Non-Iterative Valid Blind Deconvolution of Sparsifiable Images using an Inverse Filter

Andrew E. Yagle
Department of EECS, The University of Michigan, Ann Arbor, MI 48109-2122

Abstract— **We propose a new non-iterative algorithm for the blind deconvolution problem of reconstructing both a sparsifiable image and a point-spread function (PSF) from their valid 2D convolution. No support constraint is needed for either the image or the PSF, nor is non-negativity of the image. The only requirements are that the PSF be modelled as having a finite-support inverse or equalizing filter (IF), and that the product of the (known) numbers of nonzero pixels in this inverse filter and the sparsified image be less than the total number of image pixels. The algorithm requires only the solution of a large linear system of equations and a small rank-one matrix decomposition.**

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

I. INTRODUCTION

A. Background

Many imaging problems in optics, medicine and astronomy require computation of an image from its 2D convolution with a 2D impulse response or point-spread function (PSF). The PSF can sometimes be determined by imaging a bead or an isolated star; then the image can be obtained by deconvolution. In blind deconvolution, both the image and the PSF are unknown, which at first glance seems to be one equation in two unknowns, without unique solution.

If both the image and PSF have finite or compact support (they are known to be zero outside a finite region), then blind deconvolution theoretically has a unique solution in dimensions higher than one, subject to trivial ambiguities noted below. Iterative algorithms such as Ayers-Dainty and Lucy-Richardson have been developed for this case. But these algorithms converge to the solution only if they are initialized close to it. The Ayers-Dainty algorithm also requires two deconvolutions in each iteration; these are often poorly-conditioned problems, which can lead to error propagation as it progresses.

Another approach is to model the PSF as having a finite-support inverse or equalizing filter (IF). This approach has been used successfully, along with image finite support and nonnegativity constraints, in the NAS-RIF algorithm. This avoids deconvolution and it is globally convergent to the solution, since the cost functional is convex. But it does not yield a unique solution without an image support constraint.

ARMA modelling models the image as having a 2D AR model and the PSF as being finite, or vice-versa. Again, this requires a known support constraint.

B. Contribution of This Paper

This paper replaces the image support constraint with a sparsifiability constraint: the image has a sparse (mostly zero) representation in some known basis, such as a wavelet basis. The locations of nonzero basis coefficients are unknown. This model has been used successfully in compressed sensing, since many real-world images do in fact have sparse representations. It also uses a finite inverse or equalizing filter (IF) model for the unknown PSF.

The only other requirement is that the product of the number of nonzero pixels in the sparsified image and in the inverse filter is less than the total number of image pixels. The algorithm is non-iterative: it requires only the solution of a single large linear system of equations and a small rank-one matrix decomposition. While iterative algorithms can be used for each of these operations, non-iterative algorithms (such as Gaussian elimination) are still available.

II. Problem Formuation

A. Problem Statement

The goal of this paper is to solve the valid blind deconvolution problem, formuated as follows:
Given: The known *valid* 2D convolution

$$y_{i,j} = h_{i,j} * x_{i,j} = \sum_{m=0}^{\tilde{L}-1} \sum_{n=0}^{\tilde{L}-1} h_{m,n} x_{i-m,j-n} \qquad (1)$$

- Unknown image $x_{i,j}$ is $\tilde{M} \times \tilde{M}$;
- Unknown PSF $h_{i,j}$ is $\tilde{L} \times \tilde{L}$;
- Known "'valid" $y_{i,j}$ is N×N;
- No support constraint: N=$\tilde{M}$–$\tilde{L}$+1.

Model: The image and PSF are modelled as:

$$x_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} \Phi_{i,j,m,n} z_{m,n} \qquad (2)$$

$$\delta_{i,j} = \sum_{m=0}^{L-1}\sum_{n=0}^{L-1} g_{m,n} h_{i-m,j-n} \quad (3)$$

The sizes of various images are as follows:

- Unknown $x_{i,j}$ is now on an M×M subset of $\tilde{M}\times\tilde{M}$;
- Unknown IF $g_{i,j}$ is L×L; likely L<< $\tilde{L}$;
- Unknown sparse image $z_{i,j}$ is M×M;
- M=N–L+1=$\tilde{M}$–$\tilde{L}$–L+2.

