Image Hiding Algorithms based on Halftoning Technique

Xia Zhuge, Koji Nakano

Department of Information Engineering, Hiroshima University, Kagamiyama, Higashi-Hiroshima 739-8527, Japan

Abstract: Halftoning technique is used to convert a continuous-tone image into a binary image with pure black and white pixels. This technique is necessary when printing or displaying a monochrome or color image by a device with limited color levels. The main contribution of this paper is to present a halftoning method that conceals a small binary image into a large binary image. More specifically, two distinct gray scale images are given, such that the smaller one of them should be hidden in another larger gray scale image. Our halftoning method generates two binary images that reproduce the tone of the corresponding original two gray scale images. Each pixel of the small binary image is hidden into some pixel of the large binary image through our halftoning method. The small hidden image can be seen when we pick out the pixels of the large binary image at premeditated locations, or we cannot see the hidden image if we have no location information. Another contribution of this paper is to extend our halftoning method to hide a small image of any size into a corresponding large size image. The resulting images show that our halftoning method hides and recovers the original images. Hence, our halftoning technique can be used for watermarking as well as amusement purpose.

Key words: Digital Halftoning, Error Diffusion, Image Hiding, Watermarking.

1. Introduction

Halftoning is an important process to convert a continuous-tone image into a binary image which takes either black pixels or white pixels and is perceived as a continuous-tone image when seen by human eyes [1]. This process is necessary when printing or displaying a monochrome or color image by a device with limited number of color levels. It is required to generate a binary image that reproduces the tone and the details of the original gray scale image. Many kinds of halftoning methods have been proposed: screening [2], error diffusion [3], and search-based iterative methods [4-7]. In screening, the binary values of the halftone image are obtained by comparing pixels values of the continuous-tone image with the corresponding threshold values in a screen. In error diffusion, the pixel-by-pixel comparison with a threshold is also required, but after a pixel is halftoned, the rounding error is propagated to its unprocessed neighboring pixels according to some fixed ratios. Search-based iterative methods are to find the best values of binary pixels which can minimize a perceived error between the continuous-tone image and the halftone image. Error Diffusion, as one of the most famous halftoning methods, is comparatively simple and of good quality.

Digital watermarking is to embed messages into digital contents (audio, video, images, text) which can be detected or extracted later. One use of digital watermarking is to embed copyright information of the content which is supposed not to be visible, but can be retrieved by electronic devices [8]. Another use of digital watermarking is known as steganography, which hides messages in content without typical citizens or public authorities noticing its presence, and just special recipients can decode the hidden messages. Many methods for digital watermarking about halftone images have been proposed. One kind of these methods is to embed an invisible watermark into

Corresponding author: Xia Zhuge(1979-), graduate student, research fields: image processing. Email: zhugexia@cs.hiroshima-u.ac.jp.

Koji Nakano (1964-), Ph.D., professor, research fields: image processing, parallel processing, and embedded systems.

a halftone image and the hidden information can be extracted with some special procedure [9-12]. Another kind of digital watermarking methods is to embed information in two or more than two halftone images and the hidden information can be seen by overlapping the public images [13-14].

The main contribution of this paper is to present an Error-Diffusion-based halftoning method for two images, say, A and B. A is the smaller image and B is the larger image which is four or more than four times of image A in size. Our halftoning method generates two binary images A' and B' that reproduce original gray scale images A and B respectively. Further, smaller binary image A' is hidden in larger binary image B' by the way that each pixel of image A' is hidden into pixels of image B' at some premeditated locations. We can see image A', if we pick out pixels from image B' at these locations. In Figure 1, we show an example. Image B' is four times of image A' in size, and each pixel of A' is hidden in B' at odd row and odd column locations. Image A' can be seen when we get pixels from image B' at odd row and odd column locations or we can see A' by downsizing image B' in the corresponding way.

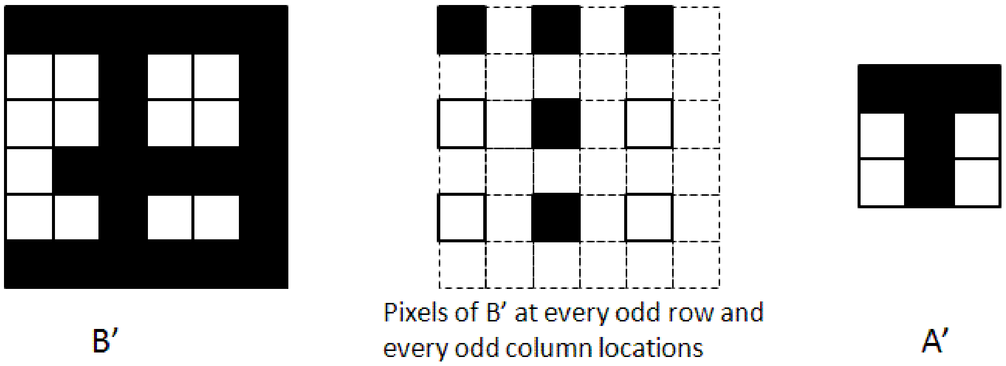


Fig. 1 Two binary images A' and B' satisfying Condition 2.

There are several possible applications for our halftoning method. For example, it can be used for watermarking as follows. Image B' is a public image and image A' is an image to be hidden. Image A' can be a hidden image that indicates the copyright information of image B'. Also, our halftoning method can be used for amusement purpose, like picture puzzle. One piece of the picture puzzle denotes one pixel of B'. When we pick out pieces from B' at premeditated locations, we can compose image A'.

