Abstract—A halftone watermarking method of high quality, robustness, and capacity flexibility is presented in this paper. An objective halftone image quality evaluation method based on the human visual system obtained by a least-mean-square algorithm is also introduced. In the encoder, the kernels-alternated error diffusion (KAEDF) is applied. It is able to maintain the computational complexity at the same level as ordinary error diffusion. Compared with Hel-Or using ordered dithering, the proposed KAEDF yields a better image quality through using error diffusion. We also propose a weighted lookup table (WLUT) in the decoder instead of lookup table (LUT), as proposed by Pei and Guo, so as to achieve a higher decoded rate. As the experimental results demonstrate, this technique is able to guard against degradation due to tampering, cropping, rotation, and print-and-scan processes in error-diffused halftone images.

Index Terms—Error diffusion, halftoning, lookup table, watermarking.

1. Introduction

Digital halftoning [1] produces a two-tone texture pattern that, through the human low-passed visual system, approximates the original multi-tone images, preserving a significant level of the original information content. The technique is widely used in computer printer-outs, printed books, newspapers, and magazines, because these printing processes are limited to the black-and-white format. There are many halftone methods, and the most popular ones are the ordered dithering [1], error diffusion [2]-[4], and least-squares [5]. Among these, error diffusion produces good visual quality and reasonable computational complexity.

Watermarking and data hiding have many usages, including: protecting ownership rights of an image, protecting against the use of an image without permission, and authenticating an image to prove that it has not been altered. Generally speaking, watermarking should take the robustness issue into consideration. Currently, numerous methods using halftones to embed watermarks have been studied.

These techniques can be used for printing security documents such as identification (ID) cards, currency as well as confidential documents, and preventing from illegal duplication and forgeries by further scanning these documents to digital forms. In general, these methods can be divided into two categories.

Techniques in the first category embed invisible digital data into halftone images, which can be retrieved by applying some extraction algorithms to the scanned images. These methods include using a number of different dither cells to create a threshold pattern in the halftoning process [6], employing the concept of vector quantization (VQ) to embed watermarks into the most or least significant bit (MSB/LSB) of the error diffusion images [7], or applying modified data hiding error diffusion (MDHED) to embed data into error diffusion images [8].

Methods in the second category embed hidden visual patterns into two or more halftone images. The hidden visual patterns can be directly perceived when the halftone images are overlaid each other. These techniques include using stochastic screen patterns [9], hybrid pixel-based data hiding, and block-based watermarking [10].

In [6], the author proposed an extremely robust watermarking technique for the ordered dithering halftone image. The correct decoded rate claimed in that case was 100%. However, nowadays, error diffusion is the preferred method of the printer industry to produce higher image quality. Therefore, in this paper, we propose the watermarking with an error-diffused halftone image. For this, an objective halftone image quality evaluation method is also given based on the human visual system obtained by a least-mean-square algorithm. As demonstrated in the experiments, the image quality is superior to the order-dither-based watermarking [6]. Pei and Guo [10] applied the kernels-alternated error diffusion (KAEDF) so as to keep the computational complexity the same as the ordinary error diffusion. A robust watermarking decoder was also proposed with a lookup table (LUT). However, the LUT
employed in [10] simply adopted two sets of “independent matched patterns” to determine whether a cell was processed by Jarvis or Stucki error kernel. In this paper, the correct decoded rate is improved by involving the proposed weighted lookup table (WLUT) decoding technique. As the experimental results demonstrate, this technique is superior to the previous LUT method, and is also able to guard against degradation due to tempering, cropping, rotation, as well as the print-and-scan process in error-diffused halftone images.

