USC–SIPI REPORT #265

A Mutliscale Error-Diffusion Technique for Digital Halftoning

by

Ioannis Katsavounidis and C.-C. Jay Kuo

August 1994

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 404
Los Angeles, CA 90089-2564 U.S.A.

A Multiscale Error-Diffusion Technique for Digital Halftoning*

Ioannis Katsavounidis† and C.-C. Jay Kuo †

August 10th, 1994; EDICS: IP 3.2

Abstract

A new digital halftoning technique based on multiscale error diffusion is examined in this research. We use an image quadtree to represent the input gray level image and the output halftoned image. An iterative algorithm is developed that searches the darker region of a given image via "maximum intensity guidance" for assigning dots and diffuses the quantization error noncausally at each iteration. To measure the quality of halftoned images, we adopt a new criterion based on hierarchical intensity distribution. The proposed method provides very good results both visually and in terms of the hierarchical intensity quality measure.

1 Introduction

Halftoning is one of the oldest applications of image processing since it is essential for the printing process. With the evolution of computers and their gradual introduction to typesetting, printing and publishing, the field of halftoning that was previously limited to the so-called halftoning screen [9] evolved into its successor, digital halftoning. Digital halftoning plays a key role in almost every discipline that involves printing and displaying today. All newspapers, magazines, books are printed with digital halftoning. It is used in image display devices capable of reproducing two-level outputs such as scientific workstations, laser printers, digital typesetters. It is also important for facsimile transmission and compression.

There exist many methods to perform digital halftoning. They can be grouped in two major categories, i.e. ordered dithering [6], [9] and error diffusion [2], [3], [7], [8]. Dithering means the addition of some kind of noise prior to the quantization of a signal which in our case is an image. The amount of noise to be added is simply determined by the order of the pixel, i.e. its spatial coordinates. The ordered (or classical) dithering techniques are attractive in the sense that they are very simple to implement, especially in parallel architectures and that they are computationally inexpensive. This is because they involve a two stage process that can be performed independently for every pixel. Their performance is however poor when compared to the error-diffusion technique. Error diffusion revolutionized the digital halftoning area and has given the spark for a great number of new methods. Error diffusion is based on the simple principle that once a pixel has been quantized, thus introducing some error, this error should affect the quantization of the neighboring pixels. The way the error is affecting the quantization of its neighboring pixels is referred to as *diffusion*, meaning that the error is split in a few components and then added to the

*This work was supported by the National Science Foundation Presidential Faculty Fellow (PFF) Award 93-50309.

†The authors are with the Signal and Image Processing Institute and the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089-2564. E-mail: katsavou@sipi.usc.edu and cckuo@sipi.usc.edu.

gray level values of the neighbors. By diffusing the error, the system performs as a self-correcting, negative feedback system.

We propose a new digital halftoning technique based on multiscale error diffusion in this research. In comparison with classical error diffusion methods, our method has the following three major differences. First, all existing error-diffusion methods are applied to every pixel in a sequential pre-determined order. Our approach scans the image pixels in a way determined by their local intensity. Roughly speaking, we treat first the darker regions of a given image that require more dots. We achieve this deterministic yet image-dependent scanning via "maximum intensity guidance". Second, existing methods distribute the error by using a causal filter. Our method uses a generalized non-causal filter. Third, error-diffusion acts as a local deterring mechanism. Upon quantization of an image pixel, the diffusion of the quantization error prohibits the accumulation of the error locally. Our method achieves a more global distribution of the quantization error. In other words, it acts as a local and global deterring mechanism by prohibiting the accumulation of the quantization error over a range of resolutions. To achieve the last point, we utilize a multiresolutional treatment of the image data to be quantized.

2 Review of Existing Digital Halftoning Techniques

Digital halftoning can be phrased as a problem of 1-bit quantization of a 2-D signal as follows. Let $W(i,j)$ be an array of size $K \times L$ whose values are within $[0,1]$, corresponding to a certain gray level. We want to find an array $X(i,j)$ of the same size which takes binary values *only* (0 and 1) such that the error introduced, given by

$$E = W - X,$$

minimizes a certain criterion. What we normally require is that E is as close to a zero matrix as possible. Therefore, the problem of defining a distance between matrices rises.

