A MATHEMATICAL DESCRIPTION OF THE DOT DIFFUSION ALGORITHM IN IMAGE HALFTONING, WITH APPLICATION IN INVERSE HALFTONING

Murat Meşe and P.P. Vaidyanathan

Department of Electrical Engineering 136-93,
California Institute of Technology,Pasadena, CA 91125 USA,
Phone (626) 395-4681 Fax: (626) 795-8649
E-mail: mese@systems.caltech.edu, ppvnath@sys.caltech.edu

ABSTRACT

The dot diffusion method for digital halftoning has the advantage of parallelism unlike the error diffusion method. The method was recently improved by optimization of the so-called class matrix so that the resulting halftones are comparable to the error diffused halftones. In this paper, we will give a mathematical description of the dot diffusion method. This description is then applied in inverse halftoning.

1. INTRODUCTION

Digital halftoning is the rendition of continuous tone images on bilevel displays. There are many good methods for this, e.g., ordered dither and error diffusion [8], neural-net methods [1], dot-diffusion [3], and direct-binary search [6]. The dot-diffusion method proposed by Knuth [3] offers an attractive compromise between error diffusion (which is an entirely serial method with good image quality) and ordered dither (which offers parallelism but suffers from periodicity patterns). In [4] and [5] it was shown that images resulting from dot diffusion method can be significantly improved by optimizing the so-called class matrix to produce blue noise. The method is comparable to error diffusion in image quality but offers significant parallelism for implementation.

In this paper we first review the dot diffusion method in Sec. 2. We then present a mathematical description of the dot diffusion method (Sec. 3) which has hitherto not been analyzed formally. We present an error propagation model which also allows us to define the so-called inverse halftone set precisely. This set is very useful in developing inverse halftoning algorithms for dot diffusion. Its usefulness in the development of the well-known POCS algorithm for inverse halftoning is justified and demonstrated in Sec. 4.

2. REVIEW OF DOT DIFFUSION

The dot diffusion method for halftoning has only one design parameter, called the **class matrix C**. It determines the order in which the pixels are halftoned. Thus, the pixel positions (n_0, n_1) of an image are divided into $L = IJ$ classes according to (n_0 mod I, n_1 mod J) where I and J are constant integers. Table 1 shows an example of the class matrix for $I = J = 8$. This is the class matrix optimized to produce blue noise in the halftone image [4]. Let $x(n_0, n_1)$ be the 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 pixel locations (n_0, n_1) belonging to class k and define the halftone pixels to be

$$h(n_0, n_1) = \begin{cases} 1 & \text{if } x(n_0, n_1) \geq 0.5 \\ 0 & \text{if } x(n_0, n_1) < 0.5 \end{cases}$$

We also define the error $q(n_0, n_1) = x(n_0, n_1) - h(n_0, n_1)$. We then look at the eight neighbors of (n_0, n_1) 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

$$x(i, j) + 2q(n_0, n_1)/w \quad \text{(for orthogonal neighbors)} \quad (1(a))$$

$$x(i, j) + q(n_0, n_1)/w \quad \text{(for diagonal neighbors)} \quad (1(b))$$

where w is such that the sum of errors added to all the neighbors is exactly $q(n_0, n_1)$. 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 $x(n_0, n_1)$ which have the next class number $k+1$ are then similarly processed. The pixel values $x(n_0, n_1)$ are of course not the original contone values but the adjusted values according to earlier diffusion steps (1). When the algorithm terminates, the signal $h(n_0, n_1)$ is the desired halftone. Usually an image is enhanced [4] before dot diffusion is applied.

3. MATHEMATICAL DESCRIPTION OF DOT-DIFFUSION

Let L denote the number of classes. Let $\mathbf{x}_k$ denote a vector whose elements are the pixels of the original contone image belonging to class k in some order. Let $\mathbf{x}$ denote a vector

[1]This work was supported by Office of Naval Research Grant N00014-93-1-0231.

[2]http://www.systems.caltech.edu/mese/halftone/

whose elements are the pixels, in some order, of the contone image. For example, $\mathbf{x} = [\mathbf{x}_1{}^T\mathbf{x}_2{}^T \ldots \mathbf{x}_L{}^T]^T$. Each of the vectors $\mathbf{x}_k$ is a polyphase component [9] of the contone image.

