Open access • Proceedings Article • DOI:10.1117/12.584430

A Multiresolution Halftoning Algorithm for Progressive Display — Source link

Mithun Mukherjee, Gaurav Sharma

Institutions: Rochester Institute of Technology, University of Rochester

Published on: 17 Jan 2005 - Color Imaging Conference

Topics: Computer data storage, Continuous tone, Binary image and Image processing

A Multiresolution Halftoning Algorithm for Progressive Display

Mithun Mukherjee[a] and Gaurav Sharma[b]

[a]Rochester Institute of Technology, Rochester, USA;
[b]University of Rochester, Rochester, USA

ABSTRACT

We describe and implement an algorithmic framework for memory efficient, 'on-the-fly' halftoning in a progressive transmission environment. Instead of a conventional approach which repeatedly recalls the continuous tone image from memory and subsequently halftones it for display, the proposed method achieves significant memory efficiency by storing only the halftoned image and updating it in response to additional information received through progressive transmission. Thus the method requires only a single frame-buffer of bits for storage of the displayed binary image and no additional storage is required for the contone data. The additional image data received through progressive transmission is accommodated through in-place updates of the buffer. The method is thus particularly advantageous for high resolution bi-level displays where it can result in significant savings in memory.

The proposed framework is implemented using a suitable multi-resolution, multi-level modification of error diffusion that is motivated by the presence of a single binary frame-buffer. Aggregates of individual display bits constitute the multiple output levels at a given resolution. This creates a natural progression of increasing resolution with decreasing bit-depth.

Keywords: Halftoning, multiresolution progression, error diffusion, screening

1. INTRODUCTION

Since its invention in 1852 by Talbot,[1] halftoning has commonly been used in printing systems and is still the dominant technology for hardcopy reproduction of documents. Halftoning has also been used extensively in bi-level and multi-level computer displays, where its use has decreased with the emergence of true-color high-bit depth displays. Algorithms for both application areas have undergone significant research and development contributing to an extensive literature covered in patents[2] as well as technical papers.[3,4] Halftoning methods in both areas also face new challenges and opportunities as new technologies are adopted. In this paper, we consider particularly, methods for halftoning for electronic display motivated by progressive resolution image transmission.

Consider the case of browsing through a high resolution image database. Progressive resolution modes of image transmission, which begin by communicating a coarse resolution approximation of the image and successively refine it as additional image information is received, are typically favored in such an environment. The progressive resolution transmission allows the user to first sample a low resolution preview of the original high resolution image **during** transmission, which is then improved-upon as additional image data is transmitted and received. At any stage of the progressive transmission, the user may choose to abort transmission if the image is not of interest; thus saving time as well as bandwidth. Because of these benefits, progressive modes of communication have also been incorporated in image compression standards.[5–7] If the display device at the user end is binary, the progressive approximations to the image must be halftoned prior to display. In this paper, we address this scenario and develop a new memory efficient method for the halftoning of images communicated in a progressive-by-resolution system. We refer to the method as Progressive Halftoning In-Place (P-HIP). The method combines multilevel error diffusion[8] and screening to progressively halftone and render a grayscale image with gradually increasing resolution. It utilizes only the binary display memory for image storage and thereby eliminates the need for additional memory to store contone image data, hence it offers improved memory efficiency.

We are not aware of other work in halftoning addressing the specific scenario of halftoning for display in a progressive by resolution transmission environment. There are, however, two related directions of research that have been reported in the halftoning literature. Goldschneider et al[9] present an embedded halftoning method that embeds a lower resolution halftone in a higher resolution version. Such a method would be useful for instance in scenarios where halftoned versions of an image are stored at a server and communicated to devices with differing resolutions. Since the client-server interaction required for realizing the benefits of progressive image transmission have been standardized primarily for contone images,[10] the method does not directly address the scenario of interest in this paper. In addition, techniques have been proposed that exploit multi-resolution halftone rendering but are not oriented towards a progressive image transmission environment.[11,12] The multiscale error diffusion algorithm proposed by Katsavounidis and Kuo[11] is an iterative technique that utilizes an image quad-tree to represent the difference image between the input contone image and the output halftone image. Using a 'maximum intensity guidance' rule across the quad-tree, the algorithm searches the brightest regions of the image for assigning dots. To construct the image quadtree, the algorithm requires the input contone image in its entirety, thereby rendering it unsuitable within a progressive transmission scheme. Additionally, although it produces very good results, its iterative nature as well as the storage requirements for the image quad-tree may pose hurdles in its implementation within such a scheme. Another halftoning technique based on multiscale dot distribution proposed by Wong[12] utilizes a similar quad-tree structure to select local neighborhoods over which average gray levels between the input contone image and output halftone image are matched using a global mean requirement. Again, this algorithm begins with the image at its final resolution and is not intended for a progressive by resolution transmission environment.

