Data Hiding for 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.
Email: fmsun@ust.hk, eeau@ust.hk

Abstract

With the ease of distribution of digital images, there is a growing concern for copyright control and authentication. While there are many existing watermarking and data hiding methods for natural images, almost none can be applied to halftone images. In this paper, we proposed two novel data hiding methods for halftone images. The proposed Data Hiding Pair-Toggling (DHPT) hides data by forced complementary toggling at pseudo-random locations within a halftone image. It is found to be very effective for halftone images with relatively coarse textures. For halftone images with fine textures (such as error diffusion with Steinberg kernel), the proposed Data Hiding Error Diffusion (DHED) gives significantly better visual quality by integrating the data hiding into the error diffusion operation. Both DHPT and DHED are computationally very simple and yet effective in hiding a relatively large amount of data. Both algorithms yield halftone images with good visual quality.

Keywords: data hiding, halftone image, watermarking, ordered dithering, error diffusion, toggling

1. Introduction

Nowadays, digital images can be distributed easily through the Internet and the Web. As a side effect, the problems of copyright infringement and authentication grow rapidly with the ease of distribution. One of the possible solutions is to embed some hidden or invisible watermarking data into these images. There have been quite a number of watermarking methods [4,5,6,7] designed for natural gray scale images. They mainly use the inherent redundancy of gray scale images to embed data such as using the least significant bit to embed data in the spatial domain or modifying the images within the perceptual thresholds in the frequency domain. However, few, if not none, of the algorithms can be applied to halftone images. This is because halftone images have only 2 tones and have large amount of high frequency noise resulting in little intensity redundancy. In this article, two simple methods are proposed to hide data in halftone images.

Halftoning [1] 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 data within the halftone images such as company identity, owner information, creation date and time and other information for copyright protection and authentication purposes. In this paper, some simple methods to a hide a fairly large amount of data in the halftone images with minimal degradation of the visual quality are developed.

There are two main kinds of halftoning techniques, namely, ordered dithering [2] and error diffusion [3]. 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. Figure 1 shows an example of a dithering screen. Figure 5 shows an original 8 bit gray scale image, Lena, of size 512x512 that will be used throughout this paper. Figure 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.

In *Security and Watermarking of Multimedia Contents II*, Ping Wah Wong, Edward J. Delp, Editors, Proceedings of SPIE Vol. 3971 (2000) • 0277-786X/00/$15.00

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

Figure 1 A typical ordered dithering screen

Error diffusion is an advanced technique usually used in high-end printers. It is more complicated than ordered dithering, but it can generate halftone image with higher visual quality. It is a single pass algorithm. In error diffusion, the halftoning output is obtained by comparing the image pixels with a fixed threshold of 128. However, the halftoning error is fed back to its adjacent neighbors so that each image pixel has effectively an adaptive threshold. The error feedback helps to maintain approximately equal local intensity average between the original multi-tone images and the corresponding halftone images. An essential component of error diffusion is its error feedback kernel. Different kernels can have quite different behavior. Two commonly used error feedback kernels are used in this article. They are the Jarvis Kernel shown in Figure 2 and the Steinberg kernel shown in Figure 3. The halftoned images of Lena processed with the Jarvis and Steinberg kernels are shown in Figures 9 and 13 respectively. The Jarvis kernel has a large support and it tends to generate halftone images with high contrast and coarse texture. The Steinberg kernel is a smaller kernel, which generate halftone images with contrast similar to the original image and with fine texture.

			7	5
3	5	7	5	3
1	3	5	3	1

Figure 2 Jarvis kernel

		7
5	3	1

Figure 3 Steinberg kernel

For data hiding in halftone images, there are two different situations. The first situation is the one in which a halftone image is available without the original multi-tone image and the method of halftoning is not known. In this case, any data-hiding algorithm can only operate on the available halftone image. The second situation is the one in which the original multi-tone image is known as well as the method of halftoning. In this situation, the data-hiding algorithms can be integrated into the halftoning procedure to yield higher visual quality than the first situation. In this paper, we examine a straightforward solution, the DHST, for the first situation and analyze its drawback. We then propose an improved solution, the DHPT. We also address the second situation by proposing the DHED for the error diffusion case. The details of DHST, DHPT and DHED will be given in the next two sections.

