Paper No.: krr06

Title: Data Hiding in Ordered Dithered Halftone Images

Authors: Ming Sun Fu*, Oscar C. Au** (correspondence author = Oscar C. Au)

Affiliation: Dept. of Electrical and Electronic Engineering
(for both) Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong, China.
Tel: +852 2358-7053, Fax: +852 2358-1485

Email: [*fmsun@ust.hk](mailto:fmsun@ust.hk), [**eeau@ust.hk](mailto:eeau@ust.hk)

Abstract: In this paper, we propose a novel method called Data Hiding Ordered Dithering (DHOD) to hide a relatively large amount of invisible watermarking data in ordered dithered halftone images while retaining good visual quality. We assume that both the original image and the ordered dithering screen are available. The proposed method includes two techniques: *screen modification* and *threshold selection*. Firstly, one out of every M pseudo-random locations is selected using *threshold selection* to embed one bit of hidden data. Secondly, at the selected locations, *screen modification* is applied to change the ordered dithering screen in a local neighborhood to achieve the desired data hiding while attaining good visual quality. Some custom-made quality measures are proposed to evaluate DHOD. Simulation results suggest that DHOD can hide a large amount of data while maintaining good visual quality.

Keyword: halftone image, ordered dithering, screen modification, threshold selection, data hiding, invisible watermark

Running header: Data Hiding Ordered Dithering

Acknowledgement: This work is funded by an RGC CERG grant.

Data Hiding in Ordered Dithered Halftone Images

Ming Sun Fu, Oscar C. Au

Department of Electrical and Electronic Engineering,
Hong Kong University of Science and Technology,
Clear Water Bay, Hong Kong, China.

Tel: +852 2358-7053, Fax: +852 2358-1485, Email: fmsun@ust.hk, eeau@ust.hk

1. Introduction

Nowadays, images appear routinely in massively distributed printed matters such as books, magazines, newspapers, printer outputs and fax documents. Images appear also in widely accessible web pages, and multimedia files on the Internet and in storage media such as CD-ROM and DVD. Associated with the widespread circulation of images are issues of copyright infringement, authentication and privacy. One of the possible solutions is to embed some hidden watermarking data into the images.

Digital watermarking [7] is a process that can embed some invisible digital data called watermark into an image. There are two classes of invisible watermarks: fragile watermarks and robust watermarks. Fragile watermarks [7] are designed to be broken easily by common image processing operations. The broken watermark serves as an indication of alteration of the original multimedia data and is useful for authentication. Major applications include tampering detection of images placed on the World Wide Web and authentication of images received from questionable sources. Robust watermarks [7] are required to remain in the watermarked image even after it has been attacked by attackers or processed by common image processing operations such as filtering, requantization, scaling, cropping, etc. Invisible watermarks can also be classified as private and public watermarks. A private watermark uses the original image in the watermark decoding while a public watermark does not.

In this paper, we are concerned about invisible watermarking for a special kind of image:

halftone images obtained by halftoning. Halftoning [10] is a process to change multi-tone images into 2 tone images, which look like the original multi-tone images when viewed from a distance. Halftone images are widely used in the printing of books, magazines, newspapers and in computer printers, which are very common in our daily lives. It is often desirable to hide certain invisible watermarking data within the halftone images such as company identity, owner information, creation date and time and other information for copyright protection and authentication purposes. As most halftone images are printed on papers, there are very few, if any, possible digital attacks on the watermarked halftone images though there can be many physical attacks on the papers on which the images are printed. In this paper, we assume that there are no physical attacks and the watermarked halftone images can be fully recovered without any error by scanning the printed images. We will propose a few methods for data hiding watermarking for halftone images with high data hiding capacity and minimal degradation of the visual quality.

There are two main kinds of halftoning techniques, namely, ordered dithering [2] and error diffusion [4]. Ordered Dithering is a computationally simple and effective halftoning method, usually adopted in low-end printers. It compares the pixel intensities with some pseudo random threshold patterns or screens in order to determinate its 2-tone output. Table 1 shows an example of a dithering screen [3], which will be used throughout this paper. Fig. 5 shows an original 8 bit gray scale image, Lena, of size 512x512. Fig. 6 is the halftoned version of Lena using ordered dithering. All images in this paper are available at the web site <http://www.ee.ust.hk/~eeau>. Error diffusion is more sophisticated than ordered dithering and can achieve higher visual quality by performing error diffusion or feedback. In this paper, we are concerned about ordered dithering.

It is well known that most image processing techniques such as filtering or resizing cannot be applied to a halftone image to produce another halftone image with good visual quality. Watermarking is not an exception. Most, if not all, existing watermarking schemes for natural images as mentioned earlier cannot be applied to the special class of halftone images due to the special characteristics of halftone images. Halftone images contain only two tones and have large amount of high frequency noise resulting in very little intensity redundancy. The halftone image pixels take on only two possible values, typically 0 and 255 (or black and white). Most existing watermarking schemes for natural

images would generate images that require 8 bits to represent and will take on values other than 0 and 255.