Our second contribution is to extend our halftoning method to hide a small image of any size into a large image with a size any integer times of the small image.

This paper is organized as follows. Section 2 reviews the Error Diffusion method and proposes the idea of our halftoning method. Section 3 shows our Error-Diffusion-based halftoning algorithm. Section 4 gives our experiment results. Section 5 extends our halftoning method to generate halftone images of any size. Section 6 offers concluding remarks.

2. Halftoning and Image Hiding

Suppose that an original gray scale image $A=(a_{i,j})$ of size $n \times n$ is given, where $a_{i,j}$ denotes the intensity level at position $(i, j)(0 \le i,j \le n-1)$ taking a real number intensity in the range [0, 1]. Although we assume that images are square for simplicity, it is easy to generalize them to non-square images. The goal of halftoning is to find a binary image $A'=(a'_{i,j})$ of the same size that reproduces the original image A, where each $a'_{i,j}$ is either 0(black) or 1(white).

One of the well known halftoning methods is Error Diffusion, which propagates the quantization errors to unprocessed neighboring pixels according to some fixed ratios. Pixels in the image are proposed in a raster order, that is, from left to right and top to bottom. The intensity level of each pixel is compared with a fixed threshold 0.5. It is rounded down to 0 if it is no more than the threshold, and rounded up to 1 otherwise. In other words, the binary value of $a'_{i,j}$ is determined as follows:

$$a'_{i,j} = \begin{cases} 1, & \text{if } a_{i,j} \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Clearly, the rounding error is $e=a_{i,j}- a'_{i,j}$. Note that the threshold value 0.5 is selected such that the absolute value of the rounding error $| a_{i,j} - a'_{i,j} |$ is minimized. This point of view is used in our Error-Diffusion-based algorithm later.

The key idea of Error Diffusion is to compensate the rounding error $e=a_{i,j}- a'_{i,j}$ to diffuse it over the unprocessed pixels around it with fixed ratios. Floyd and Steinberg [3] proposed a ratio matrix as follows:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{7}{16} \\ \frac{3}{16} & \frac{5}{16} & \frac{1}{16} \end{pmatrix}$$

Using this matrix, the intensity levels of neighboring pixels are updated such that $a_{i,j+1} \leftarrow a_{i,j+1}+e\times 7/16$, $a_{i+1,j-1} \leftarrow a_{i+1,j-1}+e\times 3/16$, $a_{i+1,j} \leftarrow a_{i+1,j}+e\times 5/16$, $a_{i+1,j+1} \leftarrow a_{i+1,j+1}+e\times 1/16$. Since the sum of the elements in the matrix is 1, the error *e* is proposed to neighboring four pixels.

Suppose that we are given two gray scale images $A=(a_{i,j})(0\leq i,j\leq n-1)$, $B=(b_{k,l})(0\leq k,l\leq 2n-1)$. Image A has size n×n and image B has size 2n×2n. Our goal is to generate corresponding binary images $A'=(a'_{i,j})$ and $B'=(b'_{k,l})$ such that:

Condition 1 binary images A' and B' reproduce the tone of the original gray scale images A and B, and

Condition 2 binary image A' is hidden in binary image B' by the way that each pixel of A' is hidden in B' at odd row and odd column locations. In other words, $a'_{i,j}=b'_{2i,2j}$ for all i and j.

Fig.1 shows an example of two binary images A' and B' satisfying Condition 2.

3. Our Error-Diffusion-Based Algorithm

The main purpose of this section is to show our algorithm that generates two binary images A' and B' satisfying conditions 1 and 2 for given original gray scale images A and B. Our algorithm consists of two steps as follows:

Step 1 Adjust the intensity levels of original gray scale images A and B.

Step 2 Using our new Error-Diffusion-based technique, generate two binary images A' and B' from adjusted gray scale images of A and B.

Adjusting Intensity Levels of Original Gray Scale Images

We will show how the intensity levels of original gray scale images are adjusted so that every pixel of A' should be hidden in B' at odd row and odd column locations. Suppose binary image A' has size n×n and binary image B' has size 2n×2n. Let $R_S(A')$ be a small region of m pixels in image A'. Let $R_L(B')$ be the corresponding small region of 4m pixels in image B'. The m pixels of $R_S(A')$ have the same values as that pixels in $R_L(B')$ at odd row and odd column locations.

Suppose $R_S(A')$ has white pixels in the range $[\alpha, \beta]$ $(0\leq\alpha<\beta\leq m)$, that is, the range of intensity level is $[\alpha/m, \beta/m]$. We know that all of these white pixels should be hidden in $R_L(B')$ at odd row and odd column locations. Thus we need to guarantee that $R_L(B')$ has white pixels in the range $[\beta,4m-(m-\alpha)]=[\beta,3m+\alpha]$ ($(m-\alpha)$ is the number of black pixels in $R_L(B')$ at odd row and odd column locations), that is, the range of intensity is $[\beta/4m,3/4+\alpha/4m]$. Note that we can choose any values of α and β satisfying $0\leq\alpha<\beta\leq m$. Let us show some examples how we choose values of α and β.

