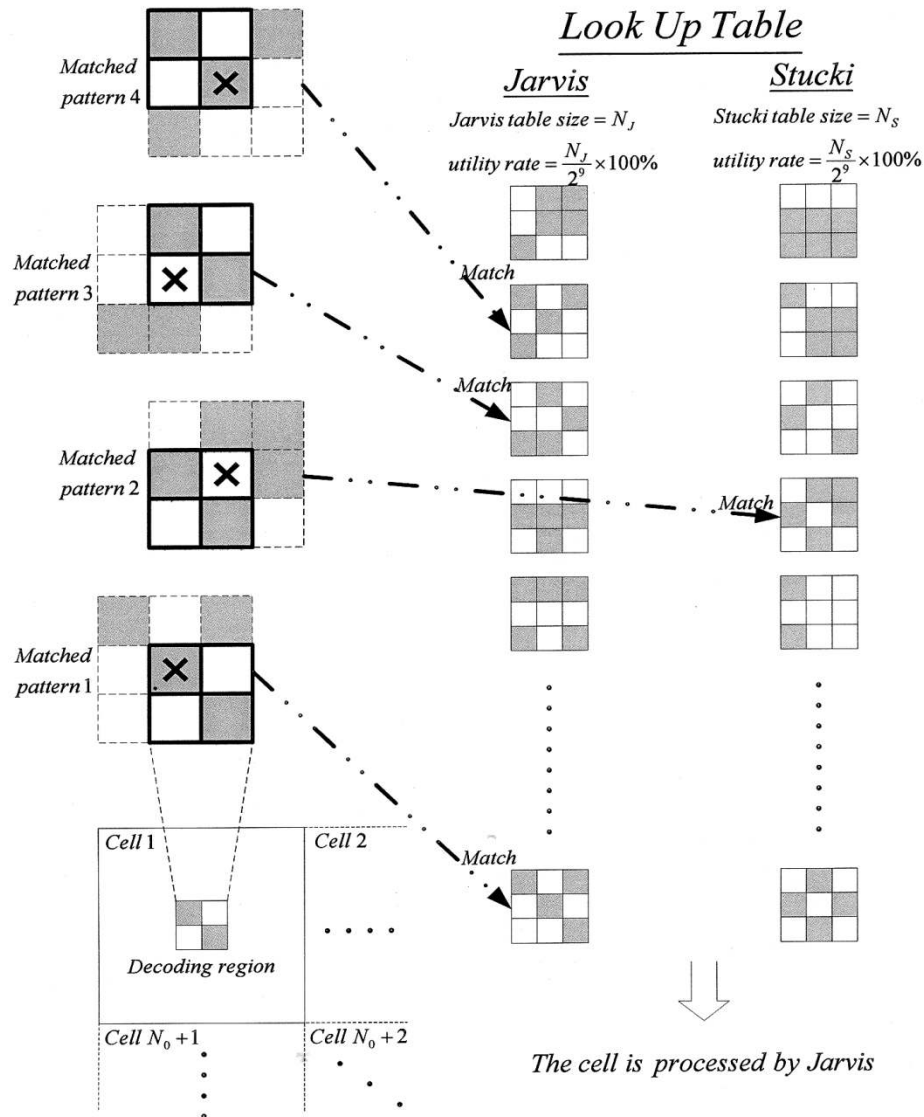


Fig. 8. LUT decoding example with decoding region size 2×2 and matched pattern size 3×3 .

TABLE II
DECODING RATE WITH DIFFERENT DECODING REGION SIZE AND PATTERN SIZE

Decoding region size		5×5	7×7	9×9	11×11	13×13
Matched pattern size	4×4	43.06	48.63	53.8	60.25	63.77
	5×5	86.23	92.48	95.12	95.77 (best)	95.56

same as in Stucki patterns; “decoding region size” means how many matched patterns (these matched patterns can overlap each other, e.g., decoding region size 3×3 for nine matched patterns) are used for majority devoting to determine whether the cell belongs to Jarvis or Stucki. The larger decoding region size offers more matched patterns for judgment, and then it can withstand more attacks or distortions to the embedded halftone images. The decoding region is selected to position at the center of a cell in this paper. Fig. 8 is an example to demonstrate how a cell decoded to Jarvis with decoding region size 2×2 and matched pattern size 3×3 . Furthermore, we should determine the optimum matched pattern size and

decoding region size. In Table II, we list the experimental average decoding rate by different pattern size and decoding region size combinations. We can see that in a matched pattern size of 5×5 and a decoding region size of 11×11 provided the highest decoding rate. So, in this paper, we use these two parameters for fast LUT decoding.

IV. HYBRID PIXEL-BASED DATA HIDING AND BLOCK-BASED WATERMARKING FOR ERROR-DIFFUSED HALFTONE IMAGES

In this section, we introduce how to combine the proposed “NBEDF” and “KAEDF” algorithms. In fact, the two tech-

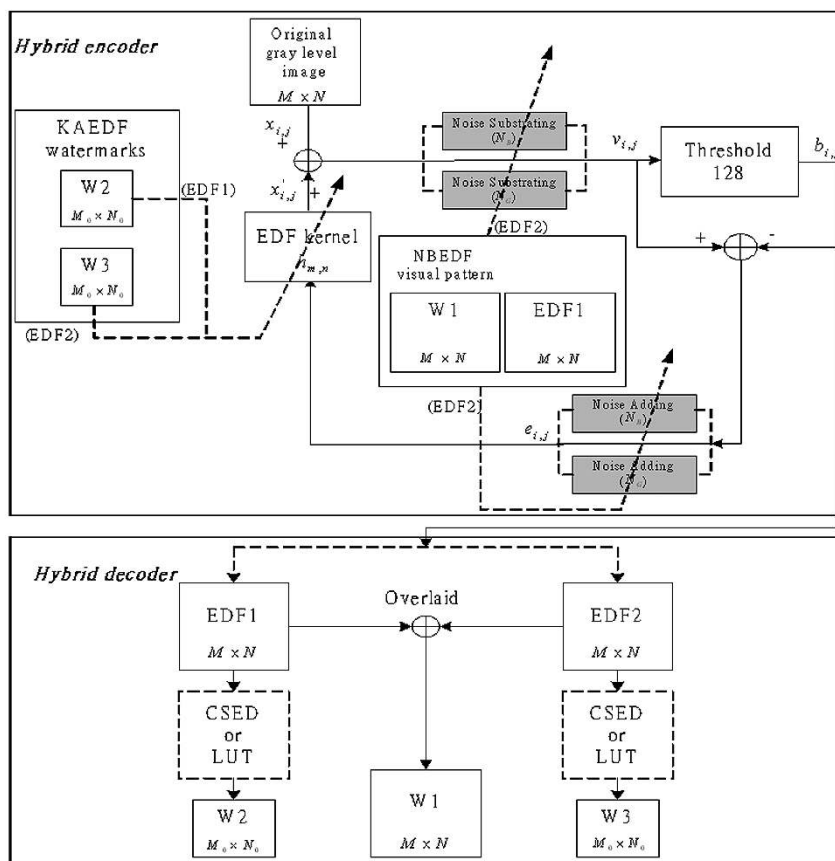


Fig. 9. Codec of the hybrid NBEDF and KAEDF schemes.

