240 × 240 portion of the luminance (Y) component of the SVD-filtered frame no. 75 (first field), with $\epsilon = 12$. (Magnified by a factor of two). This figure is almost indistinguishable from the original. For comparison, it also shows the normalized (from zero to 255) error between the original frame and the output of the SVD-based filter and the normalized error between the original frame and the output of the 3 × 3 median filter. As shown in Fig. 6, the median filter extracts both image information and noise, thus causing image blurring. Hence, such a filter is not suitable for our application, which requires near lossless reproduction. In contrast, the SVD filter preserves edge details and overall picture fidelity.

IV. Conclusions

We presented a novel noise estimation and filtering algorithm for still images and video sequences based on the theories of SVD and data compression. Our experiments show that the technique can effectively filter noisy images with no prior knowledge of either the image or the noise characteristics. This results in increased compressibility when the filtered data is subsequently processed by image and video compression schemes, such as JPEG and MPEG. For still images, comparisons with other filtering schemes, such as Butterworth filtering and wavelet-based filtering show that SVD-based filtering is better in preserving edge details. For video sequences, experiments have shown a 16% improvement in the compression ratio achieved by nearly lossless MPEG or, equivalently, a visual quality improvement of 1 dB at the same rate. This scheme can be used in conjunction with traditional motion-compensated temporal filtering techniques to further improve the overall performance of a high-quality video processing system [11].

Acknowledgment

The authors thank V. Bhaskaran and C. Herley for discussions, comments, and suggestions.

References

[1] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed. New York: Academic, 1982.

[2] M. K. Ozkan, A. T. Erdem, M. I. Sezan, and A. M. Tekalp, "Efficient multiframe Wiener restoration of blurred and noisy image sequences," *IEEE Trans. Image Processing*, vol. 1, pp. 453–476, Oct. 1992.

[3] H. C. Andrews and C. L. Patterson, "Singular value decompositions and digital image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 26–53, Feb. 1976.

[4] H.-C. Lee *et al.*, "Digital image noise suppression method using SVD block transform," U.S. Patent 5 010 504, Apr. 1991.

[5] B. K. Natarajan, "Filtering random noise from deterministic signals via data compression," *IEEE Trans. Signal Processing*, vol. 43, pp. 2595–2605, Nov. 1995.

[6] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inform. Theory*, vol. 41, pp. 613–627, May 1995.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, MD: John Hopkins Univ. Press, 1983.

[8] K. Konstantinides and K. Yao, "Statistical analysis of effective singular values in matrix rank determination," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 757–763, May 1988.

[9] B. Natarajan, K. Konstantinides, and C. Herley, "Occam filters for stochastic sources with application to digital images," in *Proc. IEEE ICIP 1996.*

[10] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures.* Norwell, MA: Kluwer, 1995.

[11] B. K. Natarajan and V. Bhaskaran, "Effective nearly lossless compression of digital video sequences via motion-compensated filtering," in *Proc. IEEE ICASSP-94*, Adelaide, Australia, pp. V-441–V-443.

A Multiscale Error Diffusion Technique for Digital Halftoning

Ioannis Katsavounidis and C.-C. Jay Kuo

***Abstract*—A new digital halftoning technique based on multiscale error diffusion is examined in this research. We use an image quadtree to represent the difference image between the input gray-level image and the output halftone image. An iterative algorithm is developed that searches the brightest 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 halftone 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.**

I. 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 [1] evolved into its successor—digital halftoning. Today, digital halftoning plays a key role in almost every discipline that involves printing and displaying. All newspapers, magazines, and 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, and digital typesetters. It is also important for facsimile transmission and compression.

There are many methods to perform digital halftoning. They can be grouped in three major categories: 1) dithering [1]–[4], 2) error diffusion [5]–[8], and 3) direct binarization [9], [10]. 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. However, their performance is poor when compared to the error diffusion technique. Error diffusion revolutionized the digital halftoning field and has given the spark for the development of 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 gray level values of the neighbors. By diffusing the error, the system performs as a self-correcting, negative feedback system. Direct binarization approaches attempt to minimize a weighted least squares criterion directly, i.e., they formulate halftoning as an optimization problem and apply standard techniques for its solution.

Manuscript received September 1, 1994; revised April 23, 1996. This work was supported by the National Science Foundation under Presidential Faculty Fellow Award 93-50309. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Charles A. Bouman.

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 USA (e-mail: cckuo@sipi.usc.edu).

Publisher Item Identifier S 1057-7149(97)00343-6.

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 predetermined order. Our approach scans the image pixels in a way determined by their local intensity. Roughly speaking, we first treat the brightest regions of a given image that require more white 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 noncausal 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.