(1) To maximize the range of black pixels in $R_S(A')$, we can choose $\alpha=0$ and $\beta=m$. If this is the case, the range of intensity of $R_S(A')$ is [0,1] and that of

R_L(B') is [1/4,3/4]=[0.25,0.75]. In this case we perform linear conversion of the intensity value x such that function GL(x)=x/2+1/4 is used for adjusting the intensity level of image B, and GS(x)=x is used for image A. In Fig.2 we show a set of examples where images B and A are adjusted by these functions respectively.

(2) We can equalize the ranges of the intensity of R_S(A') and R_L(B'). If this is the case we have β/m-α/m=3/4+α/4m-β/4m. Hence we have β-α=3m/5. If we choose β=4m/5 and α=m/5, both of the ranges of intensity for R_S(A') and R_L(B') is [1/5,4/5]=[0.2,0.8]. In this caseb we perform linear conversion of the intensity value x such that function GL(x)=3x/5+1/5 is used for adjusting intensity level of image B, and function GS(x)=3x/5+1/5 is used for image A. In Fig.3 we show the examples where images B and A are adjusted using these functions respectively.

(3) We may want to maximize the range of intensity of R_L(B') even if that of R_S(A') is small. For example, we set β=3m/5 and α=2m/5, the range of intensity of R_S(A') is [2/5,3/5]=[0.4,0.6] and that of R_L(B') is [3/20,17/20]=[0.15,0.85]. In this case we perform linear conversion of the intensity value x such that function GL(x)=7x/10+3/20 is used for adjusting intensity level of image B, and function GS(x)=x/5+2/5 is used for image A. In Fig.4 we show a set of examples where the original gray scale images A and B are adjusted by these two functions respectively.

Using an Error-Diffusion-Based Algorithm for Generating Two Binary Images

In this section we show our halftoning method based on Error Diffusion. We assume that the intensity levels of three original gray scale images have been adjusted using some linear conversions as we discussed in the previous subsection. Let A=($a_{i,j}$)(0≤i,j≤n-1) and B=($b_{k,l}$)(0≤k,l≤2n-1) be the adjusted gray scale images. We will show how halftone images A'=($a'_{i,j}$)(0≤i,j≤n-1) and B'=($b'_{k,l}$)(0≤k,l≤2n-1) are computed using our new Error-Diffusion-based halftoning algorithm.

Similar to the conventional Error Diffusion algorithm, pixels in binary images A' and B' are determined in raster order. Pixels in image B' are determined in two different ways according to locations: pixels at odd row and odd column locations $b'_{k,l}$ (k mod 2=0 and l mod 2=0) are determined jointly with pixels of image A'; pixels in image B' at other locations $b'_{k,l}$ (k mod 2≠0 or l mod 2≠0) are determined in the same way as that of conventional Error Diffusion as we introduced in Section 2. In the following part we will show how the pixels of image A' and pixels of image B' at odd row and odd column locations are determined. Recall that, in conventional Error Diffusion algorithm, the value of binary pixel is determined such that the rounding error is minimized. Our key idea is to extend this policy for two binary images such that binary values of $a'_{i,j}$ and $b'_{2i,2j}$ (0≤i,j≤n-1) are selected to minimize the total rounding error. The details are spelled out as follows. It should be clear that $a'_{i,j}$ =$b'_{2i,2j}$ holds to satisfy Condition 2. Let the total error be

$$E(a'_{i,j}, b'_{2i,2j}) = |a_{i,j} - a'_{i,j}| + |b_{2i,2j} - b'_{2i,2j}| \quad (2)$$

We select the value of $a'_{i,j}$ (i.e. $b'_{2i,2j}$) that minimizes the total error E($a'_{i,j}$, $b'_{2i,2j}$). We have the rounding errors e_a= $a_{i,j}$- $a'_{i,j}$ and e_b= $b_{2i,2j}$ –$b'_{2i,2j}$ for gray scale images A and B. We diffuse these rounding errors e_a and e_b to neighboring pixels using Floyd and Steinberg's ratio matrix. More specifically, we perform the following operations for image A: $a_{i,j+1}$←$a_{i,j+1}$+e_a×7/16, $a_{i+1,j-1}$←$a_{i+1,j-1}$+e_a×3/16, $a_{i+1,j}$←$a_{i+1,j}$+ e_a×5/16, $a_{i+1,j+1}$←$a_{i+1,j+1}$+ e_a×1/16 and do the same for image B: $b_{2i,2j+1}$←$b_{2i,2j+1}$+e_b×7/16, $b_{2i+1,2j-1}$←$b_{2i+1,2j-1}$+ e_b×3/16, $b_{2i+1,2j}$←$a_{2i+1,2j}$+ e_b×5/16, $b_{2i+1,2j+1}$←$b_{2i+1,2j+1}$+ e_b×1/16.

4. Experimental Results

In this section we give three sets of our experimental results about hiding the small image A' in the large image B' according to the three examples of linear conversions we have discussed in Section 3. In these examples, linear function GL(x) is used for adjusting the intensity value x of the larger original gray scale image B and linear function GS(x) is used for adjusting the intensity value x of the smaller original gray scale image A. In Fig.3 linear functions GL(x)=x/2+1/4 and GS(x)=x are used for adjusting B and A respectively. In Fig.4 images A and B are adjusted by linear functions GS(x)=3x/5+1/5 and GL(x)=3x/5+1/5 respectively. In Fig.5 linear functions GS(x)=x/5+2/5 and GL(x)=7x/10+3/20 are used for adjusting A and B respectively. In these examples, binary image A' has size of 256×256 and image B' has the size of 512×512. Binary image B' is obtained by our Error-Diffusion-based halftoning algorithm directly. By picking out the pixels of B' at odd row and odd column locations, we compose binary image B'. From these results, we can see our halftoning algorithm can hide a small image in a large image successfully and flexibly, and the hidden image can be achieved clearly.