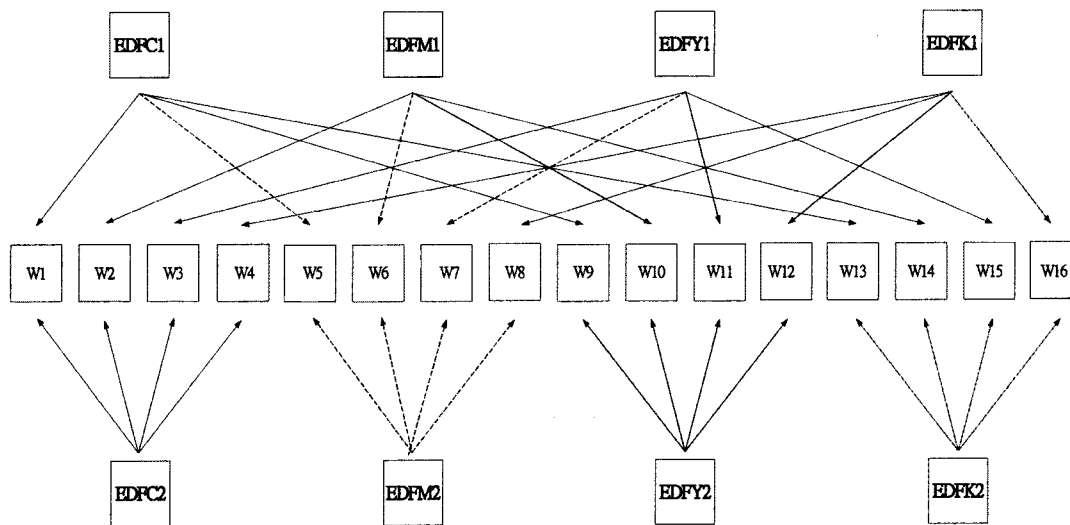


Fig. 10. Synthesized data hiding and watermarking in two color error-diffused images.

niques do not conflict each other at all, as is illustrated in Fig. 9. The EDF1 can be achieved by KAEDF, and EDF2 can be achieved by alternatively changing the diffusion kernel between Jarvis and Stucki to embed a watermark and simultaneously use NBEDF to embed a hidden visual pattern. With this arrangement, not only can a visual pattern (W_1) be decoded visually by overlaying EDF1 and EDF2, but other watermarks (W_2 and W_3) can also be extracted from EDF1 and EDF2 by

the CSED or LUT decoding schemes. Note that, if EDF1 and EDF2 belong to two totally different gray-tone images or the two watermarks W_2 and W_3 are different, then the modified version of NBEDF must be used to provide satisfactory visual decoding quality of W_1 . Results in Section V show that the quality of EDF1 and EDF2 are also good.

We now extend the synthesized algorithm to color error-diffused images, as illustrated in Fig. 10. For printing, the orig-



Fig. 11. Images are printed at 150 dpi and the sizes are equal to 512×512 . (a) Original watermark. (b) Original EDF Lena image. (c) Watermarked Lena image. ($N_F = 20$). (d) Overlaid version of (b) and (c).



Fig. 12. Images are printed at 150 dpi and the sizes are equal to 512×512 . (a) Watermarked Lena image. ($N_F = 20$) (b) Overlaid version of Fig. 11(b) and (c) and Fig. 12(a).

inal RGB color spaces should be transformed to CMYK (cyan, magenta, yellow, and black) color spaces. The RGB-to-CMYK transformation is as follows:

$$C' = 255 - R \quad (9)$$

$$M' = 255 - G \quad (10)$$

$$Y' = 255 - B \quad (11)$$

$$K = \min\{C', M', Y'\} \quad (12)$$

$$C = C' - K \quad (13)$$

$$M = M' - K \quad (14)$$

$$Y = Y' - K \quad (15)$$

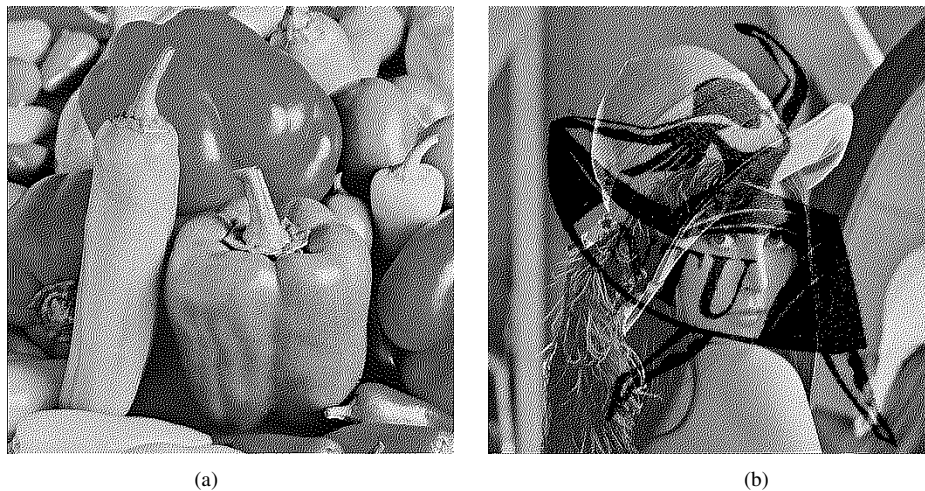


Fig. 13. Images are printed at 150 dpi and the sizes are equal to 512×512 . (a) Watermarked Peppers image; ($N_B = 25$). (b) Overlaid version of Fig. 11(b) and Fig. 13(a).

