Data Hiding in Halftone Images Using Adaptive Noise-Balanced Error Diffusion and Quality-Noise Look Up Table

Jing-Ming Guo, Member, IEEE and Jia-Jin Tsai

Department of Electrical Engineering,
National Taiwan University of Science and Technology
Taipei, Taiwan
E-mail: imguo@seed.net.tw, M9607212@mail.ntust.edu.tw

ABSTRACT
This study presents a reasonable computational complexity watermarking algorithm to embed hidden pattern into two or more halftone images with Adaptive Noise-Balanced Error Diffusion (ANBEDF). One halftone image is obtained by traditional error diffusion, and the others are obtained by ANBEDF. The visual decoded hidden pattern can be detected when the similar error-diffused images are overlaid each other. Better decoded result can be obtained by a simple XOR operation. The proposed method employs the trained Quality-Noise Look Up Table (QNLUT) and the optimized multipliers to control the adaptive noise strength according to the local variance value. The experimental results show that higher decoding rate is available under the same image quality performance as former approaches in the literature.

Index Terms — Digital halftoning, data hiding, ordered dithering, error diffusion, adaptive noise-balanced error diffusion.

1. INTRODUCTION
Digital halftoning [1] is a process to display grayscale images with two-tone texture pattern. Halftoning is mainly used in printouts such as printing magazines, newspapers, and books, since these output devices can only generate black-and-white format. Many popular halftone technique methods are proposed in the literature, such as ordered dithering [1], error diffusion [2]-[4], dot diffusion [5,6], and iteration-based approach [7], etc. Among these, error diffusion produces good image quality and reasonable computational complexity. Hence, the error diffusion is adopted in this work for hidden pattern embedding.

Watermarking and data hiding have many applications, such as protecting ownership right of an image, protecting currency as well as confidential documents, and prevent from being used for printing security documents such as ID cards, currency, etc., against the use of an image without owner’s permission. In this study, the data hiding is used to cooperate with error diffusion to protect printing documents. In practical applications, it can be used for printing security documents such as ID cards, currency as well as confidential documents, and prevent from illegal duplication and forgery by further scanning these documents to digital form.

In general, the data hiding in halftone images can be divided into two categories as described below. The first category embeds invisible digital data into halftone images that can be retrieved by scanning and applying the corresponding extraction algorithms. These methods include employing the concept of vector quantization (VQ) to embed watermarks into the most or least significant bit (MSB/LSB) of error diffusion images [8]; using joined direct binary search (DBS) halftoning and spread spectrum watermarking algorithm to achieve good image quality and good watermark detection [9].

Methods in the second category embed hidden visual patterns into two or more halftone images. The hidden visual patterns can be perceived directly when the halftone images are overlaid each other. Knox [10] employed stochastic screen patterns to embed watermark into ordered dither images. Fu and Au [11] presented stochastic error diffusion by embedding hidden visual patterns into two or more halftone images, and the hidden visual patterns can be perceived directly when the halftone images overlay each other. Pei and Guo [12, 13] presented Noise-Balanced Error Diffusion (NBEDF) to enhance the image quality. The additive noise is adopted to force the pixels in the embedded images to be either correlated or independent, according to whether the pixel value in the watermark is black or white. The average color shifting error caused by the additive noise is compensated by adding an equal amount of opposite noise to the neighborhood unprocessed pixels. The technique retains the benefit of efficient visual decoding via printing the embedded image on transparencies, and is further generalized to CMYK color images. Although Pei and Guo’s approach provide good image quality with noise-balanced error diffusion, the decoded pattern quality degrades significantly when the additive noise is small. Another observation shows that areas with different variances may have different quality degradation when the same noise strength is added. The reason is that the human visual system is not a linear system when it is used to inspect the alterations of image regions with various properties, e.g., different variances.

For this, the Adaptive Noise-Balanced Error Diffusion (ANBEDF) is proposed to solve this problem. The main strategy is to use the trained Quality-Noise Look Up Table (QNLUT) and the optimized multipliers to control the additive noise strength according to the local variance value of the host image, which can effectively provide better decoded rate with the same image quality as the former approaches in the literature, and vice versa.

The rest of this paper is organized as follows. Section 2 provides an overview of the performance evaluations and decoded rate definition. Section 3 introduces the proposed ANBEDF. The experimental results are given in Section 4, and Section 5 draws conclusions.

2. OVERVIEW OF PERFORMANCE EVALUATIONS
Figures 1(a)-(b) show the original grayscale image and its corresponding Floyd error-diffused halftone image, respectively. Two important criterions, quality and decoded rate, are defined later in this section.

