Data Hiding in Halftone Images with Parity Coding

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

Abstract

In this paper, we propose a novel halftone image data hiding method called Data Hiding in Block Parity (DHBP), which hides data in the block-sum parity. DHBP assumes that the original multi-tone image is available and the halftoning method is ordered dithering. DHBP can hide a relatively large amount of invisible watermarking data in halftone images while retaining good visual quality. In DHBP, one bit of information is hidden in a block of size $M \times N$ by forcing the parity of the sum of the MN pixels to be even or odd according to the data bit to be embedded. To alter the parity, one out of the MN pixels is chosen by minimizing the local image intensity change. Some custom-made quality measures are proposed to evaluate DHBP. Simulation results suggest that DHBP can hide a large amount of data while maintaining good visual quality.

Keyword: halftone image, data hiding, block parity, dithering, error diffusion, watermark

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 [1] is a process to embed some invisible digital data called watermark into an image. There are two classes of invisible watermarks: fragile watermarks and robust watermarks. Fragile watermarks 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 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 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 [2] 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. As a result, robustness or fragileness of the halftone image watermarking schemes is not particularly important for many applications. Instead, the data hiding capacity and the image quality are much more important in most applications. 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 method for hiding data in halftone images with high data hiding capacity and minimal degradation of the visual quality.

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 two examples of the dithering screen, the 8x8 dispersed screen [3] and the 4x4 clustered screen [3]. Figs. 2(a) and 5(a) are the 512x512 test image, Lena, dithered with the dispersed and clustered screens respectively.

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 multi-tone images that 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 [4] such that the different statistical properties due to the two dithering matrices can be detected in the future. Some used stochastic screen patterns [5] and conjugate halftone screens [6] 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 [7]. One common characteristic of these methods is that they cannot embed a large amount of data without significant perceptual distortion. In [8], we proposed a method for hiding a fairly large amount of data in halftone images without the knowledge of the original multi-tone image and the halftoning method. It is called Data Hiding Smart Pair Toggling (DHSPT). We also proposed another method called Modified Data Hiding Error Diffusion (MDHED) [9] for hiding data when the original image is available and the halftoning method is error diffusion. The data is hidden at pseudo-random locations and the error due to the data hiding is diffused to the surrounding pixels using error diffusion.

In this paper, we propose a novel method called Data Hiding in Block Parity (DHBP) which hides data in the block-sum parity. DHBP assumes that the original image is available and the halftoning method is either ordered dithering or error diffusion. In Section 2, the general idea of DHBP is introduced. In Section 3, the simulation results are given.

2. Data Hiding in Block Parity (DHBP)

In this section, we propose a novel algorithm called Data Hiding in Block Parity (DHBP) 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 known to be ordered dithering or error diffusion.

In DHBP, one bit of information is hidden in a block of size $M \times N$ by forcing the parity of the sum of the MN pixels to be even or odd according to the data bit to be embedded. Let $y(i, j)$ be the halftone value at location (i, j) , taking on values of either 0 or 255. Let the halftone pixels within an $M \times N$ block be $\{y(k+i, l+j), i=0, \dots, M-1, j=0, \dots, N-1\}$ for some (k, l) . Then the block-sum parity is $\left[\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} y(k+i, l+j) \right] \bmod 2$, which is the parity of the sum of the MN pixels within the block. 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 block-sum parity is the desired value and thus no change is needed. With a probability of 0.5, they are different. In this case, 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 MN pixels is changed in DHBP. We will call this pixel location (m, n) . To read the embedded data, one simply examines the block-sum parity. If (m, n) is chosen randomly, we will call the method DHBP-with-Random-location (DHBP-Rand).