However, there are still some existing techniques for halftone image watermarking. Some used two different dithering matrices for the halftone generation [1] such that the different statistical properties due to the two dithering matrices can be detected in the future. Some used stochastic screen patterns [6] and conjugate halftone screens [9] in which two screens were used to form two halftone images and the data was embedded through the correlations between two screens. The embedded pattern can be viewed when the two halftone images are overlaid. Some embedded data in the angular orientation of circularly asymmetric halftone dot patterns that were written into the halftone cells of digital halftone images [8]. One common characteristic of these methods is that they cannot embed a large amount of data without significant perceptual distortion. In [5], we proposed two methods for hiding a fairly large amount of data in halftone images without knowledge of the original multi-tone image and the halftoning method. The first method called Data Hiding Self Toggling (DHST) hides data by forcing pixels at pseudo-random locations to toggle. DHST gives poor visual quality. The second method called Data Hiding Pair Toggling (DHPT) improves on DHST by performing pair toggling rather than self-toggling. The visual quality of DHPT is quite good.

In this paper, we address a different situation in which the original multi-tone image is known and the halftoning method is ordered dithering. We will propose a novel method called Data Hiding Ordered Dithering (DHOD) in which we hide data by modifying the ordered dithering screen locally around pixels chosen out of some pseudo-random candidate locations. We will show that DHOD achieves higher visual quality than DHPT due to the additional information of the original image and the halftoning method.

In Section 2, we describe the proposed DHOD, analyze the error characteristics of DHOD, and propose a few objective quality measures for assessing the performance of DHOD. In Section 3, simulation results are presented and discussed.

In this paper, the data to be hidden may be pre-processed, in any arbitrary order, by one or more of these invertible operations: re-ordering, padding, scrambling, encryption, transformation, source coding or compression, error correction/control coding, etc before being embedded into the

halftone images. We will use the term 'the data to be embedded' to mean the pre-processed version of the data to be hidden. In all cases, when the embedded data is read out or recovered, the inverse of each of the operations applied during the pre-processing is applied in reverse order to recover the original data.

2. Data Hiding Ordered Dithering (DHOD)

In this section, we propose a novel algorithm called Data Hiding Ordered Dithering (DHOD) to hide a large quantity of embedded data in a halftone image when the original multi-tone image is available and the halftoning method is ordered dithering.

The block diagram of data hiding using DHOD is shown in Fig. 1. The block diagram of the corresponding data extraction is shown in Fig. 2. In DHOD, a pseudo-random number generator with a known seed is used to generate a set of N pseudo-random locations within the image. The seed will be needed for recovering the embedded data. Then one bit of the embedded data is hidden at M pseudo-random locations by forcing the parity of the sum of the M pixels to be even or odd according to the data bit to be embedded. The parity of the M -pixel-sum is changed, when needed, by two steps. The first step is *threshold selection* that selects one out of the M pixels to be changed. The second step is *screen modification* that modifies the halftoning screen around the selected pixel to get the desired parity.

Assume that the embedded data bits and the original halftone image pixels are statistically independent and that each embedded data bit is equally likely to be 0 or 1. Then, with a probability of 0.5, the original M -pixel-sum parity is the desired value and thus no change is needed. With a probability of 0.5, the original M -pixel-sum parity is different from the desired value. In this case, the *threshold selection* and *screen modification* are applied to force the M -pixel-sum parity to be the same as the desired value. To read the embedded data, one simply uses the same random number generator and the same known seed to obtain the N pseudo-random locations. Then the embedded data bits can be extracted easily by examining the M -pixel-sum parity.

2.1 Screen Modification

Let $\{(i_1, j_1), (i_2, j_2), \dots, (i_M, j_M)\}$ be M pseudo-random locations among which one data bit needs to be hidden. These locations are typically far apart, locating at different parts of the image. Let $x(i, j)$, $y(i, j)$, $T(i, j)$ be the original multi-tone image pixel value, the ordered dithered halftone value and the threshold at any location (i, j) respectively such that

$$y(i, j) = \begin{cases} 0 & x(i, j) < T(i, j) \\ 255 & \text{otherwise} \end{cases} \quad (1)$$

The parity of the sum of the pixel values at the M locations is $(\sum_{k=1}^M y(i_k, j_k)) \bmod 2$. When the parity is the same as the desired value, nothing needs to be changed. When they are different, an odd number of the halftone values $y(i, j)$ need to be changed so as to change the parity to be the desired value. To minimize the distortion due to the changes, only one of the M pixels will be changed. We will call this pixel location (m, n) . This pixel to be changed is chosen by the proposed threshold selection that will be described in the next section. In this section, we propose the screen modification to modify the screen associated with the chosen pixel $y(m, n)$ in order to change the M -pixel-sum parity.

Let the ordered dithering screen be of size $P \times Q$. For example, the screen in Table 1 has $P=Q=8$. Typically, the screen contains PQ integers, $\{0, 1, 2, \dots, PQ-1\}$. Let T_s be a number at a random location within the screen. Then T_s becomes a random variable. The T_s from the screen is scaled to $T = \frac{255T_s}{PQ-1}$ to attain the full dynamic range of the image intensity before being used as thresholds in ordered dithering in Eqn. (1). Both T_s and T are approximately uniformly distributed such that

$$\Pr\left(\frac{T}{255} \leq u\right) = \Pr\left(\frac{T_s}{PQ-1} \leq u\right) \approx u$$

where $0 \leq u \leq 1$ and the $\Pr$ is the probability. If the input is of constant intensity K , the probability of a white halftone output is

$$\Pr(y = 255) = \Pr(K \geq T) = \Pr\left(\frac{255T_s}{PQ-1} \leq K\right) = \Pr\left(\frac{T_s}{PQ-1} \leq \frac{K}{255}\right) \approx \frac{K}{255}$$

and the expected intensity within the block is

$$E(y) = 0 \cdot \Pr(y = 0) + 255 \cdot \Pr(y = 255) = 255 \cdot \frac{K}{255} = K$$

which is the same as the input.