Suppose a halftone image of size \( P \times Q \), the quality assessment used in this study is defined as below:

\[
PSNR = 10 \log_{10} \frac{P \times Q 	imes 255^2}{\sum_{i,j} (s_{i,j} - \sum_{m,n} w_{m,n} b_{i+m,j+n})^2}
\]

where \( s_{i,j} \) denotes the original grayscale image; \( b_{i,j} \) denotes the decoded result.

\( x_{i,j} \) denotes the original grayscale image; \( b_{i,j} \) denotes the decoded result.
denote the halftone image; \( w_{a,n} \) denotes the Least-Mean-Square (LMS) trained human visual filter at position \((m,n)\), and \( R \) denotes the support region of the human visual filter. In this work, the size of the filter is fixed at \( 7 \times 7 \). The LMS-trained filter can be obtained by psychophysical experiments. The other way to derive \( w \) can use a training set of both pairs of grayscale images and good halftone result of them, such as using error diffusion or ordered dithering to produce the set. Using to this quality criterion, the image quality of Fig. 1(b) is 57.06 dB.

Two criterions are used to evaluate the Correct Decoded Rate (CDR) as defined below:

\[
CDR = \frac{P \times Q - HD(W_{i,j}, W'_{i,j})}{P \times Q} \times 100\% 
\]

(2)

\[
\text{XNOR } CDR = \frac{\sum_{i,j} W_{i,j} \oplus W'_{i,j}}{P \times Q} \times 100\% 
\]

(3)

where \( W_{i,j} \) and \( W'_{i,j} \) denote the original watermark and the decoded watermark, respectively. The symbol \( HD(\cdot, \cdot) \) denotes the Hamming distance between the two patterns inside the parentheses. The symbol \( \Theta \) denotes the XNOR (Not exclusive OR) Boolean operation. The CDR is a measurement to determine the similarity between the original watermark and the corresponding decoded watermark.

3. ADAPTIVE NOISE BALANCE ERROR DIFFUSION (ANBEDF)

3-1 Structure of ANBEDF

This section introduces the proposed ANBEDF. Figure 2 shows the conceptual diagram of the ANBEDF. For simplicity, we take two halftone images for demonstration. Suppose that the size of the hidden pattern is enlarged to fix the size of the host image. The EDF1 is achieved by processing the original grayscale image with the standard error diffusion. The EDF2 is achieved using the proposed ANBEDF. The variables \( W_{a} \) and \( W_{b} \) denote the set of locations of all the white and black pixels in the watermark, respectively, and the variables \( EDF1_{w} \), \( EDF1_{b} \), \( EDF2_{w} \), and \( EDF2_{b} \) are defined likewise. The conditions and the corresponding processing of EDF2 are reorganized as below:

\[
\begin{align*}
\nu_{1} : & \begin{cases} 
W_{a} + LUT(N_{a}) & \text{if } (i,j) \in \{W_{a} \cup \{EDF1\}_{w}\} \\text{or} \{W_{a} \cup \{EDF1\}_{b}\}
\end{cases} \\
\nu_{2} : & \begin{cases} 
W_{a} - LUT(N_{a}) & \text{if } (i,j) \in \{W_{b} \cup \{EDF1\}_{w}\} \\text{or} \{W_{b} \cup \{EDF1\}_{b}\}
\end{cases} \\
\nu_{3} : & \begin{cases} 
W_{b} - LUT(N_{b}) & \text{if } (i,j) \in \{W_{a} \cup \{EDF1\}_{w}\} \\text{or} \{W_{a} \cup \{EDF1\}_{b}\}
\end{cases} \\
\nu_{4} : & \begin{cases} 
W_{b} + LUT(N_{b}) & \text{if } (i,j) \in \{W_{b} \cup \{EDF1\}_{b}\} \\text{or} \{W_{a} \cup \{EDF1\}_{w}\}.
\end{cases}
\end{align*}
\]

(4)