This criterion leads to the so-called fixed level quantization scheme that compares the pixel value with the middle gray value (in this case, 0.5). If it is higher, we quantize it to 1, if it is less we quantize it to 0. It is straightforward to see that fixed level quantization guarantees that every element of the error matrix will be bounded (in absolute value) by 0.5. This algorithm results in the minimum error for each element so that it gives the minimum mean square error solution. Although the simplest of all, the fixed level quantization produces the worst result, as can be seen in Fig. 1(a). This very poor result is due to the fact that areas of a constant gray level are quantized as either all-white or all-black. This results in an accumulation of quantization error.

The dithering technique was invented to overcome the disadvantage of the fixed level quantization approach. Dithering means the addition of some kind of noise prior to the quantization of an image. This technique was introduced as a way of breaking the monotonicity of the error in areas of constant gray level. Depending on the type of noise added, we get different types of dithering such as the clustered and dispersed order dithering methods. For a more detailed analysis of various types of dithering, we refer to [9]. In most dithering algorithms, a regular pattern is used to represent the error that is introduced at different pixel locations. Thus, the major disadvantage of dithering is that it gives rise to regular error patterns. The dispersed ordered dithering is claimed to produce a better result. We were able to verify that, but also observed some defects, as seen in Fig. 1(b). Dithering with a regular pattern is equivalent to the addition of pseudo-random (periodic) noise, followed by fixed level quantization.

The idea behind error diffusion is very simple yet attractive. After quantizing a pixel to be either 0 or 1, it is almost certain that some error is introduced - unless that pixel has gray level of exactly 0 or 1. This error should affect the quantization of the neighboring pixels. More precisely, if the error is positive, meaning that the pixel was quantized to 0, that should "increase" the gray level value of its neighbors so that they are more probable to be quantized to 1 than they would be if the error was negative. This can be interpreted as a way of keeping the local average intensity of the printed image as close to that of the original image as possible. Now, an interesting question rises: which neighboring pixels should be affected by the error introduced from quantization at a given location? The answer given by Floyd and Steinberg [2] results in the following filter:

$$\frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & -16 & 7 \\ 3 & 5 & 1 \end{bmatrix}. \tag{1}$$

Jarvis et al [3], Stucki [8] and Stevenson and Arce [7] suggested similar filters with a larger region of support. A common characteristic of all these filters is that they are all causal, i.e. their region of support is a wedge with an angle of less than 180 degrees to ensure that these filters can be applied in a sequential manner.

Even though causal filtering provides an attractive feature, it also turns out to be the reason for one disadvantage of error diffusion, i.e. directional hysteresis. Error tends to be carried to the right boundary of the image with the filter given by (1) An easy fixture for that is to use the serpentine scanning, which is done by considering two versions of the Floyd-Steinberg filter, the original one (1) used when the direction of scanning is from the left to the right of the input image, and its mirrored reflection

$$\frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 7 & -16 & 0 \\ 1 & 5 & 3 \end{bmatrix} \tag{2}$$

used when the direction is from the right to the left. The above filters are applied alternatively. Upon reaching the right boundary of the image by using the filter (1), you go to the next line, start from the right boundary and move to the left boundary by using the mirrored version of the filter as given by (2). This algorithm, known as the error diffusion with serpentine scanning, produces a very good result as shown in Fig. 1(c). There are variations based on the same error diffusion idea such as dot diffusion [4], dithering with blue noise [10] and diffusion with neural networks [1], [5].

3 Multiscale Error-Diffusion Algorithm

Our method is based on the same principal as error diffusion. The error introduced from the quantization of a given pixel is diffused to its neighbors to guarantee that the local average intensity of the printed halftoned image will resemble the local average intensity of the original gray level image. The major difference is that the order of scanning is determined through a "maximum intensity guidance algorithm". We can briefly say that the algorithm begins with the lowest resolution image (the top of the image pyramid) and proceeds by always selecting the quadrant with the highest average intensity. This procedure ends when a pixel of the original image has been reached. Thus, the order of scanning is deterministic with respect to one specific image, but it is random in the sense that it is image-dependent. The next important point is that we use a non-causal error diffusion filter.