Recall that (m, n) is the location at which the halftone pixel value $y(m, n)$ needs to be changed. In screen modification, we will change $T(m, n)$ to some value $T'(m, n)$. And it will be changed with some complementary changes to the neighboring thresholds such that the total change will always be zero. We impose a constraint

$$\sum_{(i,j) \in N} \Delta T_s(m+i, n+j) = 0 \quad (2)$$

for some neighborhood N and $\Delta T_s(m+i, n+j) \equiv T'_s(m+i, n+j) - T_s(m+i, n+j)$ such that

$$\sum_{(i,j) \in N} \Delta T(m+i, n+j) = 0 \quad \Delta T(m+i, n+j) \equiv T'(m+i, n+j) - T(m+i, n+j)$$

Some possible neighborhood definitions are shown in Fig. 3. We will refer to the screen modification using 1-, 2-, 4- and 8-neighborhoods as SM1, SM2, SM4 and SM8 respectively. We will use SM0 to refer to the control situation in which no neighboring thresholds are changed.

Consider the case of $x(m, n) < T(m, n) = 256T_s(m, n)/(PQ-1)$ such that $y(m, n) = 0$ before screen modification. To force $y(m, n)$ to be 255, we can choose $T'(m, n) = 0$ if $x(m, n)$ is unknown. The problem of this is that $\Delta T(m, n) = -T(m, n)$ tend to have a large magnitude such that there will be a large change to the neighboring thresholds due to our constraint. This may cause large changes to the final halftone outputs which are undesirable. In the proposed screen modification, we assume that $x(m, n)$ is known and we reduce the magnitude of $\Delta T(m, n)$ by choosing $T'(m, n) = x(m, n)$ such that the output $y(m, n) = 255$ according to Eqn. (1), and

$$\sum_{(i,j) \in N \setminus (0,0)} \Delta T(m+i, n+j) = -\Delta T(m, n) = T(m, n) - x(m, n) > 0$$

As for the case of $x(m, n) \geq T(m, n)$ such that $y(m, n) = 255$ before screen modification, we need to

force $y(m, n)$ to be zero. In screen modification, we choose $T'(m, n) = x(m, n) + 1$ such that $y(m, n) = 0$ according to Eqn. (1) and

$$\sum_{(i,j) \in \mathbb{N} \setminus (0,0)} \Delta T(m+i, n+j) = -\Delta T(m, n) = T(m, n) - x(m, n) - 1 \leq -1 < 0$$

Often for good perceptual quality, a large threshold is surrounded by small ones on the left, right, up and down directions in the ordered dithering screen and vice versa, as in Table 1. In other words, if $T(i, j)$ is large, then $T(i, j-1)$, $T(i, j+1)$, $T(i-1, j)$ and $T(i+1, j)$ are usually small but $T(i-1, j-1)$, $T(i-1, j+1)$, $T(i+1, j-1)$ and $T(i+1, j+1)$ are usually large. In Table 1, a small threshold may be any number less than 32 while a large threshold may be any number larger than or equal to 32. With these magnitude patterns, some neighborhood in Fig. 3 may be more effective than others. As a result, SM0, SM1, SM2, SM4 and SM8 will be investigated in our simulation.

As an example, suppose $x(m, n)$ is such that $\frac{x(m, n)}{255}(PQ-1)$ lies between screen values of 46 and 47. Suppose the corresponding value in the ordered dithering screen is $T_s(m, n) = 56$, shown as the center pixel in Table 2 such that the original $y(m, n)$ is 0. To change $y(m, n)$ to be 1, screen modification would choose $T'_s(m, n) = 46$ such that $\Delta T_s(m, n) = T'_s(m, n) - T_s(m, n) = 46 - 56 = -10$. If we modify only the left and right neighbors, then $\Delta T_s(m-1, n) = \Delta T_s(m+1, n) = -\Delta T_s(m, n) / 2 = 5$ as shown in Table 2. As a result, the $y(m-1, n)$ and $y(m+1, n)$ may or may not be changed depending on the value of $x(m-1, n)$ and $x(m+1, n)$.

2.2 Threshold Selection

As mentioned earlier, one bit is hidden in the parity of the sum of the pixel values at M pseudo-random locations in DHOD. When the original parity is the desired value, no change is needed. Otherwise, one of the locations will be selected by threshold selection to be changed. To motivate the proposed threshold selection, we will first discuss some implementation issues of screen modification and gain some theoretical insight towards screen modification before presenting the threshold selection algorithm.