3.1. 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 $\mathbf{y}_i$ of the original vectors $\mathbf{x}_i$, 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 $\mathbf{y}_1 = \mathbf{x}_1$. Let $\mathbf{h}_1$ denote the halftone vector obtained from quantizing this to two levels. The quantizer error $\mathbf{q}_1 = \mathbf{y}_1 - \mathbf{h}_1$ is then diffused to those neighbors of the pixels of $\mathbf{x}_1$, which have a higher class number. For example, $\mathbf{x}_2$ is replaced with $\mathbf{y}_2 = \mathbf{x}_2 + \mathbf{D}_{21}(\mathbf{y}_1 - \mathbf{h}_1)$ where $\mathbf{D}_{21}$ is a matrix representing the diffusion coefficients (i.e., quantities like $2/w$ and $1/w$ in equations (1(a)) and (1(b)). We then quantize $\mathbf{y}_2$ with the two level quantizer to produce the halftone $\mathbf{h}_2$ for all the pixels in class 2. The quantizer error $\mathbf{q}_2 = \mathbf{y}_2 - \mathbf{h}_2$ is then diffused to the higher class pixels. For example, consider class 3 pixels. In general these pixels receive diffused error from $\mathbf{q}_1$ and $\mathbf{q}_2$. Thus, in general, the class vector $\mathbf{x}_k$ is modified to

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{D}_{k1}(\mathbf{y}_1 - \mathbf{h}_1) + \mathbf{D}_{k2}(\mathbf{y}_2 - \mathbf{h}_2) + \ldots + \mathbf{D}_{k,k-1}(\mathbf{y}_{k-1} - \mathbf{h}_{k-1}) \quad (2)$$

and then quantized to obtain the halftone $\mathbf{h}_k$. Proceeding in this way, the halftone pixels $\mathbf{h}_k$ for all classes $1 \le k \le L$ are generated. The quantizer error vector $\mathbf{q}_k$ and halftone error vector $\mathbf{e}_k$ for class k are given by $\mathbf{q}_k = \mathbf{y}_k - \mathbf{h}_k$, $\mathbf{e}_k = \mathbf{x}_k - \mathbf{h}_k$. Subtracting $\mathbf{h}_k$ from both sides of (2) we get $\mathbf{q}_k = \mathbf{e}_k + \mathbf{D}_{k1}\mathbf{q}_1 + \mathbf{D}_{k2}\mathbf{q}_2 + \ldots + \mathbf{D}_{k,k-1}\mathbf{q}_{k-1}$, that is,

$$\begin{aligned} \mathbf{q}_1 &= \mathbf{e}_1 \\ \mathbf{q}_2 &= \mathbf{e}_2 + \mathbf{D}_{21}\mathbf{q}_1 \\ \mathbf{q}_3 &= \mathbf{e}_3 + \mathbf{D}_{31}\mathbf{q}_1 + \mathbf{D}_{32}\mathbf{q}_2 \end{aligned} \quad (3)$$

etc. By starting from the first equation, we can sequentially replace $\mathbf{q}_i$ in terms of $\mathbf{e}_i, \mathbf{e}_{i-1} \ldots$, on the right side of Eq. (3), resulting in an expression of the form

$$[\mathbf{q}_1^T \mathbf{q}_2^T \ldots \mathbf{q}_L^T]^T = \mathbf{A}_L [\mathbf{e}_1^T \mathbf{e}_2^T \ldots \mathbf{e}_L^T]^T$$

where $\mathbf{A}_L$ is a matrix depending on the elements of the smaller matrices $\mathbf{D}_{ij}$. We now show that $\mathbf{A}_L$ can be generated from $\mathbf{A}_{L-1}$ as follows: from Eq. (3)

$$\mathbf{q}_L = [\mathbf{D}_{L1} \quad \mathbf{D}_{L2} \quad \ldots \quad \mathbf{D}_{L,L-1}]\,[\mathbf{q}_1^T \mathbf{q}_2^T \ldots \mathbf{q}_{L-1}^T]^T + \mathbf{e}_L$$

which shows that

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

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

$$\mathbf{A}_3 = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{D}_{31} & \mathbf{D}_{32} & \mathbf{I} \end{bmatrix}}_{\mathbf{B}_3} \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{D}_{21} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{B}_2} \quad (4)$$

Since $\det \mathbf{A}_L = I$, A_L is invertible.

