Optimized Halftoning Using Dot Diffusion and Methods for Inverse Halftoning

Murat Meşe, *Student Member, IEEE,* and P. P. Vaidyanathan, *Fellow, IEEE*

***Abstract*—Unlike the error diffusion method, the dot diffusion method for digital halftoning has the advantage of pixel-level parallelism. However, image quality offered by error diffusion is still regarded as superior to most of the other known methods. In this paper, we show how the dot diffusion method can be improved by optimization of the so-called class matrix. By taking the human visual characteristics into account we show that such optimization consistently results in images comparable to error diffusion, without sacrificing the pixel-level parallelism. Adaptive dot diffusion is also introduced and then a mathematical description of dot diffusion is derived. Furthermore, inverse halftoning of dot diffused images is discussed and two methods are proposed. The first one uses projection onto convex sets (POCS) and the second one uses wavelets. Of these methods, the wavelet method does not make use of the knowledge of the class matrix. Embedded multiresolution dot diffusion is also discussed, which is useful for rendering at different resolutions and transmitting images progressively.**

***Index Terms*—blue noise, dot diffusion, error diffusion, halftoning, inverse halftoning, stochastic screening.**

I. Introduction

DIGITAL halftoning is the rendition of continuous-tone pictures on displays that are capable of producing only two levels. There are many good methods for digital halftoning: ordered dither [1], error diffusion [2], neural-net based methods [3], and, more recently, direct binary search (DBS) [4]. Ordered dithering is a thresholding of the continuous-tone image with a spatially periodic screen [1]. In error diffusion [2], the error is "diffused" to the unprocessed neighboring pixels.

Ordered dithering is a parallel method, requiring only pointwise comparisons. But the resulting halftones suffer from periodic patterns. On the other hand, error diffused halftones do not suffer from periodicity and offer blue noise characteristic [5] which is found to be desirable.[1]

The main drawback is that error diffusion is inherently serial.[2] Also, there occur worm-like patterns in near mid-gray regions and resulting halftones have ghosting problems [8]. Mitsa and Parker have optimized the ordered dither matrix [9] for large sizes like 256×256 to get the blue noise effect. This is a compromise between parallelism and image quality.

The dot diffusion method for halftoning introduced by Knuth [8] is an attractive method which attempts to retain the good features of error diffusion while offering substantial parallelism. However, surprisingly, not much work has been done on optimization of the so-called class matrix. In this work, we will show that the class matrix can further be optimized by taking into account the properties of human visual system (HVS). The resulting halftones will then be of the similar quality as for error diffusion. Since dot diffusion also offers increased parallelism, it now appears to be an attractive alternative to error diffusion.

In this paper, we first review the dot diffusion method in Section II. In Section III, the optimization of class matrix will be discussed and adaptive dot diffusion will be introduced. In Section IV, we will give a mathematical description of dot diffusion method. Furthermore we will address the inverse halftoning problem in Sections V–VII. Inverse halftoning has a wide range of applications such as compression, printed image processing, scaling, enhancement, etc. In these applications, operations can not be done on the halftone image directly, and inverse halftoning is mandatory. For inverse halftoning, two methods are discussed. One of the methods uses projection onto convex sets (POCS) which is an iterative algorithm. The other one is based on wavelet decomposition of images to differentiate the halftoning noise from the original image. Then a simple yet efficient algorithm for inverse halftoning of dot diffused images is proposed and compared to other methods. In Section VIII, embedded multiresolution dot diffusion is discussed. Preliminary versions of parts of this paper have been presented at recent conferences [10]–[12].

II. Review of Dot Diffusion

The dot diffusion method for halftoning has only one design parameter, called *class matrix C*. It determines the order in which the pixels are halftoned. Thus, the pixel positions (n_1, n_2) of an image are divided into IJ classes according to $(n_1 \bmod I,\ n_2 \bmod J)$ where I and J are constant integers. Table I is an example of the class matrix for $I = J = 8$, used by Knuth. There are 64 class numbers. Let $x(n_1, n_2)$ be the continuous tone (contone) image with pixel values in the normalized range $[0, 1]$. Starting from class $k = 1$, we process the pixels for increasing values of k. For a fixed k, we take all

Manuscript received December 23, 1998; revised October 24, 1999. This work was supported by the Office of Naval Research under Grant N00014-93-1-0231. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Brian L. Evans.

The authors are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: mese@systems.caltech.edu,; ppvnath@sys.caltech.edu).

Publisher Item Identifier S 1057-7149(00)02672-5.

[1]More recently, it has been shown that *green noise* is more appropriate for nonideal printers, which suffer from dot gain [6]. In this paper we consider ideal printer models.

[2]It can be shown [7] that error diffusion for an $M \times N$ image can, in principle, be implemented in $M + N$ steps by using sufficient number of parallel computations.

pixel locations (n_1, n_2) belonging to class k and define the halftone pixels to be

$$h(n_1, n_2) = \begin{cases} 1, & \text{if } x(n_1, n_2) \geq 0.5 \\ 0, & \text{if } x(n_1, n_2) < 0.5. \end{cases} \quad (1)$$

We also define the error $e(n_1, n_2) = x(n_1, n_2) - h(n_1, n_2)$. We then look at the eight neighbors of (n_1, n_2) and replace the contone pixel with an adjusted version for those neighbors which have a higher class number (i.e., those neighbors that have not been halftoned yet). To be specific, neighbors with higher class numbers are replaced with

$$x(i, j) + 2e(n_1, n_2)/w \quad \text{(for orthogonal neighbors)} \quad (2a)$$
$$x(i, j) + e(n_1, n_2)/w \quad \text{(for diagonal neighbors)} \quad (2b)$$

where w is such that the sum of errors added to all the neighbors is exactly $e(n_1, n_2)$. The extra factor of two for orthogonal neighbors (i.e., vertically and horizontally adjacent neighbors) is because vertically or horizontally oriented error patterns are more perceptible than diagonal patterns.

The contone pixels $x(n_1, n_2)$ which have the next class number $k + 1$ are then similarly processed. The pixel values $x(n_1, n_2)$ are of course not the original contone values but the adjusted values according to earlier diffusion steps (2). When the algorithm terminates, the signal $h(n_1, n_2)$ is the desired halftone.

This diffusion process is illustrated in Fig. 1. The numbers in the matrix are elements of a class matrix and the integers in the bubbles are relative weights of diffusion coefficients. The neighbors of 33 with higher class numbers are those labeled as 58, 45, 42, 40, 63, and 47. The error created at 33 is divided by the sum of relative weights of diffusion coefficients, which is $2+1+2+1+2+1 = 9$ in this case. The result of the division, ϵ, is the error to be diffused to diagonal neighbors, and 2ϵ is diffused to orthogonal neighbors. Since there are 64 classes, the algorithm completes the halftoning in 64 steps.

Usually, an image is enhanced [8] before dot diffusion is applied. For this the continuous image pixels $C(i, j)$ are replaced by $C'(i, j) = (C(i, j) - \alpha\overline{C}(i, j))/(1 - \alpha)$ where $\overline{C}(i, j) = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} C(u, v)/9$. Here, the parameter α determines the degree of enhancement. If $\alpha = 0$, there is no enhancement, and the enhancement increases as α increases. If $\alpha = 0.9$ then the enhancement filter simplifies to

$$C'(i, j) = 8C(i, j) + C(i, j) - \sum_{0<(u-i)^2+(v-j)^2<3} C(u, v).$$

This algorithm is completely parallel requiring nine additions per pixel, and no multiplications.

III. Optimization of Class Matrix

Knuth introduced the notion of *barons* and *near-barons* in the selection of his class matrix. A baron has only low-class neighbors, and a near-baron has one high class neighbor. The quantization error at a baron cannot be distributed to neighbors, and the error at a near-baron can be distributed to only one neighbor. Knuth's idea was that the number of barons and near-barons should therefore be minimized. He exhibited a class matrix with

TABLE I
Class Matrix C Used in Knuth's Method

35	49	41	33	30	16	24	32
43	59	57	54	22	6	8	11
51	63	62	46	14	2	3	19
39	47	55	38	26	18	10	27
29	15	23	31	36	50	42	34
21	5	7	12	44	60	58	53
13	1	4	20	52	64	61	45
25	17	9	28	40	48	56	37

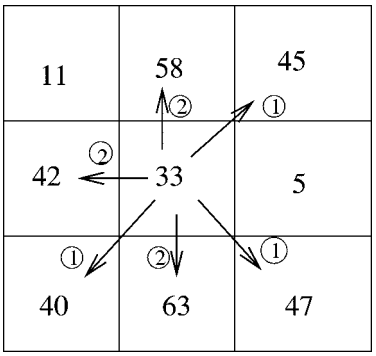

Fig. 1. Error diffusion from a point to the neighbor points.

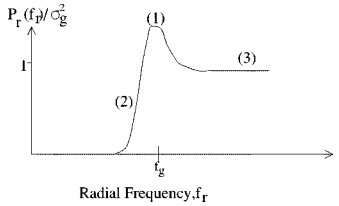

Fig. 2. Typical desired radial spectrum characteristics.

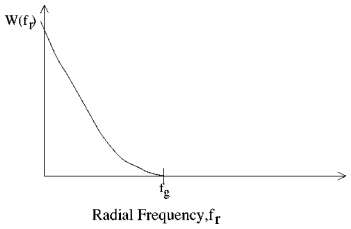

Fig. 3. Weight function used in the optimization.

two barons and two near-barons (Table I). The quality of the resulting halftones still exhibits periodic patterns similar to ordered dither methods (see Fig. 6). Knuth has also produced a class matrix with one baron and near-baron, but unfortunately these were vertically lined up to produce objectionable visual artifacts. In our experience, the baron/near-baron criterion does not appear to be the right choice for optimization. To explain this, define a k-baron to be a position which has k high-level neighbors. Thus $k = 0$ corresponds to a baron, $k = 1$ to a near baron, and $k = 8$ to an *antibaron*. We have produced a class matrix which minimizes the number of k-barons sequentially for $0 \leq k \leq 8$. The resulting halftone quality was found in most cases to be slightly worse than Knuth's original results,

Fig. 4. Original image, peppers.

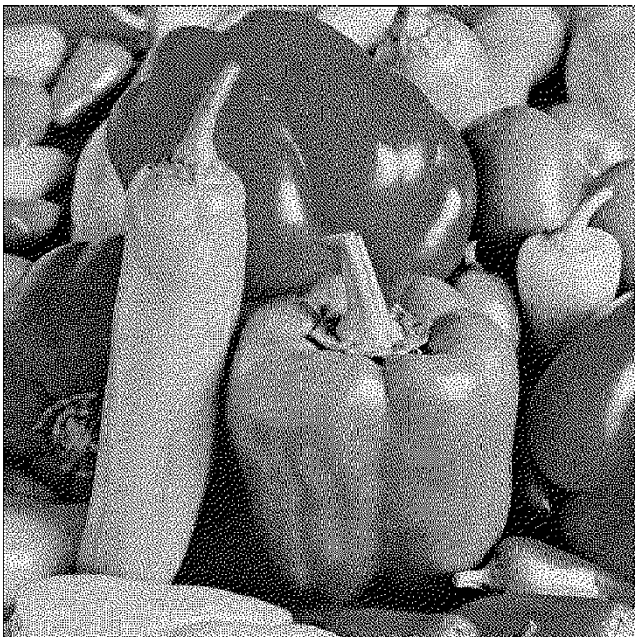

Fig. 5. Floyd–Steinberg error diffusion.

leading us to conclude that baron minimization is not the right approach. In Section III-A we introduce a different optimization criterion based on the HVS, and show that the image quality is significantly improved, though the class matrix does not minimize barons.

A. Objective Function Based on Blue Noise

It has been observed in the past that the error in a good halftone should have the *blue noise* property [5]. This means that the noise energy should mostly be in the high frequency region where it is known to be less perceptible. We will show how to incorporate blue noise characteristics into the class matrix optimization.

Imagine that we have a constant gray image $x(n_1, n_2) = g$ where $0 \leq g \leq 1$. Let $h(n_1, n_2)$ denote the halftoned version. Since the halftone is supposed to create the perception of the gray level, the average number of dark pixels should be equal to the original gray level.[3] Typically, therefore, the dark pixels are spatially distributed with a certain average frequency f_g called the *principal frequency*, which increases with gray level g. The preference for blue noise [5] (high-frequency white noise) in halftoning arises because noise energy at a significantly higher spatial frequency than f_g is less perceivable. Thus, we can optimize a halftoning method for a particular gray level g by forcing the noise spectrum to be concentrated above f_g.