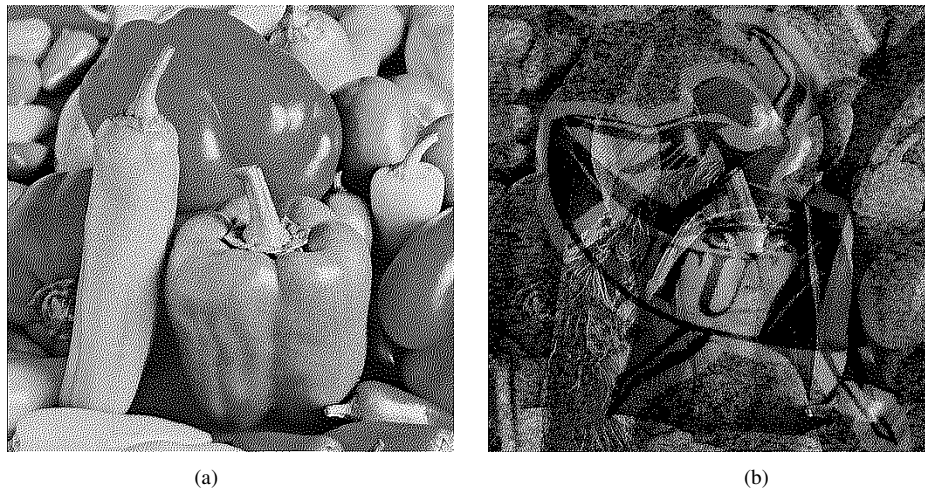


Fig. 14. Images are printed at 150 dpi and the sizes are equal to 512×512 . (a) Watermarked Peppers image ($N_B = 25$, $N_G = 20$, dilation mask size = 31×31). (b) Overlaid version of Fig. 11(b) and Fig. 14(a) with improved contrast.

In Fig. 10, the upper four blocks, EDFC1, EDFM1, EDFY1, and EDFK1 represent the four color spaces of EDF1, and the lower four blocks EDFC2, EDFM2, EDFY2, and EDFK2 represent the four color spaces of EDF2. The middle 16 blocks (W1 to W16) stand for 16 different embedded hidden patterns. From the above discussion, we can apply the KAEDF algorithm to embed 8 different watermarks into the eight color spaces of EDF1 and EDF2, and simultaneously use the NBEDF algorithm to embed 16 different visual patterns into EDF2. The detail NBEDF encoding is described below. First, the morphological dilation is applied to the 16 embedded hidden patterns. If the processing position (i, j) of EDFC2 satisfies the conditions

$$\begin{aligned}
 & ((i, j) \in W1_B \text{ AND } (i, j) \in (EDFC1)_W) \text{ OR} \\
 & ((i, j) \in W2_B \text{ AND } (i, j) \in (EDFM1)_W) \text{ OR} \\
 & ((i, j) \in W3_B \text{ AND } (i, j) \in (EDFY1)_W) \text{ OR} \\
 & ((i, j) \in W4_B \text{ AND } (i, j) \in (EDFK1)_W) \quad (16)
 \end{aligned}$$

then it should be processed with (6) and (7). The variables $W1_G$, $W2_G$, $W3_G$, and $W4_G$ represent the gray regions of W1, W2, W3, and W4, respectively, and the variables $(EDFC1)_B$, $(EDFM1)_B$, $(EDFY1)_B$, and $(EDFK1)_B$

indicate the black regions of EDFC1, EDFM1, EDFY1, and EDFK1, respectively. With this process, each color space of EDF1 superimposes with EDFB2 will produce a different visual decoding pattern. In the same way, EDFG2 and EDFR2 can be achieved with the same procedure. With this arrangement, we can embed a large amount of information into two color NBEDF images. The concept is similar to secret sharing scheme [25]–[27]. Here is an application example. Assume that a bank has 16 vaults that must be opened by 16 different secret keys. The bank employs eight senior tellers, but the bank does not need to trust or rely on any of them individually. Hence, they would like to design a secret sharing system such that any two of the eight senior tellers can open one vault together.

V. EXPERIMENTAL RESULTS

In this section, we apply the proposed NBEDF and KAEDF techniques to demonstrate the performance of our algorithm. The embedded hidden visual pattern for NBEDF is shown in Fig. 11(a). A standard 512×512 Jarvis EDF Lena image is produced, shown in Fig. 11(b). Fig. 11(c) shows the EDF image that was processed with the proposed NBEDF algorithm. The

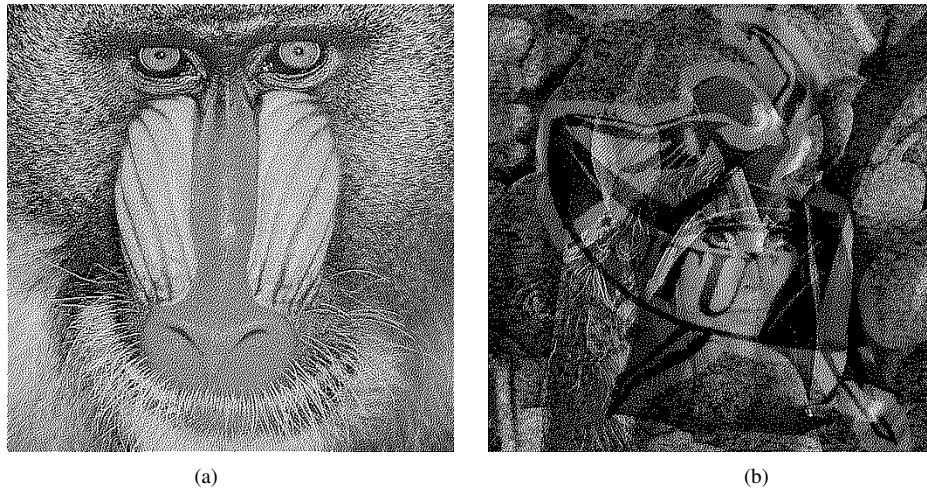


Fig. 15. Images are printed at 150 dpi and the sizes are equal to 512×512 . (a) Watermarked Mandrill image. ($N_B = 40$, $N_G = 30$, dilation mask size = 31×31) (b) Overlaid version by superimposing Fig. 15(a) and "EDF1" that obtained by replicating the black square in Fig. 15(a) then tiled to be as the same size as 15(a).