The first valid convolution $y_{i,j}$=$h_{i,j} * x_{i,j}$ means $y_{i,j}$ is known on an N×N subset of the larger $\tilde{M}\times\tilde{M}$ region on which $x_{i,j}$ is defined. The second valid convolution $x_{i,j}$=$g_{i,j} * y_{i,j}$ means $x_{i,j}$ is computed on an M×M subset of the larger N×N region on which $y_{i,j}$ is observed. On the sparsification of $x_{i,j}$ to $z_{i,j}$:

- Known $\Phi_{i,j,m,n}$ is an M×M matrix of basis vectors;
- $\Phi_{i,j,m,n}$ is orthonormal if the basis is orthonormal;
- We assume: $z_{i,j}$ has only K–1 nonzero values;
- We require: $M^2 > KL^2$.

Goal: Compute $g_{i,j}, z_{i,j}$, hence $h_{i,j}, x_{i,j}$, from $y_{i,j}$. In fact, we have the following:

- Computing $g_{i,j}$ suffices, since $x_{i,j}$=$y_{i,j} * g_{i,j}$;
- We deal with only an M×M sparsifiable subset of the larger $\tilde{M}\times\tilde{M}$ image;
- Different parts can be blurred by different $h_{i,j}$.

B. Trivial Ambiguities

The blind deconvolution problem in all dimensions is known to have the following 3 trivial ambiguities:

- **Translation:** If $\{h_{i,j}, x_{i,j}\}$ is a solution, then for any a,b $\{h_{i+a,j+b}, x_{i-a,j-b}\}$ is also a solution;
- **Scale Factor:** If $\{h_{i,j}, x_{i,j}\}$ is a solution, then for any constant c $\{\frac{1}{c}h_{i,j}, cx_{i,j}\}$ is also a solution;
- **Exchange:** There is no way to tell which of $\{h_{i,j}, x_{i,j}\}$ is the image and which is the PSF.

The problem is considered solved when $\{h_{i,j}, x_{i,j}\}$ is determined to within these three ambiguities.

III. Problem Solution

A. Sparsity Equation

We consider only the 1D case for clarity. The 2D or 3D problem can be uwrapped to the 1D problem using the Agarwal-Cooley fast convolution algorithm. This regards the 2D or 3D problem indices as a residue number system representation of the 1D problem indices. The problem size M×M must be adjusted to $M_1 \times M_2$, with M_1, M_2 relatively prime.

Let z_n=0 except for n∈ $\{n_{i_1} \dots n_{i_{K-1}}\}$. So n_{i_j} are the locations of nonzero values of sparse signal z_n.

Let s_n be the indicator function for nonzero z_n:

$$\begin{cases} s_n = 0 & \text{if } z_n \neq 0 \\ s_n \neq 0 & \text{if } z_n = 0 \end{cases} \quad \begin{cases} S_k \neq 0 & 0 \leq k \leq K \\ S_k = 0 & \text{otherwise} \end{cases} \quad (4)$$

where S_k is the M-point DFT of s_n. Then we have

$$s_n z_n = 0 \rightarrow \sum_{k=0}^{M-1} Z_k S_{m-k} = 0, m = 0 \dots M-1 \quad (5)$$

where Z_k is the M-point DFT of z_n.

The polynomial of degree K–1 with coefficients S_k has zeros $\{e^{jn_i}\}$, where n_i are the locations of the nonzero values of z_n. So the inverse DFT of S_k is zero at locations of nonzero z_n, indicating them.

B. Derivation of Linear System

Applying the inverse filter to data $y_{i,j}$ gives:

$$z_{i,j} = \sum_{m=0}^{M-1}\sum_{n=0}^{M-1} \Phi^{-1}_{i,j,m,n}(g_{m,n} * y_{m,n}) \quad (6)$$

Inserting this into the sparsity equation

$$0 = \sum_{k=0}^{M-1} S_{m-k}\mathrm{DFT}\{z_{i,j}\} \quad (7)$$

gives a linear system of M^2 equations in the KL^2 unknowns consisting of all possible products $\{S_k g_{i,j}\}$. If $M^2 > KL^2$ this system has a unique solution.