II. Review of Existing Digital Halftoning Techniques

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

$$E = X - B \tag{1}$$

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.

The choice of mean squared error (MSE) criterion leads to the so-called fixed-level quantization scheme that compares the input 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 squared error solution. Although the simplest of all, fixed-level quantization produces the worst result, as can be seen in Fig. 1(a) and (i), due to the fact that areas of a constant gray level are quantized as either all white or all black. This results in a spatial accumulation of quantization error in areas of constant gray level.

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 error accumulation 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 ordered dithering methods. For a more detailed analysis of various types of dithering, we refer to [1] and [2]. In most dithering algorithms, a regular pattern is used to represent the noise 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) and (j). Dithering with a regular pattern is equivalent to the addition of pseudorandom (periodic) noise, followed by fixed-level quantization. Other techniques that generate halftones with an error-diffusion-like appearance have been introduced, such as the one that uses a blue-noise threshold array [3] and the "void-and-cluster" method [4].

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 a 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 current pixel was quantized to 0, the gray level value of its neighbors should be increased so that they are more probable to be quantized to 1 than they would have been if the error were 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 arises, namely, which neighboring pixels should be affected by the quantization error of a pixel at a given location of the input image. The answer given by Floyd and Steinberg [5] is the filter

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

Jarvis *et al.* [6], Stevenson and Arce [7], and Stucki [8] suggested similar filters with a larger region of support. A common characteristic of these filters is that they are causal, i.e., their region of support is a wedge with an angle of less than 180° to ensure that these filters can be applied in a sequential manner.

Even though causal filtering is an attractive feature, it is the reason for one disadvantage of error diffusion known as directional hysteresis. Error tends to be diffused to the right boundary of the image with the filter given by (2). An easy fix for this is to use serpentine scanning, which is done by using two versions of the Floyd and Steinberg filter. The original one of (2) is used when the direction of scanning is from the left to the right of the input image, and its mirrored reflection

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

is used when the direction is from the right to the left. The above filters are applied alternately. This algorithm, known as error diffusion with serpentine scanning, produces a very good result, as shown in Figs. 1(c), 1(k), 2(a), and 2(c). There are variations based on the same error diffusion idea, such as error diffusion with perturbed weights [1], [11], dot diffusion [12], and diffusion with neural networks [13]. For a comparison of different halftoning approaches including error diffusion, "void-and-cluster" and a direct binarization method using various printers and eye models, we refer to [10].

A very nice work that uses an adaptive version of error diffusion was proposed by Wong [14]. The same error diffusion idea is applied with the modification that the error diffusion filter is adjusted concurrently with the error diffusion process so that an error criterion is minimized. One interesting feature of his method is its capability for multiresolution rendering. The halftone image can be rendered at multiple resolutions by subsampling. The resulting images resemble the original image at certain resolutions. Our work also uses the multiresolution concept to create halftones. The major difference is that we do not address the issue of rendering the halftone at different resolutions. Roughly speaking, Wong's method obtains a pyramidal representation of the output halftone image, while our method obtains a pyramidal representation of the input image. Finally, Peli [15] presented another multiresolution halftone algorithm. In his work, one iteratively obtains the halftone by minimizing a weighted averaged error criterion, where a pyramidal representation of both input and output images is used.