In DHBP, the location (m, n) is chosen by examining the similarity between the original multi-tone image and the thresholds used in ordered dithering. Let $x(i, j)$ and $T(i, j)$ be the original multi-tone pixel value and the threshold value at location (i, j) such that

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

Let the ordered dithering screen be of size $P \times Q$. For example, the dispersed 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 Eq. (1). Both T_s and T are approximately uniformly distributed,

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

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

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

and the expected intensity within the block is

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

which is the same as the input. Typically, the average halftone intensity is very close to the average multi-tone intensity.

$$\frac{1}{PQ} \sum_{i=0}^{P-1} \sum_{j=0}^{Q-1} y(k+i, l+j) \approx \frac{1}{PQ} \sum_{i=0}^{P-1} \sum_{j=0}^{Q-1} x(k+i, l+j)$$

In DHBP, when the original block-sum parity does not match the desired value, one location (m, n) within the $M \times N$ block needs to be chosen for toggling. Consider any location (i, j) within the $M \times N$ block. Suppose $x(i, j) \geq T(i, j)$ such that the halftone value is $y(i, j) = 255$. At this location, it is unnatural to force $y(i, j)$ to be zero. To force $y(i, j)$ to be zero while still satisfying Eq. (1), we need to force $x(i, j)$ to be decreased by $\Delta x_1(i, j) = x(i, j) - T(i, j) + 1$ such that $x(i, j) - \Delta x_1(i, j) = T(i, j) - 1 < T(i, j)$. However, the decrease in intensity by $\Delta x_1(i, j)$ can be visually disturbing if the magnitude of $\Delta x_1(i, j)$ is large, as is probably the case in DHBP-Rand.

For the case of $x(i, j) < T(i, j)$ such that $y(i, j) = 0$, it is unnatural to force $y(i, j)$ to be 255. To do so while still satisfying Eq. (1), we need to force $x(i, j)$ to be increased by $\Delta x_2(i, j) = T(i, j) - x(i, j)$ such that $x(i, j) + \Delta x_2(i, j) = T(i, j) \geq T(i, j)$. However, the increase in the intensity by $\Delta x_2(i, j)$ can be visually disturbing if the magnitude of $\Delta x_2(i, j)$ is large.

Regardless of whether $y(i, j)$ is 0 or 255, we need to change $x(i, j)$ by approximately $\Delta x(i, j) \equiv |T(i, j) - x(i, j)|$ in order to change the $y(i, j)$ using Eq. (1). This $\Delta x(i, j)$ may be large at some locations (i, j) and small at other locations. To minimize the impact of the effect of the change on the original image, the proposed DHBP chooses the location (m, n) within the $M \times N$ block where $\Delta x(i, j) \equiv |T(i, j) - x(i, j)|$ is minimum, and applies the $\Delta x(m, n)$ to toggle $y(m, n)$ naturally through Eq. (1). As such, the proposed DHBP has very low computational complexity.

If the thresholds (T) are uniformly distributed between 0 and 255 in the $M \times N$ block such that the pixel intensity dynamic range (from 0 to 255) is divided into $(MN-1)$ cells of equal size of $\frac{255}{MN-1}$, an upper bound of $\Delta x(m, n)$ is

$$\Delta x(m, n) \leq \frac{255}{2(MN-1)}$$

If $M=N=8$, then $\Delta x(m, n) \leq 2.02$. If $M=N=4$, then $\Delta x(m, n) \leq 8.50$. As the pixel values are independent of the threshold values, the $\Delta x(m, n)$ can be modeled as uniformly distributed between

$-\frac{255}{2(MN-1)}$ and $+\frac{255}{2(MN-1)}$ with a variance of $\frac{(255)^2}{12(MN-1)^2}$ and zero mean. The variance is 1.37 for $M=N=8$ and 24.08 for $M=N=4$. Smaller M and N give larger variance and upper bound which are undesirable. But smaller M and N also give higher data-hiding capacity which is good, as one bit is hidden in each $M \times N$ block.