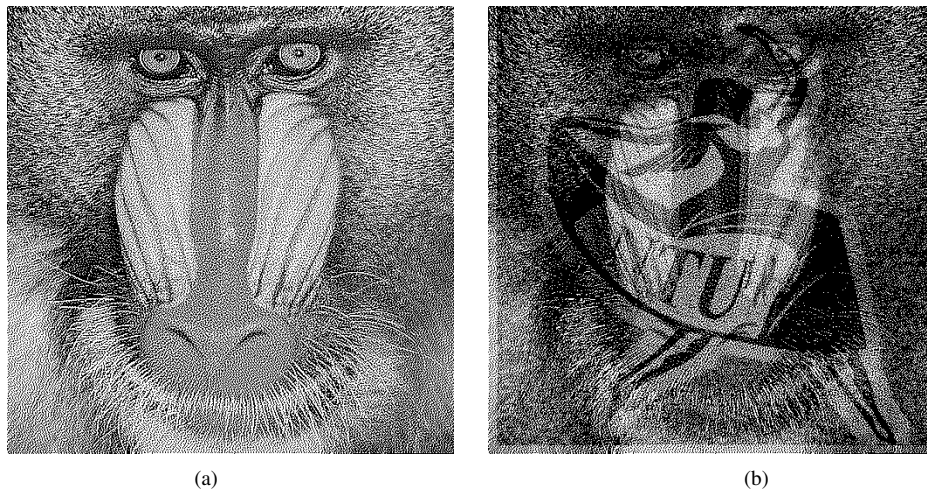


Fig. 16. Shifted self-decoding technique. (a) Embedded Lena image. ($N_B = 40$, $N_G = 30$, dilation mask size = 31×31 , $D_x = 10$, $D_y = 10$). (b) Overlaid version by superimposing Fig. 16(a) and its shifting version.

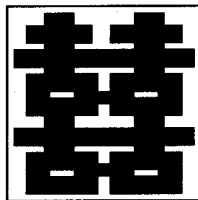


Fig. 17. Watermark with size equal to 32×32 .

balanced noise used here is 20. The superimposed version of Fig. 11(b) and (c) is shown in Fig. 11(d). In Fig. 12(b), we show the result from superimposing one standard EDF and two NBEDF images [Fig. 11(b) and (c) and 2(a)]. In Fig. 12(b), we see that the contrast of visual decoding pattern is much better than in Fig. 11(d).

Fig. 13 shows that if EDF1 and EDF2 were made from two total different gray-tone images, the overlaid visual pattern can still be perceived. Fig. 13(a) shows the NBEDF Peppers image, and Fig. 13(b) is the superimposed halftone version of Fig. 11(b) and Fig. 13(a). The additive noise strength used in Fig. 13(a) is 25. However, the contrast of visual decoding

pattern in Fig. 13(b) is lower than Fig. 11(d). Fig. 14(a) was achieved by processing the gray-tone Peppers image with the modified NBEDF algorithm, and Fig. 14(b) was achieved by overlaying Fig. 11(b) and Fig. 14(a). The values chosen here for N_B and N_G are 25 and 20, respectively, and the dilation mask size is 31×31 . In Fig. 14(b) we can clearly see that the black pixels surround the visual pattern becomes less than in Fig. 13(b) to improve its contrast. Therefore, the visual decoding pattern in Fig. 14(b) is perceived more clearer and sharper than Fig. 13(b).

In Figs. 15 and 16, we show the performance by applying the self-decode technique. The embedded pattern is the same as above. Fig. 15(a) is the embedded NBEDF image with RSDT. Fig. 15(b) is obtained by superimposing Fig. 15(a) with "EDF1", where EDF1 is available by replicating the black square in left bottom of Fig. 15(a) and tiled to be as the same size as Fig. 15(a). Fig. 16(a) is the embedded NBEDF image with the shifted self-decoding technique (SSDT). Fig. 16(b) is obtained by superimposing Fig. 16(a) with its shifted version ($i - D_x$, $j - D_y$), where the variables D_x and D_y used in this paper are fixed as 10 pixels.



Fig. 18. Original EDF images and watermarked KAEDF images with different kernel combinations. (a) Jarvis. (b) Stucki. (c) Floyd. (d) Jarvis and Stucki. (e) Floyd and Jarvis. (f) Floyd and Stucki.

Now we demonstrate the performance of the proposed KAEDF scheme. The 32×32 watermark is shown in Fig. 17. In Fig. 18, we show the three original EDF images and the watermarked version of KAEDF images. Fig. 18 (a)–(c) represents the EDF images that are processed by Jarvis, Stucki, and Floyd, respectively. Fig. 18(d) represents the KAEDF image composed of Jarvis and Stucki kernels. Fig. 18 (e) and (f) are made up by Floyd and Jarvis and Floyd and Stucki, respectively. In Fig. 18,

we find that there are obvious blocking effects in Fig. 18(e) and (f), whereas, this does not occur in Fig. 18(d). This means that the Jarvis and Stucki kernels are inherently very compatible, so in this paper we use this combination for the watermark encoding scheme.

Furthermore, we check if the average cumulative squared Euclidean distance of Jarvis ($ACSED_J$) is always larger than Stucki ($ACSED_S$). Here, we separately use Jarvis and Stucki