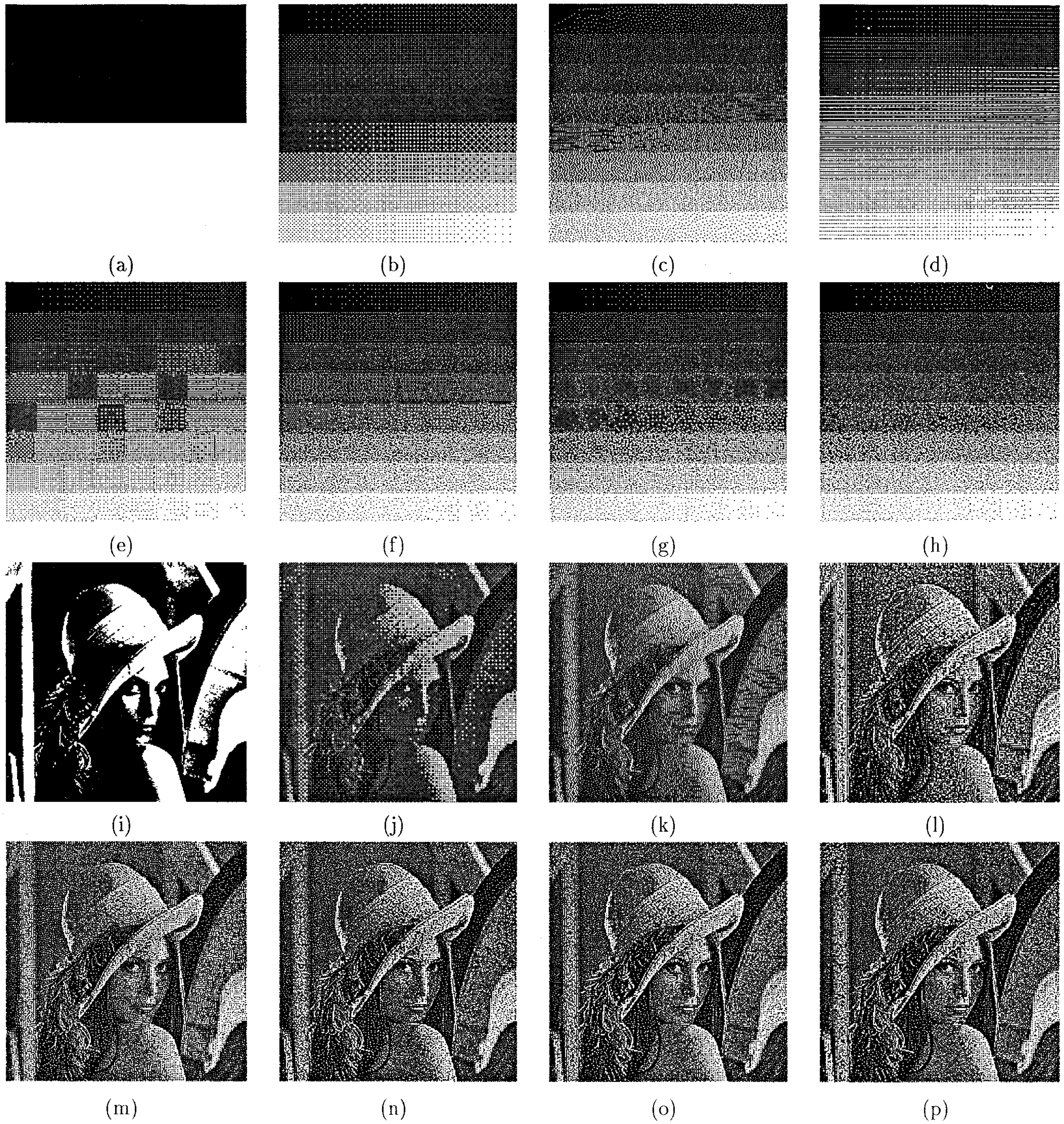


Fig. 1. Comparison of different digital halftoning techniques: (a)–(h) grayscale ramp image (256 × 256); (i)–(p) Lenna image (256 × 256). (a), (i) fixed-level quantization; (b), (j) ordered dithering; (c), (k) Floyd and Steinberg's error diffusion; (d), (l) multiscale error diffusion (MED), 1 × 1 filter; (e), (m) MED, 3 × 3 filter; (f), (n) MED, 5 × 5 filter; (g), (o) MED, 7 × 7 filter; (h), (p) MED, 9 × 9 filter.

III. Multiscale Error Diffusion Algorithm

Our method is based on the same principle 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 halftone 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 noncausal error-diffusion filter followed by an update of the image pyramid.

A. Image Quadtree Representation

Let X, B, and E be arrays of dimension $K \times L$, corresponding to the input image, output binary image, and the difference (or error) image, as defined by (1). Without loss of generality, we consider the case of square images, i.e., $K = L = N = 2^r$, for the rest of this paper. Our approach is to apply error diffusion to the pixels of the error image, followed by an update of its quadtree representation. To do so, we consider a collection of image arrays X_k with $0 \leq k \leq r$ and where $r = \log_2 N$. Thus, $r+1$ is the total number of different levels of the image array X to be viewed, and X_r denotes the array of the largest size of dimension $N \times N$, X_{r-1} denotes that of size $N/2 \times N/2$, 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., $X_r(i,\, j), i,\, j = 0, \cdots, N-1$, are the actual pixels of the X array. The elements of the coarser resolution arrays are defined by

$$X_k(i_k,\, j_k) = \sum_{i=0}^{1}\sum_{j=0}^{1} X_{k+1}(2i_k + i,\, 2j_k + j), \qquad i_k,\, j_k = 0, \cdots, 2^k - 1;\; k = r-1, \cdots, 0. \tag{4}$$

These arrays correspond to different visualizations of the same image at different resolutions, which correspond to different viewing distances. The coarsest resolution X_0 is simply a one-element array, whose value is the total intensity of the whole input image.