As far as implementation is concern, the proposed screen implementation is hard to implement. When the regular ordered dithering is applied, the same screen is used in every $P \times Q$ block and thus it can be implemented efficiently in a loop. For the proposed screen modification, the threshold patterns within each $P \times Q$ block need to be changed and the location of change is not fixed. The irregular structure makes it hard to implement. A better way to implement screen modification is to modify the original image rather than the screen. When a threshold $T(i, j)$ is changed to $T'(i, j) = T(i, j) + \Delta T(i, j)$, the same halftone output can be obtained according to Eqn. (1) by changing the image pixel from $x(i, j)$ to $x'(i, j) = x(i, j) + \Delta x(i, j)$ with $\Delta x(i, j) = -\Delta T(i, j)$ while keeping the threshold $T(i, j)$ unchanged. In this way, the ordered dithering can still be implemented efficiently in a loop since the same screen is used for all the $P \times Q$ blocks. In other words, the screen modification is effectively a modification of the original image. The constraint in Eqn. (2) can be interpreted as requiring the local changes to the image to sum up to zero so that there is no change to the local average image intensity. And the choice of the neighborhood in SM1, SM2, SM4 and SM8 can be interpreted as a choice of different basis functions (or shape) for the additive noise to implement the data hiding. Some basis functions would be visually more pleasing than others. For SM1 and SM2, the basis functions are $[1 \ -1]$ and $[-0.5 \ 1 \ -0.5]$ respectively. For SM4 and SM8,

the basis functions are $\begin{bmatrix} 0 & -1/4 & 0 \\ -1/4 & 1 & -1/4 \\ 0 & -1/4 & 0 \end{bmatrix}$ and $\begin{bmatrix} -1/8 & -1/8 & -1/8 \\ -1/8 & 1 & -1/8 \\ -1/8 & -1/8 & -1/8 \end{bmatrix}$ respectively. The additive

noise to the original image in SM1, SM2, SM4 and SM8 are basically the basis function scaled by $\Delta x(m, n) = -\Delta T(m, n)$.

We now describe the proposed threshold selection. When M locations $\{(i_1, j_1), (i_2, j_2), \dots, (i_M, j_M)\}$ are available, we want to find the best location (m, n) such that the distortion at (m, n) is visually least disturbing among the M locations. As any distortion in DHOD is caused by modifying the screen at and around (m, n) , or equivalently, by adding to the original image some additive noise scaled by $\Delta T(m, n)$, the least distortion to the original image would be achieved by choosing the location with minimum $|\Delta T(m, n)|$. When $|\Delta T(m, n)|$ is minimized, the change to the

neighboring thresholds is also minimized due to our constraint in Eqn. (2). As a result, the proposed threshold selection is to choose the location with minimum $|\Delta T(m, n)|$.

In the rest of the paper, we will use TS2 and TS4 to refer to threshold selection with $M=2$ and $M=4$ respectively. We will use TS1 to refer to the control situation in which $M=1$. As M increases, the performance should improve due to more location choices.

The proposed DHOD is the combination of SM4 and TS4.

2.3 Distortion Characteristics and Quality Measures for DHOD

The peak-signal-to-noise ratio (PSNR) is a common objective quality measure in many image processing tasks. But PSNR is not suitable for halftone images because the PSNR of halftone images is dominated by the high frequency halftone noise. Such high frequency halftone noise can be removed by low pass filtering. A more appropriate quality measure is the Modified Peak Signal-to-Noise Ratio (MPSNR), which is basically the PSNR of the original multi-tone image and a lowpass version of the halftone images, as shown in Fig. 4. The MPSNR measures effectively the distortions to the low-frequency image content. As shown in Table 3, the MPSNR is very similar for all the algorithms suggesting that MPSNR is not a good differentiating quality measure for the proposed algorithm. In this section, we analyze the distortion characteristics of DHOD and propose four scores S_1 , S_2 , S_3 and S_4 for selected locations as quality measure of the algorithms.

Let $\mathcal{A}$ be those locations among the N pseudo-random locations of which the screen value is modified. Thus the cardinality of $\mathcal{A}$ is less than or equal to N . Typically the cardinality is about $N/2$. As screen modification is applied to the locations within $\mathcal{A}$ and their nearby neighbors, there is no change to the rest of the image. In other words, there is no need to measure the distortion of the rest of the image. It suffices to measure the distortion around the locations within $\mathcal{A}$. The distortion due to the proposed algorithm appears mainly in the form of ‘salt-and-pepper’ artifacts due to local clusters of pixels as shown in the 512x512 Lena image in Fig. 8 to 12. Many clusters are formed by 2, 3, 4 or 5 connected pixels of identical color, some white and some black. These small clusters tend to be rich in high frequency energy due to its small size and thus cannot be measured effectively by MPSNR. In

general, large clusters are visually more disturbing than small clusters. Thus a good way to measure performance is to measure the size of the ‘salt-and-pepper’ clusters.

To measure the size of the clusters, we classify the elements of $\mathcal{A}$ into four classes:

1. black halftone pixel in bright region ($y(m,n) = 0$, $x(m,n) \geq 128$)
2. white halftone pixel in bright region ($y(m,n) = 255$, $x(m,n) \geq 128$)
3. black halftone pixel in dark region ($y(m,n) = 0$, $x(m,n) \leq 127$)
4. white halftone pixel in dark region ($y(m,n) = 255$, $x(m,n) \leq 127$)

Both class 1 and class 4 pixel clusters can be visually disturbing, while class 2 and class 3 pixel clusters are significantly less disturbing. Class 1 corresponds to originally white halftone pixels in bright regions being forced to be black by DHOD. They can be visually disturbing due to the huge contrast between the new pixel color (black) and the background color (bright). Similarly, class 4 pixels can be disturbing. On the contrary, class 2 corresponds to originally black pixels in bright background being forced to be white by DHOD. Given they are in bright regions, the black pixels are likely to be isolated black pixels. They tend not to be visually disturbing because of the nice blending of the new color (white) into the background color (bright). Similarly, class 3 pixels tend to be significantly less visually disturbing than class 1 and class 4 pixels.