3.1 Image Quadtree Representation

Our approach is to apply the error-diffusion not only to the pixel of the original grid level, but also at coarser grid levels. To do so, we consider a collection of image arrays W_k, with $0 \leq k \leq r$ and where $r = \log_2[\min(K, L)]$. Thus, $r + 1$ is the total number of different levels of the image array W to be viewed, and W_r denotes the array of the largest size of dimension $K \times L$, W_{r-1} denotes that of size $K/2 \times L/2$, and so on. The collection of these image arrays of different resolutions for the same image is called an image pyramid or image quadtree. The pixels associated with the finest level of the image, i.e. $W_r(i,j)$, $i = 0, \ldots, K-1$, $j = 0, \ldots, L-1$ are the actual pixels of the W array. The elements of the coarser resolution arrays are defined by

$$W_k(i_0, j_0) = \sum_{i=0}^{1}\sum_{j=0}^{1} W_{k+1}(2i_0 + i, 2j_0 + j), \quad k = 0, \ldots, r-1, \quad i_0, j_0 = 0, \ldots, 2^k - 1.$$

These arrays correspond to different visualizations of the same array at different resolutions from different viewing distances. If $K = L = 2^r$, the coarsest resolution W_0 is simply a 1×1 array that consists of one element, whose value is the average intensity of the whole input image.

Similarly, we can represent the output dot distribution X of the same size $K \times L$ with a number of arrays of different resolutions. The main difference is that this array can only takes binary values, i.e. 0 or 1, at the finest resolution. For coarser resolutions, we have

$$X_k(i_0, j_0) = \sum_{i=0}^{1}\sum_{j=0}^{1} X_{k+1}(2i_0 + i, 2j_0 + j), \quad k = 0, \ldots, r-1, \quad i_0, j_0 = 0, \ldots, 2^k - 1.$$

where X_k denotes the output dot distribution at the kth level, $k = 0$ corresponds to the coarsest level and $k = r$ corresponds to the finest level which is identical to the array X we want to print or display. Without loss of generality, we consider the case of square images, i.e. $K = L = N$, for the rest of this paper.

It is intuitive to require that the two sets of arrays are as close as possible on all levels. That is, we require that

$$E_k = W_k - X_k$$

minimizes a certain criterion for $0 \leq k \leq r$ so that the printed image resembles the original at every resolution. One such criterion is the hierarchical intensity distribution criterion introduced on Section 4.

3.2 Algorithm

To achieve this goal, we consider an iterative multiscale error diffusion algorithm. Each iteration of the algorithm consists of the following two steps.

Step 1: "Maximum intensity guidance" in an image pyramid

Start from the coarsest level W_0 which consists of one element $W_0(0,0)$. Consider the four subimages $W_1(i_1,j_1)$ with $i_1 = 0,1$ and $j_1 = 0,1$ at level 1, each of which covers an array of size of $N/2 \times N/2$ of the original array, and choose the one with the highest value, i.e. the quadrant with the highest local intensity. It is obvious that the determination of the maximum intensity value requires 3 comparisons among the 4 intensity levels. Then, consider its four subimages at level 2, i.e. $W_2(i_2,j_2)$. Continue this procedure until we reach the finest resolution $W_r(i_r,j_r)$. Since there are exactly $r+1$ levels of the image tree, the computational complexity of the whole step is $3\log_2(N)$ (in binary comparisons). At the end of this procedure, we have chosen and kept in some kind of stack one element from every level, namely, $W_k(i_k,j_k)$ with $0 \le k \le r$.

Step 2: Multiscale error diffusion in an image quad-tree

In this step, we apply the quantization followed by multiscale error diffusion in the constructed image quad-tree. Given the pixel chosen from Step 1, we quantize this pixel by setting $X_r(i_r,j_r) = 1$, i.e. assigning a dot at the corresponding location of the output raster. By doing so, the quantization error is

$$E_r(i_r,j_r) = W_r(i_r,j_r) - X_r(i_r,j_r).$$

Then, we diffuse the error to its neighbors by using the following weighting coefficients:

$$H_{\text{center}} = \frac{1}{12}\begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{3}$$

The filter given above is only applicable at the pixels in the interior region of an image. For the side and corner (i.e. boundary) pixels we apply the following filters together with their reflections to fit all possible side and corner orientations:

$$H_{\text{corner}} = \frac{1}{5}\begin{bmatrix} 0 & 0 & 0 \\ 0 & -5 & 2 \\ 0 & 2 & 1 \end{bmatrix} \quad \text{and} \quad \mathrm{H}_{\text{side}} = \frac{1}{8}\begin{bmatrix} 0 & 0 & 0 \\ 2 & -8 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{4}$$

After the quantization and error distribution for a given pixel, we have to update the image quad-tree W_i so that the values at all resolutions are in accordance with the new error-diffused values at the finest resolution. Note that each pixel affects the value of just one element at every resolution level, since a given pixel belongs to exactly one 2×2 region of the original image and one 4×4 region, and so on. Thus, we need to update exactly $\log_2(N) + 1$ elements of the image tree for every pixel whose value is affected via error diffusion. There are at most 9 pixels whose gray level is changed for every pixel quantized, thus the complexity of this update is bounded by $9\,[\log_2(N) + 1]$ for every quantized pixel. The number of quantized pixels is roughly proportional to the average intensity of the input image W and in any case is bounded by the total number N^2 of image pixels.