Similarly, we can represent the output dot distribution B of the same size $N \times N$ with a pyramid formed by a number of arrays of different resolutions B_k with $k = 0, \cdots, r$. The main difference is that the elements of the output-image quadtree have integer entries; namely, $B_k(i_k,\, j_k) \in \{0, \cdots, 4^{r-k}\}$. Thus, $B_r(\cdot, \cdot)$ can only take binary values. Finally, we define in a similar way the error-image pyramid $E_k(\cdot, \cdot)$ with $k = 0, \cdots, r$.

We require that the input and output pyramids are as close as possible on all levels. That is

$$E_k = X_k - B_k \tag{5}$$

minimizes a certain criterion for $0 \leq k \leq r$ so that the printed image resembles the original at every resolution. Since our multiscale error diffusion algorithm uses this quadtree representation and multiscale manipulation of the error values, we show very good performance in terms of the multiscale error norm. One such criterion is the hierarchical intensity distribution criterion introduced in Section IV.

B. Algorithm

To achieve this goal, we propose an iterative multiscale error diffusion algorithm. In the beginning of the algorithm, the entire output-image pyramid is blank ($B_k = 0$) and, thus, the error-image pyramid E_k is identical to the input-image pyramid X_k. Remember that the value at the root of this tree, $E_0(0,\, 0) = X_0(0,\, 0)$, is just the sum of the intensities of all the image pixels, which tells us how many dots should be assigned to the output dot raster B_r in order for this raster to resemble the original image at the coarsest resolution. Unfortunately, this information alone is not sufficient to determine the *location* of these dots. Similarly, the four values at the second-coarsest level, $E_1(i,\, j), i,\, j \in \{0,\, 1\}$, tell us how many dots should be assigned to each quadrant of the original image, but not their exact location, and so on.

For the application of our halftoning algorithm we use the error-image pyramid, E_k, with $k = 0, \cdots, r$, and the output-image array, B. The input-image pyramid, X_k, and the output-image pyramid, B_k, are not needed for the application of our algorithm; we will later use them in Section IV for the evaluation of the hierarchical intensity distribution quality measure. Using a two-step procedure, we focus on the subimages that have the largest intensity, and thus the greatest need for dots on the output image B. At each iteration of the algorithm, we introduce a white dot (value = 1) at some location of the output image B. The location is chosen in a greedy way by traversing the error tree E_k, top-to-bottom. We then diffuse the error to the neighbors of that pixel in the E_r array, ensuring that there is no error leakage and also that the total decrease in the E_r array is equal to 1. Finally, we update the error-image pyramid with the new error values, thus decreasing the values of each level in total by 1. This procedure is applied iteratively until the intensity of the root of the error tree is less than 0.5, which implies that the global error is bounded in absolute value by 0.5.

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 E_0, which consists of one element $E_0(0,\, 0)$. Consider the four subimages $E_1(i_1,\, j_1)$, where $i_1 = 0,\, 1$ and $j_1 = 0,\, 1$, at level 1, each of which covers an array of size $N/2 \times N/2$ of the original array, and choose the subimage with the highest value, i.e., the quadrant with the highest local intensity. Next, consider its four subimages at level 2, i.e., $E_2(i_2,\, j_2)$, and find the subarray with the highest value. Continue this procedure until the finest resolution $E_r(i_r,\, j_r)$ is reached. At the end of this procedure, we have chosen and kept the location $(\hat{i}_r,\, \hat{j}_r)$ of one pixel of the original image and the value $E_r(\hat{i}_r,\, \hat{j}_r)$ at the corresponding location of the error-image pyramid.

Step 2) Multiscale Error Diffusion in an Image Quadtree: In this step, we apply quantization followed by multiscale error diffusion in the constructed error-image quadtree. Given the pixel chosen in Step 1, we quantize this pixel by setting $B(\hat{i}_r,\, \hat{j}_r) = 1$, i.e., we assign a white dot at the corresponding location of the output raster. The quantization error is given by

$$e_q = E_r(\hat{i}_r,\, \hat{j}_r) - 1. \tag{6}$$

We set $E_r(\hat{i}_r,\, \hat{j}_r) = e_q$ and diffuse the error to $(\hat{i}_r,\, \hat{j}_r)$'s neighbors in the error E_r by using a noncausal diffusion filter. The algorithm works with any choice of filter, producing different results. For the rest of this section, we consider the following 3×3 diffusion filter, which has provided good halftones for our algorithm.

$$H_{\text{center}} = \frac{1}{12}\begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{7}$$

This filter is only applicable to pixels in the interior region of the error image. For the side and corner (i.e., boundary) pixels we apply the following filters and their reflections to fit all possible side and corner orientations.

$$H_{\text{corner}} = \frac{1}{5}\begin{bmatrix} 0 & 0 & 0 \\ 0 & -5 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$
$$H_{\text{side}} = \frac{1}{8}\begin{bmatrix} 0 & 0 & 0 \\ 2 & -8 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{8}$$

It is easy to see that, as a result of the error diffusion, the error value of the pixel just visited takes the new value 0, i.e., $E_r(\hat{i}_r,\, \hat{j}_r) = 0$, while the error values of the neighboring pixels are decreased according to their position relative to the center pixel and also in proportion to the quantization error. Moreover, since all the above filters have zero mean, it follows that the total decrease in the error image E_r is equal to 1, i.e., there is no error leakage.