3.2. Expression for diffused image

The image $\mathbf{h}$ whose pixels come from $\mathbf{h}_k$ is the **halftone image**. The pixels from the original contone, diffused, and halftone images can be arranged in the form of vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{h}$, where $\mathbf{y}$ and $\mathbf{h}$ are defined similar to $\mathbf{x}$. The quantizer error vector $\mathbf{q}$ and halftone error vector $\mathbf{e}$ are defined as $\mathbf{q} = \mathbf{y} - \mathbf{h}$ and $\mathbf{e} = \mathbf{x} - \mathbf{h}$. We can now express the diffused image $\mathbf{y}$ in terms of the original contone image $\mathbf{x}$ and the halftone image $\mathbf{h}$ as follows: $\mathbf{y} = \mathbf{q} + \mathbf{h} = \mathbf{A}_L\mathbf{e} + \mathbf{h} = \mathbf{A}_L(\mathbf{x} - \mathbf{h}) + \mathbf{h}$, that is,

$$\mathbf{y} = \mathbf{A}_L\mathbf{x} + (\mathbf{I} - \mathbf{A}_L)\mathbf{h} \quad (5)$$

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 $\mathbf{y}$ and $\mathbf{h}$. Since y_i is directly quantized to yield h_i we see that

$$y_i \begin{cases} \ge 0.5 & \text{if } h_i = 1 \\ < 0.5 & \text{if } h_i = 0 \end{cases} \quad (6)$$

Given a halftone image $\mathbf{h}$ and the halftone algorithm, the **inverse halftone set** $\mathcal{C}$ is the collection of all image vectors $\mathbf{x}$ which yield the halftone image $\mathbf{h}$. That is, an image $\mathbf{x}$ belongs to $\mathcal{C}$ if and only if the vector $\mathbf{y}$ computed using (5) satisfies Eq. (6). Substituting $\mathbf{x} = \mathbf{h}$ in Eq. (5) , we see that $\mathbf{h}$ belongs to $\mathcal{C}$.

3.3. A closed convex subset of $\mathcal{C}$

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 Sec. 4. For convenience of discussion let us renumber the elements of the halftone $\mathbf{h}$ such that it can be partitioned as $\mathbf{h} = [\mathbf{1}^T\mathbf{0}^T]^T$. where $\mathbf{1} = [11 \ldots 1]^T$. The elements of $\mathbf{x}, \mathbf{y}$ and the matrix $\mathbf{A}_L$ are also renumbered accordingly. Then the diffused image vector $\mathbf{y}$ is

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

where $\mathbf{A}_a, \mathbf{A}_b$ and $\mathbf{c}$ do not depend on $\mathbf{x}$ or $\mathbf{y}$. Here $\mathbf{y}_a \ge 0.5 \times \mathbf{1}$ and $\mathbf{y}_b < 0.5 \times \mathbf{1}$ (vector inequalities are interpreted componentwise). That is, the inverse halftone set $\mathcal{C}$ is the set of all $\mathbf{x}$ satisfying $\mathbf{A}_a\mathbf{x} \ge \mathbf{d}_a$ and $\mathbf{A}_b\mathbf{x} < \mathbf{d}_b$. Given two image vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ satisfying this, we can readily verify that the linear combination $\alpha\mathbf{x}^{(1)} + (1 - \alpha)\mathbf{x}^{(2)}$ also satisfies the above equation whenever $0 \leq \alpha \leq 1$. This shows that the set $\mathcal{C}$ is convex. However, since $\mathcal{C}$ is the intersection of the closed set $\mathbf{A}_a\mathbf{x} \geq \mathbf{d}_a$ and the open set $\mathbf{A}_b\mathbf{x} < \mathbf{d}_b$, it is not closed.

The 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 $\mathbf{h}$ is certainly a member of $\mathcal{D}$. With $\mathbf{x}$ chosen from this digitized subset $\mathbf{D}$, the elements y_i of $\mathbf{y}$ 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 $(0.5 - \epsilon, 0.5)$. So Eq. (6) is replaced by

$$y_i \begin{cases} \geq 0.5 & \text{if } h_i = 1 \\ \leq 0.5 - \epsilon & \text{if } h_i = 0 \end{cases} \tag{7}$$

for some fixed $\epsilon > 0$ that can be precalculated from $\mathbf{A}$ and $\mathbf{h}$. We see that if $\mathbf{x}$ is in the digitized subset $\mathcal{D}$, then

$$\mathbf{A}_a\mathbf{x} \geq \mathbf{d}_a \qquad \text{and} \qquad \mathbf{A}_b\mathbf{x} \leq \mathbf{d}_b, \tag{8}$$

where the vector $\mathbf{d}_b$ now depends on ϵ as well.