2. Quality Evaluation

The quality evaluation employed in this paper is defined as follows. For an image with the size of \( P \times Q \), the quality evaluation of halftone images is defined as

\[
\text{PSNR} = 10 \log \left( \frac{PQ}{\sum_{i=1}^{P} \sum_{j=1}^{Q} (x_{i,j} - \hat{x}_{i,j})^2} \right)^2
\]

where PSNR denotes peak signal to noise ratio, \( w_{m,n} \) is the pixel value in original image, \( h_{i,m,j,n} \) is the pixel value in a halftone image, and \( R \) is the support region of the human visual system coefficients. In this paper we fixed \( R \) at the size of 15 \times 15. The human visual system \( w \) can be obtained by psychophysical experiments [11]. The other way to derive \( w \) is using a training set of both pairs of gray level images and good halftone results, such as using error diffusion or ordered dithering to produce the set. Here we use least-mean-square (LMS) to derive \( w \) as follows

\[
\hat{x}_{i,j} = \sum_{m,n} w_{m,n} h_{i,m,j,n}
\]

where \( \hat{x}_{i,j} \) represents the current input pixel value, \( e_{i,j} \) is the mean square error (MSE) between \( x_{i,j} \) and \( \hat{x}_{i,j} \), \( \hat{x}_{i,j} \) is the reconstructed image, \( w_{m,n,\text{opt}} \) is the optimum LMS coefficient, \( w_{m,n}^{(k+1)} \) is the trained filter at the \((k+1)\)th iteration, \( w_{m,n}^{(k)} \) is the trained filter at the \(k\)th iteration, \( e_{i,m,j,n} \) is the error at position \((i+m, j+n)\), and \( \mu \) is the adjusting parameter used to control the convergent speed of the LMS optimum procedure, which is set to be \( 1 \times 10^{-5} \) in our experiments.

There are 8 images used in our training process: Lena, Mandrill, Yosemite, Paris, Airplane, Peppers, Milk, and Lake images. The Floyd error diffusion [2] and Bayer-5 dispersed-dot halftone screen [11] are used to produce the corresponding halftone training results. The trained human visual filter is shown in Fig. 1. Note that this filter has several basic human visual system characteristics, which includes: 1) the diagonal has less sensitivity than the vertical and horizontal directions and 2) the center portion has the highest sensitivity and it decreases while moving away from the center.

3. Error Diffusion

Error diffusion (EDF) is a key step in this paper. In this section, a brief overview of EDF is provided. The standard EDF can be described as (7) and (8).

\[
v_{i,j} = x_{i,j} + \hat{x}_{i,j}
\]

where \( v_{i,j} \) is the temporary calculated pixel value:

\[
\hat{x}_{i,j} = \sum_{m,n} e_{i,m,j,n} h_{m,n}
\]

\[
d_{i,j} = v_{i,j} - b_{i,j}
\]

where

\[
b_{i,j} = \begin{cases} 0 & \text{if } v_{i,j} < 128 \\ 255 & \text{if } v_{i,j} \geq 128 \\ \end{cases}
\]

and \( x_{i,j} \) is the diffused error sum added up from the neighboring processed pixels, \( b_{i,j} \) means binary output in position \((i,j)\), and the Jarvis error kernel \( h_{m,n} \) is used here. The other well-known Stucki error kernel is also shown in Fig. 2.

![Fig. 2. Stucki error diffusion kernel \((h_{m,n})\).](image-url)
values in the watermark.

alternated between two different kernels according to the proposed WLUT decoding.

KAEDF encoding and LUT decoding, and then introduce

Stucki error-diffused image with PSNR

From experiments, we find that the Jarvis and Stucki kernels are the most compatible. The results in other combinations of these three kernels are not as good as Jarvis and Stucki, which were discussed in [10].