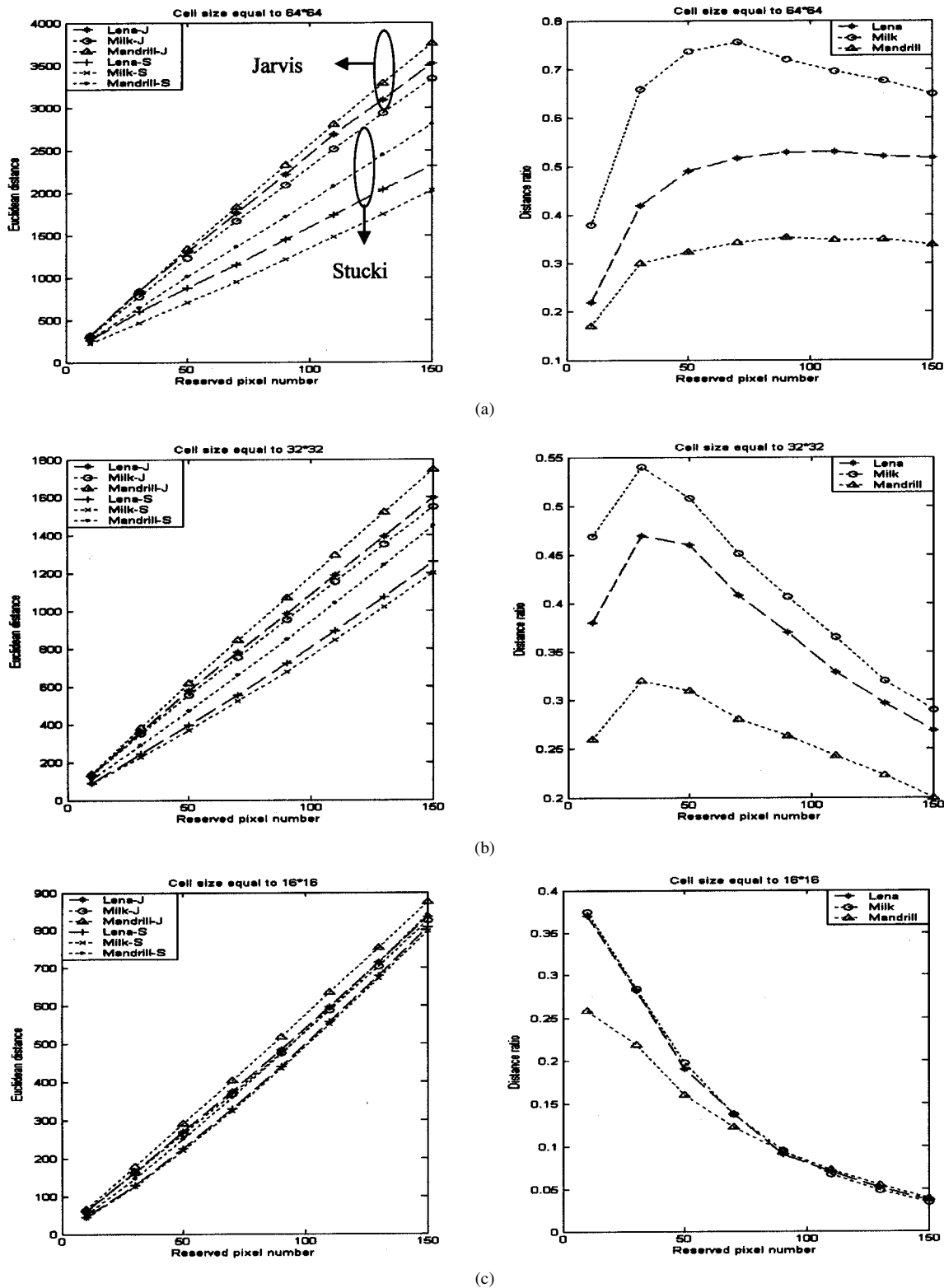


Fig. 19. Average cumulative Euclidean distances (ACED) versus reserved pixel number with three different watermark sizes. Left column: Jarvis and Stucki average cumulative Euclidean distances. Right column: distance ratio $((ACED_J - ACED_S) / ACED_S)$. (a) Cell size 64×64 . (b) Cell size 32×32 . (c) Cell size 16×16 .

kernels to process 8 testing 512×512 images as described above. Then we divide each of the halftoned images into 64×64 , 32×32 , and 16×16 cells, respectively. Each of the cells feeds into 2-D FFT process. The experimental results for three testing images, Lena, Milk, and Mandrill, are shown in Fig. 19.

The horizontal axis represents the number of higher energy entries reserved in a cell. Fig. 19(a)–(c) represents the results from dividing the testing halftone images into 64×64 , 32×32 , and 16×16 , respectively. The left column of Fig. 19 shows the distances $ACSED_J$ and $ACSED_S$, indicating that Jarvis

TABLE III
AVERAGE CORRECT DECODING RATES OF EIGHT TEST IMAGES

Embedded image form	Average decoding rates	
	CSED	LUT
Bitmap	92.16	95.77
Scanned	750 dpi	91.43
	450 dpi	90.82
	150 dpi	88.43

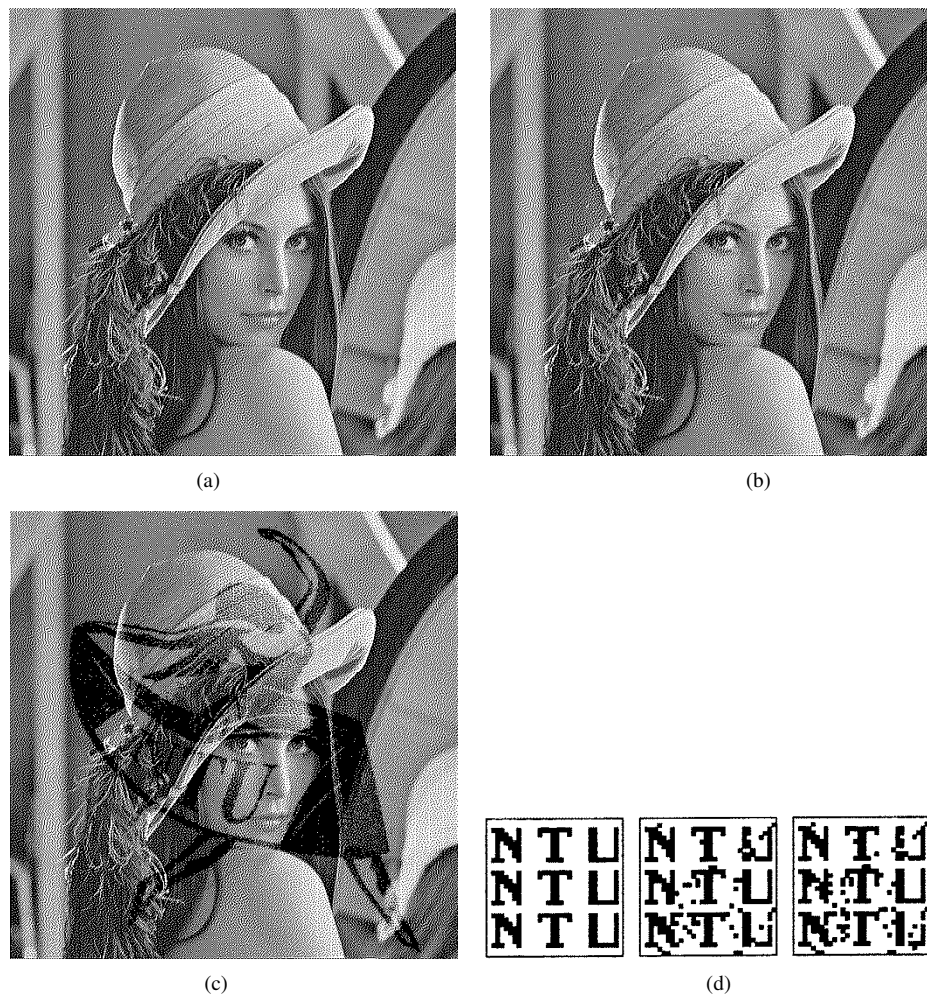


Fig. 20. Results achieved by the hybrid KAEDF and NBEDF algorithms. (a) Embeds the watermark in left most of Fig. 20(d) by KAEDF. (b) Embeds the watermarks in left most of Fig. 20(d) and the visual pattern in Fig. 11(a) by hybrid KAEDF and NBEDF algorithms. (c) Overlaid result by superimposing Fig. 20(a) and (b). (d) Original watermark and the LUT decoding patterns of Fig. 20(a) and (b), respectively.