Then the solution $\{S_k g_{i,j}\}$ is arranged into a rank one $K \times L^2$ matrix. A rank one decomposition yields S_k and $g_{i,j}$ to scale factors, the scale factor ambiguity. Distinguishing $S_k, g_{i,j}$ is the exchange ambiguity.

IV. Numerical Example

We present a small numerical example to illustrate the algorithm. The problem is kept small to enable the reader to follow construction of the matrices used in the algorithm Matlab code given below.

A. Problem Specification

Image: The image $x_{i,j}$ is 53×54. It consists of an 11×12 block letter "E" surrounded by random pixels. The block letter is the only part of the image that can be sparsified. $x_{i,j}$ is shown in Figure #1.

PSF: The PSF $h_{i,j}$ is the 41×41 truncation of $h_{i,j}$=$(\frac{1}{2})^{(|i|+|j|)}$. This PSF has four-fold symmetry. The PSF, unknown to the algorithm, is shown in Figure #2. Many real-world PSFs look like this.

The inverse filter $g_{i,j}$ is computed to be

$$g_{i,j} = \frac{1}{9}\begin{bmatrix} 4 & -10 & 4 \\ -10 & 25 & -10 \\ 4 & -10 & 4 \end{bmatrix} \quad (8)$$

The IF is *assumed* to have the form

$$g_{i,j} = \begin{bmatrix} a & b & a \\ b & c & b \\ a & b & a \end{bmatrix} \quad (9)$$

reflecting the assumed four-fold symmetry.

Data: The valid convolution $y_{i,j}=h_{i,j} * x_{i,j}$ is 13×14, where 13=53–41+1 and 14=54–41+1. The data is shown in Figure #3. Note that this includes contributions from the random part of the image.

Sparsification: The 11×12 sparsifiable part of the image $x_{i,j}$ is assumed to be sparsifiable by valid convolution with the corner detecting PSF

$$\Phi^{-1}_{i-m,j-n} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (10)$$

The resulting sparsified image $z_{i,j} = \Phi^{-1}_{i,j} * x_{i,j}$ is 10×11, where 10=11–2+1 and 11=12–2+1. The sparsified image $z_{i,j}$ is shown in Figure #4.

Goal: Tbe goal is to compute the sparsifiable region of the image $x_{i,j}$ from the following items:

- Knowledge of only the blurred image $y_{i,j}$;
- Assumed four-fold form of $g_{i,j}$;
- The image is sparsifiable to 12 nonzero values.

B. Problem Solution

The 10×11 2D problem is unwrapped to a (10)(11)=110-point 1D problem using the Agarwal-Cooley algorithm. Since $g_{i,j}$ is parametrized by only three variables a,b,c, the total number of unknowns is (12+1)(3)=39. The large linear system is thus heavily overdetermined, with a 110×39 system matrix. The structure of this matrix is shown in Figure #5.

Despite overdetermination, its singular values are

$$\sigma_1 = 123.; \quad \sigma_{38} = 0.12; \quad \sigma_{39} = 0.000022. \quad (11)$$

so the matrix null vector is fairly well defined.

The length-39 vector is arranged into a 13×3 matrix, which has the singular values

$$\sigma_1 = 1.; \sigma_2 = 0.00000021; \sigma_3 = 0.000000074. \quad (12)$$

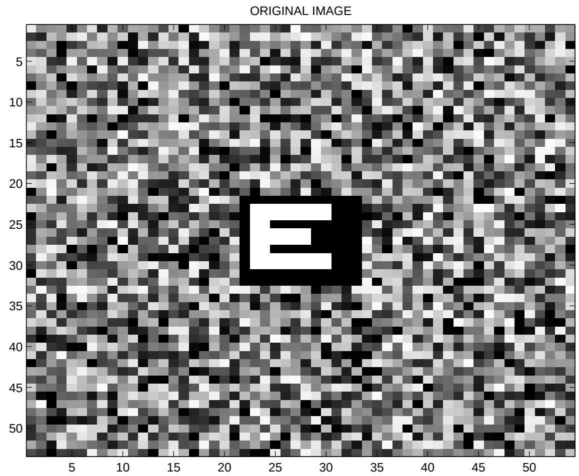
so the rank-one decomposition is well-defined. This decomposition yields, after scaling,