In the above 2-step procedure, we focus on the subimages that have the largest intensity, and thus the greatest need for dots on the output image X. This procedure is applied iteratively until the intensity of the root of the image tree is less than 0.5, which implies the global error is bounded in absolute value by 0.5. It is easy to see that the complexity of our algorithm is bounded by $O(N^2 \log N)$. For comparison, most existing methods are $O(N^2)$. The storage required by our method is bounded by $O(N^2)$, since the number of elements of a full quad-tree with $K = L = 2^r$ terminal nodes is

$$\sum_{i=0}^{r} |W_i| = \sum_{i=0}^{r} 4^i = \frac{4^{r+1} - 1}{4 - 1} \approx \frac{4}{3} \times 4^r.$$

3.3 Discussion

It is worthwhile to make some discussion, in order to provide more insights into the above algorithm.

The image quad-tree plays a fundamental role in our algorithm. During the "maximum intensity guidance" step, we perform a top-to-bottom descent along the tree. During the error diffusion step, we perform the bottom-to-up ascent in updating the values at the nodes of the tree. The 2-step procedure is performed sequentially on the image quad-tree that is kept updated. What makes our approach distinct from existing ones is that we seek the subregions of W_i that have the largest intensity and thus the greatest need for dots on X_i via maximum intensity guidance at each iteration, while existing methods trace all pixel locations in a predetermined fashion. We do not concentrate on one part of the image, quantize it and then move to another part of the image. Instead, depending on the input image, we may have to jump around from one pixel location to another pixel location which is quite far away. The criterion is always to bring the average local intensities of the output image (as this is measured through a series of resolutions) as close to those of the original image as possible. Note that, in some sense, our approach is closer to the way painters do their art work. By using the maximum intensity guidance, we throw ink (i.e. assigning ones) to the regions that have the most need of it, just like a painter starts painting using, say, his blue-colored brash from the region that has the darkest shade of blue of all. All existing algorithms process *all* pixels of the input image in order to determine whether they will be assigned a 0 or 1. Our method processes only the pixels that will eventually be assigned 1; the rest of the pixels have (by default) a zero value, just like a painter works only on the part of the canvas that has a different color than that of the background. It is thus easily understood that our algorithm has the smallest processing time for an all-white image, since no dots need to be assigned at all.

The choice of these filters in (3) and (4) can be justified as follows. First, the distance between the center of a 3×3 filter mask to its 4 nearest neighbors can be taken to be 1 (in normalized units) while the distance from the center to the other 4 diagonal neighbors would be $\sqrt{2}$. By considering an isotropical diffusion process which is proportional to d^{-2} at distance d, we conclude that the filter coefficient of the diagonal positions should be half of that of the four nearest neighbor positions. Second, the sum of all the filter coefficients should be zero to ensure that the quantization error is fully taken into consideration. That is, it is completely diffused and compensated for the consequent application of the algorithm. Third, the filter coefficient for the center pixel is normalized to be -1.

4 Quality Measure via Hierarchical Intensity Distribution

We define the error-image tree as a collection of arrays

$$E_k = W_k - X_k, \quad k = 0, \ldots, r.$$

To measure how well a digital halftoning algorithm works, we propose the following quality criterion. Once we obtain some output image from a halftoning algorithm, we calculate the corresponding image pyramid as described in Section 3 and calculate the mean squared error at each resolution of the error array

$$MSE_k = \frac{1}{N^2} \sum_{i=1}^{2^k} \sum_{j=1}^{2^k} E_k(i,j)^2.$$

Putting together these MSE values from different resolutions, we obtain a vector of dimension $r = \log_2(N)$ that presents the difference between the input and output images at different resolutions. To compare two different halftoning methods, one needs to compare the two corresponding MSE vectors. A method is clearly better if the corresponding error vector has components that are all smaller than those of another method. However, if some components are smaller for one while others are higher, then it depends on the application to determine which method is better. In the digital halftoning application, it seems that the coarser the resolution level is, the more important the error is. A supporting evidence for this statement is that the fixed level quantization method achieving the smallest MSE at the finest resolution with a threshold of 0.5 is known to be the worst method.