The codec flow chart is depicted in Fig. 5. Without loss of generality, the size of the original gray-tone image is defined as $M \times N$, and the watermark is of the size $M_0 \times N_0$. With this definition, the original gray-tone image is divided into cells, each of the size $M_1 \times N_1$, 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 watermark is 0 (black), it is processed with the Jarvis kernel. If the cell corresponding to the watermark is 1 (white), it is processed with the Stucki kernel. Note that, in order to avoid the blocking effect, the amount of errors produced in the periphery of the cell should still be diffused to the neighboring cells. It should also be noted that the codec in Fig. 5 is somewhat different from that in [10]. In this paper, the watermark is pre-processed with pseudo random permutation and daubed with gray as shown in Fig. 5. This is done in order to withstand a number of attacks discussed in Section 5. An example of KAEDF is given in Fig. 3, where Fig. 3 (a) and (b) are obtained from Jarvis and Stucki error kernels with PSNR of 32.48 dB and 33.31 dB, respectively. Fig. 3 (c) shows a watermark of size 32×32, (d) embedded error-diffused halftone image obtained by KAEDF and Hel-Or[6] with PSNR of 32.3 dB and 30.08 dB, respectively. It is clear that the image quality with the proposed KAEDF is better than that in [6] regardless of whether subjective or objective quality evaluation is used.

4. Watermarking in an Error-Diffused Halftone Image

In this section, we briefly describe the previous KAEDF encoding and LUT decoding, and then introduce the proposed WLUT decoding.

4.1 KAEDF Encoding

We now describe the KAEDF embedding technique, in which the computational complexity of the encoding process is not increased at all. The encoding used here is similar to that in [6], where Hel-Or used a pair of dithering cells embedding watermarks into dithered images. However, the method can not be applied to error diffusion images because of the limitations of the decoding rule established there. The flow chart of the proposed encoding is the same as that in Fig. 4, but the error kernel $h_{n,m}$ employed here is alternated between two different kernels according to the values in the watermark.

First, two compatible error kernels are required for halftone image production. Some well-known kernels are Floyd-Steinberg[2], Jarvis-Judice-Ninke[3], Stucki[4], etc. However, not all of these kernels are “compatible” with each other. That is, if we randomly choose two of them alternately to process the gray-tone image, the produced halftone image may decrease in quality due to unnatural blocking effects. From experiments, we find that the Jarvis and Stucki kernels are the most compatible. The results in other combinations of these three kernels are not as good as that in Fig. 4.

Fig. 4. Standard error diffusion scheme.

Fig. 5. Codec of the block-based kernels-alternated error diffusion.
The dashed pointers indicate decoder.

3. The dashed pointers indicate decoder.

4.2 WLUT Decoding

Since the detail textures of Jarvis and Stucki patterns are somewhat different in the spatial domain (e.g., Jarvis has coarse textures and Stucki has finer textures), the LUT technique\(^\text{[10]}\) can be used for rapidly decoding the KAEDF embedded images. However, the LUT constructed in [10] eliminates the patterns that simultaneously appear in both sides of the Jarvis and Stucki tables. Those patterns are considered “useless” when majority voting is applied in the decoder.

In this paper, each pattern is accompanied by a weight which represents the number accumulated in the LUT constructed procedure, and none of the pattern in the LUT is eliminated. This is the major difference from the previous LUT method. Here we use 8 training images described above to establish the WLUT. The decoded performance comparison of LUT and WLUT with the embedded error diffused Lena image is demonstrated in Table 1. In Table 1, 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 LUT (WLUT). If there are more matched patterns similar to the patterns in Jarvis LUT, then we conclude the cell has been processed by Jarvis or vice versa. The “decoding region size” indicates how many matched patterns (these matched patterns can overlap each other, e.g., a decoding region size 3 \times 3 for 9 matched patterns) are used for the majority devoting to determine whether the cell belongs to Jarvis or Stucki. The decoding region is selected to be positioned at the center of a cell in this paper. It is clear from Table 1 that the WLUT has higher correct decoded rate than LUT.