Another interpretation of DHBP is as follows. Suppose the block-sum parity of a particular $M \times N$ block needs to be changed. The proposed DHBP is equivalent to making the smallest possible change to the DC value of the block such that one of the pixels would be changed naturally according to Eq. (1). The location chosen in this way would be the location (m, n) with minimum $\Delta x(i, j) \equiv |T(i, j) - x(i, j)|$ and the DC change is $\Delta x(m, n)$. In this interpretation, the block DC value

is changed by applying the identical intensity change $\Delta x(m,n)$ to all the original multi-tone pixels within the block. As the chosen $\Delta x(i,j)$ is minimum within the $M \times N$ block, the intensity change is not large enough to cause other halftone pixels to be changed. With such a uniform intensity change, the content within the block would be uniformly darker or brighter but otherwise perfectly natural. The only problem is the possible DC mismatch at the block boundary. But this mismatch is minimized by minimizing $\Delta x(i,j)$. The DC mismatch is upper bounded by $255/(MN-1)$. Consequently, the halftone patterns of DHBP should be visually pleasing. The DHBP image is simply the ordered dithered halftone version of the image with slightly changed block DC values.

3. Results and Discussions

The proposed algorithms are simulated on a 512x512 test images 'Lena'. The DHBP is simulated with $M=N=2,3,4,5,6,7,8$ on ordered dithered images dithered with the 8x8 dispersed screen and the 4x4 clustered screen in Table 1. For example, with $M=N=4$ and an image size of 512x512, a total of 16384 bits (6.25%) are hidden in the images. The 16384 bits are a relative large quantity of hidden data, which are chosen to show the performance of the algorithms under stress. The performance measure is the Modified Peak-Signal-to-Noise-Ratio (MPSNR) as shown in Fig. 1. For halftone images dominated by high frequency halftone artifacts, PSNR is not suitable. We thus low-pass filtered the halftone images before computing the PSNR as shown in Fig. 1. The MPSNR of the algorithms are shown in Table 2.

Without DHBP, the Lena dithered with the 8x8 dispersed screen has a MPSNR of 27.2dB. The image is shown in Fig. 2(a) which has good visual quality. When DHBP is applied in different rates, corresponding MPSNR is shown in Table 2. In Fig. 3(a), with $M=N=4$, the MPSNR drops slightly to 26.3dB. The perceptual quality of DHBP is as good as the original dithered image. These suggest that the proposed DHBP can hide a large quantity of hidden data with very small, if not negligible, visual degradation. Fig. 5(a) is the plot of the performance of DHBP and DHBP-Rand. DHBP is more significant when the embedded bits increase.

The MPSNR of DHBP-Rand in Fig. 4(a) is 24.8dB which is 1.5dB lower than DHBP. The visual quality of DHBP-Rand is significantly worse than DHBP, with lots of 'salt-and-pepper' artifacts or black and white clusters. These verify that DHBP is non-trivial and is significantly better than DHBP-Rand. Although both algorithms perform toggling to individual pixels, the locations chosen by DHBP are much better than DHBP-Rand such that the toggling done by DHBP does not cause the annoying 'salt-and-pepper' artifacts and decrease visual quality. As explained in Section 2, DHBP applies effectively the minimal DC adjustment to each $M \times N$ block to force one pixel to toggle naturally. The magnitude of the DC adjustment is very small, being upper-bounded by 8.5 (since $M=N=4$). The resulting halftone patterns are thus natural without the unnatural black or white clusters. The possible DC mismatch at block boundary is minimal and is upper bounded by $2 \times 8.5 = 17$ which is small.

Without DHBP, the Lena dithered with the 4x4 clustered screen has a MPSNR of 24.2dB, which is significantly lower than the 27.2dB of the dispersed screen dithering. The image is shown in Fig. 2(b) which has significant lower perceptual quality than Fig. 2(a). When DHBP is applied in Fig. 3(b), the MPSNR drops slightly from 24.2dB to 23.5dB. The perceptual quality of DHBP is slightly worse than the original dither image. Compared with DHBP, DHBP-Rand in Fig. 4(b) has considerably worse perceptual quality with lower MPSNR. Fig. 5(b) is the plot of the performance of DHBP and DHBP-Rand. Again, DHBP is much better than DHBP-Rand.