The rest of this paper is organized as follows. Section 2 begins with a description of a straightforward approach to halftoning a multiresolution progression and then presents the proposed memory efficient alternative. Details of the proposed algorithm are included in Section 3 and Section 4 discusses its memory efficiency and computational complexity. The experimental setup and results are presented in Section 5. Section 6 provides a brief conclusion to this article.

2. HALFTONING A MULTI-RESOLUTION PROGRESSION

Consider a multiresolution image being communicated in a progressive-by-resolution system to a bi-level display. In order to avail the benefits of progressive transmission, the bi-level halftone display must constantly be updated to display the available image information. Since power and memory are both precious commodities in any portable display, for it to be capable of rendering progressive imagery, a halftoning algorithm that utilizes minimum power and memory is desirable.

2.1. Independent Reconstruction and Halftoning

A straightforward approach to halftoning a multiresolution progression is to handle the steps of multiresolution progression and halftone display independently of each other. At each stage in the transmission, the available image data is utilized to recreate the best possible contone representation which is halftoned and displayed. The actual contone image data is preserved for further refinement to finer resolution as additional detail data is received. This is shown in Figure 1.

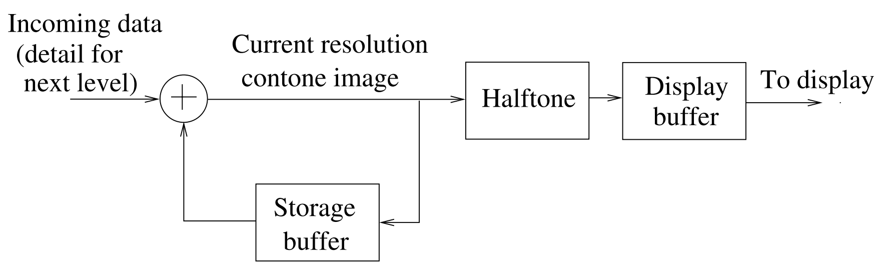

Figure 1. Halftoning a multiresolution progression using a separate storage buffer

The reconstructed image at its current resolution is stored in a buffer in memory. It is then halftoned and displayed (with potential scaling to suitable size). When the next level of detail is received, it is added to the current image thereby increasing its resolution. The higher resolution contone is stored in the buffer and the process is repeated recursively until the final resolution of the image is attained and displayed.

2.2. Halftoning in-place

In a situation where memory is at a premium, the additional memory required to store the entire contone image can be a significant system overhead. This is particularly true for high resolution/large scale devices where the large image dimensions can require a significant amount of storage memory. Since the frame buffer (or other memory) used for storing the displayed image already contains an approximate representation of the contone image, it can be used instead of a separate memory buffer, to recover the original contone image from the displayed halftone for the next stage of progression. This scheme is illustrated in Figure 2 below. It is also possible to save memory by storing only a compressed representation of the image but this entails an additional computational burden as the image reconstruction at each progression of resolution must be recomputed from scratch.

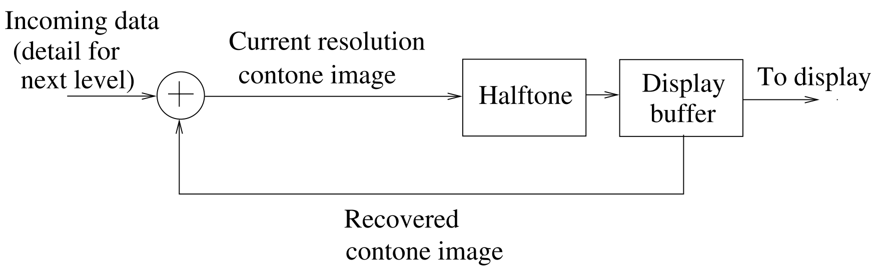

Figure 2. Progressive multiresolution halftoning scheme without the use of a storage buffer