ACSED_J is always greater than Stucki ACSED_S. Note that ACSED_J of the other five testing images are all still greater than ACSED_S. This shows that using Euclidean distance is a proper method to determine whether a halftone cell was processed with Jarvis or Stucki. With the property described above, if the CSED of a cell is greater than T_{JS} , then this cell is assume to have been processed by Jarvis, otherwise processed by Stucki, where T_{JS} is defined as $(ACSED_J - ACSED_S)/2$. 13tcsvt08-pei

Next, we should determine how many pixels are suitable to be preserved in a cell. In Fig. 19, we see that when the pixel number is in the range of 10–150, the distance between ACSED_J and ACSED_S is increased with more pixels. However it is not correct that a larger distance between ACSED_J and ACSED_S will

produce a higher correct decoding rate. The suitable number of preserved pixels is more correctly decided by the distance ratio (DR), defined as

$$DR = \frac{(ACSED_J - ACSED_S)}{ACSED_S}. \tag{17}$$

A larger value DR is more suitable for judging. On the right column of Fig. 19, we find that the positions that have the largest DR are different in different cell sizes. With 64×64 size, the largest value of DR occurs in 70–80 preserved pixels. With 32×32 and 16×16 are about 30 and 10, respectively. The number of reserved pixels that have the largest value of DR are the same in other five testing images under different sizes of cells.

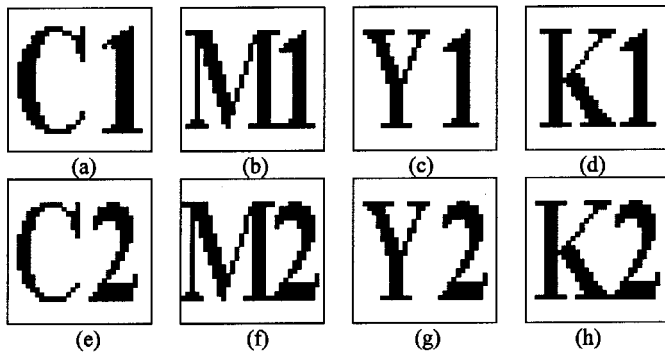


Fig. 21. Eight watermarks, each with size 32×32 , are used for embedding into the four color spaces of EDF1 and EDF2 by the KAEDF algorithm.

Since the number of reserved pixels is determined under different cell sizes, we use T_{JS} to determine whether a cell belongs to Jarvis or Stucki, where a Jarvis cell corresponds to a black dot of the watermark and a Stucki cell to a white dot. The average correct decoding rates of eight testing images in different watermark sizes are shown in Table III. Here, we also show the average correct decoding rates for printed then scanned test images. These include being printed at 150 dpi and scanned at 150, 450, and 750, respectively. Note that since halftoning is commonly used in printing books, newspapers, and magazines. The original embedded watermarked image would be largely destroyed by printing and scanning processes. Due to the following reasons, it is difficult to perfectly extract the original watermarks from a printed and scanned halftoned image.

- 1) The size of printed image is generally slightly larger than the desired one.
- 2) The shape of the printed and scanned halftone image is usually slightly different from the original one.
- 3) The resolution between the original dithering image and the scanned image is usually different.
- 4) Due to the "dot gain" of black printed pixels, the regained image would be darker than the expected ones, leading to erroneous reproduction of the watermarked image.
- 5) Almost all printed images have slight rotation, which lead to digitization errors during the scanning.

In our detection process, the fifth problem can be resolved by re-rotation. Due to the effects of the first, second, and third problems, the precise correlation pixel position could be determined by dividing into 512×512 square blocks, and the average of the pixels within a block is thresholded to recover the original halftone image pixel. However, the fourth problem cannot be solved by using ordinary printing devices. In Table III, we also list the LUT decoding rate. From the results, we see that with the two proposed decoding algorithms, the correct decoding rates of the watermarked images only decrease a little under the harm caused by printing and scanning. In other words, the proposed decoding method is proved to be extremely robust. Besides, the correct decoding rate is higher when the higher scanned resolution is applied.

Now we synthesize the NBEDF and KAEDF algorithms to simultaneously perform data hiding and watermarking. The

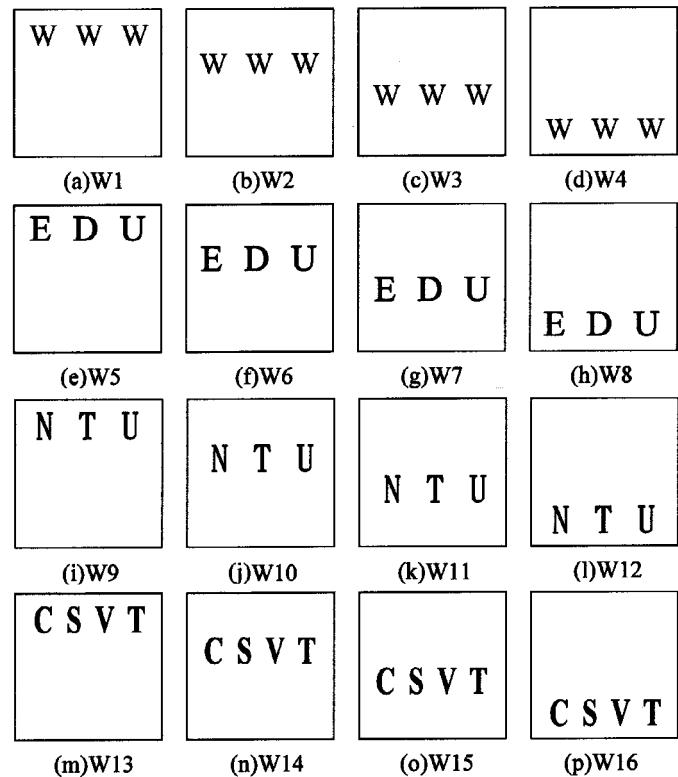


Fig. 22. 16 hidden visual patterns, each with size 512×512 , are used for embedding into the three color spaces of EDF2 by the modified NBEDF algorithm.

watermark NTU shown in left most side of Fig. 20(d) is embedded into Fig. 20(a) and (b) by the KAEDF algorithm. Note that Fig. 20(b) also uses the NBEDF to embed another hidden pattern as shown in Fig. 11(a), where the additional noise strength used is 20. Fig. 20(c) is achieved by superimposing Fig. 20(a) and 20(b). The visual decoding pattern is still clear with this synthesized technique, and Fig. 20(a) and (b) also maintain satisfactory quality. The watermarks retrieved from Fig. 20(a) and (b) by the KAEDF decoding scheme are shown in two right-most boxes of Fig. 20(d) (with a decoding rate of 94.68 and 93.7, respectively).

