Fast 2-D and 3-D blind deconvolution of bandlimited images from even point-spread functions and partial data using bandwidth extrapolation and Fourier decoupling

*Andrew E. Yagle and Siddharth Shah
Dept. of EECS, The University of Michigan, Ann Arbor, MI 48109-2122

Abstract— **We solve the 2-D and 3-D blind deconvolution problems of reconstructing an unknown image and an unknown even (symmetric) spatially-varying point-spread function (PSF) from their 2-D or 3-D convolution. The image need not have any sort of support, so that (1) the image may be part of a bigger scene; and (2) the PSF may be different for different parts of the bigger scene. The image is assumed to be spatially bandlimited; this replaces the support constraint. The procedure is direct, in that the image is reconstructed directly from the data; deconvolution using the estimated PSF is unnecessary. This permits incorporation of image regularization into the reconstruction. Small and large numerical examples are provided.**

Keywords— **Blind Deconvolution. EDICS:2-REST**
Phone: 734-763-9810. Fax: 734-763-1503
Email: aey@eecs.umich.edu.

I. INTRODUCTION

A. Blind Deconvolution Overview

The problem of reconstructing a 2-D image from its 2-D convolution with an unknown blurring or point-spread function (PSF) arises in several disciplines [1], including image restoration from an unknown blurring agent, remote sensing through the atmosphere, and medical imaging. For a good review of the history and applications of this problem, see [1].

Most images are approximately bandlimited to the extent that they may be spatially sampled. This leads to the discrete version of this problem, in which a discrete image is to be reconstructed from its discrete convolution with an also-unknown discrete point-spread function (PSF). If the PSF is known, this becomes the solution of a linear system of equations, which is often ill-conditioned. When the PSF is unknown, the problem is even harder.

A common approach for blind deconvolution problems is to use an iterative transform algorithm [1],[3] which alternates between spatial and wavenumber domains. However, these algorithms often stagnate, failing to converge to a solution [3],[4]. Other approaches require the computationally expensive and extremely unstable numerical operation of tracking zero sheets of algebraic functions [4], or statistical estimation algorithms that also may not converge. Another iterative approach (NAS-RIF) [5] is guaranteed to converge, but it requires that an inverse filter for the PSF exist and have small spatial support. We will not attempt to list all approaches here.

The problem addressed in this paper should be distinguished from the problem of *multiple-blur* blind deconvolution, which has been addressed in many recent papers, e.g., [6]-[8]. In the latter problem, a single unknown signal or image is filtered with several unknown PSFs (which are assumed to be linearly independent), resulting in several known outputs. This is conceptually a much simpler problem than the *single-blur* blind deconvolution problem addressed in this paper. To see this quickly, note that an unknown 1-D signal with compact support can be recovered from its convolutions with two unknown 1-D PSFs with compact support by simply computing the greatest common divisor of the z-transforms of the two convolutions.

B. Partial Data Problem

In many applications, in particular remote sensing, the unknown image does not have compact support. Rather, it is just part of a bigger image of indefinite spatial extent. The blurred image that is the data is actually smaller than the image to be reconstructed. This is called the partial data problem [7].

The difficulty of the partial data problem can be seen by noting that *even if the PSF is known, the image cannot be uniquely determined*. This is evident since the deconvolution problem with known PSF becomes an underdetermined linear system of equations. It is clear that in the single blur partial data problem considered in this paper, some sort of image model is required.

On the other hand, the partial data problem essentially partitions the blind deconvolution into smaller sub-problems. Each sub-problem is independent, so PSFs for each sub-problem may be different.

Hence an algorithm solving the partial data problem also solves the problem with a *spatially-varying PSF* which is required not to vary only within a sub-problem. The sub-problems need only share a single image edge, in order to fix relative scale factors.