5. Our Halftoning Method for Generating Two Binary Images of Free Size

The main purpose of this section is to extend our halftoning method to generate a small binary image and a large binary image of which the image size is any integer times of that of the small image. Suppose $S=(s_{i,j})$ $(0\le i,j\le n-1)$ and $G=(g_{k,l})$ $(0\le k,l\le pn-1)$ are the small gray scale image and large gray scale image of sizes n×n and pn×pn respectively. Our purpose is to generate two binary images $S'=(s'_{i,j})$ $(0\le i,j\le n-1)$ and $G'=(g'_{k,l})$ $(0\le k,l\le pn-1)$. Condition 1 and Condition 2 given in Section 2 should be modified as follows:

Condition 1 binary images S' and G' reproduce the tone of the original gray scale images S and G, and

Condition 2 binary image S' is hidden in binary image G' by the way that each pixel of S' is hidden in G' at locations (pi, pj).

In order to satisfy Conditions 1 and 2, we need to do the following two steps:

Step 1 Adjust the intensity levels of gray scale images S and G. Similar to the way that we have discussed in Section 3, suppose the small region $R_S(S')$ of m pixels in image S' and the corresponding region $R_L(G')$ of p^2m pixels in image G', $R_S(S')$ has white pixels in the range $[\alpha, \beta]$ $(0\le\alpha<\beta\le m)$ and $R_L(G')$ should have white pixels in the range $[\beta, p^2m-(m-\alpha)]$ to satisfy Condition 2. Thus the intensity level of S is in the range $[\alpha/m, \beta/m]$ and that of G is in the range $[\beta/(p^2m), (p^2-1)/p^2+ \alpha/p^2m]$. We can choose any α and β that satisfy $0\le\alpha<\beta\le m$. For example, we take α=0 and β=m to make the intensity of S in the range [0, 1] and that of G in the range $[1/p^2, (p^2-1)/p^2]$. In this case we perform the linear conversion of the intensity x such that function $GL(x)=(p^2-2)x/p^2+1/p^2$ is used for adjusting intensity levels of G and function GS(x)=x is used for adjusting S.

Step 2 Using our Error-Diffusion-based technique, generate binary images S' and G' from the adjusted gray scale images S and G. Pixels in G' are determined in two different ways according to their locations: pixels $g'_{pi,pj}$ are determined jointly with pixels of S'; the other pixels are determined in the same way as the conventional Error Diffusion. Similarly we extend the error minimization policy for two binary images such that binary values of $s'_{i,j}$ and $g'_{pi,pj}$ $(0\le i,j\le n-1)$ are selected to minimize the total rounding error. It should be clear that $s'_{i,j} = g'_{pi,pj}$ holds to satisfy Condition 2. Let the total error be

$$E(s'_{i,j}, g'_{pi,pj}) = \left|s_{i,j} - s'_{i,j}\right| + \left|g_{pi,pj} - g'_{pi,pj}\right| \quad (3)$$

We select the value of $s'_{i,j}$ (i.e. $g'_{pi,pj}$) that minimizes the total error $E(s'_{i,j}, g'_{pi,pj})$.

If we take p=2, this is the case we discussed in Section 3.

In Fig.5 we show examples of two binary images S' (128×128) and G' (512×512) generated by our Error-Diffusion-based algorithm , where n=128, p=4, and we use the adjusting functions listed in the above Step 1 in this Section: GL(x)=7x/8+1/16 for G and GS(x)=x for S. Binary image G' is generated directly by our Error-Diffusion-based algorithm, and S' is obtained by picking out pixels $g'_{pi,pj}$. We can see the contrast of sailboat is improved significantly than that given in Fig. 2.

6. Concluding Remark

In this paper, we have presented an Error-Diffusion-based algorithm that conceals a small binary image into a large binary image of which the size is four times of that of the small hidden image, and have extended this algorithm to hide a small image in a large image of free size. Our image hiding method can hide and recover images well, and as the size difference of the two images increases, the contrast of public images improves.

References

[1] D. Lau and G. Arce. Modern Digital Halftoning. Marcel Dekker, 2001.

[2] B. E. Bayer. An optimum method for two-level rendition of continuous-tone pictures. IEEE international conference on communications, vol.26, pp.11-15, 1973.

[3] R. Floyd and L. Steinberg. An adaptive algorithm for spatial gray scale. SID 75 Digest, Society for information display, pages 36-37, 1975.

[4] M. Analoui and J. Allebach. Model-based halftoning by direct binary search. In Proc. SPIE/IS&T Symposium on Electronic Imaging Science and Technology, volume 1666, pages 96-108, 1992.

[5] Y. Ito and K. Nakano. FM screening by the local exhaustive search, with hardware acceleration. In Proc. of Workshop on Advances in Parallel and Distributed Computational Models (CD-ROM of International Parallel and Distribution Processing Symposium), 2004.

[6] Y. Ito and K. Nakano. FM screening by the local exhaustive search, with hardware acceleration. International Journal of Computer Science, pages 89-104, February, 2005.