If the processing position \((i,j)\) satisfies the condition \((i,j) \in W_{a} \) and \((i,j) \in \{EDF1\}_{w} \), then the corresponding position in EDF2 is expected to be black, thus producing a black overlaid result by EDF1 and EDF2. Consequently, \(- LUT(N_{a}) \) is added to the top part of Eq. (4), where the notation \( LUT(N_{a}) \) is temporarily used to represent that the value of additive noise \( N_{a} \) is controlled by the trained look up table which will be introduced in the next subsection. If the processing position \((i,j)\) meets the condition \((i,j) \in W_{a} \) and \((i,j) \in \{EDF1\}_{w} \), then the corresponding position in EDF2 is expected to be white, thus producing a white overlaid result by EDF1 and EDF2. Consequently, \(+ LUT(N_{b}) \) is added to the bottom part of Eq. (4). If the processing position \((i,j)\) satisfies the condition \((i,j) \in W_{b} \) and \((i,j) \in \{EDF1\}_{w} \), the overlaid result is always black, regardless it is black or white in EDF2. Consequently, \(- LUT(N_{a}) \) is added to the top part of Eq. (4) in order to receive a compensating term \(+ LUT(N_{a}) \) in the neighborhood in the top part of Eq. (5), increasing the probability of the neighbors becoming white. If the processing position \((i,j)\) satisfies the condition \((i,j) \in W_{a} \) and \((i,j) \in \{EDF1\}_{b} \), then the overlaid result is always black, regardless of whether it is black or white in EDF2. In general, if a pixel in \((i,j) \in W_{a} \), the neighborhood is more likely to be black. Consequently, \(+ LUT(N_{a}) \) is added in the bottom part of Eq. (4) in order to receive a compensating term \(- LUT(N_{a}) \) to the neighborhood in the bottom part of Eq. (5), increasing the probability of the neighbors becoming black. Notably, whatever a positive or negative additive noise is added in Eq. (4), a compensating term must be added in Eq. (5) to preserve the overall brightness. The extracted pattern becomes clearer when the value of \( LUT(N_{a}) \) is increased, yet the quality of EDF2 reduces as well. Hence, the value of \( LUT(N_{a}) \) depends on whether the application is CDR or quality oriented.

3.2 Quality-Noise Look Up Table (QNLUT) Development

This subsection introduces the development of the QNLUT. Figure 3 shows the training system diagram. Herein, eight training images are involved to provide an objective table. The main reason of developing QNLUT is that human vision can discern a scrap of modification in the low frequency region of an image. In this work, the variance is used to measure the local area frequency of an image. The mean and the variance are defined as:

\[
\begin{align*}
\bar{b}_{ij} &= \frac{\sum_{n \in u_{ij}} b_{n}}{g} \\
\sigma_{ij}^2 &= \frac{\sum_{n \in u_{ij}} (b_{n} - \bar{b}_{ij})^2}{g}
\end{align*}
\]

(6)

(7)

where \( \bar{b}_{ij} \) and \( \sigma_{ij}^2 \) denote the mean and the variance of a block. As shown in Fig. 3, each image is divided into 3x3 overlapped blocks, and the variance of each of the block is calculated. Since the larger block size increases the complexity, this study adopts blocks of size 3x3 to maintain the overall algorithm with a reasonable complexity.

To build the QNLUT, the variance values are quantized into limited steps, and each step has equal area (equal block number). To construct the relationship between PSNR and additive noise strength, each block is processed with boundary treatment, which extends the size of a block from \( 3x3 \) to \( 9x9 \). The reason that we conducted the boundary treatment is that the size of the LMS-trained filter is \( 7x7 \), which is used to evaluate the PSNR of a halftone pattern. Hence, when the center of the LMS filter covers the boundary of the original block, the extended part can map to the exceeded LMS filter coefficients. The traditional NBEDF [12] is then applied to the extended block with randomly adding 50% noise, and the strength of the noise is varied from 1 to 26. By doing this, we can construct the QNLUT, as shown on the right hand side of Fig. 3, which is formally organized in Table 1. Table 1 will be carefully explained in Section 4.

When the QNLUT is developed, the ANBEDF introduced in subsection 3.1 is applied to embedded hidden pattern. In the decoder, the hidden pattern can be visually decoded by printing the original EDF image and the ANBEDF image on two transparencies, and then superimposing both of them. Another approach is to perform the XNOR operation with the two images.

4. EXPERIMENTAL RESULTS

In this section, the proposed ANBEDF is applied using
QNLUT to demonstrate the performance.

Figures 4 shows the relationships among PSNR, additive noise strength, and variance, with trained blocks of size. These results are obtained by adding 50% noise using typical NBEDF [12]. One important feature is observed from the three figures: When the strength of additive noise is increased from 1 to 26, the PSNR of Area 1 (the variances are classified into 10 Areas) is reduced from 41dB to 26dB. Conversely, the PSNR of Area 10 is reduced from 36dB to 26dB. The dynamic range of PSNR is reduced when the variance of a block is increased. As we expected, those blocks with higher variances can add higher additive noise strength with smaller quality degradation than those blocks with lower variances.