In many applications in optics, acoustics and electromagnetics, the PSF may be assumed to be an even function of its spatial variables, by reciprocity. To see this, let $u(x_o)$ be the electromagnetic or acoustic field strength at spatial position x_o. The response $u(y_o)$ to an excitation or source $s(x_o)$ is $u(y_o) = \int G(y_o, x_o)s(x_o)dx_o$, where $G(y_o, x_o)$ is the Green's function. If the Green's function is translation-invariant, then $G(y_o, x_o) = G(y_o - x_o)$. If reciprocity holds, then $G(y_o, x_o) = G(x_o, y_o)$ and $G(\cdot)$ is an even function.

C. Approach Used in This Paper

This paper solves the 2-D and 3-D partial data blind deconvolution problems by assuming that the (sampled) discrete image is bandlimited, i.e., that the original image was oversampled. This is certainly a reasonable assumption in most image processing problems. The bandlimit assumption amounts to a support constraint in the wavenumber domain; this replaces the support constraint in the spatial domain.

The approach used can be summarized as follows:

1. Partition the original problem into sub-problems, for each of which the PSF is even and spatially invariant, and the sub-image is bandlimited;
2. Extrapolate the missing edge data in each blurred sub-image. This can be done since each sub-image is assumed to be bandlimited, so that the blurred sub-image is also bandlimited, since the PSF is spatially invariant. This converts the partial data sub-problem into a full-data sub-problem;
3. Partition each full data sub-problem into 1-D full data problems, using the discrete Fourier transform;
4. Solve each 1-D problem directly for the sub-image using resultants. This makes the overall procedure a "direct" procedure; the PSF is never used. Image regularization techniques can be applied here;
5. Recombine the reconstructed sub-images, fixing the relative scale factors as described below.

The Fourier decoupling and solution of 1-D problems has been presented previously in [10] and [11] for the 2-D and 3-D problems, respectively. However, both of these assumed an image having compact spatial support. This paper shows how the assumption of compact spatial support can be replaced with an assumption of compact wavenumber support, which allows application of our previous results to the partial data problem. The technique presented here for estimation in noisy data is also new.

This paper is organized as follows. Section II formulates the problem and the image and PSF models. Section III presents the bandwidth extrapolation algorithm, and uses the discrete Fourier transform to decouple the 2-D or 3-D problem into 1-D problems. Section IV presents a small illustrative example. Section V presents and discusses numerical results. Section VI concludes with a summary and suggestions for future work.

II. PROBLEM FORMULATION

A. Problem Statement

The 2-D discrete blind deconvolution problem is as follows [1]. We observe

$$y(i_1, i_2) = h(i_1, i_2) * *u(i_1, i_2) + n(i_1, i_2) \qquad (1)$$

where $**$ denotes convolutions in variables i_1 and i_2. The 1-D convolution $*$ is defined here as

$$h(n) * u(n) = \sum_{i=0}^{n} h(n-i)u(i) = \sum_{i=0}^{n} h(i)u(n-i) \quad (2)$$

We assume $h(i_1, i_2) \neq 0$ only for $0 \leq i_1, i_2 \leq L-1$. We do **not** assume that the image $u(i_1, i_2)$ has compact support. The 2-D blind deconvolution problem is to reconstruct the image $u(i_1, i_2)$ (and presumably the PSF $h(i_1, i_2)$) from the known data $y(i_1, i_2)$, hence the term "blind deconvolution." No stochastic assumptions are made about either the image or the point-spread function. This precludes use of methods based on cumulants, ARMA or Poisson image models or stochastic equalization.

To solve the overall 2-D blind deconvolution problem, we partition it into sub-problems. *For each sub-problem*, we make the following assumptions:

1. $u(i_1, i_2) \neq 0$ only for $0 \leq i_1, i_2 \leq M-1$;
2. $h(i_1, i_2) \neq 0$ only for $0 \leq i_1, i_2 \leq L-1$;
3. $h(i_1, i_2) = h(L-1-i_1, L-1-i_2)$ (even PSF);
4. $y(i_1, i_2) \neq 0$ only for $L-1 \leq i_1, i_2 \leq M-1$;
5. $n(i_1, i_2)$ is a zero-mean 2-D white Gaussian noise random field;
6. All quantities are real functions.