This does not, however, imply optimality at other gray levels. Interestingly however, if the gray level g during the optimization phase is chosen carefully, the resulting halftones for arbitrary natural images are excellent. For example we optimized the class matrix in the dot diffusion method for the gray level $g = \frac{1}{16}$ and obtained very good halftones for natural images as we demonstrate in this section.

[3]Note that a grey level of 0 represents white and a grey level of 1 represents black.

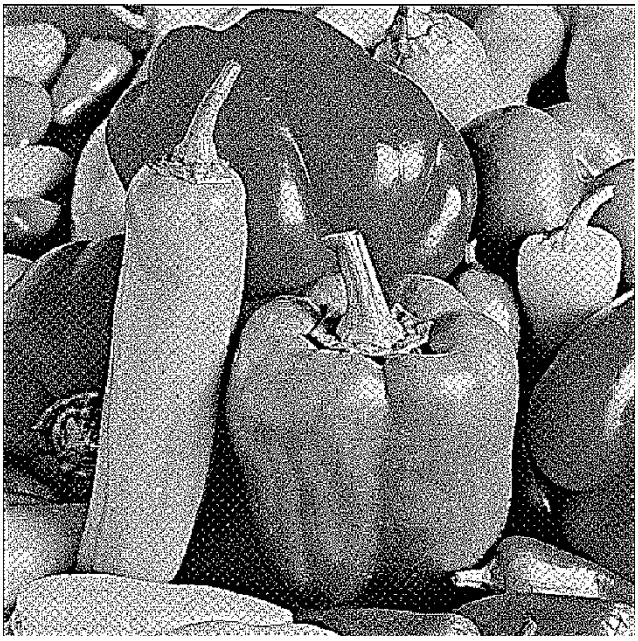

Fig. 6. Dot diffusion with Knuth's class matrix.

Calculating the Noise Spectrum: In order to implement the optimization, we first need to compute the noise spectrum. The halftone pattern $h(n_1, n_2)$ for the gray level $x(n_1, n_2) = g$ has the error $e(n_1, n_2) = g - h(n_1, n_2)$, which is an $N \times N$ image. Imagine that this is divided into $L \times L$ blocks so there are $B = (N/L)^2$ blocks. (In our experiment $N = 256$, $L = 64$, $B = 16$.) Let $E_m(l_1, l_2)$ be the $L \times L$ DFT of the mth block of $e(n_1, n_2)$. We define the average noise spectrum as

$$P(l_1, l_2) = \frac{1}{B} \sum_{m=0}^{B-1} |E_m(l_1, l_2)|^2.$$

TABLE II
CLASS MATRIX C OBTAINED BY PARABOLIC WEIGHTING FUNCTION

59	12	46	60	28	14	32	3
21	25	44	11	58	45	43	30
24	20	13	42	33	5	54	8
64	52	55	40	63	47	7	18
35	57	9	15	50	48	4	36
41	17	6	61	22	49	62	34
2	53	19	56	39	23	26	51
16	37	1	31	29	27	38	10

From this we compute the so-called *radially averaged power spectrum* $P_r(k_r)$ where k_r is a scalar called the radial frequency. Since $|l_1|$ and $|l_2|$ range from 0 to $L/2$, k_r ranges from 0 to $L/\sqrt{2}$. We take specific integer values for k_r and calculate $P_r(k_r)$ as follows. For each chosen k_r define an annulus $\mathcal{A}(k_r)$ in the (l_1, l_2) plane by the equation

$$\left|\sqrt{l_1^2 + l_2^2} - k_r\right| < \Delta/2.$$

The quantity Δ, which determines the width of the annulus, is chosen as unity in our calculation. With $N(k_r)$ denoting the number of elements in $\mathcal{A}(k_r)$, the radially averaged power spectrum of the error for gray level g is then

$$P_r(k_r) = \frac{1}{N(k_r)} \sum_{l_1, l_2 \in \mathcal{A}(k_r)} P(l_1, l_2).$$

The class matrix in the dot diffusion method should be optimized such that this radial spectrum is appropriately shaped for a well-chosen fixed gray level g. In terms of the radial frequency variable k_r, the principal frequency for the halftone of gray level g is given by

$$f_g = k_{\max}\sqrt{g}$$

where $k_{\max} = L$. In fact, for $g > 0.5$, since black pixels are more in number, the halftone is perceived as a distribution of white dots [5] and we have to take $f_g = k_{\max}\sqrt{1-g}$.

The aim of the optimization is to shape $P_r(k_r)$ by choice of the class matrix $\boldsymbol{C}$ so that most of its energy is moved to the region $k_r > f_g$ (as demonstrated in Fig. 2). We therefore define the cost function

$$\Phi(\boldsymbol{C}, g) = \int_0^{f_g} P_r(k_r) w(k_r)\, dk_r.$$

The idea is to choose the weighting function $w(k_r)$ such that upon minimization of the above function, $P_r(k_r)$ has a low frequency cutoff at principal frequency f_g, sharp transition region, and a flat high frequency region. The weight function was chosen to be $w(k_r) = (k_r - f_g)^2$ for $0 \le k_r \le f_g$ and zero outside (see Fig. 3). (In Section III-B we consider more sophisticated weighting functions.) In the optimization the integral was replaced with a discrete sum. The choice of the class matrix that minimizes this sum was performed using the *pairwise exchange algorithm* [13] described as follows.

1) Randomly order the numbers in the class matrix.
2) List all possible exchanges of class numbers.
3) If an exchange does not reduce cost, restore the pair to original positions and proceed to the next pair.
4) If an exchange does reduce cost, keep it and restart the enumeration from the beginning.
5) Stop searching if no further exchanges reduce cost.

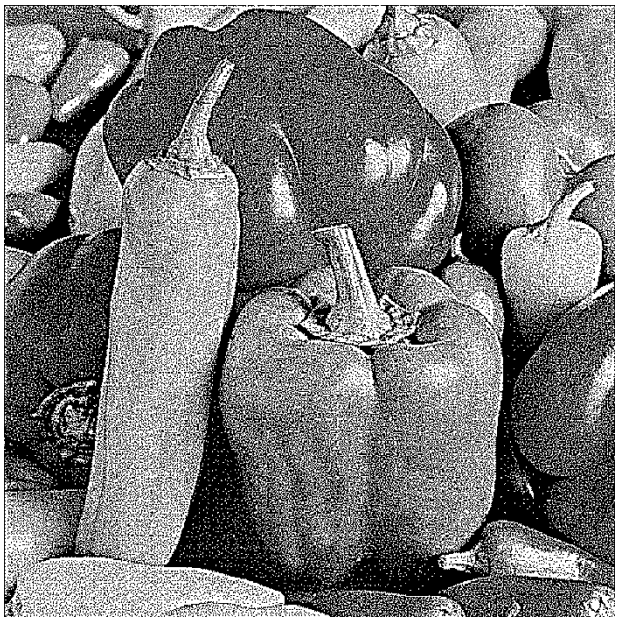

Fig. 7. Dot diffusion with enhancement and 8×8 class matrix optimized using parabolic weighting function.

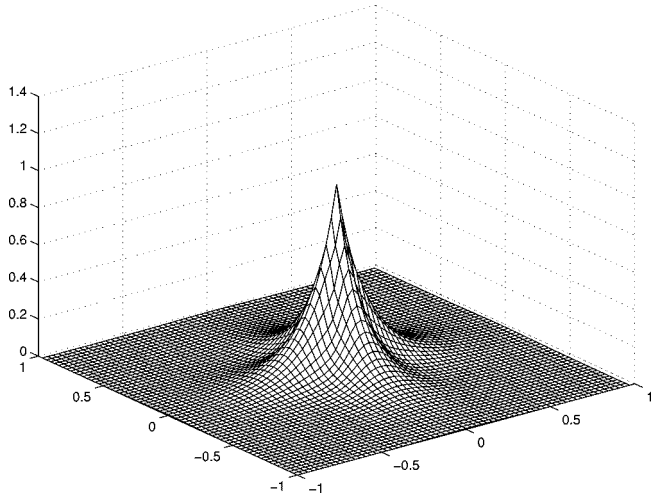

Fig. 8. HVS function $H(u, v)$ for $T = 0.2$. The axes are u/π and v/π.

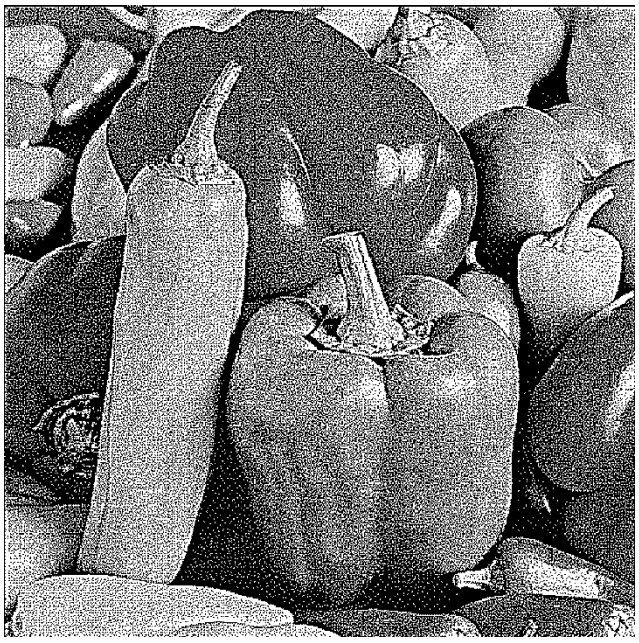

Fig. 9. Dot diffusion with enhancement and 8×8 class matrix optimized using HVS function.

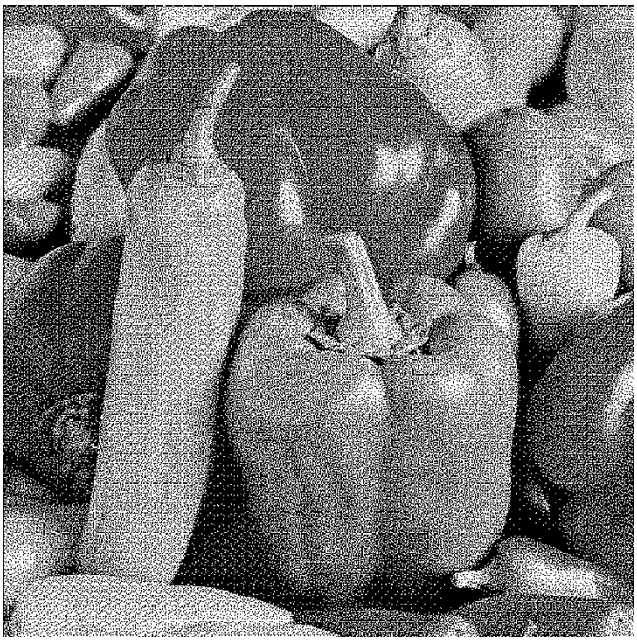

Fig. 10. Dot diffusion with no enhancement and 8×8 class matrix optimized using HVS function.

6) Repeat the above steps a fixed number of times and keep the best class matrix.[4]

Choice of Gray Level: Since the algorithm can be applied only to a given gray level, the gray level should be chosen wisely, in order to get good halftones for other gray levels also. In our experience if we perform this optimization for a fixed small gray level (e.g., $g = \frac{1}{16}$, $\frac{1}{8}$, etc.), we get good halftones for natural images also. Class matrices obtained from optimization with a very small gray level will not work, because there is not much error to diffuse to other points during the dot diffusion process. Mid gray levels are not suitable, first because there are huge diffusions between points, and second, even unoptimized algorithms yield perceptually pleasing halftones for mid gray anyway. The actual gray level used in the optimization was $g = \frac{1}{16}$, and it was found experimentally. The optimized class matrix is shown in Table II. Notice that the optimal class matrix has *several barons and near barons*.

Example: The 512×512 continuous tone peppers image (Fig. 4) was halftoned by using Knuth's class matrix (Fig. 6), and by the optimized class matrix (Fig. 7). It is clear that the new method is visually superior to unoptimized dot diffusion method. In fact, the new method offers a quality comparable to Floyd–Steinberg error diffusion method (Fig. 5). Error diffused images suffer from worm-like patterns which are not in the original image, whereas dot diffused halftones do not contain these artifacts. Notice that the artificial periodic patterns in Fig. 6 are absent in Fig. 5 and in the new method (Fig. 7).[5]

[4]Note that pairwise exchange algorithm yields a local minimum of the cost function. We start the pairwise exchange with random class matrices and take the class matrix having the least local minimum in order to get closer to the global minimum. Global minimum is not guaranteed.