[7] X. Zhuge, Y. Hirano and K. Nakano. A new hybrid multitoning based on the direct binary search. In Proc. of International Multi Conference of Engineers and Computer Scientists, volume 1, pages 627-632, March, 2008.

[8] S. Katzenbeisser and F. Petitcolas (Editors). Information hiding techniques for steganagraphy and digital watermarking. Artech House, 1999.

[9] J. M. Guo, M. F. Wu and Y. C. Kang. Watermarking in conjugate ordered dither block truncation coding images. Signal Processing, ELSEVIER, vol.89, pp.1864-1882, 2009.

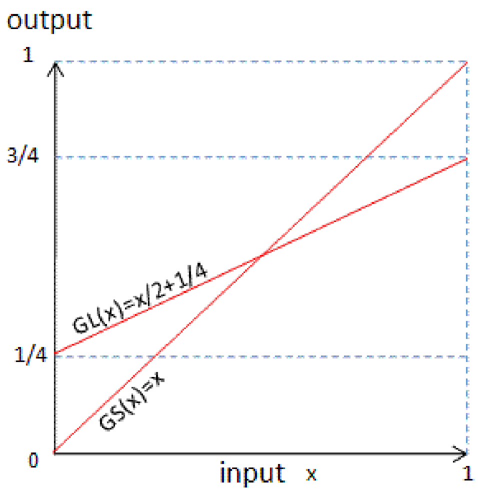
[10] Z. Baharav and D. Shaked. Watermarking of dither halftoned images. IS&T/SPIE Conference, Security and Watermarking of Multimedia Contents, 3957: 307-316, January, 1999.

[11] S. Wang and K. Knox. Embedding digital watermarks in halftone screens. Proceedings of SPIE, Security and watermarking of multimedia contents II, 3971: 218-227, 2000.

[12] D. Kacker and P. Alleach. Joint halftoning and watermarking. IEEE Transactions on Signal Processing, 51 (4): 1054-1068, April, 2003.

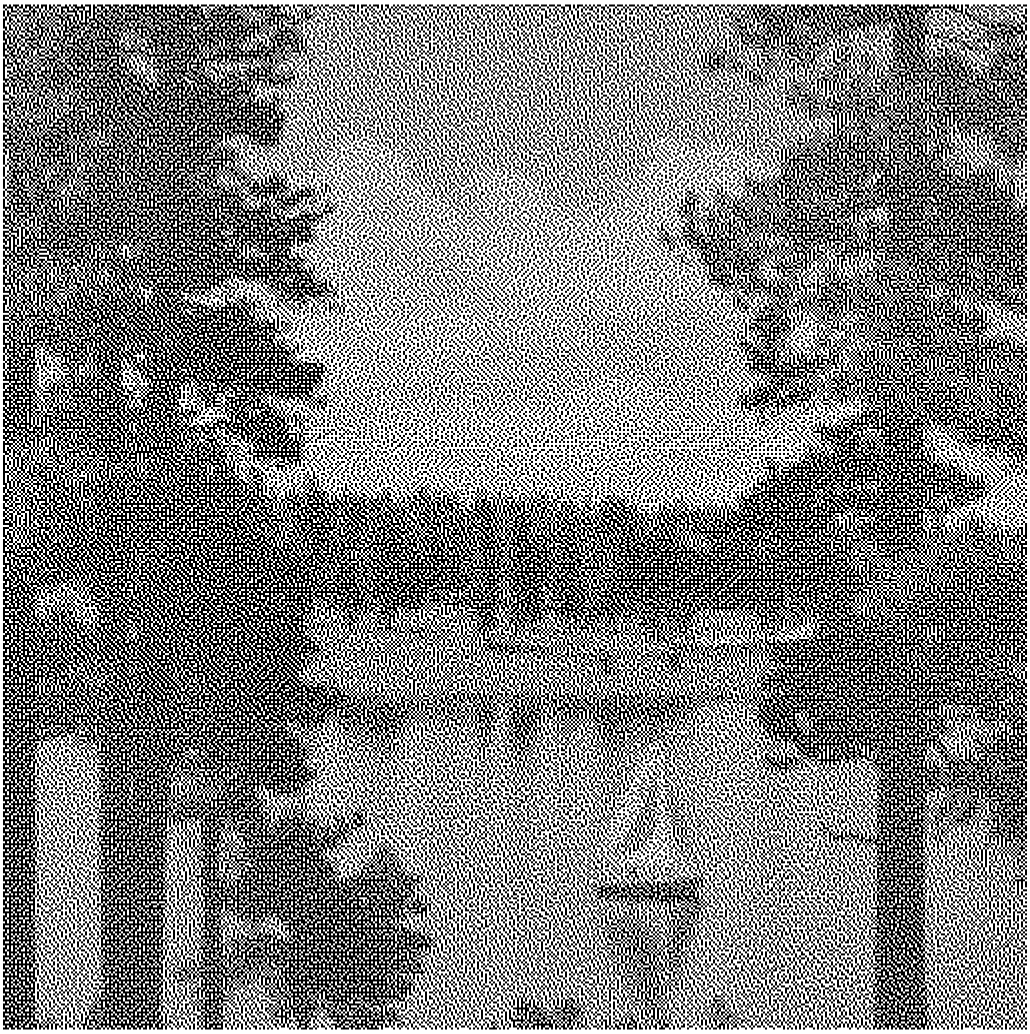
[13] S. Pei and J. Guo. Data hiding in halftone images with noise-balanced error diffusion. IEEE Signal Processing letters, 10 (12): 349-351, December, 2003.