Finally, we extend the proposed algorithm to the color images as described in Section IV. The eight images of size 32×32 shown in Fig. 21(a)–(h) are embedded into the four color spaces of EDF1 and EDF2 by the KAEDF encoding scheme. Of these, Fig. 21(a)–(d) are embedded into EDF1, and 20(e)–(h) are embedded into EDF2, respectively. Fig. 22(a)–(p) shows the 16 hidden visual patterns, each with size of 512×512 , are embedded into the four color spaces of EDF2 by the modified NBEDF encoding scheme. The embedding method is as described in Section IV. Fig. 23(a)–(d) shows the original color EDF Lena images and watermarked color EDF images, where Fig. 23(a) is processed by Jarvis kernel and 23(b) is processed by Stucki kernel. Fig. 23(c), referred to as EDF1, is processed by KAEDF, and the four color spaces C, M, Y, and K embed watermarks as in Fig. 21(a)–(d), respectively. Fig. 23(d), referred to as EDF2, is processed by the synthesized KAEDF and modified NBEDF algorithm. The four color spaces-C, M, Y, and K of EDF2



Fig. 23. Original color EDF Lena images and watermarked color EDF images. These are all printed at 150 dpi. (a) Processed by Jarvis kernel. (b) Processed by Stucki kernel. (c) Processed by KAEDF, the four color spaces embed four watermarks as in Fig. 21(a)–(d), respectively. (d) Processed by the synthesized KAEDF and modified NBEDF algorithm, the four color spaces embed 4 and 16 watermarks as in Fig. 21(e)–(h) and Fig. 22 (a)–(p), respectively. (a) Jarvis. (b) Stucki. (c) KAEDF (EDF1). (d) KAEDF+NBEDF (EDF2).

embed watermarks as in Fig. 21(e)–(h) and visual patterns as in Fig. 22(a)–(p). Note that, since the KAEDF embedded watermarks of the four color spaces of EDF1 are different from that of EDF2, thus we cannot use normal NBEDF to process EDF2 but must use the modified NBEDF scheme as described in Section II. Fig. 24 shows the visual decoding patterns achieved by alternatively superimposing the four color spaces of Fig. 23(c) and (d). Among the results, the visual decoded patterns are clearly revealed in Fig. 24 (a)–(c), (e)–(g), and (i)–(k). However, in Fig. 24(d), (h), (l), and (m)–(p), the patterns are not as clear as those described above. This is because the intensity of the color spaces K in EDF1 and EDF2 are too high, and therefore fewer black dots can be contributed in overlaid results. So we find that the color space K is not suitable for embedding watermark by NBEDF technique. The four color spaces of EDF1 and EDF2 are shown in Fig. 25. However, if the printing is not the final output method, RGB color spaces are all can be used to embed watermarks, and the

qualities are also good. The eight LUT decoded watermarks are shown in Fig. 26.

VI. CONCLUSIONS

In this paper, a pixel-based data-hiding technique NBEDF is proposed with low computational complexity to hide information into error-diffused images. The visual decoding pattern can be perceived when two or more NBEDF images are overlaid, even in the high activity region of the halftone image. Furthermore, the two halftone images can be made from two totally different gray-tone images and still provide a clear and sharp visual decoding pattern. Besides, with the proposed two self-decode techniques RSDT and SSDT, we can decode the pattern by only one modified NBEDF image. The block-based watermarking technique, referred to as KAEDF, is also described in this paper. We find the two well-known kernels proposed by Jarvis and Stucki are very compatible



Fig. 24. Visual decoding patterns achieved by alternately superimposing the four color spaces of Fig. 23(c) and (d). (a) $C1 + C2$. (b) $M1 + C2$. (c) $Y1 + C2$. (d) $K1 + C2$. (e) $C1 + M2$. (f) $M1 + M2$. (g) $Y1 + M2$. (h) $K1 + M2$. (i) $C1 + Y2$. (j) $M1 + Y2$. (k) $Y1 + Y2$. (l) $K1 + Y2$. (m) $C1 + K2$. (n) $M1 + K2$. (o) $Y1 + K2$. (p) $K1 + K2$.

for alternately processing EDF images. The results exhibit as a conventional error-diffused image. The computational complexity of encoding is not increased at all. At the decoder, we use the CSED and LUT techniques to decode the watermark. From the simulation results, the correct decoding rate is

high and robust even under the distortion caused by printing and scanning. Furthermore, we synthesize the NBEDF and KAEDF algorithms to embed a large amount of information into two color EDF images, which can be used for secret sharing applications.



Fig. 25. Eight color spaces belonging to Fig. 23(c) and (d), respectively.

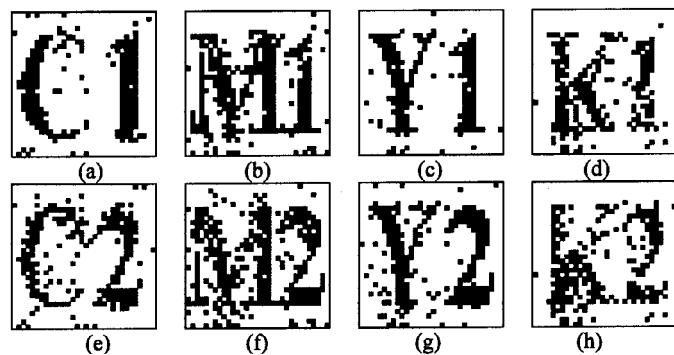


Fig. 26. Eight LUT decoded watermarks with decoding rates: (a) 95.36; (b) 88.91; (c) 95.82; (d) 88.62; (e) 90; (f) 86.1; (g) 92.43; and (h) 87.37.