A halftone display trades off spatial resolution against the number of gray levels. If one interprets the gray level as a spatial average then there is a natural progression of increasing tonal resolution with decreasing spatial resolution in a halftone display. Consider a binary display of size $2^k \times 2^k$ pixels. If one considers 2×2 blocks of pixels within this image, each block as a whole may represent 5 gray levels i.e. the number of 'on' pixels may vary from 0 to 4. Thus, by reducing the spatial resolution by half it is possible to increase the tonal resolution from binary (2 levels) to 5 gray levels. The image dimensions have fallen from the original size of $2^k \times 2^k$ to an equivalent size of $2^{k-1} \times 2^{k-1}$. As a generalization, decreasing the spatial resolution by a factor r by picking blocks of size $2^r \times 2^r$ shall lead to an increase in tonal resolution from 2 binary levels to $2^{2r} + 1$ gray levels. The spatial resolution thus reduces from an original image size of $2^k \times 2^k$ to a final image size of $2^{k-r} \times 2^{k-r}$. The inherent averaging or low pass involved in this process may be considered an approximation to that within the human visual system. The image decomposition in spatial terms assumes the structure of a quadtree. Figure 3 below shows a graphical example of this inversely proportional relationship between tonal resolution and spatial resolution. P-HIP exploits this inherent relationship.

3. PROGRESSIVE HALFTONING IN-PLACE (P-HIP) ALGORITHM

We utilize a simple multiresolution decomposition based on the Haar wavelet transform to illustrate a progressive by resolution transmission environment. The highest level of wavelet decomposition corresponds to the lowest resolution or the first stage of progression and vice versa. This convention shall be used in the following description of the P-HIP algorithm.

P-HIP operates in two steps - a progression based multilevel error diffusion step followed by a fill order based screening operation on this quantized output for updating the display. We consider each of these in turn.

The progression based multilevel error diffusion step is shown in Figure 4. The multilevel image at the current stage of progression, or decomposition level $(i + 1)$, is combined with the incoming image detail for the level i.

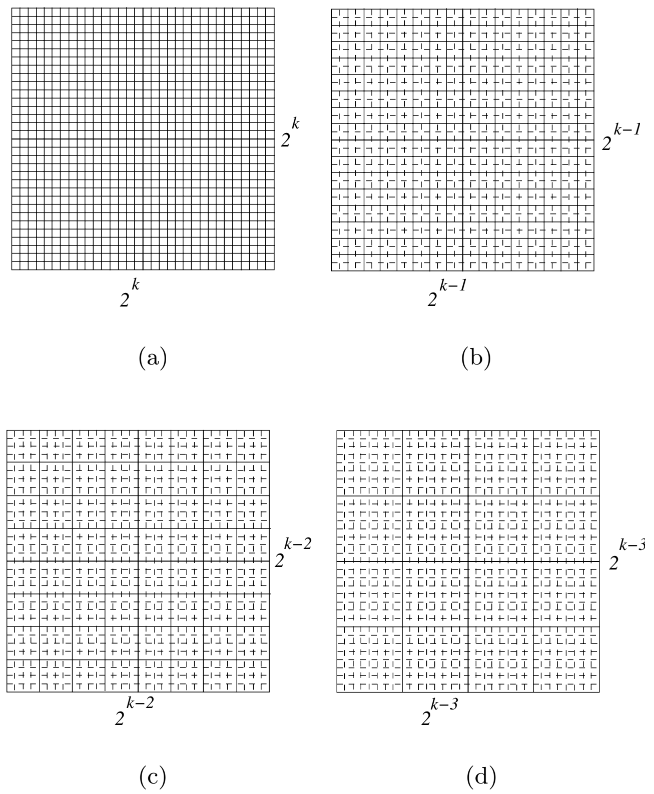

Figure 3. Relation between tonal and spatial resolution between levels of progression. (a)Tonal resolution: Binary (2 levels), Spatial resolution: $2^k \times 2^k$. (b)Tonal resolution: 5 gray levels, Spatial resolution: $2^{k-1} \times 2^{k-1}$. (c)Tonal resolution: 17 gray levels, Spatial resolution: $2^{k-2} \times 2^{k-2}$. (d)Tonal resolution: 65 gray levels, Spatial resolution: $2^{k-3} \times 2^{k-3}$.

Multilevel error diffusion is performed on the image produced to obtain the higher resolution but lower bit-depth image at the the next, i.e. i^{th}, level of progression.