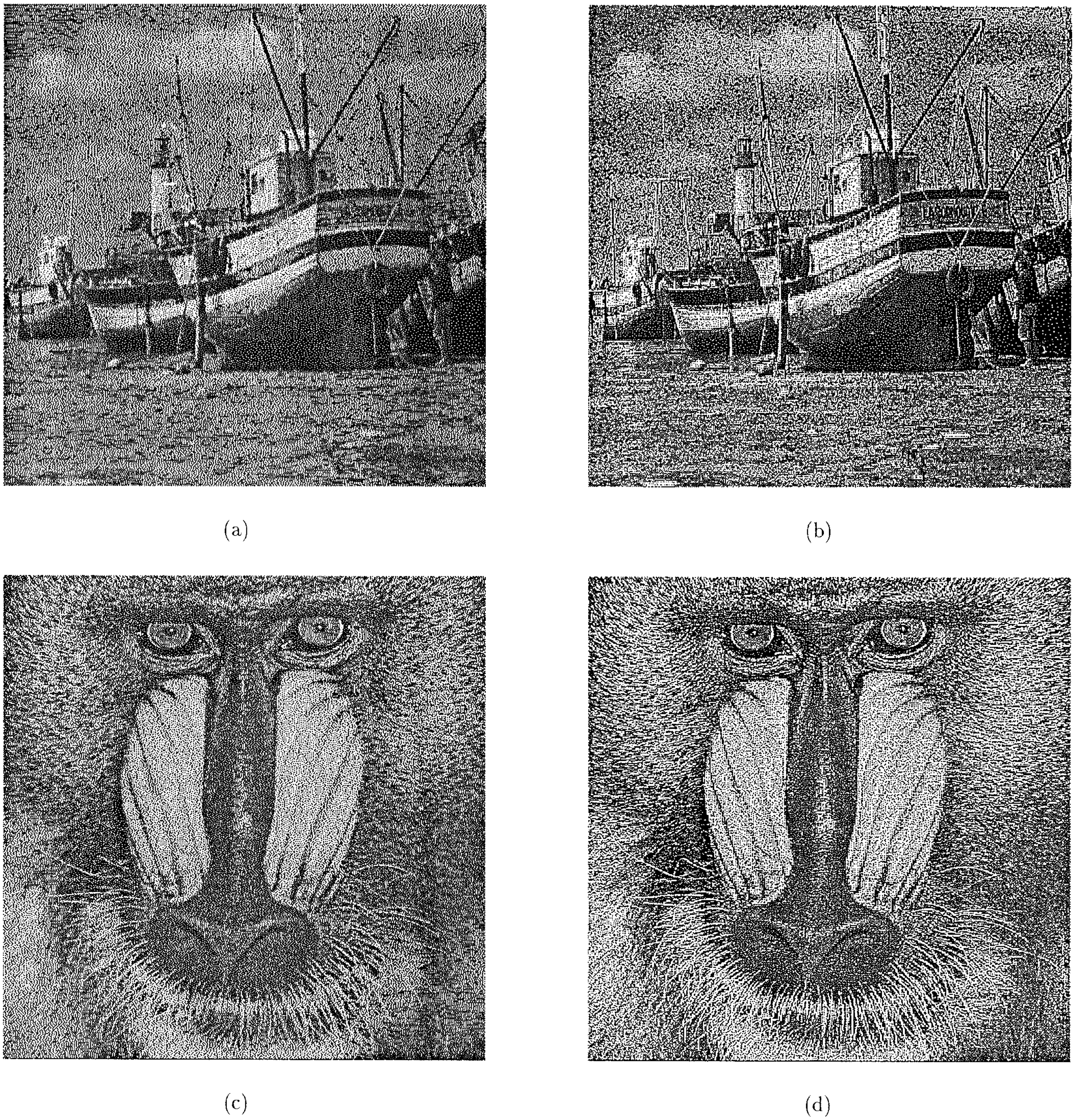