2. Data Hiding Self-Toggling (DHST) and Data Hiding Pair-Toggling (DHPT)

In this section, we focus on the situation in which only a halftone image is available but not the original multi-tone image and the method of halftoning is unknown. Without the original image and the method of halftoning, we can only hide data in the halftone image by modifying it in such a way that the visual quality is least compromised while the hidden data can be extracted in the future.

A straightforward data-hiding method for this situation is Data Hiding Self-Toggling (DHST). The key idea of DHST is to identify a set of predefined pseudo-random locations and then change the pixel at that location according to the data to be hidden. To extract the hidden data, the predefined pseudo-random locations are examined and the data is recovered according to the reverse of initial data hiding rule. The predefined pseudo-random locations can be generated with a random number generator using a predefined initial key. The key may or may not be dependent on the image content. In our simulation using the 512x512 Lena image, we hide one bit within every 8x8 block using an image independent random key. In this way, the random locations would not be too crowded.

The exact algorithm of Data Hiding Self-Toggling (DHST) is

1. Select N pseudo random locations using the key (e.g. N may be such that there is one location in each 8x8 block).
2. For each pseudo random location, if the halftone pixel value matches the hidden data bit (black for bit 0 and white for bit 1), do nothing.
3. If the halftone pixel value does not match the hidden data bit, perform self-toggling to change the pixel value.

One advantage of DHST is that it is extremely simple with very low computation requirement. Another advantage is that the amount of hidden data can be controlled easily by adjusting the amount of random locations to be examined. One important disadvantage of DHST is that the random locations and the pixel output are generated without paying attention to the image content. The visual quality cannot be expected to be too high. The visual quality can be particularly bad when more data are hidden.

As the hidden data are statistically independent of the halftone image, typically half of the halftone pixel values at the predefined pseudo-random locations would be the same as the intended values and thus no change of the halftone pixel values would be necessary. For the rest of the halftone pixels, the halftone pixels are altered which change the local average intensity and thus can result in potential visible artifacts.

DHST is applied to the halftone image of Lena in Figures 6, 9 and 13, obtained by various halftoning algorithms. One bit of information is hidden in every 8x8 blocks. As there are 4096 (=512*512/8/8) 8x8 blocks in the 512x512 image, there are a total of 4096 bits hidden. On the average, the 4096 bits are hidden in 1.526% (=4096/512/512) of the pixels of the image, which is a rather large percentage compared with other watermarking algorithms. This percentage is intentionally set to be large to show the possible visual degradation due to the data hiding procedure. In real applications, the percentage should probably be set to be smaller to get better visual quality. The resulting images are shown in Figures 7, 10 and 14 respectively.

It can be observed that DHST generates many unpleasant and abnormal clusters of dots over the image. Consider the halftone image by ordered dithering. In Figure 6, the original halftone image using ordered dithering is not of very high visual quality as compared with the other halftone images in Figures 9 and 13. The halftone image shown in Figure 7 with data hidden using DHST has similar contrast as Figure 6, as expected. However, there are a lot of 'salt and pepper' noise in Figure 7, some black and some white. These clusters are created because DHST forces some pixels to be white or black regardless of the image content. As a result, large clusters of blacks pixels are created as some white pixels are forced to be black, and vice versa. Such forced self-toggling changes the local average intensity making the clusters visually disturbing. Similar comments can be made on the error diffused images with data hidden using DHST in Figures 10 and 14.

Here we propose an algorithm, the Data Hiding Pair Toggling (DHPT), to improve on the DHST. The major problem of DHST is the abrupt change of average local intensity due to the forced self-toggling. Instead of perform only one forced toggling, we propose to perform a pair of complementary forced toggling. For example, when a black pixel at the pseudo-random location is required to self-toggle from black to white, a neighboring white pixel is chosen randomly to self-toggled from white to black at the same time. Although two errors are introduced instead of one, the two complementary errors (one positive and one negative) tend to mask out each other. In particular, with the complementary pair toggling, the local average intensity can be preserved and this should give much better visual quality than DHST. If there are M white pixels in the 3x3 neighborhood, one of the M white pixels is chosen randomly. In the special case that M equals zero (i.e. all pixels within the 3x3 neighborhood are black), no complementary toggling is performed. In other words, in this special case, only one self-toggling is performed. This should be a very rare case occurring only when the local image content is completely black or completely white.