REFERENCES

[1] R. Ulichney, *Digital Halftoning*. Cambridge, MA: MIT Press, 1987.
 [2] J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A survey of techniques for the display of continuous-tone pictures on bilevel displays," in *Comp. Graph. Image Proc.*, vol. 5, 1976, pp. 13–40.
 [3] P. Stucki, "MECCA-A Multiple-Error Correcting Computation Algorithm for Bilevel Image Hardcopy Reproduction," IBM Res. Lab., Zurich, Switzerland, Res. Rep. RZ1060, 1981.
 [4] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray scale," in *Proc. SID 75 Dig.*: Society for information Display, 1975, pp. 36–37.
 [5] A. Zaknor, S. Lin, and F. Eskafi, "A new class of B/W and color halftoning algorithm," in *Proc. ICASSP-91*, vol. 4, Toronto, ON, Canada, May 1991, pp. 2801–2804.
 [6] M. Analoui and J. P. Allebach, "Model based halftoning using direct binary search," in *Proc. SPIE, Human Vision, Visual Proc., Digital Display III*, vol. 1666, San Jose, CA, Feb. 1992, pp. 96–108.
 [7] D. Anastassion, "Error diffusion coding for A/D conversion," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1175–1186, Sept. 1989.
 [8] D. Anastassiou and S. Kollias, "Digital image halftoning using neural networks," in *Proc. SPIE*, vol. 1001, 1988.

[9] J. B. Mulligan and A. J. Ahumada Jr., "Principled halftoning based on models of human vision," in *Proc. SPIE, Human Vision, Visual Proc., Digital Display III*, vol. 1666, San Jose, CA, Feb. 1992, pp. 109–121.
 [10] D. E. Knuth, "Digital halftones by dot diffusion," *ACM Trans. Graph.*, vol. 6, no. 4, Oct. 1987.
 [11] M. Mese and P. P. Vaidyanathan, "Optimized halftoning using dot diffusion and methods for inverse halftoning," *IEEE Trans. Image Processing*, vol. 9, pp. 691–709, Apr. 2000.
 [12] H. Z. Hel-Or, "Watermarking and copyright labeling of printed images," *J. Electron. Imaging*, vol. 10, no. 3, pp. 794–803, July 2001.
 [13] J. R. Goldschneider, E. A. Riskin, and P. W. Wong, "Embedded color error diffusion," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 1996, pp. 565–568.
 [14] —, "Embedded multilevel error diffusion," *IEEE Trans. Image Processing*, vol. 6, pp. 956–964, July 1997.
 [15] M. S. Fu and O. C. Au, "Hiding data in halftone image using modified data hiding error diffusion," in *Proc. SPIE Conf. Visual Communication and Image Processing*, vol. 4067, 2000, pp. 1671–1680.
 [16] S. G. Wang and K. T. Knox, "Embedding digital watermarks in halftone screens," in *Proc. SPIE Conf. Security and Watermarking of Multimedia Contents II*, vol. 3971, 2000, pp. 218–227.
 [17] M. S. Fu and O. C. Au, "Data hiding by smart pair toggling for halftone images," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, June 2000, pp. 2318–2321.
 [18] —, "Halftone image data hiding with intensity selection and connection selection," *Signal Processing: Image Commun.*, vol. 16, pp. 909–930, 2001.
 [19] —, "Data hiding watermarking for halftone images," *IEEE Trans. Image Processing*, vol. 11, pp. 477–484, Apr. 2002.
 [20] C. W. Wu, "Multimedia data hiding and authentication via halftoning and coordinate projection," *Eurasip J. Appl. Signal Processing*, vol. 2002, no. 2, pp. 143–151, Feb. 2002.
 [21] K. T. Knox, "Digital Watermarking Using Stochastic Screen Patterns," U.S. Patent 5 734 752.
 [22] S. G. Wang, "Digital Watermarking Using Conjugate Halftone Screens," U.S. Patent 5 790 703.
 [23] M. S. Fu and O. C. Au, "Data hiding in halftone images by stochastic error diffusion," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 3, Salt Lake City, UT, May 2001, pp. 1965–1968.
 [24] K. T. Knox and R. Eschbach, "Threshold modulation in error diffusion," *J. Electron. Imaging*, vol. 2, no. 3, pp. 185–192, Jul 1993.

- [25] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, pp. 612–613, 1979.
- [26] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. National Computer Conf.*, vol. 48, 1979, pp. 313–317.
- [27] —, "Safeguarding cryptographic keys," in *American Federation of Information Processing Societies Proceedings*, vol. 48, 1979, pp. 313–317.



Soo-Chang Pei (M'86–SM'89–F'99) was born in Soo-Auo, Taiwan, R.O.C., in 1949. He received the B.S. degree from National Taiwan University, Taiwan, R.O.C., in 1970 and the M.S. and Ph. D. degrees from the University of California, Santa Barbara, in 1972 and 1975, respectively, all in electrical engineering.

He was an Engineering Officer in the Chinese Navy Shipyard from 1970 to 1971. From 1971 to 1975, he was a Research Assistant with the University of California, Santa Barbara. He was

a Professor and Chairman in the Electrical Engineering Departments of Tatung Institute of Technology (1981–1983) and National Taiwan University (1995–1998), respectively. Presently, he is a Professor and Dean with the Electrical Engineering Department, College of Electrical Engineering and Computer Science, National Taiwan University. His research interests include digital signal processing, image processing, optical information processing, and laser holography.

Dr. Pei received the National Sun Yet-Sen Academic Achievement Award in Engineering in 1984, the Distinguished Research Award from the National Science Council in 1990–1998, the Outstanding Electrical Engineering Professor Award from the Chinese Institute of Electrical Engineering in 1998, the Academic Achievement Award in Engineering from the Ministry of Education in 1998, the IEEE Fellow in 2000 for contributions to the development of digital eigenfilter design, color image coding and signal compression, and electrical engineering education in Taiwan, the Pan Wen-Yuan Distinguished Research Award in 2002, and the National Chair Professor Award from the Ministry of Education in 2002. He has been President of the Chinese Image Processing and Pattern Recognition Society in Taiwan from 1996–1998. He is a member of Eta Kappa Nu and the Optical Society of America.



Jing-Ming Guo was born in Kaohsiung, Taiwan, R.O.C., in 1972. He received the B.S.E.E. and M.S.E.E. degrees from National Central University, Chung-Li, Taiwan, in 1984 and 1986, respectively. He is currently working toward the Ph.D. degree in Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

He was a second lieutenant officer in the Chinese army from 1987 to 1988. He is now also a Lecturer in Department of Electronics Engineering at Jin Wen Institute of Technique. He has been granted National Science Council scholarship for advanced research in the Department of Electrical and Computer Engineering, University of California, Santa Barbara, for 2003–2004. His research interests include digital signal processing, image processing, low-bit-rate video coding, digital halftoning, and digital watermarking.