For each sub-problem, the problem is to reconstruct the $M \times M$ portion of the image from the $(M-L+1) \times (M-L+1)$ portion of the data. Note that the blurred image $y(i_1, i_2)$ is **smaller** than the image $u(i_1, i_2)$. This means that $y(i_1, i_2)$ for $L-1 \leq i_1, i_2 \leq M-1$ does not depend on any part of the image outside the square $0 \leq i_1, i_2 \leq M-1$. Hence, without loss of generality, we can set the image equal to zero outside this, making the sub-problem easier to visualize.

B. Problem Ambiguities

There are three trivial ambiguities in the 2-D blind deconvolution problem:

1. *Scale factor:* If $\{h(i_1,i_2),u(i_1,i_2)\}$ is a solution, then $\{ch(i_1,i_2),\frac{1}{c}u(i_1,i_2)\}$ is also a solution for any real constant c. If c cannot be determined from the image energy, it usually is irrelevant. We consider the problem to be solved when the image is determined to a scale factor;
2. *Translation:* If $\{h(i_1,i_2),u(i_1,i_2)\}$ is a solution, then $\{h(i_1+d_1,i_2+d_2),u(i_1-d_1,i_2-d_2)\}$ is also a solution for any constants d_1,d_2. We eliminate this ambiguity by specifying the supports;
3. *Exchange:* We need to be able to distinguish $h(i_1,i_2)$ from $u(i_1,i_2)$. Since we need $M>L$ above, this is not a problem here.

We also assume that the 2-D z-transforms

$$H(x,y)=\sum_{i_1=0}^{L-1}\sum_{i_2=0}^{L-1}h(i_1,i_2)x^{-i_1}y^{-i_2}$$
$$U(x,y)=\sum_{i_1=0}^{M-1}\sum_{i_2=0}^{M-1}u(i_1,i_2)x^{-i_1}y^{-i_2} \tag{3}$$

are irreducible (they cannot be factored) for each subproblem. This is almost surely true [1].

C. Image and PSF Models

We assume that the sub-image $u(i_1,i_2)$ is bandlimited. This replaces the support constraint often imposed in the spatial domain. In particular, we assume that, for some integer K,

$$U(k_1,k_2)=0 \text{ for } \frac{M}{2}-K\le i_1,i_2\le\frac{M}{2}+K \tag{4}$$

where the $(M\times M)$ 2-D discrete Fourier transform (DFT) is defined as

$$U(k_1,k_2)=\sum_{i_1=0}^{M-1}\sum_{i_2=0}^{M-1}u(i_1,i_2)e^{-j2\pi\frac{i_1k_1+i_2k_2}{M^2}}. \tag{5}$$

Note that bandlimitation of an image does not imply bandlimitation of its sub-images. In practice the sub-images also turn out to be bandlimited, provided they are not too small.

The PSF is assumed to be an even function, as noted above, so that

$$h(i_1,i_2) \;=\; h(L-1-i_1,L-1-i_2). \tag{6}$$

In practice the PSF is often an even function, for reasons noted above. Its significance in this paper is that a 2-D or 3-D blind deconvolution problem with an even PSF can be decoupled into 1-D problems with even PSFs using the DFT. This decoupling greatly speeds up the algorithm.

However, we do *not* need to know the size L of the PSF. This is significant, since this information is often not available. All we need is

$$\begin{aligned}((\frac{M}{2}+K)-(\frac{M}{2}-K)+1)>\\ ((M+L-1)-(M-L+1)+1)\\ \to K>(L-1)\end{aligned} \tag{7}$$

so that the number of wavenumbers (k_1,k_2) at which $U(k_1,k_2)=0$ exceeds the number of unknown blurred image pixels $y(i_1,i_2)$ in the partial data problem. This suggests that the above quantities should be squared or cubed, but since the 2-D or 3-D problem is decoupled into 1-D problems, the above expression is pertinent for our algorithm.