If a class 1 pixel is an isolated black pixel not connected to any other black pixels, it does not look very bad because ordered dithering would produce isolated black pixels naturally, though not necessarily at that location. However, if it is connected to many other black pixels to form a large cluster, it would look strange and annoying because ordered dithering would not generate such clusters naturally in the bright regions. Similar comments apply to class 4 pixels as well. We thus define the following four scores for the class 1 and class 4 elements of $\mathcal{A}$:

$$S_1 = \sum_{i=0}^4 N_i \quad S_2 = \sum_{i=0}^4 (i+1)N_i \quad S_3 = \frac{S_2}{S_1} \quad S_4 = \sum_{i=2}^4 N_i$$

where N_i is the total number of the class 1 and class 4 elements in $\mathcal{A}$ having i neighbors with same pixel values in the 4-neighborhood after DHOD is performed. The N_0 corresponds to the number of isolated class 1 or class 4 elements in $\mathcal{A}$. The S_1 gives the total number of class 1 and class 4 elements

of $\mathcal{A}$. The S_2 gives approximately the total area covered by the clusters associated with the class 1 and class 4 elements of $\mathcal{A}$. The S_3 gives approximately the average area of the clusters. The S_4 is the number of class 1 and class 4 elements of $\mathcal{A}$ associated with clusters of size 3 or more, which is useful because clusters of size 1 or 2 are not very visually disturbing. Algorithms with smaller scores of S_1 , S_2 , S_3 and S_4 are better.

3. Results and Discussions

In our simulation, a 512x512 gray-scale images ‘Lena’ is halftoned using ordered dithering with the screen in Table 1. The original Lena and the halftoned Lena are shown in Figs. 5 and 6 respectively. The proposed *screen modification* with various neighborhood size (SM0, SM1, SM2, SM4 and SM8) is applied without threshold selection (TS1) to embed 4096 bits into the halftones in Figs. 8 to 12. For SM1, the 1-neighborhood can take on the left, right, top or bottom neighbors. From our experiments, we found that random choice gave similar performance as a fixed neighbor. Thus the fixed neighbor shown in Fig. 3 is used for SM1. Similarly, the 2-neighborhood for SM2 can take on any two of the left, right, top or bottom neighbors. From our experiments, top and bottom neighbors do not perform as well as left and right. Thus the left and right neighbors are used for SM2. The 4096 embedded bits are generated randomly to simulate encrypted data and the same 4096 data bits are used in all simulations (TS1, TS2, TS4, DHPT). If encryption algorithms such as DES are used, 4096 encrypted bits can be used to represent 4096 data bits and the encrypted 4096 bits would appear as random bits. For the sake of comparison, the DHPT [14] is also simulated for comparison. The Lena with the same 4096 bits hidden using DHPT is shown in Fig. 7. The 4096 bits are a relatively large amount of data to be hidden in a 512x512 image. The amount is chosen to show the performance of various algorithms under stress.

Compared with the original halftone image in Fig. 6, the DHPT in Fig. 7 has many ‘salt-and-pepper’ artifacts which are undesirable. The SM0 without screen modification in Fig. 8 has poorer visual quality than DHPT with even more ‘salt-and-pepper’ artifacts. This is expected because SM0 is effectively DHST [14] upon which DHPT improves. The SM0 forces the pixels at the pseudo-random

locations to be 0 or 255 according to the embedded data bits regardless of the image content or the screen thresholds. This causes a great distortion in the local average intensity. The DHPT fixed the problem by performing pair toggling to preserve the local intensity. When screen modification is applied to the 1-neighborhood in SM1 in Fig. 9, there are slightly fewer ‘salt-and-pepper’ artifacts than SM0. When the neighborhood size increases from 1 to 2, 4 and 8 in Figs. 10 to 12, the subjective visual quality appears to be best in SM4. And SM4 appears to have similar, if not slightly better, visual quality than DHPT.

These observations are confirmed quantitatively by Table 4. Out of the 4096 pseudo-random locations, screen modification was applied to 2056 locations in all the TS1 algorithms. DHPT is also applied to the same 2056 locations. In other words, the cardinality of the set $\mathcal{A}$ is 2056 for these algorithms. According to Table 4, 1317 (S_1) out of the 2056 locations are class 1 or class 4 locations for these algorithms. The S_1 is expected to be larger than half of the cardinality of the set $\mathcal{A}$ (1028 in this case) for the following reason. For any element of $\mathcal{A}$ in a bright region (class 1 and 2 elements), it is more likely for the original halftone output to be white (class 1) than black (class 2). Thus there should be more class 2 elements than class 1 elements in $\mathcal{A}$. Similarly, for any element of $\mathcal{A}$ in dark regions (class 3 or 4 elements), it is more likely for the original halftone output to be black (class 4) than white (class 3). Thus there should be more class 4 elements than class 3 elements in $\mathcal{A}$. Consequently, the sum of class 1 and class 4 elements should be more than the sum of the class 2 and 3 elements. In other words, the S_1 which is the sum of the class 4 and class 1 elements should be more than half of the cardinality of $\mathcal{A}$.