[14] X. Zhuge and K. Nakano. Halftonng based algorithms for image hiding. Journal of communication and computer, vol.6, No.9, pp.39-45, Sept. 2009.



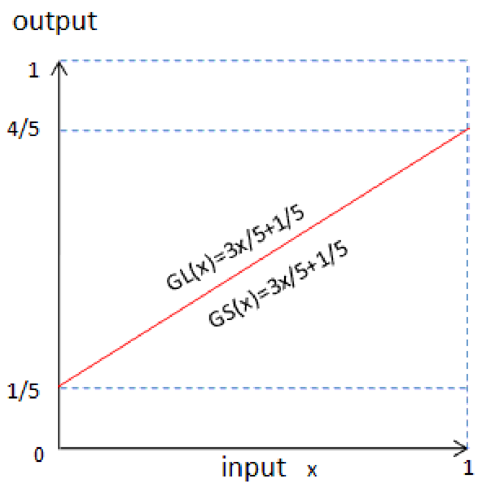
Linear function used for images A and B

Binary image A': airplane 256×256

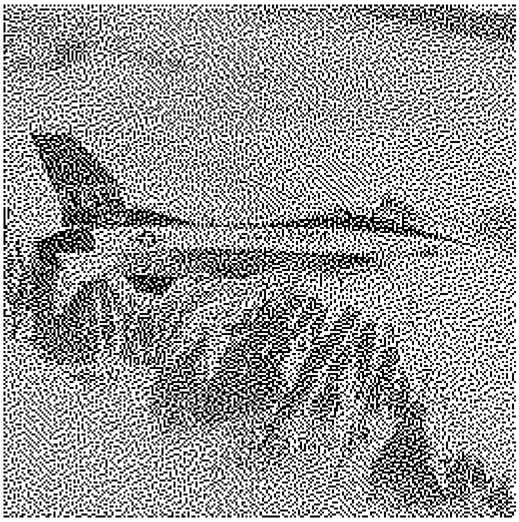


Binary image B': sailboat 512×512

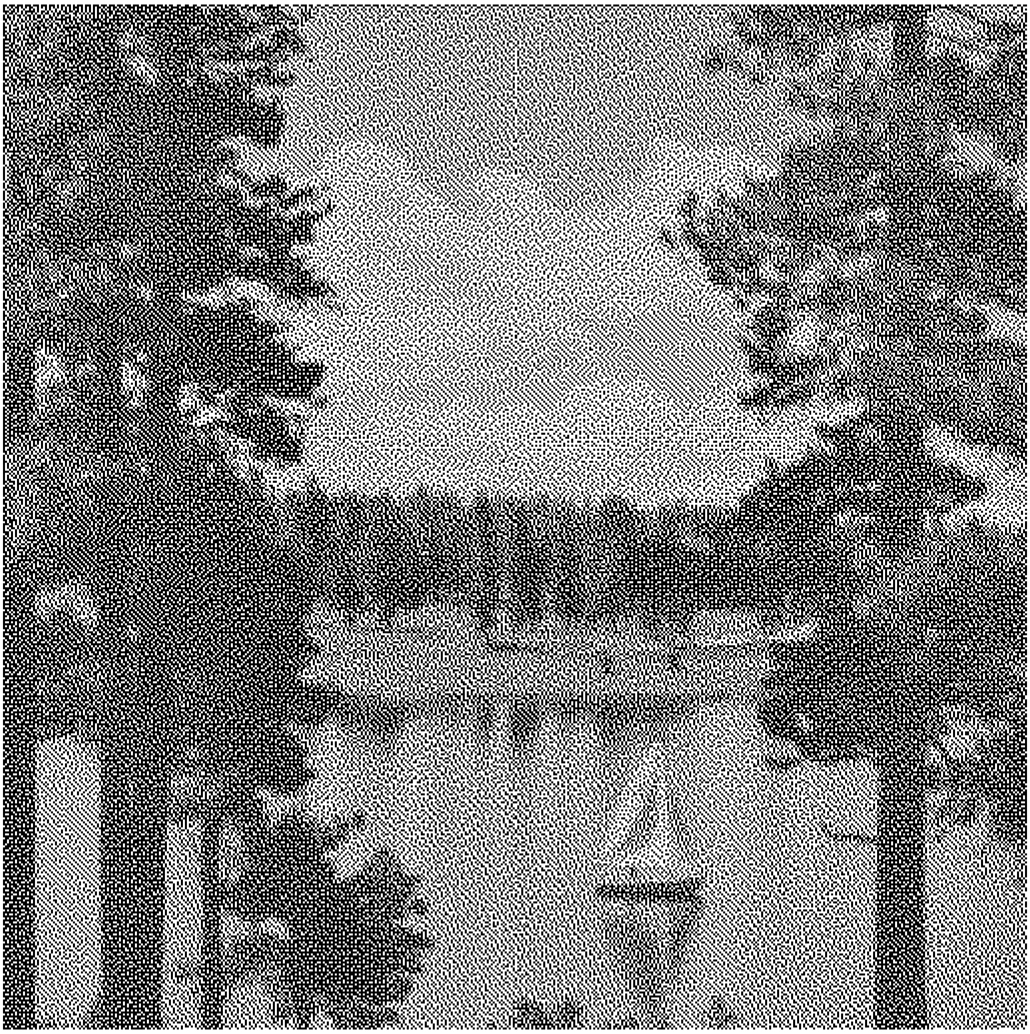
Fig. 2 Two binary images Sailboat and Airplane generated by our Error-Diffusion-based algorithm.