In fact, the curves in Fig. 4 construct the actual values of QNLUT as shown in Table 1. Notably, the additive noise strength as addressed in Fig. 4 cannot provide the optimized result from the experimental results. For example, for a given PSNR, if the noise strengths of different area are multiplied with appropriate multipliers, the overall CDR can be improved. For this, we conduct extensive experiments, where each multiplier associates to each area is varied from 0.5 to 2, the varying step is 0.01. The optimized multipliers for various areas are organized in Table 2. The practical usage of the multipliers is given as blow:

\[
\text{ANS}(k) = N_{\text{NS(PSNR)}} \times \alpha(k), \quad k = 1 \sim 10
\]

where \(\text{ANS}(k)\) denotes the adaptive noise strength of Area \(k\); \(N_{\text{NS(PSNR)}}\) denotes the noise strength of the Fig. 4 when a specific PSNR is selected, and \(\alpha(k)\) denotes the multiplier of Area \(k\).

Figure 5 shows the average XNOR CDR and CDR results between ANBEDF and NBEDF [12] using eight test images. According to experimental results, the ANBEDF provides better XNOR CDR and CDR than that in NBEDF when the image quality is fixed the same.

Figures 6 and 7 show the practical results processed by NBEDF and ANBEDF, respectively. Figure 6(a) shows the hidden pattern of size 512x512. Figures 6(b) and 7(a) show the NBEDF image and ANBEDF image, respectively, which have the same PSNR. Figure 6(c) shows the overlaid results from Figs. 1(b) and 6(b), and Fig. 7(b) shows the overlaid results from Figs. 1(b) and 7(a). The CDRs of Figs. 6(c) and 7(b) are 56.86% and 57.17%, respectively. Apparently, Figure 7(b) has less unwanted deckle-edge-like artifacts on the right hand side of the decoded hidden pattern. Figure 6(d) shows the XNOR results from Figs. 1(b) and 6(b), and Fig. 7(c) shows the XNOR results from Fig. 1(b) and 7(a). The CDRs of Figs. 6(d) and 7(c) are 81.81% and 82.45%, respectively. It is clear that the ANBEDF significantly reduces the unwanted deckle-edge-like artifacts.

5. CONCLUSIONS

Human visual system is not a linear system when it is used to inspect the alterations of image regions with various properties, e.g., different variances. Former approaches related to embed hidden pattern into multiple host images used fixed additive noise strength, which cannot enjoy the benefit of increasing the CDR by exploiting the human vision response. For this, this study presents a reasonable computational complexity watermarking algorithm to embed hidden pattern into two or more halftone images with Adaptive Noise-Balanced Error Diffusion (ANBEDF). The Quality-Noise Look Up Table (QNLUT) and the optimized multipliers are developed to control the adaptive noise strength according to the local variance value of the host image. The experimental results demonstrate that the proposed approaches can effectively improve the decoding rate of the hidden pattern than former approaches in the literature.

REFERENCES

Fig. 3. QNLUT development system diagram.

Fig. 4. Relationships among PSNR, additive noise strength, and variance.

Fig. 5. Average CDR and XNOR CDR comparisons between ANBEDF and NBEDF [12] under the same PSNR (a) XNOR CDR. (b) Overlaid CDR of ANBEDF and NBEDF.

Fig. 6. NBEDF results. (a) The hidden pattern. (b) The Mandrill image processed with NBEDF (fixed noise=26). (c) Overlaid result from Figs. 1(b) and 6(b). (d) XNOR result from Figs. 1(b) and 6(b).

Fig. 7. ANBEDF results. (a) The Mandrill image processed with ANBEDF. (LUT PSNR=26) (b) Overlaid result from Figs. 1(b) and 7(a). (c) XNOR results from Fig. 1(b) and 7(a).

TABLE 1. EQUIVALENT EMBEDDED QUALITY USING ANBEDF AND NBEDF.

<table>
<thead>
<tr>
<th>Area</th>
<th>Variance range</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[0.00,1.99]</td>
<td>0.92</td>
</tr>
<tr>
<td>2</td>
<td>(1.99,3.55]</td>
<td>0.94</td>
</tr>
<tr>
<td>3</td>
<td>(3.55,6.02]</td>
<td>0.96</td>
</tr>
<tr>
<td>4</td>
<td>(6.02,12.44]</td>
<td>0.98</td>
</tr>
<tr>
<td>5</td>
<td>(12.44,24.54]</td>
<td>1.00</td>
</tr>
<tr>
<td>6</td>
<td>(24.54,51.09]</td>
<td>1.02</td>
</tr>
<tr>
<td>7</td>
<td>(51.09,120.22]</td>
<td>1.04</td>
</tr>
<tr>
<td>8</td>
<td>(120.22,312.00]</td>
<td>1.08</td>
</tr>
<tr>
<td>9</td>
<td>(312.00,820.76]</td>
<td>1.5</td>
</tr>
<tr>
<td>10</td>
<td>&gt;820.76</td>
<td>2.0</td>
</tr>
</tbody>
</table>

TABLE 2. OPTIMIZED MULTIPLIER FOR QNLUT.