The cardinality of $\mathcal{A}$ and the score S_1 are the same for all TS1 cases and DHPT because they depend on the hidden data and the natural halftone pixel values of the original halftone ‘Lena’. As the same 4096 data bits are embedded and the same 4096 pseudo-random locations of the same original halftone ‘Lena’ are used, the cardinality of $\mathcal{A}$ and the S_1 are the same among the TS1 cases and DHPT. The sum of the area of all the associated visually disturbing class 1 and class 4 clusters in SM0 is 4089 (S_2), which is the largest among all algorithms tested. This explains why SM0 has poor visual quality. The average cluster size (S_3) for class 1 and class 4 clusters in SM0 is 3.10 pixels per

cluster, which is very large. In other words, the ‘salt-and-pepper’ clusters look the largest in SM0 on the average. The number of class 1 and class 4 locations with neighborhood size of 3 or more is 840 for SM0, the worst among all algorithms test. With screen modification applied, the scores S_2 , S_3 and S_4 of SM1 are lower than that of SM0 by 12%, 12% and 10% respectively suggesting that SM1 should have better quality than SM0. However, its S_2 , S_3 and S_4 are still significantly higher than those of DHPT implying that DHPT should have better quality than SM1. When the neighborhood size increases from 1 to 2, the scores S_2 , S_3 and S_4 of SM2 are lower than those of SM1 by 6%, 6%, and 3% respectively suggesting that SM2 should have slightly better quality than SM1. Similarly, SM4 should have better quality than SM2 because its scores S_2 , S_3 and S_4 are 5%, 5% and 20% respectively lower than SM2. In particular, the amount of visually disturbing larger clusters is reduced drastically in SM4. When the neighborhood size increases from 4 to 8, the scores S_2 , S_3 and S_4 of SM8 is larger than SM4 by 14%, 14% and 22% respectively suggesting that SM8 should have worse quality than SM4. The scores of SM8 are similar to those of SM1. One possible reason for the poor performance of SM8 is that the ΔT becomes so small in each of the 8 neighbors that the performance of SM8 is approaching that of SM0 which does not change the neighbors at all ($\Delta T = 0$). In other words, among all the neighborhood size, the 4-neighborhood in SM4 performs the best and is thus chosen in the proposed DHOD. The S_2 , S_3 of SM4 are 4% higher than DHPT while the S_4 of SM4 is 13% smaller. These suggest that SM4 and DHPT should have similar quality, with SM4 being slightly better due to the fewer large clusters as reflected by the smaller S_4 .

When *threshold selection* is applied to SM4 for $M=2$ (TS2) and $M=4$ (TS4), the results are shown in Figs. 13 and 14 respectively. The TS2 in Fig. 13 appears to have better visual quality than SM4 in Fig. 11 with fewer ‘salt-and-pepper’ artifacts at the shoulder and face of ‘Lena’ and some smooth regions in the background. The TS4 in Fig. 14 is even better than TS2 at the shoulder and face and the smooth regions. Compared with DHPT, the TS4 has significantly better visual quality with significantly fewer ‘salt-and-pepper’ artifacts at the shoulder and face of ‘Lena’ and at the smooth regions in the background. These observations are confirmed in Table 4.

While the same 4096 hidden data bits remain unchanged, 8192 and 16384 pseudo-random locations are generated in TS2 and TS4 respectively, compared with the 4096 locations in TS1. Because of the different pseudo-random locations, the set $\mathcal{A}$ is different for TS1, TS2 and TS4 and consequently the S_1 is different also. When the M increases from 1 to 2, the S_1 decreases from 1317 to 1110. In other words, the number of visually disturbing class 1 and class 4 elements of $\mathcal{A}$ reduces suggesting better quality of TS2. Actually, the number of larger clusters (S_4) reduces by even more (18%). The area per cluster (S_3) also reduces slightly by 4%. These lead to a significant reduction of total cluster area (S_2) by 19%, from 3226 to 2619, which is very significant. These scores suggest that TS2 should have considerably improved quality over TS1. When M increases from 2 to 4, the scores S_1 , S_2 , S_3 and S_4 of TS4 are lower than those of TS2 by 9%, 20%, 12% and 30% respectively. In other words, TS4 is superior to TS1 and TS2. All these results suggest that threshold selection really gives rise to improved visual quality. The TS4 in Fig. 14 is also significantly better than DHPT in Fig. 7 with the scores S_1 , S_2 , S_3 and S_4 reduced by 24%, 32%, 12%, and 49% respectively. With these numbers, the TS4 should have significantly better visual quality than DHPT. These observations agree with the subjective results.