Linear functions used for images A and B

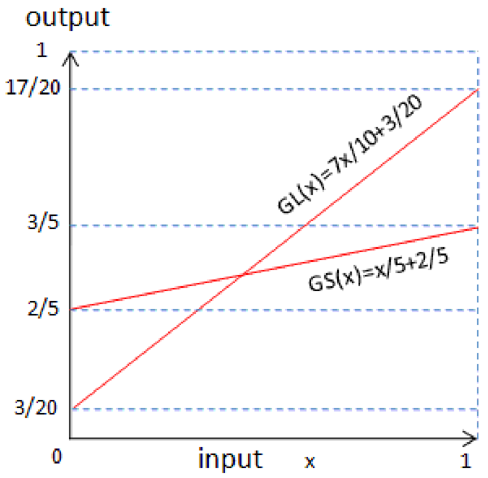


Binary image A’: airplane 256×256

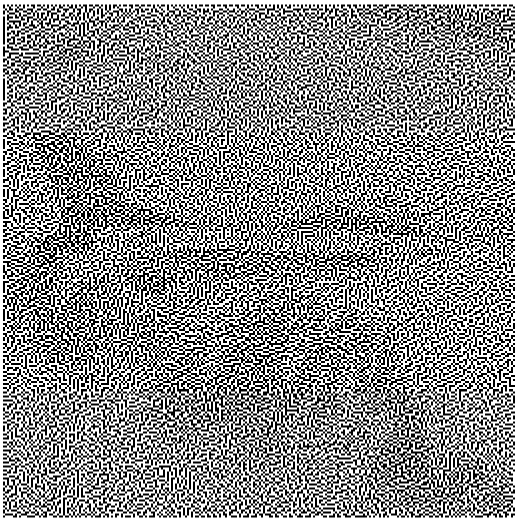


Binary image B’: sailboat 512×512

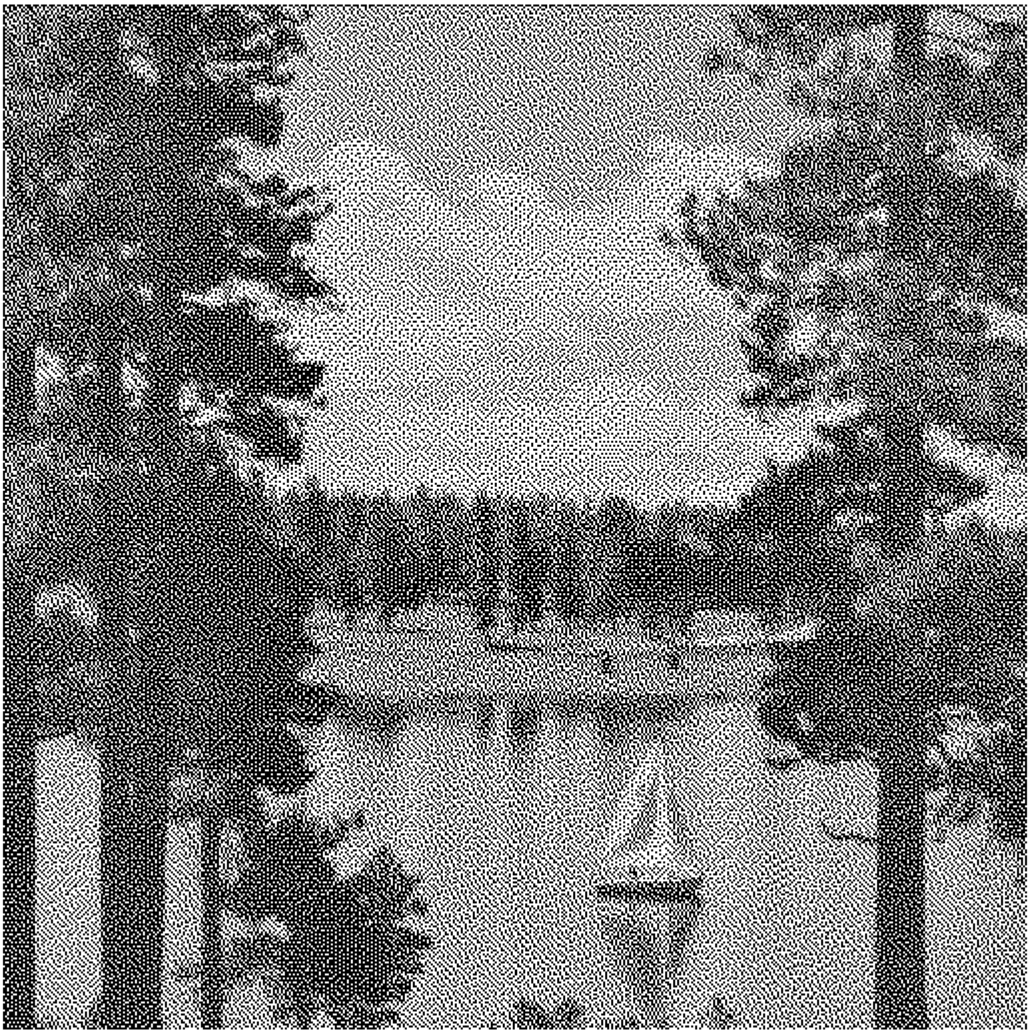
Fig. 3 Two binary images Sailboat and Airplane generated by our Error-Diffusion-based algorithm.