The hierarchical intensity distribution quality measure can be naturally obtained by applying the Haar wavelet transform to the error image. In this case, the energy values of different low-pass filtered error images give the MSE values defined above. Note also that different resolutions use low-pass filters of different length. The highest resolution convolves with the 1×1 identity filter while the coarsest resolution is obtained with a filter of size $N \times N$. In between, consecutive resolutions convolve with filters that have widths with one octave difference. This logarithmic law, underlining the hierarchical intensity distribution quality measure, makes it a very good candidate as a substitute for the "subjective quality measure" that has been used almost exclusively in evaluating halftoning algorithms.

5 Experimental Results

The test image used to produce all the results as presented in Figs. 1 (a)-(d) is the well known "Lena" image of size 256×256. It is clear from the figure that the actual two competitors are the error diffusion method with serpentine scanning and our multiscale error-diffusion. For that reason, we present the results obtained from a boat image of size 512×512 for a more detailed comparison in Figure 2. Our method gives a better result in representing the lines such as poles and ropes between poles. We also present the result from the baboon image of size 512×512 in Figure 3. The baboon image has more textured regions than the other two images. For all three experiments, we used 8-bit gray scale images as the input. No preprocessing was performed, since we would like to compare the methods without the effect of edge crispening or contrast stretching. We deliberately printed all the results using a high-quality laser printer, but tuned to its lowest resolution (75 dpi) so that the individual dots can be clearly printed and the effect of dot overlapping is not dominant.

By comparing these figures, we can say that the proposed multiscale recursive error diffusion method produces reasonably good results. It is evident that the proposed method produces more clear and crisp halftones than traditional error diffusion, while keeping all the desirable characteristics of the later, mainly excellent gray-level rendition and no periodic patterns. For example, the hair of "Lena" or the letters on the "boat" are significantly more clear with the new method. Similarly, the textured pattern of the "baboon" image is better represented by using multiscale error diffusion. Furthermore, the overall contrast of the result produced by the new method is higher and more pleasing.

We calculate the error energy at various resolutions for all these 3 images and list them in Table 1. As shown in the table, it is obvious that the method that achieves the smallest MSE at the finest resolution is the fixed level quantization. This was explained in Section 2. However, as the resolution becomes coarser and coarser, the accumulation of error over large regions becomes more and more severe - a fact that was pointed out earlier. By comparing the other three methods, i.e. ordered dithering, traditional error diffusion and multiscale error diffusion, we can easily verify that our method outperforms the other two on *every* resolution level.

6 Conclusions

In this research, we proposed a new digital halftoning algorithm based on iterative multiscale error diffusion. The algorithm can be easily implemented in hardware, making it a very attractive candidate for an "in the device" halftoning method. The method performs significantly better than some of the best existing methods in terms of hierarchical intensity matching, and the visual quality of the resulting halftoned image is excellent. Almost all of the existing methods require some pre-processing of the input image (usually contrast stretching and/or edge crispening) in order to give their best result. Our method requires no such pre-processing since it preserves the contrast of the original image and it does not tend to oversmooth the image.

References

[1] D. Anastassiou, "Error diffusion coding for A/D conversion," *IEEE Trans. on circuits and systems*, Vol. 36, pp. 1175–1186, Sept. 1989.

[2] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey scale," in *SID Int. Sym. Digest of Tech. papers*, pp. 36–37, Winner, New York, 1975.

[3] J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A survey of techniques for the display of continuous-tone pictures on bilevel display," *Computer Graphics and Image Processing*, Vol. 5, pp. 13–40, 1976.

[4] D. E. Knuth, "Digital halftones by dot diffusion," *ACM Trans. on Graphics*, Vol. 6, pp. 245–273, Oct. 1987.

[5] S. Kollias and D. Anastassiou, "A unified neural network approach to digital image halftoning," *IEEE Trans. on signal processing*, Vol. 39, pp. 980–984, Apr. 1991.

[6] W. F. Schreiber, *Fundamentals of Electronic Imaging Systems: Some Aspects of Image Processing*, New York: Springer-Verlag, 1986.