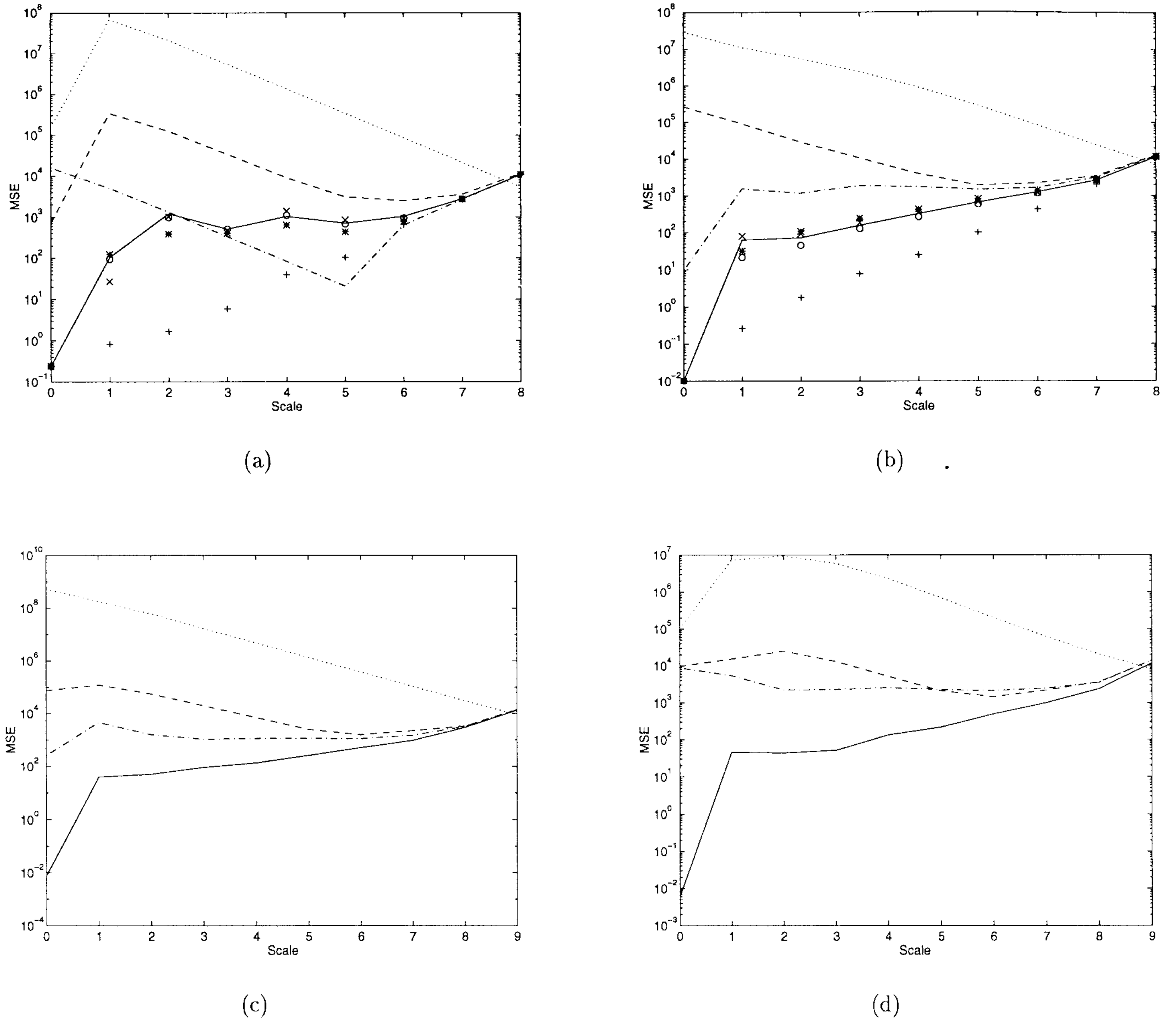