Linear functions used for images A and B

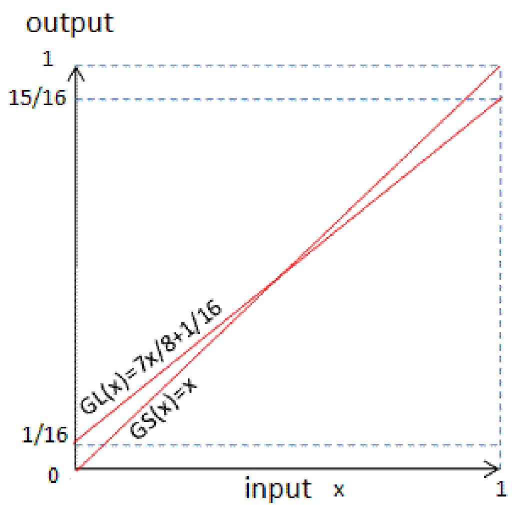


Binary image A’: airplane 256×256

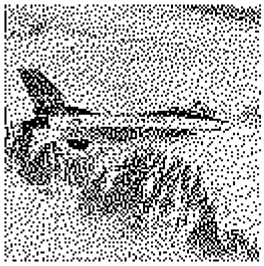


Binary image B’: sailboat 512×512

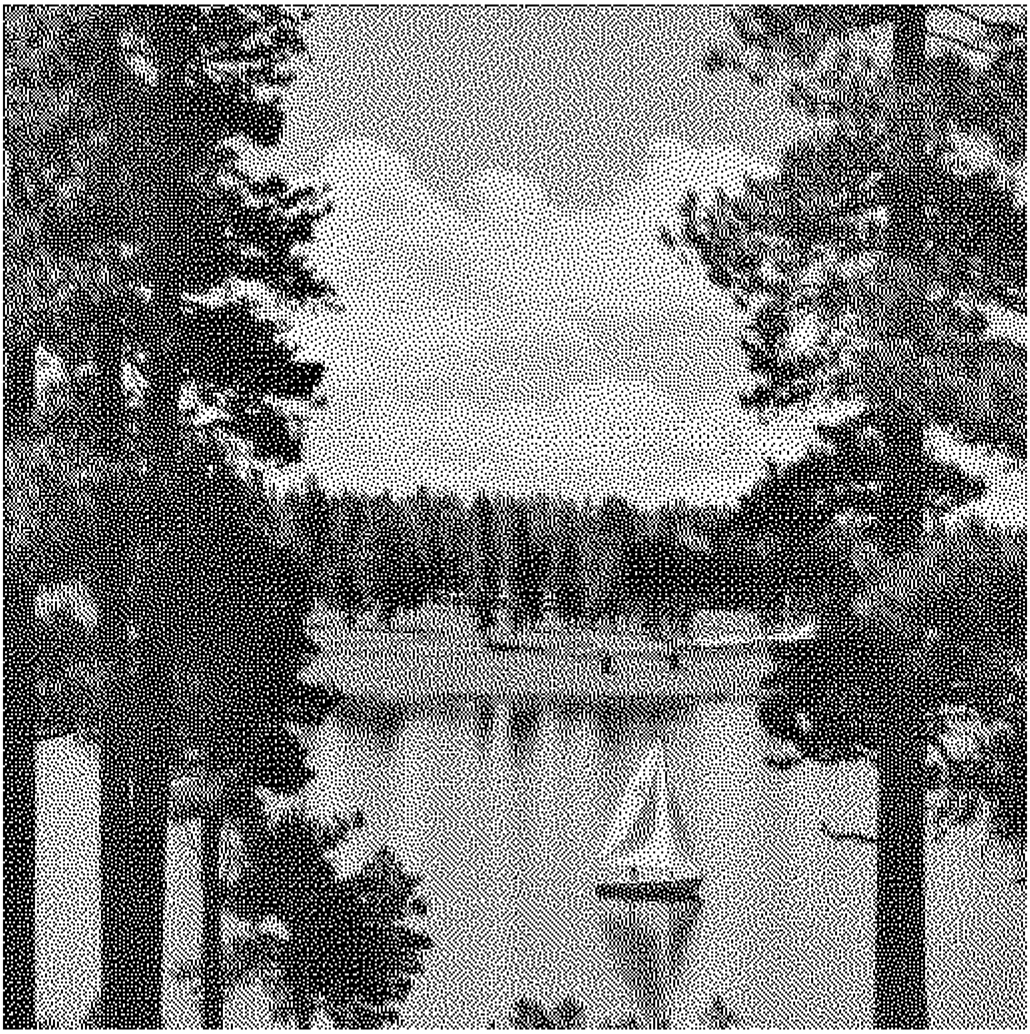
Fig. 4 Two binary images Sailboat and Airplane generated by our Error-Diffusion-based algorithm.



Linear function used for images S and G



Binary image S': airplane 128×128



Binary image G': sailboat 512×512

Fig. 5 Two binary images Sailboat and Airplane generated by our Error-Diffusion-based algorithm.