The coarseness of quantization within the multilevel error diffusion block is gradually increased with increasing resolution. i.e the degree of quantization is finer at lower resolutions and coarser at higher resolutions. In other words, the number of gray levels or tones are higher at lower resolutions and vice versa. With each progression of incoming detail, the quantization step size is quadrupled to mimic a multiresolution quadtree. At the final stage, when the last remaining image detail has been received, the quantization is at its coarsest, thus representing the equivalent of a simple bi-level error diffusion halftoning operation. This process is well matched to the inherent trade-off between the tonal and spatial resolutions as illustrated in Figure 3.

Following the multilevel error diffusion, a bi-level representation of the lower bit-depth multilevel image is displayed using a halftone screen as shown in Figure 5. Aggregates of individual display bits constitute the multiple output levels at a given resolution. The multilevel contone image can be recovered from this bi-level representation by simply collecting these aggregates. This recovered contone image is then combined with the next level of image detail. This process continues till level 0, at which point all available image detail has been received and the image has been rendered at its final resolution.

In this sense, the halftoning for the purpose of display may be considered simply as the storage of the multilevel error diffusion output in the bits afforded by the corresponding area in the binary display.

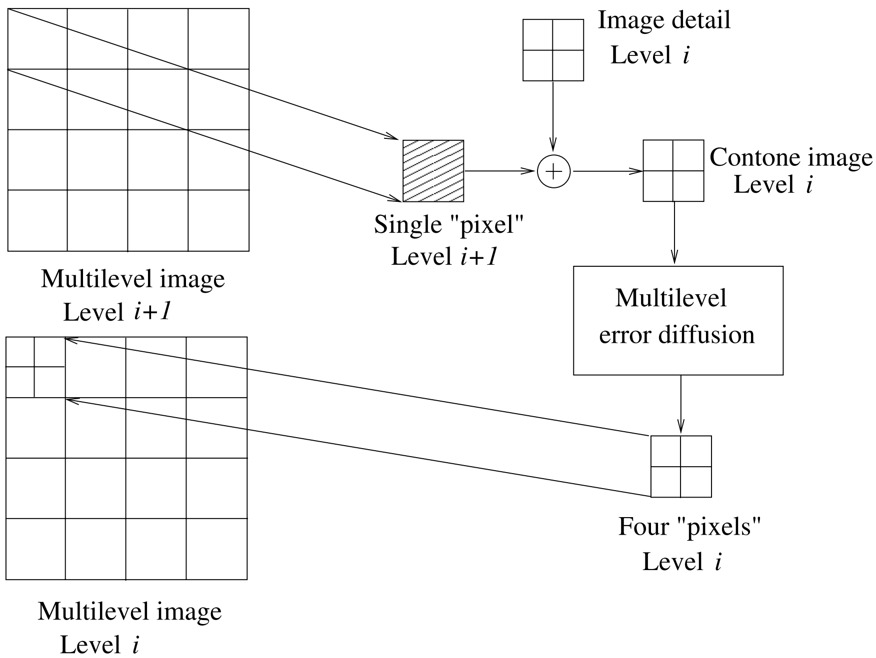

Figure 4. Progressive image detail addition and multilevel halftoning within P-HIP