III. BANDWIDTH EXTRAPOLATION AND FOURIER DECOUPLING

A. 1-D Extrapolation

Consider the problem of extrapolating a 1-D bandlimited signal:

1. $x(n)=0$ outside $0\le n\le M-1$;
2. $x(n)$ is known for $K\le n\le M-1-K$;
3. $x(n)$ is unknown for $0\le n\le K-1$;
4. $x(n)$ is unknown for $M-K\le n\le M-1$;
5. $X(k)=0$ for $\frac{M}{2}-K\le k\le\frac{M}{2}+K$

Here $X(k)$ is the M-point 1-D DFT of $x(n)$, defined as above. The goal is to compute the unknown values of $x(n)$ from the known ones, using the fact that $x(n)$ is bandlimited.

The solution satisfies the linear system of equations

$$\sum_{n=0}^{K-1}x(n)e^{-j2\pi\frac{nk}{M}}+\sum_{n=M-K}^{M-1}x(n)e^{-j2\pi\frac{nk}{M}}=$$
$$-\sum_{n=K}^{M-1-K}x(n)e^{-j2\pi\frac{nk}{M}} \text{ for } \frac{M}{2}-K\le k\le\frac{M}{2}+K \tag{8}$$

which is a linear system of $2K$ equations (if M is odd) in $2K$ unknowns ($2K+1$ equations if M is even, due to the extra equation at $k=M/2$). By taking periodic extensions, the unknown values of $x(n)$ can be viewed as being contiguous, so that the system matrix is Vandermonde and thus guaranteed to be nonsingular.

In fact, the extrapolation can be performed in closed form [12],[13]. Let $s(n)$ be the signal of length

$M-2K$ defined from the coefficients of

$$\begin{aligned} S(z) &= 1+\prod_{k=K+1-\frac{M}{2}}^{\frac{M}{2}-K-1}(z-e^{\frac{j2\pi k}{M}}) \\ S(z) &= \sum_{n=0}^{M-2K-1} s(n)z^n. \end{aligned} \quad (9)$$

Note that the M-point DFT $S(k)$ of $s(n)$ has the property that

$$\begin{aligned} S(k)=1 \quad &\text{for} \quad \left\{ \begin{array}{c} 0\le k\le \frac{M}{2}-K-1 \\ \frac{M}{2}+K+1\le k\le M-1 \end{array} \right\} \\ S(k)\ne 1 \quad &\text{for} \quad \frac{M}{2}-K\le k\le \frac{M}{2}+K \end{aligned} \quad (10)$$

The statement above that $x(n)$ is bandlimited can be rewritten as

$$\begin{aligned} X(k)S(k) &= X(k) \\ x(n)*s(n) &= x(n) \end{aligned} \quad (11)$$

which is an autoregression on $x(n)$. This can be used to compute the unknown values of $x(n)$ from the known values of $x(n)$.

It should be noted that extrapolation is in general a poorly-conditioned problem, in that the extrapolated $x(n)$ are sensitive to variations in the known $x(n)$. This is reflected in the large numbers that can arise in $s(n)$ [13]. This can be handled by regularization of the linear system above.