[7] R. L. Stevenson and G. R. Arce, "Binary display of hexagonally sampled continuous-tone images," *J. Opt. Soc. Amer. A*, Vol. 2, No. 7, pp. 1009–1013, 1985.

[8] P. Stucki, "MECCA -A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction," *Research Report RZ1060, IBM Research Laboratory*, 1981.

[9] R. A. Ulichney, *Digital Halftoning*, Cambridge, Massachusetts: MIT Press, 1987.

[10] R. A. Ulichney, "Dithering with blue noise," *Proc. IEEE*, Vol. 76, pp. 56–79, Jan. 1988.

Table Captions

Table 1: The MSE at different resolutions of the image pyramid for three test images.

Figure Captions

Figure 1: Comparison of different digital halftoning techniques: (a) fixed threshold quantization; (b) dispersed order dithering using 8×8 mask; (c) Floyd-Steinberg's error diffusion with serpentine scanning; (d) multiscale error diffusion.

Figure 2: Comparison of digital halftoned boat images by using (a) Floyd-Steinberg's error diffusion with serpentine scanning and (b) multiscale error diffusion.

Figure 3: Comparison of digital halftoned baboon images by using (a) Floyd-Steinberg's error diffusion with serpentine scanning and (b) multiscale error diffusion.

Image	Resolution	Fixed Level	Ordered Dithering	Floyd-Steinberg	Multiscale
Lena	1×1	2.923×10^7	9.547×10^0	2.765×10^5	1.031×10^{-2}
	2×2	1.100×10^7	1.538×10^3	9.892×10^4	1.476×10^1
	4×4	5.401×10^6	1.171×10^3	3.046×10^4	1.355×10^2
	8×8	2.383×10^6	1.860×10^3	1.124×10^4	2.408×10^2
	16×16	8.754×10^5	1.721×10^3	4.004×10^3	4.862×10^2
	32×32	2.778×10^5	1.469×10^3	1.978×10^3	7.691×10^2
	64×64	8.286×10^4	1.629×10^3	2.309×10^3	1.321×10^3
	128×128	2.386×10^4	3.136×10^3	3.465×10^3	2.672×10^3
	256×256	7.229×10^3	1.263×10^4	1.298×10^4	1.192×10^4
Boat	1×1	5.294×10^8	2.568×10^2	7.895×10^4	7.725×10^{-3}
	2×2	1.791×10^8	4.593×10^3	1.165×10^5	1.865×10^1
	4×4	6.166×10^7	1.579×10^3	5.536×10^4	5.089×10^1
	8×8	1.695×10^7	1.051×10^3	2.021×10^4	1.035×10^2
	16×16	4.777×10^6	1.129×10^3	6.934×10^3	1.782×10^2
	32×32	1.329×10^6	1.135×10^3	2.462×10^3	3.854×10^2
	64×64	3.777×10^5	1.110×10^3	1.574×10^3	6.854×10^2
	128×128	1.074×10^5	1.496×10^3	2.192×10^3	1.254×10^3
	256×256	3.064×10^4	3.269×10^3	3.380×10^3	2.980×10^3
	512×512	9.109×10^3	1.408×10^4	1.426×10^4	1.346×10^4
Baboon	1×1	9.975×10^4	8.782×10^3	1.134×10^4	6.104×10^{-3}
	2×2	7.264×10^6	5.394×10^3	1.382×10^4	6.328×10^1
	4×4	9.334×10^6	2.207×10^3	2.337×10^4	5.046×10^1
	8×8	5.887×10^6	2.283×10^3	1.212×10^4	6.534×10^1
	16×16	2.305×10^6	2.547×10^3	5.177×10^3	1.521×10^2
	32×32	6.947×10^5	2.259×10^3	1.994×10^3	3.104×10^2
	64×64	2.017×10^5	2.127×10^3	1.439×10^3	6.430×10^2
	128×128	6.107×10^4	2.462×10^3	2.148×10^3	1.312×10^3
	256×256	2.051×10^4	3.514×10^3	3.610×10^3	2.492×10^3
	512×512	8.047×10^3	1.399×10^4	1.389×10^4	1.212×10^4

Table 1: The MSE at different resolutions of the image pyramid for three test images.