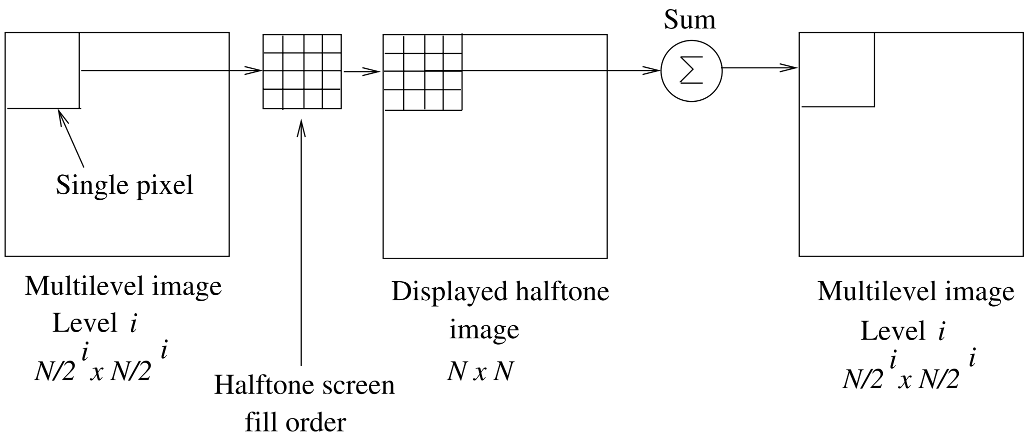

Figure 5. Screening for display and storage

4. MEMORY AND COMPUTATIONAL COMPLEXITY

A major advantage of the P-HIP algorithm is the saving in memory achieved by utilizing a single frame buffer as compared to the method of independent reconstruction and halftoning which requires separate memory for storage of the contone image at its current resolution. With P-HIP, at any given point in time, only a single display size buffer is required to store the halftone as well as to display it. This can be a useful feature in low cost, portable display devices where memory is at a premium. Since a single buffer is used for display as well as storage, for a display size of $M \times N$ pixels, this accounts for a memory saving of $M \times N$ bytes (assuming 8 bit images).

Recovery of the contone image from the fill order screened output halftone can impose an additional computational burden. This computation involves counting the number of 'on' pixels in each block of halftone output corresponding to a single pixel in the contone image. A significant reduction in this computation can be achieved by incorporating a binary search operation within this process. For a screen size of $N \times N$, straightforward counting requires a maximum of N^2 operations. Instead, a binary search within the screen using the known fill order for the screen will reduce this figure to $\log_2(N^2)$ operations. We begin the search at the midpoint of the tonal range. For example, for an 8 bit image(256 gray levels), we begin at the midpoint of the tonal range i.e. 128. If the pixel in the halftone block corresponding to the location marked 128 in the fill order is 'on', it is obvious that the pixel value in the contone image being recovered is greater than 128. Therefore, we restrict the search to the range of values between 128 and 255. If not, we restrict the search to values between 0 and 127. This process is repeated recursively until the final value is obtained.The use of suitable hardware that automatically interprets the bits within the chosen block of binary values as a single multilevel value may allow further improvements in efficiency.

4.1. Computational simplification

Section 2.2 described the multiresolution progression of a halftone display based on the relationship between increasing tonal resolution (increasing screen/block sizes) and decreasing spatial resolution. Beyond a certain block size, the number of pixels in each block may equal the maximum number of gray levels in the original image. Since this size represents all the gray levels in the original image, larger block sizes beyond this stage may be considered redundant. Thus, the block size may be fixed beyond this point to still lower the computation required. For example, for an 8 bit image (256 gray levels), the block size may be fixed at 16×16 beyond the fourth level of decomposition.

5. EXPERIMENTAL SETUP AND RESULTS

For illustration we consider a simple multiresolution decomposition based on the Haar wavelet transform. To simulate a progressive-by-resolution image transmission environment the input image is decomposed using an elementary Haar wavelet transform. The process is reversible and the original image can be recovered completely from the decomposed low resolution image by inversion. The computation of the Haar wavelet coefficients is illustrated in Figure 7.

In our implementation, the step-by-step addition of image detail to the decomposed low resolution image represents the progression of resolution. For a single level of decomposition, the low resolution image corresponds to the LL subband and the next level of detail corresponds to a combination of the LH, HL and HH subbands. In the beginning, the low resolution image undergoes multilevel error diffusion to yield a quantized output image with fewer gray levels, thus reducing its tonal resolution. This quantized image is then displayed on screen using a Bayer fill pattern[13] of order n where $2^n + 1$ is the maximum number of gray levels in the image.

For example a 65-level image is represented using an 8×8 Bayer fill pattern. Beginning with an all-zero blank image, the fill pattern outputs a particular number of white dots in a given region corresponding to one pixel depending upon that particular pixel's gray level intensity. The number of white dots are arithmetically counted to recover the original gray level from the halftoned output. The image detail for the next stage is then added thus increasing its spatial resolution. The process is repeated, gradually reducing tonal resolution and increasing spatial resolution at each stage. At the final stage the input image is simply error diffused to produce the final halftoned bi-level output image.