[5]The halftone and inverse halftone images can be found in [36].

B. Other Choices for the Weighting Function

For simplicity we have chosen our weighting function above to be the parabola $w(k_r) = (k_r - f_g)^2$ for $0 \leq k_r \leq f_g$ and zero outside. Another alternative is to use the HVS function as the weighting function. The HVS function has been derived in [14] and [15] experimentally. In the frequency domain the HVS function is approximated well by

$$H(u, v) = aL^b \exp\left(-\frac{1}{s(\phi)} \frac{\sqrt{u^2+v^2}}{c \log(L) + d}\right)$$

where $a = 131.6$, $b = 0.3188$, $c = 0.525$, and $d = 3.91$. We used $L = 0.091$ in our experiments where L is the average luminance. Furthermore, the phase dependent function $s(\phi)$ is defined as $s(\phi) = ((1-w)/2)\cos(4\phi) + ((1+w)/2)$ where $w = 0.7$ and $\phi = \operatorname{atan}(u/v)$. With $h(x, y)$ denoting the inverse Fourier transform of $H(u, v)$, the discretized version $h[m, n] = h(Tm, Tn)$ is used in the calculations. In Fig. 8, the HVS function is shown for $T = 0.2$. Notice that the HVS weighting filter has three basic properties:

1) it is a decaying function in terms of frequency;
2) HVS response along the $45°$ line is $1/\sqrt{2}$ of the response to horizontal and vertical lines;
3) weights are nonzero for all frequencies.

Note that the parabolic weight function is circularly symmetric, and becomes zero after a certain frequency. So, the parabolic weighting function does not have properties 2 and 3. We optimized the class matrix with this HVS function as the weighting function. The result is shown in Table III. The dot diffused image of a constant gray level $g = \frac{1}{16}$ with the class matrix optimized using HVS function and the dot diffused image of the same constant gray level with the class matrix optimized using the parabolic weight function are shown in Fig. 11. From

TABLE III
CLASS MATRIX C OBTAINED BY PARABOLIC HVS FUNCTION

48	32	52	25	28	46	6	22
38	64	54	12	23	5	2	34
62	1	58	17	27	30	47	9
21	15	10	63	19	42	39	7
18	14	26	16	56	49	53	59
4	8	3	33	31	35	57	61
29	41	37	40	50	44	36	11
55	24	51	13	43	60	45	20

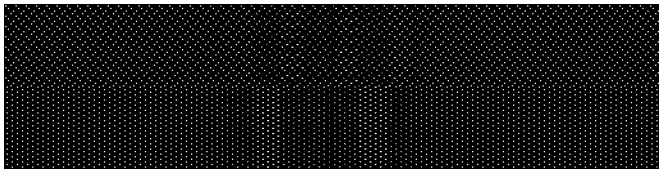

Fig. 11. Optimized dot patterns for $g = \frac{1}{16}$. Top: Using HVS function and bottom: using the parabolic weighting function.

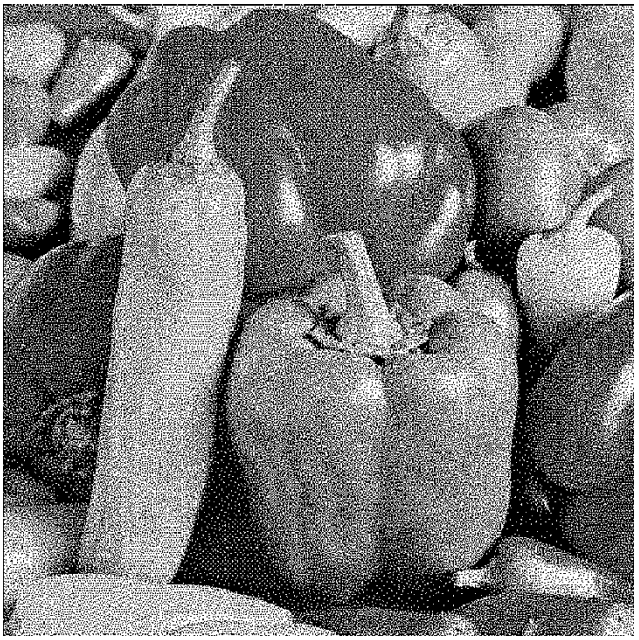

Fig. 12. Adaptive dot diffusion with no enhancement and 8×8 class matrices optimized using HVS function.

the dot patterns it can be seen that the HVS function aligns the pattern in diagonal directions, and the dot pattern looks more irregular. The dot diffused image of the peppers image using the class matrix optimized with the HVS function is shown in Fig. 9. Notice the dark area between the tall pepper and the fat pepper on the left bottom: The vertical patterns in Fig. 7 do not exist in Fig. 9. So, we can conclude that the experimentally derived HVS weighting function is a better cost function.

C. Effect of Diffusion Constants

The diffusion constant β is the ratio between the horizontal diffusion coefficient and the diagonal diffusion coefficient of the dot diffusion process. The diffusion constant has been chosen to be 2.0 by Knuth. The reasons for this selection are that 1) it is desirable to diffuse more errors in the vertical and horizontal directions, and keep more errors in the diagonal directions where the eye sensitivity is known to be lower and 2) it reduces the number of multiplications if β is chosen as a power of two.

In Knuth's method, the choice $\beta = 2$ was rather crucial because there was no optimization of the class matrix. We have found experimentally that when the class matrix is optimized for a chosen β, the results are relatively insensitive to β as long as it is in the range $1 \leq \beta \leq 2$. However there are slight vertical and horizontal patterns when $\beta = 0.5$. For $\beta \leq 1/4$ and $\beta \geq 4$ more serious artifacts are noticeable. In principle the diffusion coefficient can be chosen with greater degree of freedom. For example we can choose it such that it depends on the class number as well as the direction of diffusion. At this time however we do not have an optimization algorithm to optimize such a general set of diffusion coefficients.

D. Remarks

The optimization of class matrix was done in Section III-A for constant gray level images only. For this, it is necessary to pick the constant gray level strategically such that, for most natural images with multiple gray levels, the halftone quality is good. A natural question here is whether we can we make the class matrix adaptive. For example, imagine a library of optimal class matrices $\{C_i\}$ with C_i optimized for the ith gray level. We can divide the image into 8×8 blocks, and for each block a different class matrix can be used depending on the average gray level there. We have done experiments with this idea. Assuming that the image is enhanced prior to halftoning (as in Section II), we found the advantage of the adaptive scheme to be insignificant. However if there is no enhancement prior to halftoning, then adaptive dot diffusion is significantly better than non adaptive (compare Figs. 10 and 12). Besides the obvious advantages brought into play by adaptation, there is another reason why this is so: the grid of pixels formed by a given class number is not a periodic grid in the adaptive case. Any periodicity artifacts created by the periodic class matrix are therefore broken. But there is another problem, namely the boundary effects between the blocks having different class matrices are apparent.

Another parameter in the dot diffusion is the enhancement filter. The enhancement lessens the objectionable halftoning artifacts in other grey levels which are not close to $g = \frac{1}{16}$. This can be seen from the resulting image obtained by dot diffusion with enhancement (Fig. 7) and the image obtained by the dot diffusion without enhancement (Fig. 10). The periodic patterns show up almost everywhere in the dot diffused images if enhancement is not done prior to halftoning. The enhancement filter used has a parameter α (see Section II) which controls the degree of enhancement where $\alpha = 0$ means no enhancement and $\alpha = 0.9$ is the value used in our experiments. The enhancement parameter α can be lessened to 0.8 without any perceivable difference.

E. Dot Diffusion Without Enhancement

In all the discussions so far, the halftoning step is preceded by an enhancement filter (described in Section II). The enhancement step reduces halftoning noise, but might be objectionable

TABLE IV
THE 16×16 CLASS MATRIX

208	1	14	18	29	56	19	103	82	98	75	145	150	170	171	173
4	7	24	37	57	51	66	88	146	131	138	159	183	185	196	222
8	15	25	38	68	70	87	6	107	153	151	166	184	193	225	2
16	27	44	54	52	102	116	132	140	137	167	120	209	224	227	5
23	40	53	72	85	104	165	136	158	174	114	191	223	226	228	17
41	86	73	84	105	118	168	134	169	181	201	220	232	229	13	22
48	121	55	106	124	133	147	177	180	203	221	231	246	3	21	42
77	74	128	110	139	135	179	182	207	197	230	245	247	20	43	50
81	100	113	148	143	172	178	204	219	233	244	249	248	34	49	69
109	108	141	144	186	164	205	218	234	243	250	256	45	46	71	80
111	142	89	76	176	206	215	235	242	251	255	39	47	78	117	101
112	149	161	175	202	216	236	241	252	253	254	62	63	94	95	126
152	160	190	200	198	217	237	240	26	32	61	83	93	96	125	115
157	189	192	210	214	238	239	30	33	60	65	92	119	79	129	156
188	195	199	213	10	11	31	36	59	64	91	97	123	130	155	162
194	211	212	9	12	28	35	58	67	90	99	122	127	154	163	187

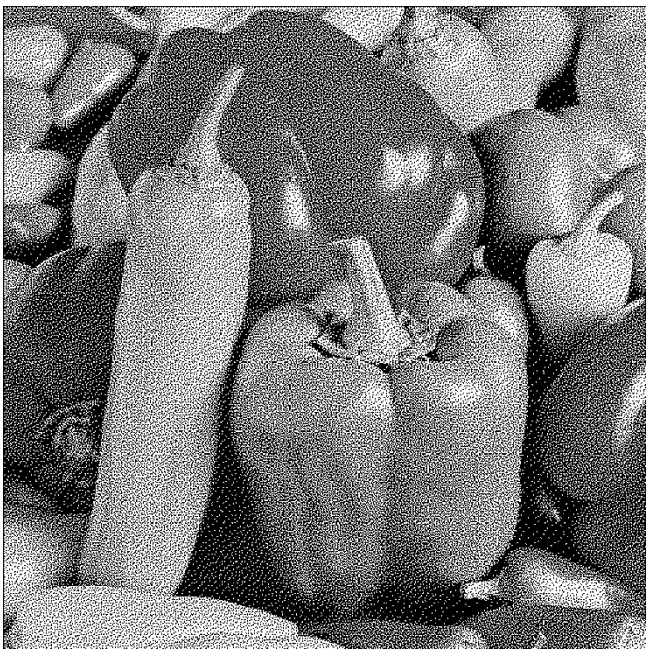

Fig. 13. Dot diffusion with no enhancement and 16×16 class matrix optimized using HVS function.

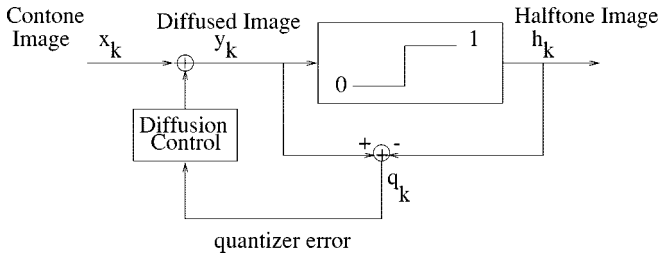

Fig. 14. Schematic representation of the dot diffusion process. Here x_k represents a vector of all pixels belonging to class k.

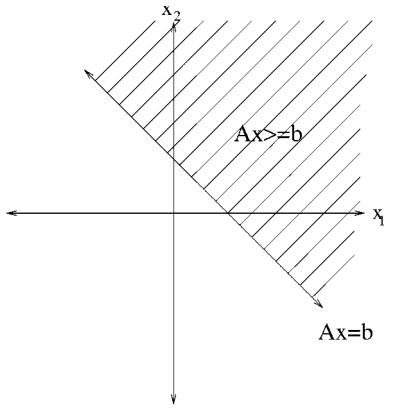

Fig. 15. $Ax \geq b$ is a closed set.