B. 2-D and 3-D Extrapolation

Now consider the 2-D version of the 1-D problem:

1. $x(i_1,i_2)=0$ outside $0\le i_1,i_2\le M-1$;
2. $x(i_1,i_2)$ is known for $K\le i_1,i_2\le M-1-K$;
3. $x(i_1,i_2)$ is unknown for $0\le i_1,i_2\le K-1$;
4. $x*i_1,i_2)$ is unknown for $M-K\le i_1,i_2\le M-1$;
5. $X(k_1,k_2)=0$ for $\frac{M}{2}-K\le k_1,k_2\le \frac{M}{2}+K$

Here $X(k_1,k_2)$ is the M-point 2-D DFT of $x(i_1,i_2)$, defined above. The goal is to compute the unknown values of $x(i_1,i_2)$ from the known ones, using the fact that $x(i_1,i_2)$ is bandlimited.

Of course, this can be solved by setting up a linear system as before. But note that a 2-D bandlimited image is bandlimited in each spatial index *separately*. So the 2-D extrapolation problem can be solved by:

1. Extrapolate each row of the image, regarding it as a 1-D extrapolation problem;
2. Then, extrapolate each column of the image, regarding it as a 1-D problem. Also, extrapolate the extra columns formed by the row extrapolations.

The 2-D extrapolation problem completely decouples into 1-D problems. This is very significant in saving computational time and storage.

The 3-D extrapolation problem also completely decouples into 2-D problems, which in turn decouple into 1-D problems. In this case the computational savings become enormous.

C. Fourier Decoupling

Here we follow [10]. Taking 2-D z-transforms (defined above) of the basic equations for 2-D blind deconvolution yields

$$\begin{aligned} Y(x,y) &= H(x,y)U(x,y) \\ (xy)^N Y(\frac{1}{x},\frac{1}{y}) &= (xy)^L H(\frac{1}{x},\frac{1}{y})(xy)^M U(\frac{1}{x},\frac{1}{y}) \\ H(x,y) &= (xy)^L H(\frac{1}{x},\frac{1}{y}) \end{aligned} \quad (12)$$

These can be combined into

$$\begin{aligned} Y(x,y)(xy)^M U(\frac{1}{x},\frac{1}{y}) &= \\ U(x,y)H(x,y)(xy)^M U(\frac{1}{x},\frac{1}{y}) &= \\ U(x,y)(xy)^L H(\frac{1}{x},\frac{1}{y})(xy)^M U(\frac{1}{x},\frac{1}{y}) &= \\ U(x,y)(xy)^N Y(\frac{1}{x},\frac{1}{y}) & \end{aligned} \quad (13)$$

Let $x_k=e^{j2\pi k/N}$ and y_k similarly. Setting $y=y_k$,

$$Y(x,y_k)(xy_k)^M U(\frac{1}{x},\frac{1}{y_k})=(xy_k)^N Y(\frac{1}{x},\frac{1}{y_k})U(x,y_k) \quad (14)$$

Since $y(i_1,i_2)$ and $u(i_1,i_2)$ are real, by conjugate symmetry we have

$$U(\frac{1}{x},\frac{1}{y_k})=U^*(\frac{1}{x^*},\frac{1}{y_k^*})=U^*(\frac{1}{x^*},y_k) \quad (15)$$

and similarly for $Y(\cdot)$, since $y_k y_k^*=1$. Hence

$$\begin{aligned} Y(x,y_k)(y_k x)^M U^*(\frac{1}{x^*},y_k) &= \\ (y_k x)^N Y^*(\frac{1}{x^*},y_k)U(x,y_k) & \end{aligned} \quad (16)$$

We recognize this as a decoupled (in k) set of 1-D complex-valued blind deconvolution problems. Each of these can be solved in parallel, as we show in the next section.

IV. Small Illustrative Example

A. 1-D Problem Solution

The equations above can be solved by equating coefficients of x^n, since two polynomials are equal if and only if their corresponding coefficients are equal. This results in a Sylvester linear system of equations, with the (complex) coefficients $\tilde{u}(i_1, k)$ of $U(x, y_k)$ as unknowns. This is repeated for each value of y_k. An inverse 1-D DFT gives $u(i_1, i_2)$.