The reason that the correct decoded rate of LUT in Table 1 is about 4% lower than the data given in [10] is the pseudo random permutation of the watermark, which causes the distribution of white and black dots of the watermark to be more disordered. The scramble watermark leads to an alternated arrangement of Jarvis and Stucki patterns in an embedded halftone image, so that the Jarvis texture will be more easily to affect the matched pattern characteristics in the Stucki cell, and vice versa. This is the main reason for the lowering of the correct decoding rate.

Fig. 6 is an example demonstrating how a cell is decoded to Jarvis with a decoding region size of 2 \times 2 and a matched pattern size of 3 \times 3. The dashed pointers indicate when a matched pattern occurs in both Jarvis and Stucki training images. The cell has been processed by Jarvis if the value of the weighting (\(W_{\text{J2}}\) to \(W_{\text{JSN}}\)) is used to determine which error kernel is preferred.

Furthermore, we determine the optimum matched pattern size and decoding region size. Table 2 lists the average decoding rate under different pattern sizes and decoding region sizes combinations with 8 different testing images described above. It is clear that a matched pattern of the size 5 \times 5 combined with a decoding region of the size 11 \times 11 gives the highest decoding rate. So in this paper these two parameters are used for the fast WLUT decoding. Generally speaking, the larger decoding region size offers more matched patterns for the majority voting. So, it can withstand more attacks or distortions to the embedded halftone images. However, the sum of the matched pattern size and the decoding region size is suggested to be smaller than the size of a cell. Otherwise, some matched patterns will spill over to the neighboring cells, then cause error decoding. The reason that a decoding region of the size 7 \times 7 generally has a lower decoded rate than that of the size 5 \times 5 under different matched pattern sizes is the training space (training images) being insufficient to cover more types of matched patterns in 7 \times 7 for the LUT (WLUT) construction procedure. When comparing Table 2 (a) and (b), it can be seen that the proposed WLUT has a better decoding performance than LUT, especially with a matched pattern of the size 4 \times 4.

### Table 1: Decoding performance of LUT and WLUT.

<table>
<thead>
<tr>
<th>Matched pattern size</th>
<th>Decoding region size</th>
<th>Correct decoding rate (%)</th>
<th>Decoded watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUT 5 \times 5</td>
<td>11 \times 11</td>
<td>92.47</td>
<td></td>
</tr>
<tr>
<td>WLUT 5 \times 5</td>
<td>11 \times 11</td>
<td>94.53</td>
<td></td>
</tr>
</tbody>
</table>

4.2 WLUT Decoding

Since the detail textures of Jarvis and Stucki patterns are somewhat different in the spatial domain (e.g., Jarvis has coarse textures and Stucki has finer textures), the LUT technique\(^\text{[10]}\) can be used for rapidly decoding the KAEDF embedded images. However, the LUT constructed in [10] eliminates the patterns that simultaneously appear in both sides of the Jarvis and Stucki tables. Those patterns are considered “useless” when majority voting is applied in the decoder.

In this paper, each pattern is accompanied by a weight which represents the number accumulated in the LUT constructed procedure, and none of the pattern in the LUT is eliminated. This is the major difference from the previous LUT method. Here we use 8 training images described above to establish the WLUT. The decoded performance comparison of LUT and WLUT with the embedded error diffused Lena image is demonstrated in Table 1. In Table 1, 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 LUT (WLUT). If there are more matched patterns similar to the patterns in Jarvis LUT, then we conclude the cell has been processed by Jarvis or vice versa. The “decoding region size” indicates how many matched patterns (these matched patterns can overlap each other, e.g., a decoding region size 3 \times 3 for 9 matched patterns) are used for the majority devoting to determine whether the cell belongs to Jarvis or Stucki. The decoding region is selected to be positioned at the center of a cell in this paper. It is clear from Table 1 that the WLUT has higher correct decoded rate than LUT.