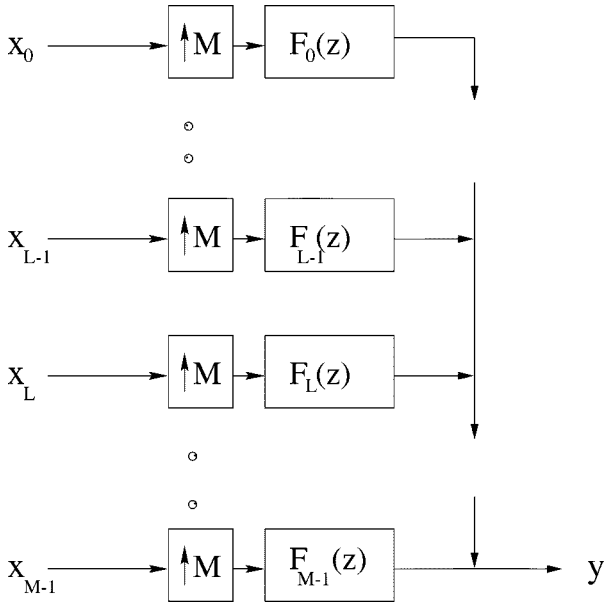

Fig. 16. Synthesis section of an M channel filter bank.

in some applications because of its very visible sharpening effect (e.g., see Fig. 9). It turns out that we can get good halftones without use of the enhancement step provided we make the class matrix larger than the standard 8×8 size. The price paid for the larger class matrix is that the parallelism of the algorithm is compromised. We found that if a 16×16 matrix is used, the halftone images resulting from the optimization of this matrix

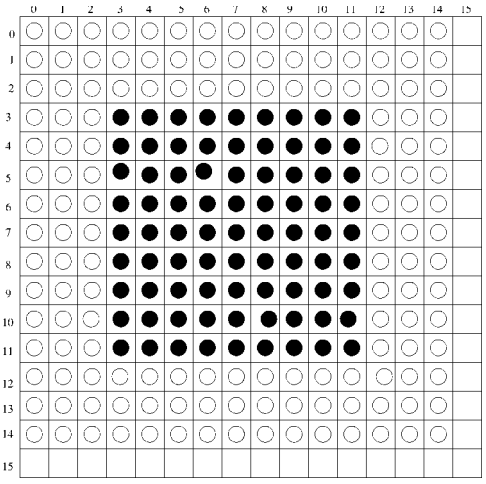

Fig. 17. Overlapping blocks used in approximating the QP problem.

are very good even without the enhancement step. (For comparison we note here that whenever enhancement is used, the class matrix can be as small as 5×5 without creating noticeable periodicity patterns.) Such optimization was carried out using a gray scale ramp as the training image. The HVS function of Section III-A was used in the optimization, and the associated cost was optimized as in Section III-B using the pairwise exchange algorithm. The 16×16 optimized class matrix is shown in Table IV.

The peppers image halftoned with the resulting class matrix is shown in Fig. 13. There are no periodic artifacts in this result. While the overall visible noise level appears to be higher than for error diffusion, the problematic halftone patterns of error diffusion in the mid gray level are eliminated here (examine the body of the middle pepper in Fig. 5). By comparing Figs. 6 and 13, we see that 16×16 dot diffusion without enhancement is also superior to 8×8 enhanced dot diffusion using Knuth's matrix because there are no noticeable periodic patterns any more, and there are no enhancement artifacts.

In order to obtain an authentic comparison with good printing quality we have produced three images in Figs. 31–33 using 150 dpi resolution. These are halftoned versions of the Parrot image. Fig. 31 shows the result of error diffusion, Fig. 32 the result of direct binary search (DBS) obtained from the website of the authors of [15], and Fig. 33 shows the result of using the 16×16 optimized dot diffusion described above. In terms of image quality the DBS method is evidently the best one. The dot diffusion output appears to be comparable to error diffusion in most areas of the image. Dot diffusion has the advantage that the complexity is much lower than that of DBS. Moreover it offers parallelism of implementation unlike error diffusion.

IV. Mathematical Description of Dot-Diffusion

We have defined the dot diffusion process in Section II, but we also want to give a mathematical description of the dot diffusion. With the aid of this description, we can relate the quantizer error to halftone error. In addition to providing further insight, this will also be useful in Section VI for inverse halftoning.

Let us denote the number of classes by L. For example if the class matrix is 8×8 as in Section II, then $L=64$. Let $\boldsymbol{x}_k$ denote a vector whose elements are the pixels of the original contone image belonging to class k in some order. Let $\boldsymbol{x}$ denote a vector whose elements are the pixels, in some order, of the contone image. For example

$$\boldsymbol{x}=\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \vdots\\ \boldsymbol{x}_L\end{bmatrix}.$$

Each of the vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L,$ can be regarded as a polyphase component [16] of the contone image.

A. Quantizer Error q and Halftone Error e

In the dot diffusion process, the pixels which are quantized by the two-level quantizer are modified versions $\boldsymbol{y}_i$ of the original vectors $\boldsymbol{x}_i$, the modification being that we diffuse the quantization errors from lower classes processed earlier. Since the pixels in class 1 are quantized directly, we have

$$\boldsymbol{y}_1=\boldsymbol{x}_1.$$

Let $\boldsymbol{h}_1$ denote the halftone vector obtained from quantizing this to two levels. The quantizer error $\boldsymbol{q}_1=\boldsymbol{y}_1-\boldsymbol{h}_1$ is then diffused to those neighbors of the pixels of $\boldsymbol{x}_1$, which have a higher class number. For example, $\boldsymbol{x}_2$ is replaced with

$$\boldsymbol{y}_2=\boldsymbol{x}_2+\boldsymbol{D}_{21}(\boldsymbol{y}_1-\boldsymbol{h}_1)$$

where $\boldsymbol{D}_{21}$ is a matrix representing the diffusion coefficients [i.e., quantities like $2/w$ and $1/w$ in [(2a)] and (2b)]. We then quantize $\boldsymbol{y}_2$ with the two-level quantizer to produce the halftone $\boldsymbol{h}_2$ for all the pixels in class 2. The quantizer error $\boldsymbol{q}_2=\boldsymbol{y}_2-\boldsymbol{h}_2$ is then diffused to the higher class pixels. For example, consider class 3 pixels. In general these pixels receive diffused error from $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ so that the modified class 3 pixels are represented by the vector

$$\boldsymbol{y}_3=\boldsymbol{x}_3+\boldsymbol{D}_{31}(\boldsymbol{y}_1-\boldsymbol{h}_1)+\boldsymbol{D}_{32}(\boldsymbol{y}_2-\boldsymbol{h}_2).$$

Two-level quantization of $\boldsymbol{y}_3$ then produces the halftone $\boldsymbol{h}_3$ for class 3 pixel positions, and so forth. Thus, in general, the class vector $\boldsymbol{x}_k$ is modified to

$$\begin{aligned}\boldsymbol{y}_k=&\boldsymbol{x}_k+\boldsymbol{D}_{k1}(\boldsymbol{y}_1-\boldsymbol{h}_1)+\boldsymbol{D}_{k2}(\boldsymbol{y}_2-\boldsymbol{h}_2)+\cdots\\ &+\boldsymbol{D}_{k,k-1}(\boldsymbol{y}_{k-1}-\boldsymbol{h}_{k-1})\end{aligned}\tag{3}$$

and then quantized to obtain the halftone $\boldsymbol{h}_k$. Proceeding in this way, the halftone pixels $\boldsymbol{h}_k$ for all classes $1\le k\le L$ are generated. The quantizer error vector $\boldsymbol{q}_k$ and halftone error vector $\boldsymbol{e}_k$ for class k are given by

$$\begin{aligned}\boldsymbol{q}_k&=\boldsymbol{y}_k-\boldsymbol{h}_k \qquad \text{(quantizer error)}\\ \boldsymbol{e}_k&=\boldsymbol{x}_k-\boldsymbol{h}_k \qquad \text{(halftoning error).}\end{aligned}$$

Subtracting $\boldsymbol{h}_k$ from both sides of (3), we get $\boldsymbol{q}_k=\boldsymbol{e}_k+\boldsymbol{D}_{k1}\boldsymbol{q}_1+\boldsymbol{D}_{k2}\boldsymbol{q}_2+\cdots+\boldsymbol{D}_{k,k-1}\boldsymbol{q}_{k-1}$, that is,

$$\begin{aligned}\boldsymbol{q}_1&=\boldsymbol{e}_1\\ \boldsymbol{q}_2&=\boldsymbol{e}_2+\boldsymbol{D}_{21}\boldsymbol{q}_1\\ \boldsymbol{q}_3&=\boldsymbol{e}_3+\boldsymbol{D}_{31}\boldsymbol{q}_1+\boldsymbol{D}_{32}\boldsymbol{q}_2\\ &\vdots\end{aligned}\tag{4}$$

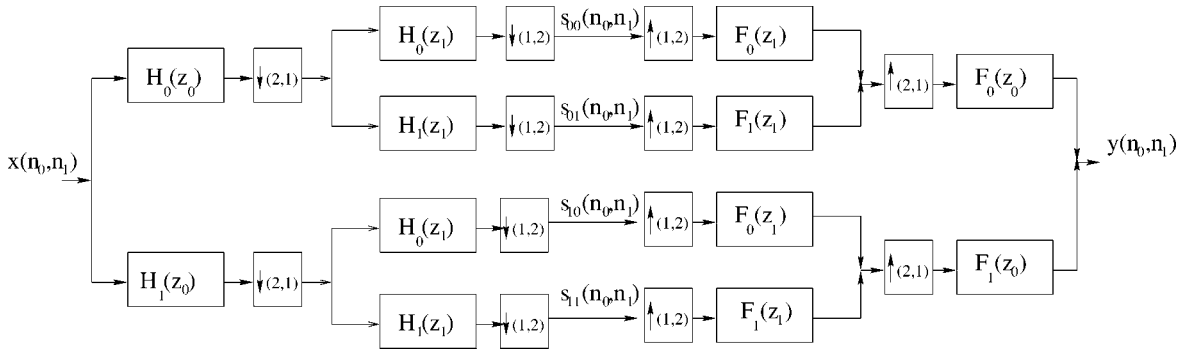

Fig. 18. Two-dimensional separable PR filter bank.

Fig. 19. Inverse halftoned peppers with POCS (transform domain projection applied before halftoning).

Fig. 20. Inverse halftoned Lena with POCS (transform domain projection applied before halftoning).

By starting from the first equation, we can sequentially replace $\boldsymbol{q}_i$ in terms of $\boldsymbol{e}_i$, $\boldsymbol{e}_{i-1}$, $\cdots$, on the right side of (4), resulting in an expression of the form

$$\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{q}_L \end{bmatrix} = \boldsymbol{A}_L \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_L \end{bmatrix}$$

where $\boldsymbol{A}_L$ is a matrix depending on the elements of the smaller matrices $\boldsymbol{D}_{ij}$. We now show that $\boldsymbol{A}_L$ can be generated from $\boldsymbol{A}_{L-1}$ as follows: from (4) we can express $\boldsymbol{q}_L$ as

$$\boldsymbol{q}_L = [\boldsymbol{D}_{L1} \quad \boldsymbol{D}_{L2} \quad \cdots \quad \boldsymbol{D}_{L,L-1}] \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{q}_{L-1} \end{bmatrix} + \boldsymbol{e}_L$$

$$= [\boldsymbol{D}_{L1} \quad \boldsymbol{D}_{L2} \quad \cdots \quad \boldsymbol{D}_{L,L-1}] \boldsymbol{A}_{L-1} \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_{L-1} \end{bmatrix} + \boldsymbol{e}_L$$

Fig. 21. Inverse halftoned peppers with POCS (transform domain projection is not applied before halftoning).

Fig. 22. Inverse halftoned Lena with POCS (transform domain projection is not applied before halftoning).

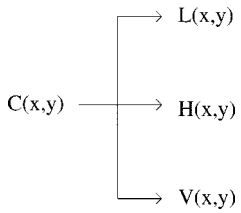

Fig. 23. Wavelet decomposition of an image.