Solution of the 1-D blind deconvolution problem amounts to finding the polynomial greatest common divisor of $Y(x, y_k)$ and its complex conjugate reversal $x^N Y^*(\frac{1}{x^*}, y_k)$. There is no ambiguity (besides scale factor), provided $Y(x, y_k)$ has no zeros in conjugate reciprocal pairs. In particular $Y(x, y_k)$ should have no zeros on the unit circle. Note that bandlimited signals seldom have zeros on (vs. near) the unit circle (other than at -1, and this can be avoided by using a DFT of odd order).

B. Illustrative Example: Problem Statement

The procedure is best illustrated with a simple and small example. Consider the 2-D blind deconvolution problem

$$\begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & p \\ q & r & s & t & u \\ v & w & x & y & z \end{bmatrix} * \begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix} = \begin{bmatrix} * & * & * & * & * & * \\ * & 106 & 331 & 179 & 242 & * \\ * & 403 & 579 & 298 & 171 & * \\ * & 219 & 298 & 563 & 379 & * \\ * & 258 & 259 & 419 & 138 & * \\ * & * & * & * & * & * \end{bmatrix} \quad (17)$$

where $*$ denotes an unknown observation. This is a partial data problem; the 5×5 "image" could be part of a larger image, since no edge information is used in the given observations of the blurred image. Note that the 2×2 PSF is an even function, as required by our algorithm.

Although we have no compact support constraint for the image, we do know that the (6×6)-point 2-D DFT of the (zero-padded) image has form

$$U(k_1, k_2) = \begin{bmatrix} * & * & 0 & * & 0 & * \\ * & * & 0 & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & 0 & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & 0 & * & 0 & * \end{bmatrix} \quad (18)$$

This means that the 5×5 image is known to have its 2-D discrete space Fourier transform components at $\omega_x = \pi/3$ and $\omega_x = \pi/3$ equal to zero. That is,

$$\begin{aligned} U(2, k_2) = U(k_2, 2) &= 0 \\ U(4, k_2) = U(k_2, 4) &= 0 \end{aligned} \quad (19)$$

where $U(k_1, k_2)$ is the (6×6)-point 2-D DFT of the zero-padded image. Note that the highest wavenumber component is not assumed to be zero.

C. Illustrative Example: Extrapolation

The second column of the blurred image is extrapolated as follows. Since $Y(2, k_2) = Y(4, k_2) = 0$,

$$\begin{aligned} &y(0,2) + y(5,2)e^{-j2\pi\frac{5\cdot 2}{6}} = \\ &-106e^{\frac{-j2\pi 1\cdot 2}{6}} - 403e^{\frac{-j2\pi 2\cdot 2}{6}} \\ &-219e^{\frac{-j2\pi 3\cdot 2}{6}} - 258e^{\frac{-j2\pi 4\cdot 2}{6}} \end{aligned} \quad (20)$$

Taking real and imaginary parts yields

$$\begin{aligned} y(0,2) - y(5,2)\frac{1}{2} &= \frac{1}{2}(106 + 403 + 258) - 219 \\ y(5,2)\frac{\sqrt{3}}{2} &= \frac{\sqrt{3}}{2}(106 - 403 + 258) \end{aligned} \quad (21)$$

which yields

$$y(0,2) = 145; \quad y(5,2) = -39. \quad (22)$$

Extrapolating the other columns, and then rows, of the blurred image yields the full blurred image

$$y(i_1, i_2) = \begin{bmatrix} 111 & 145 & 292 & 35 & 1 & -146 \\ 169 & 106 & 331 & 179 & 242 & 17 \\ 276 & 403 & 579 & 298 & 171 & -5 \\ 35 & 219 & 298 & 563 & 379 & 300 \\ -23 & 258 & 259 & 419 & 138 & 137 \\ -130 & -39 & 11 & 300 & 209 & 159 \end{bmatrix} \quad (23)$$