Besides ‘Lena’, the algorithms are tested on four other 512x512 test images, including the ‘Boat’, ‘Barbara’, ‘Peppers’ and ‘Harbor’. The results are shown in Tables 5 to 8. Some selected images are shown in Figs. 15 to 22. As far as screen modification is concern, the SM4 is found to be better than SM0, SM1, SM2 and SM8 in all the images with lowest S_2 , S_3 and S_4 . Compared with DHPT, SM4 typically has slightly larger S_2 and S_3 , but smaller S_4 than DHPT. In particular, the S_4 of SM4 is significantly lower than DHPT in ‘Boat’. These suggest that the visual quality of SM4 is similar to DHPT, with SM4 being slightly better due to the smaller number of the larger clusters (S_4). Concerning threshold selection, the TS2 is found to achieve significant improvement in all of S_1 , S_2 , S_3 and S_4 over TS1, and TS4 is in turn significantly better than TS2. Compared with DHPT, the S_1 , S_2 , S_3 and S_4 of TS4 are reduced by 26%, 38%, 17%, and 57% respectively. The 57% reduction in the amount of larger clusters (S_4) is a particular performance improvement. For

the 'Boat' shown in Figs. 15 to 18, the SM4 is found to have similar DHPT as suggested by the scores. But the TS4 is significantly better than both SM4 and DHPT at the black bottom of the boat with much fewer 'salt-and-pepper' artifacts. For the 'Peppers' shown in Figs. 19 to 22, the SM4 has better than DHPT with fewer 'salt-and-pepper' artifacts. And TS4 is significantly better than SM4 with even fewer 'salt-and-pepper' artifacts, which agrees with the observation from Tables 4 to 8.

4. Conclusion

In this paper, we proposed a novel way, the Data Hiding Ordered Dithering (DHOD), to hide data in halftone images generated by ordered dithering when both the original multi-tone image and the ordered-dithering screens are known. The proposed DHOD contains two main steps which are screen modification and threshold selection. The distortion characteristics of DHOD were analyzed and some specialized error distortion scores were defined to measure the visual quality of the processed halftone images. For the screen modification, it was found that the 4-neighborhood was the best. The image quality of screen modification with the 4-neighborhood alone was found to be comparable to the existing DHPT. When threshold selection was added to screen modification to form the DHOD, the resulting visual quality is significantly better than DHPT and the scores S_1 , S_2 , S_3 and S_4 of DHOD are significantly better than DHPT.

5. References

1. Z. Baharav, D. Shaked, "Watermarking of Dither Halftoned Images", *Proc. of SPIE Security and Watermarking of Multimedia Contents*, pp. 307-313, Jan 1999.
2. B. E. Bayer, "An Optimum Method for Two Level Rendition of Continuous Tone Pictures," *Proc. of IEEE Int. Communication Conf.*, pp2611-2615, 1973.
3. L.M. Chen, H.M. Hang, "An Adaptive Inverse Halftoning Algorithm", *IEEE Trans. on Image Processing*, Vol. 6, No. 8, August 1997.
4. R.W. Floyd, L. Steinberg, "An Adaptive Algorithm for Spatial Grayscale," *Proc. SID*, pp. 75-77,

- 1976.
5. Ming S. Fu, Oscar C. Au, "Data Hiding for Halftone Images", *Proc. of SPIE Conf. On Security and Watermarking of Multimedia Contents II*, Jan. 2000.
 6. K.T. Knox, "Digital Watermarking Using Stochastic Screen Patterns", *United States Patent Number 5,734,752*.
 7. F. Mintzer, et al., "Effective and Ineffective Digital Watermarks", *Proc. of IEEE Int. Conf. on Image Processing*, Vol. 3, pp. 9-13, Oct. 1997.
 8. R. T. Tow, "Methods and Means for Embedding Machine Readable Digital Data in Halftone Images", *United States Patent Number 5,315,098*.
 9. S. G. Wang, "Digital Watermarking Using Conjugate Halftone Screens", *United States Patent Number 5,790,703*.
 10. R. A. Ulichney, *Digital Halftoning*, Cambridge, MA, MIT Press, 1987.

0	32	8	40	2	34	10	42
48	16	56	24	50	18	58	26
12	44	4	36	14	46	6	38
60	28	42	20	62	30	44	22
3	35	11	43	1	33	9	41
51	19	59	27	49	17	57	25
15	47	7	39	13	45	5	37
63	31	45	23	61	29	43	21

Table 1 '8x8 dispersed-dot' ordered dithering screen matrix

32	8	40
16	56	24
44	4	36

Screen before screen modification

32	8	40
21	46	29
44	4	36

Screen after modifying left, right neighbors

Table 2 An example of screen modification using 2-neighborhood

TS1					TS2	TS4	DHPT
SM0	SM1	SM2	SM4	SM8	SM4	SM4	
26.46	26.67	26.62	26.69	26.66	26.68	26.73	26.93

Table 3 MPSNR of DHOD with SM0/SM1/SM2/SM4/SM8 and TS1/TS2/TS4

i	0	1	2	3	4	S_1	S_2	S_3	S_4
N_i for SM0, TS1	317	160	173	402	265	1317	4089	3.10	840
N_i for SM1, TS1	367	197	288	354	111	1317	3596	2.73	753
N_i for SM2, TS1	460	123	401	192	141	1317	3382	2.57	734
N_i for SM4, TS1	459	269	189	338	62	1317	3226	2.45	589
N_i for SM8, TS1	354	242	185	410	126	1317	3663	2.78	721
N_i for SM4, TS2	536	90	108	301	75	1110	2619	2.36	484
N_i for SM4, TS4	646	21	51	203	87	1008	2088	2.07	341
N_i for DHPT	477	171	404	264	1	1317	3092	2.35	669