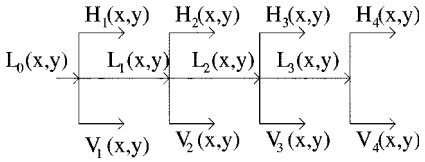

Fig. 24. Wavelet tree used in inverse halftoning.

$$= [\boldsymbol{D}_{L1} \quad \boldsymbol{D}_{L2} \quad \cdots \quad \boldsymbol{D}_{L,L-1} \quad \boldsymbol{I}] \begin{bmatrix} \boldsymbol{A}_{L-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_{L-1} \\ \boldsymbol{e}_L \end{bmatrix}$$

which shows that

$$\underbrace{\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{q}_L \end{bmatrix}}_{\boldsymbol{q}} = \underbrace{\begin{bmatrix} \boldsymbol{I} & & & \boldsymbol{0} \\ \boldsymbol{D}_{L1} & \boldsymbol{D}_{L2} & \cdots & \boldsymbol{D}_{L,L-1} \quad \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_{L-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{A}_L} \cdot \underbrace{\begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_L \end{bmatrix}}_{\boldsymbol{e}}$$

that is

$$\boldsymbol{q} = \boldsymbol{A}_L \boldsymbol{e}.$$

Similarly, $\boldsymbol{A}_{L-1}$ can be expressed in terms of $\boldsymbol{A}_{L-2}$, and so forth. This gives an expression for $\boldsymbol{A}_L$ as a product of simple matrices, that is, $\boldsymbol{A}_L = \boldsymbol{B}_L \boldsymbol{B}_{L-1} \cdots \boldsymbol{B}_2 \boldsymbol{B}_1$, where $\boldsymbol{B}_k$ represents diffusion of error to class k from all lower classes, and $\boldsymbol{B}_1 = \boldsymbol{I}$. For example

$$\boldsymbol{A}_4 = \underbrace{\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{D}_{41} & \boldsymbol{D}_{42} & \boldsymbol{D}_{43} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{B}_4} \underbrace{\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{D}_{31} & \boldsymbol{D}_{32} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{B}_3} \cdot \underbrace{\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{D}_{21} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{B}_2}. \tag{5}$$

The matrix $\boldsymbol{A}_L$ has determinant equal to unity as seen from the factored expression (5). It is therefore invertible, and we can obtain

$$\boldsymbol{e} = \boldsymbol{T}_L \boldsymbol{q}$$

where $\boldsymbol{T}_L = \boldsymbol{A}_L^{-1}$. Here $\boldsymbol{T}_L$ can be regarded as the transfer function from the quantizer error $\boldsymbol{q}$ to the actual halftoning error. The total halftoning error-squared, defined as $\boldsymbol{e}^T\boldsymbol{e}$, can readily be computed from this if we know the quantizer error $\boldsymbol{q}$.

B. Expression for Diffused Image

Here is a summary of the main points of the preceding discussions. The *original contone image* is made from pixels in the vectors $\boldsymbol{x}_k$. The *diffused image* $\boldsymbol{y}$ is made from the pixels in $\boldsymbol{y}_k$ which are inputs to the two level quantizer. The image $\boldsymbol{h}$ whose pixels come from $\boldsymbol{h}_k$ is the *halftone image*. The pixels from the original contone, diffused, and halftone images can be arranged in the form of vectors $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{h}$ as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_L \end{bmatrix}, \qquad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_L \end{bmatrix}, \qquad \boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \vdots \\ \boldsymbol{h}_L \end{bmatrix}.$$

The quantizer error vector $\boldsymbol{q}$ and halftone error vector $\boldsymbol{e}$ are defined as

$$\boldsymbol{q} = \boldsymbol{y} - \boldsymbol{h}, \qquad \boldsymbol{e} = \boldsymbol{x} - \boldsymbol{h}.$$

The diffusion process is schematically depicted in Fig. 14. We can now express the diffused image $\boldsymbol{y}$ in terms of the original contone image $\boldsymbol{x}$ and the halftone image $\boldsymbol{h}$ as follows: $\boldsymbol{y} = \boldsymbol{q} + \boldsymbol{h} = \boldsymbol{A}_L\boldsymbol{e} + \boldsymbol{h} = \boldsymbol{A}_L(\boldsymbol{x} - \boldsymbol{h}) + \boldsymbol{h}$that is

$$\boldsymbol{y} = \boldsymbol{A}_L\boldsymbol{x} + (\boldsymbol{I} - \boldsymbol{A}_L)\boldsymbol{h}. \tag{6}$$

This expression allows us to characterize the so-called inverse halftone set in a nice way. Let y_i and h_i denote, respectively, the

ith scalar component of $\boldsymbol{y}$ and $\boldsymbol{h}$. Since y_i is directly quantized to yield h_i we see that

$$y_i \begin{cases} \geq 0.5, & \text{if } h_i = 1 \\ < 0.5, & \text{if } h_i = 0. \end{cases} \tag{7}$$

Given a halftone image $\boldsymbol{h}$ and the halftone algorithm [e.g., dot diffusion with class matrix (as in Table II)], the *inverse halftone set* $\mathcal{C}$ is the collection of all image vectors $\boldsymbol{x}$ which yield the halftone image $\boldsymbol{h}$. That is, an image $\boldsymbol{x}$ belongs to $\mathcal{C}$ if and only if the vector $\boldsymbol{y}$ computed using (6) satisfies (7).

Notice that if we set $\boldsymbol{x} = \boldsymbol{h}$ in (6) then we get $\boldsymbol{y} = \boldsymbol{h}$ which implies in particular that (7) holds. Thus the halftone image $\boldsymbol{h}$ itself is a member of $\mathcal{C}$. That is, if we perform dot diffusion again on the halftone image $\boldsymbol{h}$, the result is still $\boldsymbol{h}$.

C. Closed Convex Subset of the Inverse Halftone Set

We will see now that the inverse halftone set $\mathcal{C}$ is convex but not closed. We then show how to construct a subset $\mathcal{S}_1 \subset \mathcal{C}$ which is closed and convex. This will be useful in Section VI where we create a contone image from the dot-diffused halftone using the POCS method. For convenience of discussion let us renumber the elements of the halftone $\boldsymbol{h}$ such that it can be partitioned as

$$\boldsymbol{h} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \qquad \text{where} \quad \mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

The elements of $\boldsymbol{x}$, $\boldsymbol{y}$, and the matrix $\boldsymbol{A}_L$ are also renumbered accordingly. Then the diffused image vector $\boldsymbol{y}$ has the form

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_a \\ \boldsymbol{y}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_a \\ \boldsymbol{A}_b \end{bmatrix} \boldsymbol{x} + \boldsymbol{c}$$

where $\boldsymbol{A}_a$, $\boldsymbol{A}_b$, and $\boldsymbol{c}$ do not depend on $\boldsymbol{x}$ or $\boldsymbol{y}$.

The diffused image vector $\boldsymbol{y}$ is such that $\boldsymbol{y}_a \geq 0.5 \times \mathbf{1}$ and $\boldsymbol{y}_b < 0.5 \times \mathbf{1}$ where the vector inequalities in the previous equation should be interpreted on an element by element basis. That is, the inverse halftone set $\mathcal{C}$ is the set of all image vectors $\boldsymbol{x}$ such that

$$\boldsymbol{A}_a \boldsymbol{x} \geq \boldsymbol{d}_a \qquad \text{and} \qquad \boldsymbol{A}_b \boldsymbol{x} < \boldsymbol{d}_b. \tag{8}$$

Given two image vectors $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ satisfying (8), we can readily verify that the linear combination $\alpha \boldsymbol{x}^{(1)} + (1-\alpha)\boldsymbol{x}^{(2)}$ also satisfies (8) whenever $0 \leq \alpha \leq 1$. This shows that the set $\mathcal{C}$ is convex.

The set of all $\boldsymbol{x}$ satisfying $\boldsymbol{A}_a \boldsymbol{x} \geq \boldsymbol{d}$ is interpreted geometrically as in Fig. 15 for the case where $\boldsymbol{x}$ is a two-dimensional (2-D) vector. Since the boundary $\boldsymbol{A}_b \boldsymbol{x} = \boldsymbol{d}$ is included, this is a closed set [17]. The set $\boldsymbol{A}_b \boldsymbol{x} < \boldsymbol{d}$ has a similar interpretation, but since the boundary is not included, it is not closed. The intersection of the two sets described by $\boldsymbol{A}_a \boldsymbol{x} \geq \boldsymbol{d}_a$ and $\boldsymbol{A}_b \boldsymbol{x} < \boldsymbol{d}_b$ is therefore not closed. Summarizing, *the inverse halftone set $\mathcal{C}$ for a dot-diffused halftone $\boldsymbol{h}$ is a convex set but it is not closed.*

D. Digitized Subset

Now consider a subset $\mathcal{D} \subset \mathcal{C}$ such that all images in $\mathcal{D}$ are digitized to, say, 8 bits/pixel. The set $\mathcal{D}$ is clearly not empty because the halftone image $\boldsymbol{h}$ is certainly a member of $\mathcal{D}$. With $\boldsymbol{x}$ chosen from this digitized subset $\mathcal{D}$, the elements y_i of $\boldsymbol{y}$ also take values from a discrete set. So we can always find an $\epsilon > 0$ such that none of the y_i's fall in the open interval $(0.5-\epsilon, 0.5)$. That is, not only is (7) satisfied, but the following stronger condition holds:

$$y_i \begin{cases} \geq 0.5, & \text{if } h_i = 1 \\ \leq 0.5 - \epsilon, & \text{if } h_i = 0 \end{cases} \tag{9}$$

for some fixed $\epsilon > 0$ that can be precalculated from $\boldsymbol{A}$ and $\boldsymbol{h}$. By following the kind of reasoning that resulted in (8), we see that if $\boldsymbol{x}$ is in the digitized subset $\mathcal{D}$, then

$$\boldsymbol{A}_a \boldsymbol{x} \geq \boldsymbol{d}_a \qquad \text{and} \qquad \boldsymbol{A}_b \boldsymbol{x} \leq \boldsymbol{d}_b \tag{10}$$

where the vector $\boldsymbol{d}_b$ now depends on ϵ as well. Notice that the strict inequality $<$ of (8) has now been replaced with $\leq$.

E. Closed Convex Subset

We have just shown that every element of the digitized set $\mathcal{D}$ satisfies (10). The set $\mathcal{D}$ is trivially "closed" because it is finite [18, p. 15]. However, $\mathcal{D}$ is evidently not convex because a linear combination $\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2$ of 8 bit images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is not an 8 bit image for arbitrary α in $0 \leq \alpha \leq 1$. Now consider a set $\mathcal{S}_1$ that is bigger than $\mathcal{D}$ by defining it to be the *set of all image vectors $\boldsymbol{x}$* for which (10) holds, or equivalently, (9) holds. By slight modification of the arguments at the end of Section IV-D we see that this set is both closed and convex. Since (9) holds, it is also clear that (7) holds which shows that $\boldsymbol{x}$ continue to belong in the inverse halftone set $\mathcal{C}$. Summarizing, we have three sets $\mathcal{D}$, $\mathcal{S}_1$, and $\mathcal{C}$ with the inclusion relationship

$$\mathcal{D} \subset \mathcal{S}_1 \subset \mathcal{C}.$$

$\mathcal{C}$ is the set of all image vectors which result in the given halftone image $\boldsymbol{h}$ using the given dot diffusion algorithm. The set $\mathcal{C}$ is convex but not closed. The digitized subset $\mathcal{D}$ is closed but not convex. The intermediate set $\mathcal{S}_1$ is closed and convex. Notice finally that the closed convex set $\mathcal{S}_1$ described by (10) can be described more compactly as

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$$

where $\boldsymbol{A} = \begin{bmatrix} -\boldsymbol{A}_a \\ \boldsymbol{A}_b \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} -\boldsymbol{d}_a \\ \boldsymbol{d}_b \end{bmatrix}$. Note that the above vector-inequality is interpreted componentwise.

These ideas will be useful in Section VI where we apply the iterative POCS technique to derive an approximation of the original contone image from a halftone. Assuming that an 8-bit image is a good approximation of the original contone, the distinction between the three sets is minor. However the fact that we can work with the closed convex set $\mathcal{S}_1$ without much loss of generality is significant as we shall see in Section VI. It allows us to assume that the method of POCS converges to a good approximation of the contone image.

V. Inverse Halftoning

Inverse halftoning is the reconstruction of a continuous tone image from its halftoned version. Since there can be more than one continuous tone image giving rise to a particular halftone image, there is no unique inverse halftone of a given halftoned

Fig. 25. Result of simple de-enhancement of dot diffused Lena image.

Fig. 26. Result of inverse halftoning using previous method [33].

image. Nevertheless, using the "mostly lowpass" characteristics of a natural image, good inverse halftones can be obtained. The basic aim in inverse halftoning is to separate the halftoning noise from the original image. In good halftoning algorithms, the noise introduced by halftoning is *blue*, i.e., it is concentrated in the high frequencies. Thus, simple low pass filtering can remove most of the halftoning noise, but it also removes the edge information.