Fig. 2. Comparison of different digital halfoning techniques. (a) Boat image (512 × 512), Floyd and Steinberg's error diffusion. (b) Boat image (512 × 512), multiscale error diffusion, 9 × 9 filter. (c) Baboon image (512 × 512), Floyd and Steinberg's error diffusion. (d) Baboon image (512 × 512), multiscale error diffusion, 9 × 9 filter.

After the quantization and error distribution for a given pixel has been done, we have to update the error-image quadtree E_k, so that the values at all resolutions are in accordance with the new error-diffused values at the finest resolution.

It is obvious that the determination of the maximum intensity value at a certain level of the error-image tree requires three comparisons among the four error values. Since there are exactly $r+$ 1 levels of the error-image tree, the computational complexity of the whole Step 1 is 3 $\log_2 N$ (in binary comparisons). To determine the complexity of Step 2, 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 error-image tree for every pixel whose value is affected via error diffusion. There are at most 9 pixels (for the case of 3 × 3 diffusion filter) whose error value is changed for every pixel quantized, thus the complexity



(a)

(b)

(c)

(d)

Fig. 3. Comparison of different digital halftoning techniques based on the multiscale MSE vector. (a) Gray-scale image (256 × 256). (b) Lenna image (256 × 256). (c) Boat image 512 × 512. In (a)–(d), the dotted, dash-dotted, dashed, and solid lines are used to represent the fixed-level quantization, ordered dithering, Floyd and Steinberg's error diffusion,and multiscale error diffusion with the 9 × 9 filter, respectively. The multiscale error diffusion method with other filter sizes are also plotted in (a) and (b), where the "+," "*," "o," and "x" lines denote filters of sizes 1 × 1, 3 × 3, 5 × 5, and 7 × 7, respectively.

of this update is bounded by 9 ($\log_2 N+$ 1) for every quantized pixel. The total number of quantized pixels is roughly proportional to the average intensity of the input image X and in any case is upper bounded by the total number N^2 of image pixels. 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 quadtree with $N^2 = 4^r$ terminal nodes is

$$\sum_{i=0}^{r}|E_i| = \sum_{i=0}^{r}4^i = \frac{4^{r+1}-1}{4-1} \approx \tfrac{4}{3}\times 4^r = \tfrac{4}{3}\times N^2. \qquad (9)$$

C. Discussion

The image quadtree 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 two-step procedure is performed sequentially on the image quadtree that is kept updated. What makes our approach distinct from existing ones is that we seek via maximum intensity guidance at each iteration the subregions of E_k that have the largest value, and thus the greatest need for dots on B, 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 in a deterministic manner. Instead, depending on the input image, we jump from one pixel location to another, which may be spatially quite far away. The criterion is always to bring the average local intensities of the output image (as measured by a multiresolution representation) as close to those of the original image as possible. Note that, in some sense, our approach is closer to the way painters create artwork. By using the maximum intensity guidance, we throw ink (i.e., assign 1's) to the regions that have the most need of it, just like a painter starts using, say, his blue-colored brush by painting 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 be assigned 1; the rest of the pixels have (by default) a 0 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 black image, since no dots need to be assigned.

The choice of filters in (7) and (8) can be justified as follows. First, the distance between the center of a 3 × 3 filter mask to its four nearest neighbors can be taken to be 1 (in normalized units) while the distance from the center to the other four diagonal neighbors would be $\sqrt{2}$. By considering an isotropic 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 subsequent application of the algorithm. Third, the filter coefficient for the center pixel is normalized to be -1. Using these three principles, we can obtain filters with larger region of support (5 × 5, 7 × 7, 9 × 9, etc.) that produce similar but still different results. A very interesting variation of the algorithm presented above can be obtained by using a 1 × 1 diffusion filter, i.e., without performing any error diffusion on the neighbors of the selected pixel. Thus, after we choose a pixel using Step 1 of the presented algorithm, we set a dot on the output raster and subtract 1 from the value of the error-image pyramid, followed by an update of the error-image pyramid. The effects of different filter choices are presented and discussed in Section V.

IV. Quality Measure via Hierarchical Intensity Distribution

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 pyramids as described in Section III-A and calculate the mean squared error (MSE) at each resolution of the error array

$$\begin{aligned} MSE_k &= \frac{1}{N^2}\sum_{i=1}^{2^k}\sum_{j=1}^{2^k} E_k(i,\, j)^2 \\ &= \frac{1}{N^2}\sum_{i=1}^{2^k}\sum_{j=1}^{2^k}[X_k(i,j) - B_k(i,j)]^2, k = 0, \cdots, r. \end{aligned} \quad (10)$$

We form a vector of dimension $r+1 = \log_2 N + 1$ of these MSE values from different resolutions, that presents the difference between the input and output images at different resolutions. To compare two different halftoning methods, we compare the two corresponding MSE vectors. We say that 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 method while others are higher, then the application determines which method is better. In the digital halftoning area, because of the integration performed by the human visual system, it seems that the coarser the resolution level is, the more important the MSE. We caution the reader that this is a general qualitative statement. A generalized MSE value obtained by assigning weights to the components of the MSE vector may be a more appropriate choice for an error criterion, and further experiments are needed to determine proper weighting parameters in different spectral bands. Supporting evidence for this statement is that the fixed-level quantization method, which achieves the smallest MSE at the finest resolution with a threshold of 0.5, is known to be the worst halftoning 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 the different lowpass filtered error images give the MSE values defined above. Note also that different resolutions use lowpass filters of different length. The highest resolution image is obtained by being convolved with the 1 × 1 identity filter, while the coarsest resolution image is obtained with a filter of size $N \times N$. The other resolutions are obtained by convolution with filters that have widths 1/2 of the others. This logarithmic law, underlying 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.