The exact algorithm of Data Hiding Pair-Toggling (DHPT) is

1. Select N pseudo random locations using the key (e.g. N may be such that there is one location in each 8x8 block).

2. For each pseudo random location, if the halftone pixel value matches the hidden data bit (black for bit 0 and white for bit 1), do nothing.
3. If the halftone pixel value does not match the hidden data bit, perform self-toggling to change its value and perform complementary toggling on one of the pixels in the 3x3 neighborhood. If more than one complementary toggling candidate exist, choose one randomly. If no complementary toggling candidate exists, the complementary toggling is not performed.

The incremental complexity of DHPT over DHST should be minimal. As the data to be hidden is statistically independent of the halftone pixel values, typically half of the halftone pixel values at the predefined pseudo-random locations would be the same as the intended values and thus no change of the halftone pixel values would be necessary, similar to DHST. The incremental complexity of DHPT over DHST is due to the complementary toggling at the remaining half of the pixels at the pseudo-random locations.

DHPT is applied to the halftone images of Lena in Figures 6, 9 and 13. Again one bit of information is hidden in every 8x8 blocks such that there are a total of 4096 bits (or 1.526%) hidden in the 512x512 halftone images. Compared with the corresponding images with data embedded using DHST, these images have significantly improved visual quality over DHST, with fewer 'salt-and-pepper' noise artifacts. This is particularly obviously in the case of error diffusion using the Jarvis kernel.

Although DHPT can achieve good visual quality in halftone images by ordered dithering and error diffusion with coarse texture (Jarvis kernel), it does not work as well in error diffused halftone image with fine texture (Steinberg kernel). Figure 14 and Figure 15 are the results of DHST and DHPT on error diffused halftone images generated by Steinberg kernel. Unpleasant artifacts are still created after DHPT, though significantly fewer than in DHST. This suggests that fine textures are more susceptible to distortions due to the data hiding process. Although the local average intensity is preserved in DHPT, generation of clusters is still unavoidable in DHPT. Since the clusters by DHPT are of smaller size than those by DHST, they can be masked quite effectively by the coarse textures from ordered dithering and error diffusion with Jarvis kernel. However, the clusters cannot be masked completely by the fine textures from error diffusion with Steinberg. These unnatural artifacts are still perceptually disturbing to the human eyes. Therefore, further improvement is required. In order to improve the quality further, we propose the Data Hiding Error Diffusion (DHED).

3. Data Hiding Error Diffusion (DHED)

In the previous situation, only the halftone image is available for data hidden, but not the original multi-tone image nor the halftoning method. If the original image and the halftoning method are available, we can do better than DHPT. Here we focus on the situation in which the original image is available and the method of halftoning is error diffusion. We will integrate the data hiding operation into the error diffusion operation to obtain improved visual quality. The proposed method is called Data Hiding Error Diffusion (DHED), which is only slightly more complicated than the regular error diffusion.

When forced toggling is performed at the pseudo-random locations in DHST, the errors are not compensated in any way and thus the visual quality of DHST is bad. In DHPT, the complementary toggling is used to compensate for the error so as to preserve the local average intensity. The resulting visual quality of DHPT is significantly improved. If the original image is known and the halftoning method is error diffusion, the error due to the forced toggling can actually be diffused to the neighboring pixels to obtain even better visual quality.

The exact algorithm of Data Hiding Error Diffusion (DHED) is
1. Select N pseudo random locations using the key (e.g. N may be such that there is one location in each 8x8 block).
2. For the N pseudo random locations, forced self-toggling is applied according to the hidden data. The error due to the self-toggling is recorded for error diffusion to the surrounding pixels.
3. Perform regular error diffusion if the location is not one of the N pseudo random locations. The errors due to halftoning and self-toggling are diffused to the neighboring pixels according to the kernel being used.

The complexity of DHED is similar, if not less than, the regular error diffusion. The same kernel of error diffusion can be used. As the error due to self-toggling is diffused to the neighboring pixels, no significantly error clusters should appear in the data hidden halftone images. The DHED is applied to the two error diffused halftone images in Figures 9 and 13. The resulting images are shown in Figures 12 and 16. Figures 12 and 16 are halftone image after DHED by Jarvis kernel and Steinberg kernel respectively. Visually, the artifacts are almost masked out. DHED can generate higher quality image than DHPT.