Table 4 Statistics and scores of DHOD for classes 1 and 4 elements of set $\mathcal{A}$ of 512x512 'Lena'

i	0	1	2	3	4	S_1	S_2	S_3	S_4
N_i for SM0, TS1	326	181	218	516	115	1356	3981	2.94	849
N_i for SM1, TS1	379	220	429	291	37	1356	3455	2.55	757
N_i for SM2, TS1	467	220	416	205	48	1356	3215	2.37	669
N_i for SM4, TS1	464	451	215	200	26	1356	2941	2.17	441
N_i for SM8, TS1	384	337	251	333	51	1356	3398	2.51	635
N_i for SM4, TS2	557	191	161	171	42	1122	2316	2.06	374
N_i for SM4, TS4	702	50	58	112	29	951	1569	1.65	199
N_i for DHPT	507	218	513	116	2	1356	2956	2.18	631

Table 5 Statistics and scores of DHOD for classes 1 and 4 elements of set $\mathcal{A}$ of 512x512 'Boat'

i	0	1	2	3	4	S_1	S_2	S_3	S_4
N_i for SM0, TS1	388	204	179	426	218	1415	4127	2.92	823
N_i for SM1, TS1	440	241	322	328	84	1415	3620	2.56	734
N_i for SM2, TS1	542	177	362	222	112	1415	3430	2.42	696
N_i for SM4, TS1	571	265	222	282	75	1415	3270	2.31	579
N_i for SM8, TS1	465	243	212	361	134	1415	3701	2.62	707
N_i for SM4, TS2	620	104	120	236	54	1134	2402	2.12	410
N_i for SM4, TS4	716	48	58	180	47	1049	1941	1.85	285
N_i for DHPT	587	183	418	213	14	1415	3129	2.21	645

Table 6 Statistics and scores of DHOD for classes 1 and 4 elements of set $\mathcal{A}$ of 512x512 'Barbara'

i	0	1	2	3	4	S_1	S_2	S_3	S_4
N_i for SM0, TS1	371	168	210	458	166	1373	3999	2.91	834
N_i for SM1, TS1	421	220	358	311	63	1373	3494	2.54	732
N_i for SM2, TS1	523	139	410	224	77	1373	3312	2.41	711
N_i for SM4, TS1	510	307	250	261	45	1373	3143	2.29	556
N_i for SM8, TS1	432	262	226	368	85	1373	3531	2.57	679
N_i for SM4, TS2	599	141	172	219	38	1169	2463	2.11	429
N_i for SM4, TS4	733	38	70	130	45	1016	1764	1.74	245
N_i for DHPT	538	211	456	165	3	1373	3003	2.19	624

Table 7 Statistics and scores of DHOD for classes 1 and 4 elements of set $\mathcal{A}$ of 512x512 'Peppers'

i	0	1	2	3	4	S_1	S_2	S_3	S_4
N_i for SM0, TS1	203	78	81	547	339	1248	4485	3.59	967
N_i for SM1, TS1	222	98	326	466	136	1248	3940	3.16	928
N_i for SM2, TS1	256	104	420	273	195	1248	3791	3.04	888
N_i for SM4, TS1	275	210	212	458	93	1248	3628	2.91	763
N_i for SM8, TS1	246	86	191	521	204	1248	4095	3.28	916
N_i for SM4, TS2	333	80	119	355	126	1013	2900	2.86	600
N_i for SM4, TS4	467	37	51	253	115	923	2281	2.47	419
N_i for DHPT	272	86	543	341	6	1248	3467	2.78	890

Table 8 Statistics and scores of DHOD for classes 1 and 4 elements of set A of 512x512 'Harbor'

Figure 1 Data hiding for halftone images (DHOD)

Figure 2 Data extraction from halftone image with embedded data (for all proposed algorithms)

Figure 3 Left-to-right: Definition of 1-neighborhood, 2-neighborhood, 4-neighborhood and 8-neighborhood (white pixel=current pixel, black pixel=neighbors)

Figure 4 Definition of MPSNR

Fig. 5 Original 512x512 Lena

Fig. 6 Ordered dithered Lena (8x8 dispersed-dot)

Fig. 7 Lena after DHPT (4096 bits)

Fig. 8 Lena after DHOD (4096 bits, SM0, TS1)

Fig. 9 Lena after DHOD (4096 bits, SM1, TS1)

Fig. 10 Lena after DHOD (4096 bits, SM2, TS1)

Fig. 11 Lena after DHOD (4096 bits, SM4, TS1)

Fig. 12 Lena after DHOD (4096 bits, SM8, TS1)

Fig. 13 Lena after DHOD (4096 bits, SM4, TS2)

Fig. 14 Lena after DHOD (4096 bits, SM4, TS4)

Fig. 15 Ordered dithered 512x512 'Boat'

Fig. 16 Boat after DHOD (4096 bits, SM4, TS1)

Fig. 17 Boat after DHOD (4096 bits, SM4, TS4)

Fig. 18 Boat after DHPT (4096 bits)

Fig. 19 Ordered dithered 512x512 'Peppers'

Fig. 20 Peppers after DHOD (4096 bits, SM4, TS1)

Fig. 21 Peppers after DHOD (4096 bits, SM4, TS4)

Fig. 22 Peppers after DHPT (4096 bits)