D. Illustrative Example: Fourier Decoupling

Setting $k_2 = 1$ produces the following 1-D blind deconvolution problem:

$$\tilde{y}(n,1) = \tilde{u}(n,1) * \tilde{h}(n,1)$$

$$\text{where } \tilde{y}(n,1) = \frac{1}{6}\sum_{k=0}^{5} Y(k,1)e^{j2\pi nk/6} \text{ and}$$

$$\begin{aligned} \left[\tilde{h}(0,1), \tilde{h}(1,1)\right] &= e^{-j2\pi/6}\left[\tilde{h}(1,1)^*, \tilde{h}(0,1)^*\right] \\ \tilde{y}(0,1) = y_0 &= 071 + j504.03 \\ \tilde{y}(1,1) = y_1 &= 235 + j154.15 \\ \tilde{y}(2,1) = y_2 &= 198 + j706.68 \\ \tilde{y}(3,1) = y_3 &= 607 - j140.30 \\ \tilde{y}(4,1) = y_4 &= 443 + j209.58 \\ \tilde{y}(5,1) = y_5 &= 480 - j342.95 \end{aligned} \quad (24)$$

E. Illustrative Example: 1-D Problem Solution

Next, we would normally solve the equation

$$\begin{aligned}\tilde{y}(n,1) * \tilde{u}^*(5-n,1) &= \\ e^{-j2\pi/6}\tilde{y}^*(5-n,1) * \tilde{u}(n,1) \end{aligned} \tag{25}$$

by computing the null vector of the (10×10) Sylvester matrix

$$\begin{bmatrix} y_0 & 0 & 0 & 0 & 0 & y_5^* & 0 & 0 & 0 & 0 \\ y_1 & y_0 & 0 & 0 & 0 & y_4^* & y_5^* & 0 & 0 & 0 \\ y_2 & y_1 & y_0 & 0 & 0 & y_3^* & y_4^* & y_5^* & 0 & 0 \\ y_3 & y_2 & y_1 & y_0 & 0 & y_2^* & y_3^* & y_4^* & y_5^* & 0 \\ y_4 & y_3 & y_2 & y_1 & y_0 & y_1^* & y_2^* & y_3^* & y_4^* & y_5^* \\ y_5 & y_4 & y_3 & y_2 & y_1 & y_0^* & y_1^* & y_2^* & y_3^* & y_4^* \\ 0 & y_5 & y_4 & y_3 & y_2 & 0 & y_0^* & y_1^* & y_2^* & y_3^* \\ 0 & 0 & y_5 & y_4 & y_3 & 0 & 0 & y_0^* & y_1^* & y_2^* \\ 0 & 0 & 0 & y_5 & y_4 & 0 & 0 & 0 & y_0^* & y_1^* \\ 0 & 0 & 0 & 0 & y_5 & 0 & 0 & 0 & 0 & y_0^* \end{bmatrix} \tag{26}$$

The lower half of the null vector is $\tilde{u}(n,1)$, to a scale factor.

However, for this particular problem there is an ambiguity. Since the signal $\tilde{u}(n,1)$ does have two zeros on the unit circle at $e^{\pm j2\pi/3}$, there is an ambiguity in the 1-D blind deconvolution problem. In fact, the rank of the above (10×10) matrix is seven, as it should be here.

One way around this would be to note that since signal $\tilde{u}(n,1)$ is known to have zeros at $e^{\pm j2\pi/3}$, $\tilde{y}(n,1)$ also has these zeros, and they can be deconvolved from $\tilde{u}(n,1)$. However, this is undesirable for numerical reasons (deconvolution is often ill-conditioned).