4. Conclusion

In this paper, we propose a novel way called the Data Hiding in Block Parity (DHBP) to embed hidden data in the block-sum parity of halftone images. When the original multi-tone image is present and the halftoning method is ordered dithering with known screen, the simulation results show that the proposed DHBP can hide a lot of hidden data in the ordered dithered images with very small, if not negligible, visual degradation.

5. Acknowledgement

This work is supported by an RGC CERG Grant from the Hong Kong Government.

6. References

1. F. Mintzer, et al., "Effective and Ineffective Digital Watermarks", *Proc. of IEEE Int. Conf. on Image Processing*, Vol. 3, pp. 9-13, Oct. 1997.
2. R. A. Ulichney, *Digital Halftoning*, Cambridge, MA, MIT Press, 1987.
3. R. L. Stevenson, "Inverse Halftoning via MAP Estimation", *IEEE Trans. of Image Processing*, Vol. 6, No. 4, pp. 574-583, Apr. 1997.
4. Z. Baharav, D. Shaked, "Watermarking of Dither Halftoned Images", *Proc. of SPIE Security and Watermarking of Multimedia Contents*, pp. 307-313, Jan 1999.
5. K. T. Knox, "Digital Watermarking Using Stochastic Screen Patterns", *United States Patent Number 5,734,752*.
6. S. G. Wang, "Digital Watermarking Using Conjugate Halftone Screens", *United States Patent Number 5,790,703*.
7. R. T. Tow, "Methods and Means for Embedding Machine Readable Digital Data in Halftone Images", *United States Patent Number 5,315,098*.
8. M.S. Fu, O.C. Au, "Data Hiding in Halftone Image by Pixel Toggling", *Proc. of SPIE Electronic Imaging: Int. Conf. On Security and Watermarking of Multimedia Contents*, Vol. 3971, pp. 228-236, Jan 2000.
9. M.S. Sun, O.C. Au, "Modified Data Hiding Error Diffusion for Image Halftoning", *Proc. of SPIE Conf. On Visual Communication and Image Processing*, Vol. 3, pp. 1671-1680, 20-23 Jun 2000.

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

8x8 dispersed-dot screen

13	11	12	15
4	3	2	9
5	0	1	10
8	6	7	14

4x4 clustered-dot screen

Table 1 Ordered Dithering Screen Matrices

Type of halftoning	Bits embedded (block size)	0	4096 (8x8)	5329 (7x7)	7225 (6x6)	10404 (5x5)	16384 (4x4)	28900 (3x3)	65536 (2x2)
Dithering (dispersed)	DHBP	27.2	27.1	27.0	27.0	26.7	26.3	25.2	23.1
	DHBP-Rand	27.2	26.5	26.3	26.0	25.5	24.8	23.4	20.4
Dithering (clustered)	DHBP	24.2	24.1	24.1	24.0	23.8	23.5	23.0	21.5
	DHBP-Rand	24.2	23.8	23.6	23.5	23.2	22.7	21.7	19.5

Table 2 Comparison of MPSNR of different data rate ($M=N=4$ for DHBP)

Fig. 1 Definition of MPSNR

Fig. 2(a) Lena ordered dithered (dispersed, 27.2dB)

Fig. 2(b) Lena ordered dithered (clustered, 24.2 dB)

Fig. 3(a) DHBP (dispersed, $M=N=4$, 26.3dB)

Fig. 3(b) DHBP (clustered, $M=N=4$, 23.5dB)

Fig. 4(a) DHBP-Rand (dispersed, $M=N=4$, 24.8dB)

Fig. 4(b) DHBP-Rand (clustered, $M=N=4$, 22.7dB)

Figure 5(a) MPSNR vs. Embedded Bits (dispersed)

Figure 5(b) MPSNR vs. Embedded Bits (clustered)