The closed convex subset. Since $\mathcal{D}$ is a discrete finite set, it is closed but not convex. Now consider a set $\mathcal{S}_1$ that is bigger than $\mathcal{D}$ by defining it to be the *set of all image vectors* $\mathbf{x}$ for which Eq. (8) holds, or equivalently, Eq. (7) holds. We see that this set is both closed and convex. Since Eq. (7) holds, Eq. (6) holds which shows that $\mathbf{x} \in \mathcal{C}$. Summarizing, the three sets $\mathcal{D}, \mathcal{S}_1$ and $\mathcal{C}$ have the relationship $\mathcal{D} \subset \mathcal{S}_1 \subset \mathcal{C}$. 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, and can be described compactly as $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ where $\mathbf{A} = [\,-\mathbf{A}_a^T \mathbf{A}_b^T\,]^T$ and $\mathbf{b} = [\,-\mathbf{d}_a^T \mathbf{d}_b^T\,]^T$.

4. APPLICATION IN INVERSE HALFTONING

The method of POCS has been used very widely in many applications [7], [2]. Assume the unknown signal $\mathbf{f}$ be a vector in a Hilbert space H. Furthermore $\mathbf{f}$ is known apriori to belong to the intersection of two sets $\mathcal{S}_1$ and $\mathcal{S}_2$. Let P_i be the **projection operator** from H to $\mathcal{S}_i$. That is, for any $\mathbf{x}$ in H the vector $P_i\mathbf{x}$ is in $\mathcal{S}_i$, and moreover $P_i\mathbf{g} = \mathbf{g}$ for any $\mathbf{g}$ in $\mathcal{S}_i$. Define the composite operator $P \stackrel{\text{def}}{=} P_2P_1$ and consider the iteration $\mathbf{f}_k = P\mathbf{f}_{k-1}$, with initial vector $\mathbf{f}_0$. Then, according to the **POCS theorem** (Theorem 2.4-1 in [7]), the vector $\mathbf{f}_k$ converges weakly [1] to a point $\mathbf{f}_{lim}$ in $\mathcal{S}_1 \cap \mathcal{S}_2$.

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

The first set, $\mathcal{S}_1$ is the set of all images such that the given halftoning algorithm yields the fixed halftone $\mathbf{h}$. Evidently, the original contone image, $\mathbf{x}$, belongs to $\mathcal{S}_1$. From the results of Sec. 3, it can be shown that the halftone $\mathbf{h}$ 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 $\mathcal{S}_2$ to represent "natural images" which have certain smoothness properties. Since $\mathcal{S}_2$ is usually constructed with the help of lowpass operators, it will be called the **frequency domain constraint set.** If $\mathcal{S}_1$ and $\mathcal{S}_2$ are closed and convex, then we can start from an arbitrary initial image $\mathbf{f}_0 \neq \mathbf{0}$ in ℓ_2 and perform the projections

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

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

That is, $\mathbf{f}_k = P_2P_1\mathbf{f}_{k-1}$. 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 $\mathbf{x}$, we are done.

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 (Sec. 3), that the set $\mathcal{S}_1$ is a closed convex set. In the past, lowpass filtering has been used [2] 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 [2] 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 a multirate filter operator which is an orthogonal projection retaining the lowpass properties.

4.1. The Frequency Domain Projection

Fig. 1 shows the two dimensional 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 two-dimensional analysis filters [9]. 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. In our work we actually used Daubechies' 10-tap FIR filter for the lowpass filter $H_0(z)$. The highpass filter $H_1(z)$ was chosen in the usual way [9] to obtain the orthonormal filter bank. 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 onto a closed subspace (which is therefore a closed convex set).