The reason that the correct decoded rate of LUT in Table 1 is about 4% lower than the data given in [10] is the pseudo random permutation of the watermark, which causes the distribution of white and black dots of the watermark to be more disordered. The scramble watermark leads to an alternated arrangement of Jarvis and Stucki patterns in an embedded halftone image, so that the Jarvis texture will be more easily to affect the matched pattern characteristics in the Stucki cell, and vice versa. This is the main reason for the lowering of the correct decoding rate.

Fig. 6 is an example demonstrating how a cell is decoded to Jarvis with a decoding region size of 2 \times 2 and a matched pattern size of 3 \times 3. The dashed pointers indicate when a matched pattern occurs in both Jarvis and Stucki training images. The cell has been processed by Jarvis if the value of the weighting (\(W_{\text{J2}}\) to \(W_{\text{JSN}}\)) is used to determine which error kernel is preferred.

Furthermore, we determine the optimum matched pattern size and decoding region size. Table 2 lists the average decoding rate under different pattern sizes and decoding region sizes combinations with 8 different testing images described above. It is clear that a matched pattern of the size 5 \times 5 combined with a decoding region of the size 11 \times 11 gives the highest decoding rate. So in this paper these two parameters are used for the fast WLUT decoding. Generally speaking, the larger decoding region size offers more matched patterns for the majority voting. So, it can withstand more attacks or distortions to the embedded halftone images. However, the sum of the matched pattern size and the decoding region size is suggested to be smaller than the size of a cell. Otherwise, some matched patterns will spill over to the neighboring cells, then cause error decoding. The reason that a decoding region of the size 7 \times 7 generally has a lower decoded rate than that of the size 5 \times 5 under different matched pattern sizes is the training space (training images) being insufficient to cover more types of matched patterns in 7 \times 7 for the LUT (WLUT) construction procedure. When comparing Table 2 (a) and (b), it can be seen that the proposed WLUT has a better decoding performance than LUT, especially with a matched pattern of the size 4 \times 4.
Table 2: Decoding rate (%) of LUT and WLUT with different decoding region sizes and pattern sizes: (a) LUT and (b) WLUT.

<table>
<thead>
<tr>
<th>Matched pattern size</th>
<th>Decoding region size</th>
<th>7×7</th>
<th>9×9</th>
<th>11×11</th>
<th>13×13</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4×4</td>
<td></td>
<td>47.47</td>
<td></td>
<td>51.7</td>
<td>57.35</td>
</tr>
<tr>
<td>5×5</td>
<td></td>
<td>88.47</td>
<td>91.21</td>
<td></td>
<td>91.76 (best)</td>
</tr>
<tr>
<td>7×7</td>
<td></td>
<td>87.43</td>
<td>89.85</td>
<td>90.92</td>
<td></td>
</tr>
<tr>
<td>(b)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4×4</td>
<td></td>
<td>65.68</td>
<td>66.65</td>
<td>67.24</td>
<td>68.41</td>
</tr>
<tr>
<td>5×5</td>
<td></td>
<td>90.91</td>
<td>92.31</td>
<td></td>
<td>93.67 (best)</td>
</tr>
<tr>
<td>7×7</td>
<td></td>
<td>88.62</td>
<td>91.95</td>
<td>92.12</td>
<td></td>
</tr>
</tbody>
</table>

5. Experimental Results

In the following, we conduct a series of experiments testing issues of robustness, which include tampering, cropping, rotation, and print-and-scan attacks. The watermark of the size 32×32 as shown in Fig. 5 (c) is used in the following experiments.

Fig. 7 (a) shows the tampered watermarked image, and the corresponding decoded watermark with a decoded rate 87.99% is shown in Fig. 7 (b).

Fig. 8 (a)−(e) are the watermarked Lena images attacked and cropped of 50×50, 100×100, 150×150, 200×200, and 250×250, respectively. Fig. 8 (f)−(j) are the decoded watermarks from (a)−(e) with the decoding rates of 94.24%, 93.36%, 91.41%, 89.16%, and 86.82%, respectively.