When the original multi-tone images are not available, the halftone image can undergo inverse halftoning [8,9] first to reconstruct a multi-tone image, and then DHED can be applied. In this way, the halftone image would not suffer from the 'salt-and-pepper' noise. However, as both halftoning and inverse halftoning are lossy process, some details of the images can be lost. We have performed experiments to try this out and found that the resulting halftone images show considerable loss in contrast compared with the initial halftone image.

4. Simulation Results

The Peak Signal-to-Noise Ratio (PSNR) between a halftone image and its original multi-tone image is not a good measure of subjective quality because, while a halftone image may resemble its original multi-tone image very well when viewed from a distance, it takes on intensity values of 0 and 255 only and thus would be very different from the original image resulting in a very low PSNR always.

In order to quantify the visual quality of halftone images, we propose to use the Modified Peak Signal-to-Noise Ratio (MPSNR). The halftone image is filtered with a lowpass filter before the normal PSNR is computed, as shown in Figure 4. Notice that the MPSNR used here is only a rough indicator of the image quality and it depends on the lowpass filter used. It is not a substitute for subjective evaluation of the halftone images using the human eyes. The MPSNR of halftone images with various halftoning methods and data hiding methods are listed in Table 1. As mentioned before, the corresponding images are shown in Figures 5 to 16.

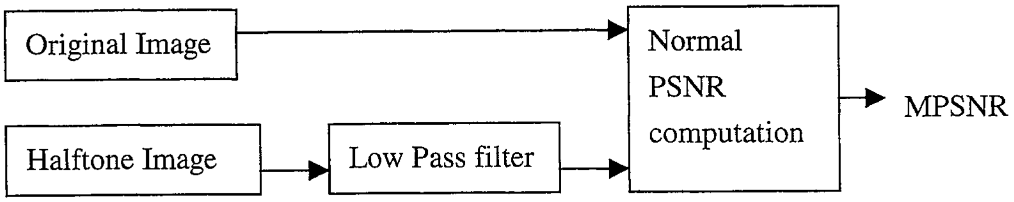

Figure 4 Definition of Modified PSNR (MPSNR)

Figures 6, 9 and 13 are halftone images without data hiding. Figure 6 is generated by ordered dithering, the simpler halftoning technique. It looks worse than Figures 9 and 13 and its MPSNR is only 27.21dB. Figures 9 and 13 are generated by error diffusion, the more sophisticated technique. Figure 9 is generated with the Jarvis kernel. It produces coarse texture and creates higher contrast. The MPSNR is 29.76 dB. Figure 13 is generated with the Steinberg kernel. It produces fine texture and maintains good contrast similar to that of the original image, and its MPSNR is the highest, being 30.69 dB.

Figures 7, 10 and 14 are halftone images with 1.5625% data hidden using DHST. Their MPSNR degradations compared with the corresponding halftone images without data hiding are 0.74 dB for Figure 7, 1.39 dB for Figure 10 and 1.73dB for Figure 14. Visually, there are many unpleasant black and white clusters all over the images. These clusters are formed by dots of same color clustered together in an unnatural manner in DHST.

Figure 8, 11 and 15 are halftone images with 1.5625% data hidden using DHPT. Their MPSNR degradations compared with the corresponding halftone images without data hiding are 0.28 dB for Figure 8, 0.62 dB for Figure 11 and 0.75 dB for Figure 15. Visually, the unpleasant white and black clusters are greatly removed in Figure 8 and 11. This is mainly because the size of those clusters is decreased, becoming comparable to the coarse texture formed by order dithering and Jarvis kernel. However, in Figure 15, due to the fine texture nature of Steinberg Kernel, those smaller clusters still cannot totally merge into the texture of the halftone image. As a result, some of them are still visible. Still, the clusters are greatly suppressed when Figure 15 is compared with Figure 14.

Figure 12 and 16 are halftone images with 1.5625 data hidden by DHED. Their MPSNR degradations compared with the corresponding halftone images without data hiding are 0.77 dB for Figure 12 and 0.19 dB for Figure 16. With DHED, the white and black clusters are greatly removed. Note that the MPSNR of Figure 16 is much larger than that of Figure 15 and is actually similar to that of Figure 13. DHED is better than DHPT for error diffused halftone image with Steinberg kernel. Although MPSNR of Figure 12 is slightly lower than that of Figure 11, it still looks nice and the black and white clusters are nearly invisible. For error diffused halftone image using the Jarvis, the DHPT is sufficiently good.