A better way is to include the zeros $e^{\pm j2\pi/3}$ in $\tilde{h}(n,1)$. Thus, compute the null vector of the (8×6) Sylvester matrix

$$\begin{bmatrix} y_0 & 0 & 0 & y_5^* & 0 & 0 \\ y_1 & y_0 & 0 & y_4^* & y_5^* & 0 \\ y_2 & y_1 & y_0 & y_3^* & y_4^* & y_5^* \\ y_3 & y_2 & y_1 & y_2^* & y_3^* & y_4^* \\ y_4 & y_3 & y_2 & y_1^* & y_2^* & y_3^* \\ y_5 & y_4 & y_3 & y_0^* & y_1^* & y_2^* \\ 0 & y_5 & y_4 & 0 & y_0^* & y_1^* \\ 0 & 0 & y_5 & 0 & 0 & y_0^* \end{bmatrix} \tag{27}$$

and then convolve the lower half of the result with $[1,1,1]$ to restore the zeros at $e^{\pm j2\pi/3}$. The null vector of the above matrix is

$$[.5, .1 - j.06, .4 + j.2, .13 - j.44, .11 - j.04, .34 - j.40]'$$

Convolving the lower half of this with $[1,1,1]$ gives

$$[.13 - j.44, .24 - j.48, .58 - j.87, .45 - j.44, .34 - j.40]'$$

which is $\tilde{u}(n,1)$ to a scale factor of $\frac{1+2.2j}{1000}$.

F. Illustrative Example: Conclusion

$\tilde{u}(n,k)$ for $k = 0,3,5$ can be found similarly; note that $\tilde{u}(n,2) = \tilde{u}(n,4) = 0$. The scale factor ambiguity of each 1-D problem can be addressed by repeating the entire procedure with horizontal and vertical axes exchanged (see [10],[11] for details). An inverse 6-point 1-D DFT of $\tilde{u}(i_1,k)$ taking k to i_2 then yields $u(i_1,i_2)$:

$$\begin{bmatrix} 37 & 73 & 146 & 109 & 73 \\ 81 & 101 & 202 & 121 & 101 \\ 146 & 218 & 372 & 226 & 154 \\ 109 & 145 & 226 & 117 & 81 \\ 65 & 117 & 170 & 105 & 53 \end{bmatrix} \tag{28}$$

to a scale factor. Although it isn't used, the PSF is

$$h(i_1,i_2) = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix} \tag{29}$$

again to a scale factor.

V. Acknowledgments

The work of the authors was supported by DARPA and ARO under grant #DAAD-19-02-1-0262.

References

[1] D. Kundur and D. Hatzinakos, "Blind image deconvolution," IEEE Sig. Proc. Magazine, vol. 13, no. 3, 43-64 (1996).

[2] S. Shah and A.E. Yagle, "2-D blind deconvolution of compact support images using Bezout's lemma and a spline-based image model," submitted to IEEE Trans. on Image Proc.

[3] G.R. Ayers and J.C. Dainty, "Iterative blind deconvolution method and its applications," Optics Letters, vol. 13, no. 7, 547-549, 1988.

[4] C.C. Wackerman and A.E. Yagle, "Use of Fourier domain real-plane zeros to overcome a phase retrival stagnation," J. Opt. Soc. Am. A, vol. 8, no. 12, 1898-1904, 1991.

[5] D. Kundur and D. Hatzinakos, "A novel blind deconvolution scheme for image restoration using recursive filtering," IEEE Trans. on Sig. Proc., vol. 46, no. 2, 375-390, 1998.

[6] H.T. Pai and A.C. Bovik, "On eigenstructure-based direct multichannel blind image restoration," IEEE Trans. Image Proc., vol. 10, no. 10, 1434-1446, 2001.

[7] G. Harikumar and Y. Bresler, "Perfect blind restoration of images blurred by multiple filters: Theory and efficient algorithms," IEEE Trans. Image Proc., vol. 8, no. 2, 202-219, 1999.

[8] G.B. Giannakis and R.W. Heath Jr., "Blind identification of multichannel FIR blurs and perfect image restoration," IEEE Trans. Image Proc., vol. 9, no. 11, 1877-1896, 2000.