Besides lowpass filtering, there are more sophisticated approaches for inverse halftoning. The method of *projection onto convex sets (POCS)* has been used by Analoui and Allebach [19] for halftone images produced by ordered dithering. For error diffused halftones, Hein and Zakhor [20] have successfully used the POCS approach. A different method called logical filtering has been used by Fan [21] for ordered dither images. Wong [22] has used an iterative filtering method for inverse halftoning of error diffused images. Finally the method of overcomplete wavelet expansions has been used in [23] to produce inverse halftones with good quality. During the process of preparing this paper, we came across another promising (perhaps the fastest) method for inverse halftoning which uses space varying filtering based on gradients obtained from the image [24].

In Section VI, we show how the POCS method can be applied for the inverse halftoning of a dot-diffused halftone. The mathematical characterization of dot-diffused images developed in Section IV will be especially useful to construct the so-called space-domain constraint set, which is a key ingredient in the development of the algorithm. Even though the wavelet method gives better results, inverse halftoning using POCS method is added because of its elegance and generality. In Section VII, we show how the wavelet method for inverse halftoning [23] can be modified for the case of dot-diffused images. While the POCS method often produces better PSNR, the wavelet method typically yields a more pleasing visual quality.

VI. Inverse Halftoning Using POCS

The method of POCS, which is an acronym for projection onto convex sets, is a powerful algorithm for the approximate recovery of a signal from partial information. It has been used very widely in many applications, as elaborated in authoritative references [25], [26]. To explain the idea in its simplest form, assume that the unknown signal is known *a priori* to belong to the intersection of two sets $\mathcal{S}_1$ and $\mathcal{S}_2$. Assume these are closed convex sets (see below). Then starting from an arbitrary initial guess for the signal and performing successive projections onto these two sets, we can converge to a point in the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$. Even though this intersection may have many elements and therefore the original signal not exactly recoverable, careful choice of $\mathcal{S}_1$ and $\mathcal{S}_2$ often leads to satisfactory results.

We will first state the POCS method and the associated convergence theorem more precisely. After this we describe how the method can be applied for recovering a continuous tone image from its halftoned version. We will see that the specific details of the convex set $\mathcal{S}_1$ depend on the details of the dot diffusion procedure and the class matrix (Section VI-D). Finally in Section VI-E, we will show experimental results.

A. Mathematical Background

The mathematical setting for the POCS method is the following. Let the unknown signal $\boldsymbol{f}$ be a vector in a Hilbert space H, e.g., ℓ_2 space of images. For example it could be a vector constructed from some arrangement of the pixels in a contone image. In view of the physical constraints that we happen to know, let us assume that $\boldsymbol{f}$ is in the intersections of known subsets $\mathcal{S}_1, \mathcal{S}_2, \cdots \mathcal{S}_m$ in H. (These may not be subspaces.) Assume that each of these is a *closed convex set*[6] and that their intersection $\mathcal{S}_{int}$ is nonempty. Let P_i be the *projection operator*

[6]A set $\mathcal{S}$ is said to be convex if $\alpha \boldsymbol{f} + (1-\alpha)\boldsymbol{g}$ belongs to $\mathcal{S}$ for any α such that $0 \leq \alpha \leq 1$, whenever $\boldsymbol{f}, \boldsymbol{g}$ are in $\mathcal{S}$. A set $\mathcal{S}$ equipped with a metric (i.e., any measure of distance) is said to be closed if the limit of any convergent subsequence $\{\boldsymbol{f}_i\}$ in $\mathcal{S}$ also belongs to $\mathcal{S}$.

Fig. 27. Inverse halftoned Lena using the modified wavelet denoising method.

Fig. 28. Inverse halftoned peppers using the modified wavelet denoising method.

from H to $\mathcal{S}_i$. The projection $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ onto $\mathcal{S}_i$ is defined to be the unique vector in $\mathcal{S}_i$ such that the error norm $||\boldsymbol{x} - \hat{\boldsymbol{x}}||$ is minimized [25, Sec. 2.2]. Define the composite operator

$$P \stackrel{\text{def}}{=} P_m P_{m-1} \cdots P_1$$

and consider the iteration $\boldsymbol{f_k} = P\boldsymbol{f}_{k-1}$, with initial vector $\boldsymbol{f}_0 \in \mathcal{H}$. Then, according to the *POCS theorem* [25, Th. 2.4-1], the vector $\boldsymbol{f}_k$ converges weakly[7] to some vector $\boldsymbol{f}_{\lim}$ in the intersection $\mathcal{S}_{int}$. This result is true regardless of the initial vector

[7]The term "weakly" means that the inner product $\langle \boldsymbol{f}, \boldsymbol{f}_k \rangle$ converges to $\langle \boldsymbol{f}, \boldsymbol{f}_{\lim} \rangle$ for any $\boldsymbol{f}$ in H. This is weaker than the requirement that $\boldsymbol{f}_k$ converges to $\boldsymbol{f}_{\lim}$, which would mean that $\|\boldsymbol{f}_k - \boldsymbol{f}_{\lim}\|$ goes to zero.

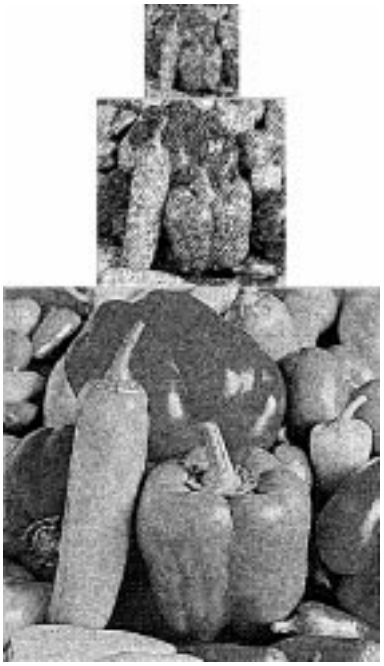

Fig. 29. Dot diffused peppers image without embedded multiresolution property.

$\boldsymbol{f}_0$, even though the limit $\boldsymbol{f}_{\lim}$ can depend on $\boldsymbol{f}_0$. If this limit is an acceptable approximation of $\boldsymbol{f}$, then we are happy.

The convergence result continues to be true even if the projection operators P_i are replaced with so-called *relaxed projections* T_i [25]. These are defined as $T_i = 1 + \lambda_i(P_i - 1)$ where $0 < \lambda_i < 2$. We shall not require this stronger version.

B. Application of the POCS Theorem for Inverse Halftoning

Consider applying the above result for the problem of inverse halftoning. The contone image $\boldsymbol{x}$ is halftoned with a known algorithm (e.g., dot diffusion with known class matrix), to yield a halftone $\boldsymbol{h}$. From this $\boldsymbol{h}$, and using our knowledge of the halftoning process, we have to construct a contone approximation $\boldsymbol{x}_{approx}$ subject to two conditions: 1) if it is halftoned, the result is again $\boldsymbol{h}$ and 2) $\boldsymbol{x}_{approx}$ should be an "acceptable" approximation of $\boldsymbol{x}$.

The first condition alone can be satisfied by many images. Let $\mathcal{S}_1$ be the set of all images such that the given halftoning algorithm yields the fixed halftone $\boldsymbol{h}$. The original contone image $\boldsymbol{x}$ evidently belongs to $\mathcal{S}_1$. Moreover, using the description of dot diffusion process in Section IV it can be shown that the halftone $\boldsymbol{h}$ itself belongs to $\mathcal{S}_1$. We say that $\mathcal{S}_1$ is the *space domain constraint set*. For the second condition we have to define a set $\mathcal{S}_2$ which represents the set of "acceptable images" in some sense. For example $\mathcal{S}_2$ could represent "natural images" which have certain smoothness properties. Since $\mathcal{S}_2$ is usually constructed with the help of lowpass operators (see below), it will be called the *frequency domain constraint set*. In the notation of Section IV the parent Hilbert space H is ℓ_2, and $\mathcal{S}_1$ and $\mathcal{S}_2$ are the two subsets. If these are closed and convex, then we can start from an arbitrary initial image $\boldsymbol{f}_0$ in ℓ_2 and perform the projections

$$\boldsymbol{g}_k = P_1 \boldsymbol{f}_{k-1} \qquad \text{(space-domain projection)}$$

$$\boldsymbol{f}_k = P_2 \boldsymbol{g}_k \qquad \text{(frequency-domain projection).}$$

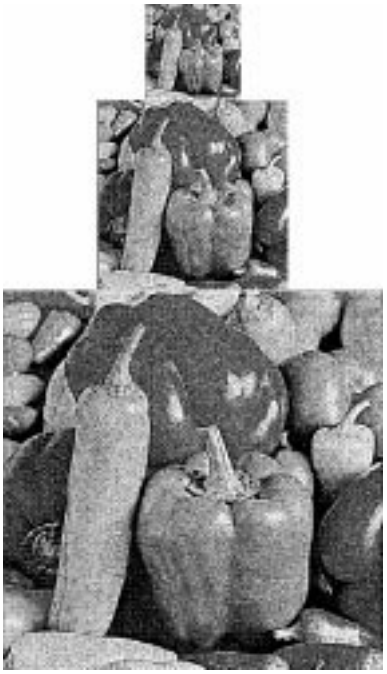

Fig. 30. Dot diffused peppers image with embedded multiresolution property.

That is, $\boldsymbol{f}_k = P_2 P_1 \boldsymbol{f}_{k-1}$. According to the POCS theorem this iteration converges to a member in the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$. If we are willing to accept any member in the intersection to be a valid approximation of the contone $\boldsymbol{x}$, then we are done.

The POCS algorithm has in the past been applied for inverse halftoning [20]. In the actual algorithm we have to identify the projection operators P_1 and P_2 which take an arbitrary image in ℓ_2 and project onto sets $\mathcal{S}_1$ and $\mathcal{S}_2$. For our application we already showed, using the mathematics of the dot diffusion algorithm (Section IV), that the set $\mathcal{S}_1$ is a closed convex set. The second point of novelty is with respect to the projection operator P_2. In the past, lowpass filtering has been used [20] as an approximation for P_2, the rationale being that, many natural images are lowpass. But unfortunately LTI filtering is not a projection operator, that is $H^2(e^{j\omega}) \neq H(e^{j\omega})$, unless $H(e^{j\omega})$ is an ideal filter with passband response of unity and stopband response of zero. In [20] the authors use partial reconstructions from DCT and SVD (singular value decomposition) as other possible choices for the projection operator. In this paper we use an operator which is not only an orthogonal projection but retains the properties of a good lowpass filter; this projection is constructed from an orthonormal multirate filter bank.

C. Implementation of the Frequency Domain Projection

Consider Fig. 16, which shows the synthesis section of an M-band uniform orthonormal filter bank [16] with the perfect reconstruction property $y(n) = x(n)$. Note that the subbands $x_0(n), \cdots, x_{L-1}(n)$ are obtained by feeding $x(n)$ to the analysis section (not shown) of the same filter bank. Suppose we delete the subband signals $x_L(n), \cdots, x_{M-1}(n)$ and perform the synthesis by retaining only the subbands $x_0(n), \cdots, x_{L-1}(n)$. Then the reconstruction $y(n) \neq x(n)$ in general, and $y(n)$ is called a partial reconstruction. If the filter bank has the orthonormal property [16], [27], [28], then we can give an orthogonal projection interpretation for this partial reconstruction. Thus, assume that the input $x(n)$ is in ℓ_2. Let $S \subset \ell_2$ be the subspace formed by the basis functions

$$\eta_{km}(n) = f_k(n - Mm), \qquad L \leq k \leq M-1, \; -\infty \leq m \leq \infty$$

from the deleted channels. Then the partial reconstruction $y(n)$ belongs to $S^{\perp}$, the orthogonal complement of S. Since orthogonal complements are closed subspaces [29], and subspaces are automatically convex, it follows that $y(n)$ is the projection of $x(n)$ onto the closed convex set $S^{\perp}$. We can take the frequency domain constraint set to be

$$\mathcal{S}_2 = S^{\perp}.$$

As explained in Section VI-B, in order to implement the POCS method we have to know how to project an arbitrary intermediate image onto the closed convex set $\mathcal{S}_2$. It is clear that this can be done by decomposing the image into subbands using an orthonormal filter bank, and partially reconstructing it as above.