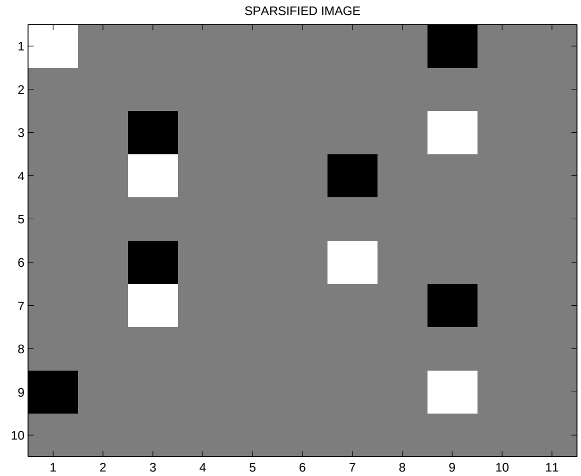
$$[a, b, c] = [-4.0000, -10.0000, 25.0000] \quad (13)$$

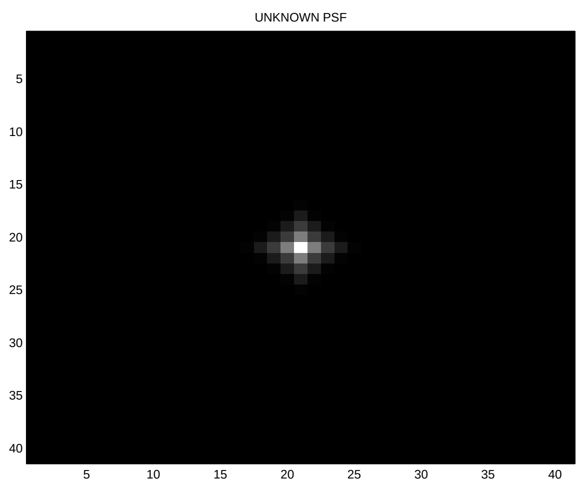
which match the actual values. Inserting these into the form of the IF $g_{i,j}$ yields the 11×12 reconstructed $x_{i,j}=g_{i,j} * y_{i,j}$, where 11=13–3+1 and 12=14–3+1. The reconstructed sparsifiable part of the image is shown in Figure #6. Compare this to Figure #1.

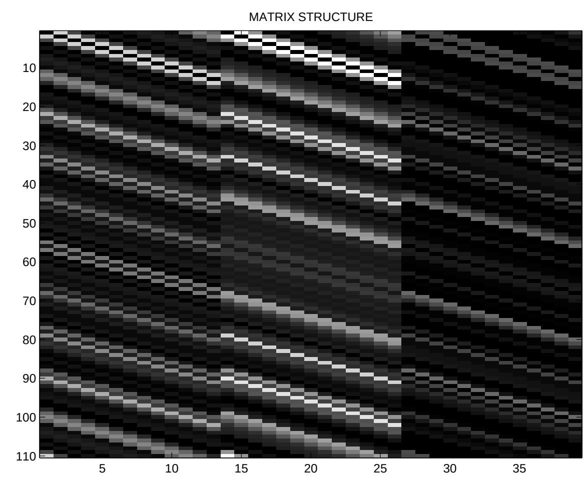
Matlab code used to generate this example:

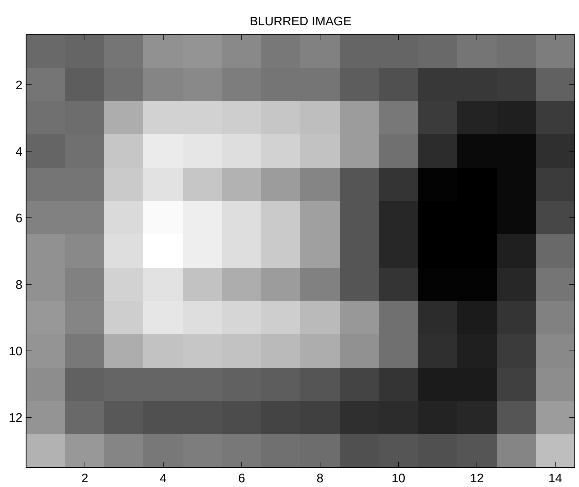
```
clear;X(11,12)=0; %CONSTRUCT BLOCK E:
X(2:9,2:3)=ones(8,2);X(2:3,4:9)=ones(2,6);
X(5:6,4:7)=ones(2,4);X(8:9,4:9)=ones(2,6);
H=[2.^[-20:-1] 1 0.5.^[1:20]];H=H'*H;
XX=rand(53,54);XX(22:32,22:33)=X;
Y=conv2(XX,H,'valid'); %GIVEN DATA Y.
D1=conv2([1 -1;-1 1],[1 0 1;0 0 0;1 0 1]);
D2=conv2([1 -1;-1 1],[0 1 0;1 0 1;0 1 0]);
D3=conv2([1 -1;-1 1],[0 0 0;0 1 0;0 0 0]);
ZZ1=conv2(Y,D1,'valid');ZZ2=conv2(Y,D2,'valid');
ZZ3=conv2(Y,D3,'valid');for I=0:109;
Z1(I+1)=ZZ1(mod(I,10)+1,mod(I,11)+1);
Z2(I+1)=ZZ2(mod(I,10)+1,mod(I,11)+1);
Z3(I+1)=ZZ3(mod(I,10)+1,mod(I,11)+1);end
FZ1=fft(Z1);T1=toeplitz(FZ1',FZ1);
FZ2=fft(Z2);T2=toeplitz(FZ2',FZ2);
FZ3=fft(Z3);T3=toeplitz(FZ3',FZ3);
TT=[T1(:,1:13) T2(:,1:13) T3(:,1:13)];
[U S V]=svd(TT);VV=reshape(V(:,39),13,3);
[U2 S2 V2]=svd(VV);N=real(V2(:,1)/V2(1,1))*4
GHAT= [1 0 1;0 0 0;1 0 1]*N(1);
GHAT=GHAT+[0 1 0;1 0 1;0 1 0]*N(2);
GHAT=GHAT+[0 0 0;0 1 0;0 0 0]*N(3);
XHAT=conv2(Y,GHAT,'valid');
figure,imagesc(XX),colormap(gray)
figure,imagesc(H),colormap(gray)
figure,imagesc(Y),colormap(gray)
figure,imagesc(abs(TT)),colormap(gray)
figure,imagesc(XHAT),colormap(gray)
```

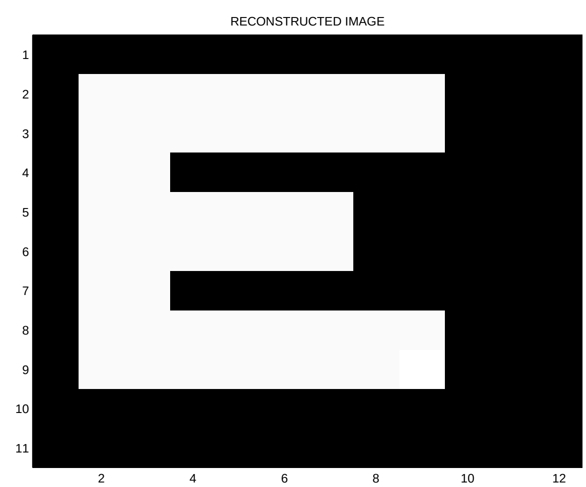

ORIGINAL IMAGE


SPARSIFIED IMAGE


UNKNOWN PSF


MATRIX STRUCTURE


BLURRED IMAGE


RECONSTRUCTED IMAGE