4.2. Implementation of Space Domain Projection

The space domain constraint on the inverse halftone is that it should lie in the closed convex set $\mathcal{S}_1$. In order to implement the POCS method we have to know how to project an arbitrary intermediate image vector $\mathbf{v} \in \ell_2$ onto $\mathcal{S}_1$. Here we use an equivalent definition of projection [7, Sec. 2.2]. The projection $\widehat{\mathbf{v}}$ of $\mathbf{v}$ onto $\mathcal{S}_1$ is the unique vector in $\mathcal{S}_1$ such

[1]The term "weakly" means that the inner product $\langle \mathbf{f}, \mathbf{f}_k \rangle$ converges to $\langle \mathbf{f}, \mathbf{f}_{lim} \rangle$

that the ℓ_2 error norm $\|\mathbf{v} - \widehat{\mathbf{v}}\|$ is minimized. So, we simply solve a minimization problem subject to the constraint $\widehat{\mathbf{v}} \in \mathcal{S}_1$. That is we find $\widehat{\mathbf{v}}$ which solves

$$\min_{\widehat{\mathbf{v}}} \|\mathbf{v} - \widehat{\mathbf{v}}\|^2 \quad \text{subject to} \quad \mathbf{A}\widehat{\mathbf{v}} \leq \mathbf{b} \qquad (9)$$

This is a quadratic programming (QP) problem and can be solved using standard techniques from the Matlab toolbox. In the interest of efficient programming, the QP problem was broken into several subproblems by partitioning the image into blocks. For this, 9×9 overlapping blocks are used. A further detail in the implementation is that the matrix $\mathbf{A}$ in the constraint equation must be modified to take into account the fact that the original image $\mathbf{x}$ is enhanced with a highpass filter, $F_{enh}(z_0, z_1) = 10-(z_0+1+z_0^{-1})(z_1+1+z_1^{-1})$ before halftoning (as described in [4]).

5. CONCLUSION AND EXAMPLES

The frequency domain projection described above implicitly assume 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 more 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. Fig. 2 shows the inverse halftoned image (psnr=29.26dB with respect to original peppers and psnr=31.73dB with respect to projection of peppers image onto $\mathcal{S}_2$). As a final remark we note that in many examples, the POCS algorithm converges to a good solution even without the preconditioning.

6. REFERENCES

[1] Dimitris Anastassiou, "Neural net based digital halftoning of images," ISCAS, pp. 507-510, 1988.

[2] S. Hein and A. Zakhor, *"Halftone to Continuous-Tone Conversion of Error-Diffusion Coded Images,"* IEEE Trans. Image Proc., vol.4, pp 208-216, February 1995.

[3] D. E. Knuth, *"Digital Halftones by Dot Diffusion,"*, ACM Transactions on Graphics, vol. 6, no. 4, pp 245-273, October 1987.

[4] M. Meşe and P.P. Vaidyanathan, *"Image Halftoning Using Optimized Dot Diffusion,"* EUSIPCO, 1998.

[5] "—", *"Image Halftoning and Inverse Halftoning for Optimized Dot Diffusion,"* Proc. of ICIP, IL, 1998.

[6] 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.

[7] Henry Stark, edited, *"Image Recovery: Theory and Application,"* Academic Press, 1987.

[8] R. A. Ulichney, *"Dithering with Blue Noise,"* Proceedings of IEEE, vol. 76, no. 1, pp.56-79, January 1988.

[9] P. P. Vaidyanathan, *"Multirate Systems and Filter Banks,"* Englewood Cliffs, NJ 1993.

Table 1: Optimized Class Matrix **C** for Dot Diffusion [4]

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

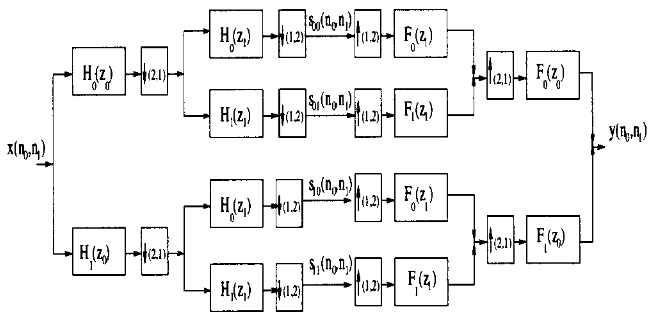

Figure 1: Two-dimensional separable PR Filter Bank

Figure 2: Inverse halftoned peppers (POCS Algorithm)