Fig. 18 shows the actual 2-D filter bank used in our work for this frequency domain projection. Here $H_0(z)$ and $H_1(z)$ are one-dimensional filters, so the filter bank has separable 2–D analysis filters [16]. The notation $\downarrow (2, 1)$ means decimation by two in the horizontal direction and no decimation in the vertical direction. The notation $\uparrow (2, 1)$ similarly stands for the separable expander. With $H_0(z)$ and $H_1(z)$ denoting a lowpass/highpass pair, the signal $s_{00}(n_0, n_1)$ is the low–low subband. If $y(n_0, n_1)$ is reconstructed using this subband alone, then we can regard it as a "multirate" lowpass version, which at the same time is an orthogonal projection in the mathematical sense. In our work we actually used Daubechies' ten-tap FIR filter [30] for the lowpass filter $H_0(z)$. The highpass filter $H_1(z)$ was chosen in the usual way [16] to obtain the orthonormal filter bank.

D. Implementation of Space Domain Projection

The space domain constraint on the inverse halftone is that it should lie in the closed convex set $\mathcal{S}_1$ defined in Section IV. This is essentially the set of all contone images which can give rise to the given halftone $\boldsymbol{h}$. As explained in Section VI-B, in order to implement the POCS method we have to know how to project an arbitrary intermediate image vector $\boldsymbol{v} \in \ell_2$ onto the closed convex set $\mathcal{S}_1$. The meaning of a projection was reviewed in Section VI-A: the projection $\hat{\boldsymbol{v}}$ of $\boldsymbol{v}$ onto $\mathcal{S}_1$ is the unique vector in $\mathcal{S}_1$ such that the error norm $\|\boldsymbol{v} - \hat{\boldsymbol{v}}\|$ is minimized. Here the notation $\|\boldsymbol{e}\|$ represents the ℓ_2 norm $\sqrt{\boldsymbol{e}^{\dagger}\boldsymbol{e}}$. In order to implement this projection, we simply solve a minimization problem subject to the constraint $\hat{\boldsymbol{v}} \in \mathcal{S}_1$. Thus, the projection $\hat{\boldsymbol{v}}$ of the image $\boldsymbol{v}$ onto the convex set $\mathcal{S}_1$ is the solution to the following constrained optimization problem:

$$\min_{\hat{\boldsymbol{v}}} \|\boldsymbol{v} - \hat{\boldsymbol{v}}\|^2 \qquad \text{subject to} \qquad \boldsymbol{A}\hat{\boldsymbol{v}} \leq \boldsymbol{b}. \tag{11}$$

This follows because the elements $\hat{\boldsymbol{v}}$ of the set $\mathcal{S}_1$ are completely characterized by the property $\boldsymbol{A}\hat{\boldsymbol{v}} \leq \boldsymbol{b}$. This is a quadratic programming (QP) problem and can be solved using standard techniques. We used the Matlab optimization toolbox to solve for

Fig. 31. Floyd–Steinberg error diffusion.

$\hat{\boldsymbol{v}}$. In the interest of efficient programming, the QP problem was broken into several subproblems by partitioning the image into blocks. For this, overlapping blocks are used. In Fig. 17, the blocks used are shown. The black circles show the pixel positions that are changed after solving the QP subproblem, and white circles show the pixels used for boundary conditions of black circles, but they are not changed after solving this QP subproblem. Afterwards, the block location is moved 8 pixels to right or left, till the whole image is covered. The sizes of QP subproblems are 9×9. A further detail in the implementation is that the matrix $\boldsymbol{A}$ in the constraint equation must be modified to take into account the fact that the original image $\boldsymbol{x}$ is enhanced with a highpass filter before halftoning (as described in Section II). Recall that the matrix $\boldsymbol{A}$ originated from the matrix $\boldsymbol{A}_L$ described in Section IV where the key equation relating the original image $\boldsymbol{x}$ and the diffused image $\boldsymbol{y}$ was $\boldsymbol{y} = \boldsymbol{A}_L\boldsymbol{x} + (\boldsymbol{I} - \boldsymbol{A}_L)\boldsymbol{h}$. In this equation we have to replace $\boldsymbol{A}_L\boldsymbol{x}$ with $\boldsymbol{A}_L\boldsymbol{x}_e$ where $\boldsymbol{x}_e$ is the enhanced version of $\boldsymbol{x}$. The enhancing filter for $\alpha = 0.9$ (see Section II) is the 2-D filter

$$F_{enh}(z_0, z_1) = 10 - (z_0 + 1 + z_0^{-1})(z_1 + 1 + z_1^{-1})$$

and we can write $\boldsymbol{x}_e = \boldsymbol{E}\boldsymbol{x}$ where $\boldsymbol{E}$ is a square matrix (neglecting boundary details, such as lengthening of a signal due to filtering). The modified $\boldsymbol{A}$ matrix in the constraint $\boldsymbol{A}\hat{\boldsymbol{v}} \leq \boldsymbol{b}$ in (11) can now be worked out.

E. Implementation Details and Experimental Results

The frequency domain projection described above implicitly assumes that the original contone image is in the subset $\mathcal{S}_2$. Given an arbitrary image $x(n_0, n_1)$, we can replace it with its projection onto $\mathcal{S}_2$ before halftoning (i.e., compute the partial reconstruction $y(n_0, n_1)$ by using $s_{00}(n_0, n_1)$ alone, and then halftone $y(n_0, n_1)$). This *preconditioning* ensures that the desired inverse halftone is indeed in the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$. We found experimentally that for most natural images, the projection onto $\mathcal{S}_2$ is nearly as good as the original image, so such an initial conditioning is not a severe loss of information. Second, we found that in many examples the POCS algorithm converges to a good solution even without such preconditioning.

For the peppers and Lena images, Figs. 19–22 show the inverse halftoned images. In Figs. 19 and 20 the original image was first projected onto the transform domain set $\mathcal{S}_2$ before halftoning. In Figs. 21 and 22, this preconditioning was omitted. For completeness we mention the PSNR values for the reconstructed images. The PSNR values are as follows: peppers with preconditioning (PSNR = 30.35 dB with respect to original peppers and PSNR = 32.39 dB with respect to projection of peppers image onto $\mathcal{S}_2$), Lena with preconditioning (PSNR = 31.19 dB with respect to original Lena and PSNR = 33.08 dB with respect to projection of Lena image onto $\mathcal{S}_2$), peppers without preconditioning (PSNR = 29.44 dB), and Lena without preconditioning (PSNR = 30.66 dB). The images are obtained after five iterations.

Fig. 32. Direct binary search (DBS).

VII. Inverse Halftoning Using Wavelets

An inverse halftoning algorithm which use wavelets was considered in [23] and [31]. In this method, wavelets are used to differentiate the halftoning noise from the edge information. The edges are detected at different scales with specific overcomplete wavelet transform. Since the edges are correlated at different scales whereas the noise is not, the halftoning noise is suppressed by thresholding operations wherever the edges are not prominent (these correspond to steps 2 and 3 in our inverse halftoning method). However, the algorithm in [23] is tailored for error diffusion, which has different characteristics than dot diffusion. If the method in [23] is used for dot diffusion, the result is not good. This can be seen from Fig. 26 which shows the result of inverse halftoning the dot diffused Lena by using the method in [23]. The image suffers from periodic patterns, which represent low frequency noise. There are basically two reasons for the inferior performance: 1) the images are enhanced in dot diffusion before halftoning and 2) there is more low-frequency noise in dot diffusion.

In the new method, the specific properties of the dot diffusion algorithm are taken into account. The image is enhanced before dot diffusion, hence in the inverse halftoning, the dot diffused image should be de-enhanced using the inverse of the filter $F_{enh}(z_1, z_2) = 10 - (z_1 + 1 + z_1^{-1})(z_2 + 1 + z_2^{-1})$. Note that $F_{enh}(e^{jw_1}, e^{jw_2}) > 0$ for all $0 \leq w_1, w_2 \leq \pi$. We use the wavelet tree built from the analysis block shown in Fig. 23. An image $X(x, y)$ is decomposed into $L(x, y)$, $H(x, y)$, and $V(x, y)$ using the undecimated wavelet transform. At scale 2^{i+1}, (which will be described below), the filtering operations are as follows:

$$\begin{aligned} L(w_1, w_2) &= F(2^i w_1)F(2^i w_2)X(w_1, w_2), \\ H(w_1, w_2) &= G(2^i w_1)F(2^i w_2)X(w_1, w_2), \\ V(w_1, w_2) &= F(2^i w_1)G(2^i w_2)X(w_1, w_2) \end{aligned}$$

where G and F are derived from quadratic spline wavelets. These are tabulated along with the synthesis filters in [32, Table 1] (our F is H in that table). The choice of filters given in [32] detect edges at different scales if they are used in the wavelet tree shown in Fig. 24 with scales 2^0, 2^1, 2^2, 2^3 from left to right. For example $H_i(x, y)$ and $V_i(x, y)$ represent the horizontal edges, and vertical edges of $L_{i-1}(x, y)$ at scale 2^{i-1}, respectively, and $L_i(x, y)$ is the low pass version of $L_{i-1}(x, y)$.

The algorithm starts with a dot diffused image, $h(x, y)$. Then $h(x, y)$ is de-enhanced with the de-enhancement filter specified above. Let us call the resulting image $L_0(x, y)$. Afterwards, a four-level wavelet decomposition is applied to $L_0(x, y)$. Then for each pixel location (x, y), the following is done.

1) Apply a symmetric FIR Gaussian filter, $f_g(n, m)$ to $V_1(x, y)$, and $H_1(x, y)$. [$f_g(n, m) = ce^{-((n^2+m^2)/2\sigma^2)}$ for $-3 \leq n, m \leq 3$, and c is chosen such that the dc gain of the filter is unity.] The first level edge images contain mostly the halftoning noise, thus low pass filtering these

Fig. 33. The 16×16 dot diffusion without enhancement.

images reduces the blue noise without harming the edges significantly.

2) Let $E_{23}(x, y) = V_2(x, y)V_3(x, y)+H_2(x, y)H_3(x, y)$. If $E_{23}(x, y) \leq T_1$ then make $V_2(x, y) = 0$ and $H_2(x, y) = 0$.
3) Let $E_{34}(x, y) = V_3(x, y)V_4(x, y)+H_3(x, y)H_4(x, y)$. If $E_{34}(x, y) \leq T_2$ then make $V_3(x, y) = 0$ and $H_3(x, y) = 0$.

Steps 2 and 3 are the denoising steps in the algorithm where T_1 and T_2 are the thresholds determined experimentally. In order to discriminate the edges from the halftoning noise, we have to locate the edges. For this, the above steps perform a cross correlation between the edges at different scales. If there is a horizontal edge at scale i at (x, y) then $H_i(x, y)$ and $H_{i+1}(x, y)$ will be of the same sign [32]. The same is also true for vertical edges. Combining the horizontal and vertical edge correlations gives better results in detecting the diagonal edges.

4) The above steps have modified the subband signals L_i, H_i and V_i in certain ways. We now use the inverse filter bank (synthesis bank) corresponding to Fig. 24, and obtain a reconstructed version $\hat{L}_0(x, y)$. The image $\hat{L}_0(x, y)$ is the desired inverse halftone image.

In inverse halftoning, dot diffusion has an advantage, namely, even the simple de-enhanced image is a quite reasonable inverse halftone (PSNR = 26.62 dB for Lena image) (Fig. 25). The de-enhanced image is further processed as described above. The parameters used in the method are found experimentally. The variance of the Gaussian filter, σ^2 is chosen to be 0.5 and the thresholds are chosen to be $T_1 = 300$ and $T_2 = 20$. The results are shown in Fig. 27 (PSNR = 30.58 dB) and in Fig. 28 (PSNR = 30.07 dB).

Even though POCS gives higher PSNR values than the wavelet method, the wavelet method gives more pleasing results than the POCS. This is due to the space domain projection step in the POCS. Another advantage of the wavelet method is that, it is not iterative, whereas the POCS is inherently iterative. Thus the wavelet method is better than the POCS method for inverse halftoning. More recently a promising faster method has emerged for inverse halftoning of error diffused images [24]. We have not tried applying the algorithm for dot diffused images.

VIII. Embedded Multiresolution DOT Diffusion

Another desired property of images is the embedded multiresolution property. If an image has embedded multiresolution property, the lower resolution images can be obtained from higher resolution images. Embedded images require less storage space, and embedding is also useful for progressive transmission.