5. Conclusion

In this paper, we propose two novel methods for hiding data in halftone images, the Data Hiding Pair-Toggling (DHPT) and the Data Hiding Error Diffusion (DHED). Both methods are capable of hiding a relatively large amount of data while retaining high visual quality.

The proposed DHPT can hide data when the original multi-tone image is not available while the proposed DHED requires the additional knowledge of the original multi-tone image and the halftoning method. From our simulation results, DHPT is found to be very effective for halftone images with relatively coarse textures, such as those generated by ordered dithering and error diffusion with Jarvis kernel. DHED is found to be better for halftone images with fine textures, such as those generated by error diffusion with the Steinberg kernel.

6. References

[1] R. A. Ulichney, "*Digital Halftoning*." Cambridge, MA: MIT Press, 1987.

[2] B. E. Bayers, "An Optimum Method for Two Level Rendition of Continuous Tone Pictures," *Proc. of IEEE Int. Communication Conf.*, pp2611-2615, 1973.

[3] R.W. Floyd and L. Steinberg, "An Adaptive Algorithm for Spatial Grayscale," *Proc. SID*, pp. 75-77, 1976.

[4] M.D. Swanson, B. Zhu and A.H. Tewfik, "Multiresolution Scene-Based Video Watermarking Using Perceptual Models", *IEEE Journal on Selected Areas in Communication*, vol. 16, no. 4, pp 540-550, May 1998.

[5] S. Craver, N. Memon, B.L Yeo and M. Yeung, "Resolving Rightful Ownerships with Invisible Watermarking Technique: Limitation, Attacks, and Implications", *IEEE Journal on Selected Areas in Communication*, vol. 16, no. 4, pp 573-586, May 1998.

[6] I.J. Cox, et al., "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Trans. of Image Processing*, Vol. 6, No. 12, pp. 1673-87, Dec 1997.

[7] Z. Baharav, D. Shaked, "Watermarking of Dither Halftoned Images", *Proc. of SPIE Security and Watermarking of Multimedia Contents*, pp. 307-313, Jan 1999.

[8] Z. Xiong, K. Ramchandran and M. Orchard, "Inverse Halftoning Using Wavelets", *Proc. Of IEEE Inl. Conf. On Image Processing*, vol. I, pp 569-572, Oct. 1996.

[9] M. S. Fu, O.C. Au, "Hybrid Inverse Halftoning using Adaptive Filtering", *Proc. of IEEE Int. Sym. on Circuits and Systems*, vol. 4, pp. 259-262, May 99.

MPSNR	No Data Hiding	DHST	DHPT	DHED
Order Dithering	27.21dB	26.46dB	26.93dB	
Error Diffusion (Jarvis kernel)	29.76dB	28.37dB	29.13dB	28.99dB
Error Diffusion (Steinberg kernel)	30.69dB	28.96dB	29.94dB	30.50dB

Table 1 MPSNR of Low passed halftone images

Figure 5 Original multi-tone Lena

Figure 6 Dithered Halftone Lena without Data Hiding (27.2091 dB)

Figure 7 Dithered Halftone Lena with Data Hiding by DHST (26.4647 dB)

Figure 8 Dithered Halftone Lena with Data Hiding by DHPT (26.9341 dB)

Figure 9 Jarvis Error Diffused Halftone Lena without Data Hiding (29.7552 dB)

Figure 10 Jarvis Error Diffused Halftone Lena with Data Hiding by DHST (28.3651 dB)

Figure 11 Jarvis Error Diffused Halftone Lena with Data Hiding by DHPT (29.1317 dB)

Figure 12 Jarvis Error Diffused Halftone Lena with Data Hiding by DHED (28.9876 dB)

Figure 13 Steinberg Error Diffused Halftone Lena without Data Hiding (30.6934 dB)

Figure 14 Steinberg Error Diffused Halftone Lena with Data Hiding by DHST (28.9590 dB)

Figure 15 Steinberg Error Diffused Halftone Lena with Data Hiding by DHPT (29.9472 dB)

Figure 16 Steinberg Error Diffused Halftone Lena with Data Hiding by DHED (30.5008 dB)