V. Experimental Results

The test image used to produce Fig. 1(a)–(h) is a 256 × 256 gray-scale ramp of 64 levels. It is interesting to see the differences among Fig. 1(d)–(h) obtained using multiscale error diffusion with filters of various sizes. Fig. 1(i)–(p) were obtained by using the Lenna image of size 256 × 256. It is obvious from these results that the two competitors are the error diffusion method with serpentine scanning and our multiscale error diffusion method. It is again interesting to see the differences among Fig. 1(l)–(p) due to the use of filters of various sizes. As the filter size increases, the overall sharpness of the output image decreases, from an oversharpened unnatural-looking image produced with the 1 × 1 filter to a very nice and pleasing image produced with the 9 × 9 filter. Combining this observation with the very good gray-level rendition in Fig. 1(h), we decided to use the 9 × 9 filter in our multiscale error diffusion algorithm.

We present the results obtained from the Boat and the Baboon 8-b grayscale images, both of size 512 × 512, for a more detailed comparison in Fig. 2. 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 medium resolution (150 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 method produces more clear and crisp halftones than traditional error diffusion, while keeping all the desirable characteristics of the latter, mainly excellent gray-level rendition and no periodic patterns. Examples include the hair of Lenna or the letters on the Boat are significantly more clear; lines in the Boat image such as poles and ropes between poles are better represented and the textured pattern of the Baboon image is better treated. 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 these four images and plot them in Fig. 3(a)–(d). The method that achieves the smallest MSE at the finest resolution is fixed-level quantization, as explained in Section II. However, as the resolution becomes coarser, the accumulation of error over large regions becomes more severe. By comparing the other three methods, i.e., ordered dithering, traditional and multiscale error diffusion, one can verify that our method outperforms the other two on *every* resolution level. An interesting observation is that, for the grayscale ramp and Lenna images, the filter that achieves the smallest MSE values is the 1 × 1 filter, which, however, produces the worst visual result among them. This indicates that the choice of a simple averaging for the creation of the error-image pyramid may not be the best and other weighted averaging filters may be more appropriate.

VI. Conclusions

In this research, we proposed a new digital halftoning algorithm based on multiscale error diffusion. The method performs significantly better than some of the best existing methods in terms of hierarchical intensity matching. Almost all of the existing methods require some preprocessing of the input image (usually contrast stretching and/or edge crispening) in order to give their best result. Our method requires no such preprocessing since it preserves the contrast of the original image, and it does not tend to oversmooth the image. By changing the size of the diffusion filter, the amount of sharpness of the output image can also be controlled.

Acknowledgment

The authors thank the anonymous reviewers who have contributed to a much improved manuscript.

References

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

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

[3] T. Mitsa and K. Parker, "Digital halftoning using a blue-noise mask," in *Proc. SPIE, Image Processing Algorithms and Techniques II,* 1991, vol. 1452, pp. 47–56.

[4] R. A. Ulichney, "The void-and-cluster method for dither array generation," *Proc. SPIE, Human Vision, Visual Processing and Digital Display IV,* vol. 1913, pp. 332–343, 1993.

[5] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey scale," in *Proc. SID Int. Symp. Digest of Technical Papers.* New York, 1975, pp. 36–37.

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

[7] R. L. Stevenson and G. R. Arce, "Binary display of hexagonally sampled continuous-tone images," *J. Opt. Soc. Amer. A.*, vol. 2, pp. 1009–1013, 1985.

[8] P. Stucki, "Mecca—A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction," IBM Res. Lab., Res. Rep. RZ1060, 1981.

[9] T. Pappas and D. L. Neuhoff, "Least-squares model-based halftoning," in *Proc. SPIE, Human Vision, Visual Processing and Digital Display III,* 1992, vol. 1666.

[10] M. Schulze and T. Pappas, "Blue noise and model-based halftoning," in *Proc. SPIE, Human Vision, Visual Processing and Digital Display V,* 1994, vol. 2179, pp. 181–194.

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

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

[13] S. Kollias and D. Anastassiou, "A unified neural network approach to digital image halftoning," *IEEE Trans. Signal Processing,* vol. 39, pp. 980–984, Apr. 1991.

[14] P. W. Wong, "Adaptive error diffusion for multiresolution rendering," in *Society for Information Display, Proc. SID '94,* San Jose, CA, June 1994, pp. 801–804.

[15] E. Peli, "Multiresolution, error-convergence halftone algorithm," *J. Opt. Soc. Amer. A.,* vol. 8, pp. 625–636, Apr. 1991.