As observed by [33], normal halftones do not have embedded multiresolution property. This can be seen from Fig. 29, where the 512×512 image is halftoned by dot diffusion and the lower resolution images are obtained by downsampling the

higher resolution images by two in each direction. The lower resolution images are not good representations of the corresponding original images. But, the embedded multiresolution property can be imposed during the halftoning process. In [33]–[35] this property is imposed on the halftone image as follows. First, the lowest resolution image is obtained using the halftoning algorithm. The higher resolution halftones are obtained from the lower resolution halftones, by retaining the lower resolution image at the corresponding pixels, and halftoning the other pixels of the higher resolution. In [34] and [35], the halftoning method was an optimization-based one whereas in [33], the method was adaptive error diffusion. We will exploit the same idea, and we will show how to impose the embedded multiresolution property for dot diffused images.

Given a contone image $C_0(m, n)$ of size $M_0 \times N_0$, we want halftone images of smaller sizes $M_k \times N_k, k = 0, 1, 2, \cdots, K$, to have a fair representation of the original image. We assume that there exist integers p_k and q_k such that $M_{k-1} = p_k M_k$ and $N_{k-1} = q_k N_k$ for $k = 1, 2, \cdots, K$. Then the halftoning algorithm will be as follows.

1) Obtain $C_K(m, n) = C(mp_1p_2 \cdots p_K, nq_1q_2 \cdots q_K)$ for $m = 1, 2, \cdots, M_k$, $n = 1, 2, \cdots, N_k$. Then initialize i as $K - 1$.
2) Halftone $C_K(m, n)$, and let the resulting image be $h_K(m, n)$.
3) Define $C_i(m, n) = C(mp_0p_1 \cdots p_i, nq_0q_1 \cdots q_i)$ for $m = 1, 2, \cdots, M_i$, $n = 1, 2, \cdots, N_i$, where $p_0 = 1$, $q_0 = 1$.
4) For each pixel location (m, n) belonging to class i do
   a) If $(m, n) = (ap_{i+1}, bq_{i+1})$ for some a, b integer, then $h_i(m, n) = h_{i+1}(m/p_{i+1}, n/q_{i+1})$ else

$$h_i(m, n) = \begin{cases} 1 & \text{if } C_i(m, n) \geq 0.5 \\ 0 & \text{if } C_i(m, n) < 0.5 \end{cases}$$

   b) $e(m, n) = C_i(m, n) - h_i(m, n)$, and diffuse the error as in (2).
5) If $i = 0$, stop else let $i = i - 1$ and go to step 3).

The multiresolution property is demonstrated in Fig. 30 where $K = 2$, $p_1 = p_2 = 2$, and $q_1 = q_2 = 2$. The lower resolution halftones are contained in higher resolution halftones and halftones at any resolution have a "fair" representation of the original contone image. For the peppers image, the embedded multiresolution constraint did not affect the image quality at the highest resolution. However, there is a slight loss of quality in the highest resolution halftone image for the Lena image because of the latter constraint. Thus, at the expense of a little quality loss, the embedded multiresolution property can be imposed on the dot diffusion method.

IX. Concluding Remarks

Even though dot diffusion offers more parallelism than error diffusion, it has not received much attention in the past. This is partly because the noise characteristics of error diffusion method are generally regarded as superior. Although the quality of dot diffused halftones is not as good as error diffusion, we have shown that it can be substantially improved over standard dot diffusion. With more optimization work, it should be possible to come even closer to error diffusion quality. Furthermore, the parallelism offered by dot diffusion is a great advantage. The dot diffusion algorithm terminates in at most 64 steps for an 8×8 class matrix, compared to $M \times N$ steps needed for error diffusion algorithm for an $M \times N$ image. Moreover, as noticed in [10], the algorithm can in fact be terminated in about 50 steps. The conclusion is that Knuth's dot diffusion method with a carefully optimized class matrix is very promising; the image quality is comparable to error diffusion, and the implementation offers more parallelism than error diffusion. Since enhancement prior to halftoning can be objectionable in some cases, we also introduced and optimized 16×16 class matrix, which eliminated the need for enhancement. In this paper, we first optimized the class matrix. Then a mathematical description of dot diffusion was derived which was particularly useful in inverse halftoning. We also presented a wavelet-based inverse halftoning algorithm which works very well, even though the class matrix information is not used. Furthermore, we have shown that the dot diffusion algorithm can be easily modified to have the embedding property. This is useful for rendering at different resolution levels and for transmitting images, progressively.

Acknowledgment

The authors would like to thank Prof. J. Allebach, Dr. P. Wong, and Prof. B. Evans for their encouragement and many useful comments. Thanks are also due to Dr. D. L. Lau for his valuable comments on a conference version of the paper. Finally, they would like to thank Dr. Xiong for the use of his software on inverse halftoning.

References

[1] B. E. Bayer, "An optimum method for two level rendition of continuous tone pictures," in *Conf. Rec. IEEE ICC*, 1973, pp. 26-11–26-15.
[2] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," *Proc. SID*, pp. 75–77, 1976.
[3] D. Anastassiou, "Neural net based digital halftoning of images," in *Proc. ISCAS*, vol. 1, 1988, pp. 507–510.
[4] M. A. Seldowitz, J. P. Allebach, and D. E. Sweeney, "Synthesis of digital holograms by direct binary search," *Appl. Opt.*, vol. 26, pp. 2788–2798, 1987.
[5] R. A. Ulichney, "Dithering with blue noise," *Proc. IEEE*, vol. 76, pp. 56–79, Jan. 1988.
[6] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Green noise digital halftoning," *Proc. IEEE*, vol. 86, pp. 2424–2444, Dec. 1996.
[7] B. Evans, private communication.
[8] D. E. Knuth, "Digital halftones by dot diffusion," *ACM Trans. Graph.*, vol. 6, pp. 245–273, Oct. 1987.
[9] T. Mitsa and K. J. Parker, "Digital halftoning technique using a blue noise mask," *J. Opt. Soc. Amer. A*, vol. 9, pp. 1920–1929, Nov. 1992.
[10] M. Meşe and P. P. Vaidyanathan, "Image halftoning using optimized dot diffusion," in *Proc. EUSIPCO*, Rhodes, Greece, 1998.
[11] ——, "Image halftoning and inverse halftoning for optimized dot diffusion," in *Proc. ICIP*, Chicago, IL, 1998.
[12] ——, "A mathematical description of the dot diffusion algorithm in image halftoning, with application in inverse halftoning," in *Proc. ICASSP*, Phoenix, AZ, 1999.
[13] J. P. Allebach and R. N. Stradling, "Computer-aided design of dither signals for binary display of images," *Appl. Opt.*, vol. 18, pp. 2708–2713, Aug. 1979.
[14] R. Nasanen, "Visibility of halftone dot textures," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-14, pp. 920–924, Dec. 1984.
[15] J. P. Allebach, "FM screen design using DBS algorithm," in *Proc. ICIP*, vol. 1, Lausanne, Switzerland, 1996, pp. 549–552.

[16] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[17] J. Franklin, *Methods of Mathematical Economics*, New York: Springer-Verlag, 1980, pp. 44–45.
[18] R. B. Ash, *Real Variables with Basic Metric Space Topology*, New York: IEEE Press, 1993.
[19] M. Analoui and J. P. Allebach, "New results on reconstruction of continuous-tone from halftone," in *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 3, San Francisco, CA, 1992, pp. 313–316.
[20] S. Hein and A. Zakhor, "Halftone to continuous-tone conversion of error-diffusion coded images," *IEEE Trans. Image Processing*, vol. 4, pp. 208–216, Feb. 1995.
[21] Z. Fan, "Retrieval of images from digital halftones," *ISCAS*, pp. 313–316, May 1992.
[22] P. W. Wong, "Inverse halftoning and kernel estimation for error diffusion," *IEEE Trans. Image Processing*, vol. 4, pp. 486–498, Apr. 1995.
[23] Z. Xiong, K. Ramchandran, and M. Orchard, "Inverse halftoning using wavelets," in *Proc. Int. Conf. Image Processing*, vol. I, Lausanne, Switzerland, 1996, pp. 569–572.
[24] T. Kite, N. D. Venkata, B. Evans, and A. C. Bovik, "A high quality, fast inverse halftoning algorithm for error diffused halftones," in *Proc. ICIP*, Chicago, IL, 1998.
[25] H. Stark, *Image Recovery: Theory and Application*. Orlando, FL: Academic, 1987.
[26] D. C. Youla and H. Webb, "Image restoration by the method of convex projections—Part I: Theory," *IEEE Trans. Med. Imag.*, vol. MI-1, pp. 81–94, Oct. 1982.
[27] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[28] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley, MA: Wellesley-Cambridge, 1996.
[29] D. Luenberger, *Optimization by Vector Spaces*, New York: Wiley, 1969, p. 52.
[30] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
[31] J. Luo, R. de Queiroz, and Z. Fan, "A robust technique for image descreening based on the wavelet transform," *IEEE Trans. Signal Processing*, vol. 46, pp. 1179–1184, Apr. 1998.
[32] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, July 1992.
[33] P. W. Wong, "Adaptive error diffusion and its application in multiresolution rendering," *IEEE Trans. Image Processing*, vol. 5, July 1996.
[34] D. Anastassiou and S. Kollias, "Progressive halftoning of images," *Electron. Lett.*, vol. 24, pp. 489–490, 1988.
[35] S. Kollias and D. Anastassiou, "A progressive scheme for digital image halftoning, coding of halftones, and reconstruction," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 944–951, June 1992.
[36] http://www.systems.caltech.edu/mese/halftone/.

Murat Meşe was born in Bursa, Turkey, in 1974. He received the B.S. degree from Bilkent University, Ankara, Turkey, in 1996, and the M.S. degree from California Institute of Technology, Pasadena, in 1997, both in electrical engineering. He is currently pursuing the Ph.D. degree in the field of digital signal processing at California Institute of Technology.

His research interests are in digital halftoning, multirate systems, wavelets, and applications to communications.

P. P. Vaidyanathan (S'80–M'83–SM'88–F'91) was born in Calcutta, India, on October 16, 1954. He received the B.Sc. (Hons.) degree in physics and the B.Tech. and M.Tech. degrees in radiophysics and electronics, all from the University of Calcutta, India, in 1974, 1977, and 1979, respectively. He received the Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 1982.

He was a Postdoctoral Fellow at the University of California, Santa Barbara, from September 1982 to March 1983. In March 1983, he joined the Electrical Engineering Department, California Institute of Technology, Pasadena, as an Assistant Professor, and, since 1993, he has been a Professor of electrical engineering. His main research interests are in digital signal processing, multirate systems, wavelet transforms, and adaptive filtering. He is a Consulting Editor for the journal *Applied and Computational Harmonic Analysis*. He has authored a number of papers in IEEE journals, and is the author of the book *Multirate Systems and Filter Banks* (Englewood Cliffs, NJ: Prentice-Hall, 1993). He has written several chapters for various signal processing handbooks.

Dr. Vaidyanathan served as Vice-Chairman of the Technical Program Committee for the 1983 IEEE International Symposium on Circuits and Systems, and as the Technical Program Chairman for the 1992 IEEE International Symposium on Circuits and Systems. He was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1985 to 1987, and is currently an Associate Editor for IEEE SIGNAL PROCESSING LETTERS. He was a Guest Editor in 1998 for special issues of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, on the topics of filter banks, wavelets, and subband coders. He was a recipient of the Award for Excellence in Teaching at the California Institute of Technology for the years 1983–1984, 1992–1993, and 1993–1994. He also received the NSF's Presidential Young Investigator Award in 1986. In 1989, he received the IEEE ASSP Senior Award for his paper on multirate perfect-reconstruction filter banks. In 1990, he was the recipient of the S. K. Mitra Memorial Award from the Institute of Electronics and Telecommuncations Engineers, India, for his joint paper in the *IETE Journal*. He was the coauthor of a paper on linear-phase perfect reconstruction filter banks in the IEEE TRANSACTIONS ON SIGNAL PROCESSING, for which the first author, T. Nguyen, received the Young Outstanding Author Award in 1993. He received the 1995 F. E. Terman Award of the American Society for Engineering Education, sponsored by Hewlett Packard Co., for his contributions to engineering education, especially the book *Multirate Systems and Filter Banks*. He has given several plenary talks including at the Eusipco'98, Asimolar'88, and SPCOM'95 conferences on signal processing. He was chosen a Distinguished Lecturer for the IEEE Signal Processing Society for the year 1996–1997.