(a) The Lena image (256 by 256 pixels) halftoned by P-HIP

(b) The Lena image (256 by 256 pixels) halftoned by Floyd-Steinberg error diffusion

(c) The Cameraman image (256 by 256 pixels) halftoned by P-HIP

(d) The Cameraman image (256 by 256 pixels) halftoned by Floyd-Steinberg error diffusion

Figure 6. Comparison between halftone results from P-HIP and Floyd-Steinberg error diffusion

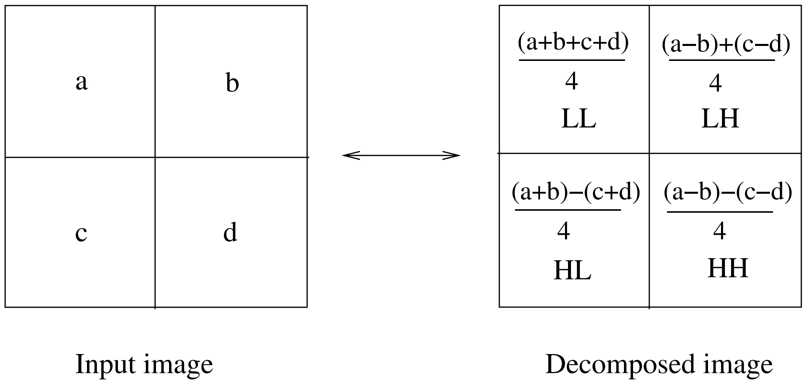

Figure 7. 2D Haar wavelet decomposition

A software application to demonstrate the P-HIP algorithm in a simulated progressive transmission environment has been implemented. We illustrate the results by comparing the output of the final stage of the P-HIP algorithm to that of the conventional Floyd Steinberg error diffusion. Figures 6(a), 6(c) and 6(b), 6(d) show halftones generated using the P-HIP algorithm and the Floyd Steinberg error diffusion algorithm respectively. From the displayed images we can see that the final images produced by the two methods are comparable in quality.

6. CONCLUSION

We describe a new halftoning approach intended for use in a progressive transmission environment. The algorithm uses a combination of multilevel error diffusion and screening to render progressively incoming image data on a bi-level display. The algorithm offers memory efficiency by utilizing a single frame buffer to store the displayed halftoned image as well as the contone image required for further image refinement with each progression of incoming data. The resulting halftones are shown to be comparable in quality to the those from the generally accepted Floyd-Steinberg error diffusion algorithm.

REFERENCES

1. W. H. F. Talbot, "Improvements in the art of engraving." British Patent Specification No. 565, 29 Oct. 1852.
2. P. Jones, "Evolution of halftoning technology in the United States patent literature," *J. Electronic Imaging* **3**, pp. 257–275, July 1994.
3. J. C. Stoffel and J. F. Moreland, "A survey of electronic techniques for pictoral image reproduction," *IEEE Trans. Comm.* **COM-29**, pp. 1898–1925, Dec. 1981.
4. J. P. Allebach, ed., *Selected Papers on Digital Halftoning*, vol. MS 154, SPIE, Bellingham, WA, 1999.
5. D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Practice and Standards*, Kluwer Academic Publishers, 2001.
6. M. Rabbani and R. Joshi, "An overview of the JPEG2000 still image compression standard," *Signal Proc.: Image Comm.* **5150**, pp. 791–805, 2003.
7. "JPEG 2000:part 9 - Interactivity tools, APIs and protocols (JPIP)." Final committee draft, published by JPEG 2000 Standards Committee, Dec. 2003.
8. R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial grayscale," *Proc. SID* **17**, pp. 75–77, Second Quarter 1976.
9. J. R. Goldschneider, E. A. Riskin, and P. W. Wong, "Embedded multilevel error diffusion," *IEEE Trans. Image Proc.* **6**, pp. 956–964, July 1997.
10. D. Taubman and R. Prandolini, "Architecture, philosophy, and performance of JPIP: Internet protocol standard for JPEG2000," in *Proc. SPIE: Visual communications and image processing*, T. Ebrahimi and T. Sikora, eds., pp. 791–805, June 2003.

11. I. Katsavounidis and J. Kuo, "A multiscale error diffusion technique for digital halftoning," *IEEE Trans. Image Proc.* **6**, pp. 483–490, Mar. 1997.
12. P. W. Wong, "Halftoning by multiscale dot distribution," in *Proc. IEEE Intl. Conf. Image Proc.*, **III**, pp. 3117–3121, Oct. 1995.
13. B. E. Bayer, "An optimum method for two level rendition of continuous tone pictures," in *IEEE Intl. Conf. Comm.*, **1**, pp. 26.11–26.15, 1973.