Now, the rotation attack question is considered. Fig. 9 (a)−(f) are watermarked images which are attacked with rotated angles of 5°, 6°, 7°, 8°, 9°, and 10°, respectively. In the decoder, the re-rotation process is applied first, and then the proposed WLUT decoding scheme is performed. The corresponding decoded watermarks are shown in Fig. 9 (g)−(l) with the decoded rates of 88.38%, 86.82%, 86.23%, 85.74%, 82.23%, and 80.57%, respectively.

In the most common applications of halftoning in printed books, newspapers, and magazines, the original embedded watermarked image is often damaged by the print-and-scan process, e.g., zooming, rotation, and dot gain. And perfectly extracting the original watermarks is very challenging. So during the extraction process, we put auxiliary synchronized black pixels in the four corners of the embedded halftone image. The print-and-scan embedded image is first re-rotated by Adobe Photoshop 7.0.
Because the size of the print-and-scan image is usually larger than expected one (when the same dot-per-inch (dpi) of printing and scanning are applied), the print-and-scan embedded image should be geometrically transformed into the size of $512 \times 512$ prior to the decoding process. To overcome this problem, the print-and-scan image is divided into 262,144 square blocks, under the assumption that the original image is $512 \times 512$, and the average of the pixels within a block is threshold to recover the original halftone image pixels. Since the laser printer often introduces dot gain, the threshold is lowered from 128 to 100 to overcome the dot gain effect. For the experiments, the HP LaserJet 4050 printer and HP OfficeJet 7100 scanner were utilized. Before the embedded halftone image is printed, the format is saved in bitmap and sent directly to the printer. For that, the printer driver does not involve any further halftone process in the image.

Table 3 shows the print-and-scan average correct decoded rates with eight embedded tested halftone images as described above (Lena, Mandrill, Yosemite, Paris, Airplane, Peppers, Milk, and Lake images). The embedded images are printed at 150 dpi and scanned at 150 dpi, 450 dpi, and 750 dpi, respectively. The results show that the proposed WLUT is still superior to LUT under the print-and-scan attack. The results also show that a higher scan resolution improves the accuracy of the decoding rates.

In summary, the experiments convincingly demonstrate that the KAEDF encoding cooperates with WLUT decoding give good tolerance for the print-and-scan distortion.

### 6. Conclusions

In this paper, we propose a high quality and robust watermarking technique with KAEDF encoding and WLUT decoding techniques. An objective halftone image quality evaluation method based on the human visual system obtained by least-mean-square is also presented. As the experimental results demonstrate, this technique is able to guard against the degradation due to tampering, cropping, rotation, as well as print-and-scan processes in error-diffused halftone images.

### References


---

**Jing-Ming Guo** was born in Taiwan in 1972. He received the B.S.E.E. and M.S.E.E. degrees from National Central University in 1995 and 1997, respectively, and the Ph.D. degree from the Institute of Communication Engineering, National Taiwan University in 2004.

From 1998 to 1999, he was an information technique officer in army. From 2003 to 2004, he was granted the National Science Council scholarship for advanced research with the Department of Electrical and Computer Engineering, University of California. He is currently a professor with the Department of Electrical Engineering, National Taiwan University of Science and Technology. His research interests include multimedia signal processing, multimedia security, computer vision, and digital halftoning.

Dr. Guo is a senior member of the IEEE Signal Processing Society. He received the Outstanding Youth Electrical Engineer Award from Chinese Institute of Electrical Engineering in 2011, the Outstanding Young Investigator Award from the Institute of System Engineering in 2011, the Best Paper Award from the IEEE International Conference on System Science and Engineering in 2011, the Excellence Teaching Award in 2009, the Research Excellence Award in 2008, the Acer Dragon Thesis Award in 2005, the Outstanding Paper Awards from Computer Vision and Graphic Image Processing in 2005 and 2006, and the Outstanding Faculty Award in 2002 and 2003.