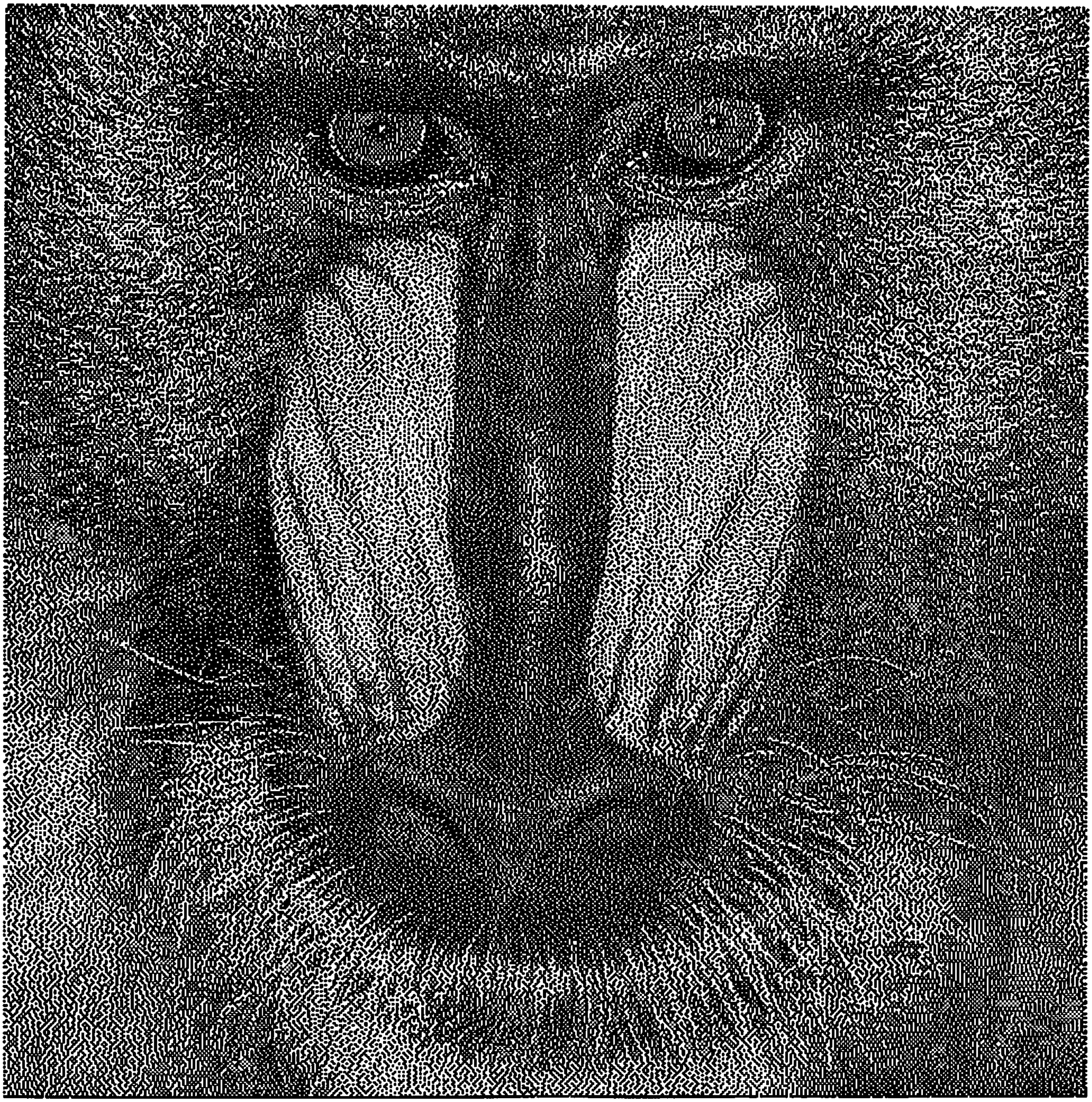
(a) (b) (c) (d)

Figure 1: Comparison of different digital halftoning techniques: (a) fixed threshold quantization; (b) dispersed order dithering using 8×8 mask; (c) Floyd-Steinberg's error diffusion with serpentine scanning; (d) multiscale error diffusion.

(a)

Figure 2: Comparison of digital halftoned boat images by using (a) Floyd-Steinberg's error diffusion with serpentine scanning and (b) multiscale error diffusion.

(b)



(a)

Figure 3: Comparison of digital halftoned baboon images by using (a) Floyd-Steinberg's error diffusion with serpentine scanning and (b) multiscale error diffusion.