Non-Iterative Compressed Sensing Using a Minimal Number of Fourier Transform Values

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

Abstract— **Reconstruction of signals or images from a few discrete Fourier transform (DFT) values has applications in MRI and SAR. Many real-world signals can be sparsified by an invertible transformation, such as wavelets, into a sparse (mostly zero, with K nonzero values at unknown locations) signal. This K-sparse signal can be reconstructed using the K lowest-frequency DFT values using Prony's method or MUSIC (2K frequencies are required for complex signals or real-valued images). However, this does not work in practice due to poor conditioning caused by the clustering of the locations of the K nonzero values. We use the scaling property of the DFT to uncluster these locations and to spread out the frequencies of known DFT values. We reconstruct a Shepp-Logan phantom using only 2K DFT values, much fewer than the number required by ℓ_1 norm minimization, and using less computation than ℓ_1 norm minimization.**

Keywords— **Sparse reconstruction**
Email: aey@eecs.umich.edu. **EDICS: 2-REST.**

I. INTRODUCTION

A. Problem Statement

The N-point DFT X_k of the length=N signal x_n is

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi nk/N}, k = 0 \dots N-1. \quad (1)$$

- There exists a sparsifying function ψ_n such that:
- $z_n = \sum_{i=0}^{N-1} x_i \psi_{(n-i)\mathrm{mod}(N)}$ is K-sparse, meaning:
- $z_n = 0$ unless $n \in \{n_1 \dots n_K\}$ (the n_i are unknown)
- $Z_k = X_k \Psi_k$ is known for $k \in \{k_1 \dots k_K\}$ and for the conjugate frequencies $\{N - k_i\}$ using $Z_{N-k} = Z_k^*$.
- The DC value $X_0 = \sum_{n=0}^{N-1} x_n$ is also known.

Otherwise, knowledge of X_k and of Z_k is equivalent.

The goal is to compute the K-sparse signal z_n, and then x_n, from DFT values X_k known at K frequencies k_i, conjugate frequencies $N - k_i$, and X_0. Note the problem is underdetermined without sparsity.

B. Problem Significance

Reconstruction of signals and images from limited frequency data occurs in various problems such as:

- Limited-angle tomography in medical imaging;
- Synthetic Aperture Radar (SAR) radar imaging;
- Magnetic Resonance Imaging (MRI) in medicine.

Many signals and images of practical interest have a sparse representation in a wavelet basis. Computation of the wavelet transform can be viewed as convolutions with scaled wavelet and scaling basis functions, which becomes multiplication in the DFT domain. Hence the results of this paper apply to both sparse signals and wavelet-sparsifiable signals.

C. Previous Approaches

A common approach to sparse reconstruction is to compute the minimum ℓ_1 norm solution, perhaps by linear programming. If the DFT frequencies k_i are randomly chosen, and if enough of them are known, then it has been shown that the minimum ℓ_1 norm solution is in fact z_n. In many practical situations we do not have the luxury of choosing the k_i at random-they are pre-specified. And the number of Z_{k_i} required is $O(K \log N)$ (the exact number is unknown).

Other approaches include thresholded Landweber iteration and orthogonal matching pursuit. These are much faster computationally, but require more of the problem in order to compute z_n. Since our approach is completely different from both of these, we refer the reader to the extensive literature on these methods.

D. New Approach

If Z_k is known for all $|k| \leq K$ ($2K+1$ consecutive values of Z_k), then z_n can be reconstructed using any of the well-known array processing techniques such as Prony's method, MUSIC, or ESPRIT. This works well if the locations n_i of nonzero z_n are *unclustered* (spaced out in n). It may seem that using only the lowest DFT frequencies should lead to an ill-conditioned problem, but the DFT basis vectors are all orthonormal, even for adjacent frequencies.

However, sparsified signals tend to have clustered n_i (see Fig. 3 below). This makes the problem ill-conditioned. Consider these two extreme cases:

- $n_i = \{0, \dots K-1\}$. z_n from $\{Z_k, |k| \leq K\}$ is well-known to be very ill-conditioned if $1 << K << N$.
- $n_i = \{0, \frac{N}{K}, 2\frac{N}{K}, \dots (K-1)\frac{N}{K}\}$. z_n from $\{Z_k, |k| \leq K\}$ is perfectly conditioned since unknown values of Z_k are just the periodic extension of the given values.

In order to use the deterministic version of MUSIC (derived below) we must *uncluster* the n_i of the sparsified signal to convert a problem like the former to one more like the latter. This may account for the lack of use of this approach in compressed sensing. We do this below using the DFT scaling property.

II. Derivation of New Algorithm

A. Sparsification of x_n to z_n

Let ψ_n be a sparsifying convolutional basis function, such as a wavelet basis function, and define the sparsified z_n as a cyclic convolution of x_n and ψ_n:

$$z_n = \sum_{i=0}^{N-1} x_i \psi_{(n-i)\,\mathrm{mod}(N)} \tag{2}$$

Taking the N-point DFT of this equation gives

$$Z_k = X_k \Psi_k \quad \text{and} \quad X_k = Z_k/\Psi_k. \tag{3}$$

- z_n=0 except at K locations $\{n_i, i = 1 \dots K\}$.
- Z_k is known at K frequencies $\{k_i, i = 1 \dots K\}$ and their complex conjugate frequencies $\{N–k_i\}$.
- Z_0=$X_0\Psi_0$=0 since Ψ_0=0. But X_0 is known.

For wavelets, Ψ_k=0 for a range of k. Then several time-scaled versions of ψ_n must be used, so that X_k may be recovered from the reconstructed Z_k for all k. No additional data is required, but the following procedure must be repeated for each wavelet scale.

B. Deterministic form of MUSIC

Define the annihilating or indicator function s_n as

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

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

$$s_n z_n = 0 \rightarrow \sum_{k=0}^{N-1} Z_k S_{(n-k)\mathrm{mod}(N)} = 0. \tag{5}$$

$$S(z) = \sum_{k=0}^{K} S_k z^k \tag{6}$$

has zeros $\{e^{jn_i/N}\}$, where n_i are the locations of K nonzero values of z_n. So the inverse DFT of S_k is zero at locations n_i of nonzero z_n, indicating them.

This equation may be arranged into the system

$$\begin{bmatrix} Z_0 & Z_1 & \cdots & Z_K \\ Z_1^* & Z_0 & \cdots & Z_{K-1} \\ \ddots & \ddots & \ddots & \ddots \\ Z_K^* & Z_{K-1}^* & \cdots & Z_0 \end{bmatrix} \begin{bmatrix} S_K \\ S_{K-1} \\ \vdots \\ S_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{7}$$

where we have used conjugate symmetry Z_k^*=Z_{N-k} to show that the matrix has Hermitian symmetry.

So the elements of the right null vector of this singular Hermitian Toeplitz matrix are the DFT S_k of the function s_n. An inverse DFT recovers s_n, and zero values of s_n indicate locations n_i of nonzero z_n.

In Prony's method the zeros $\{e^{jn_i/N}\}$ of $S(z)$ would be computed. But this is not necessary here, since there are only N possible locations of the zeros, and they can all be checked by evaluating $S(z)$ at z=$e^{jk/N}, k = 0 \dots N$–1. This can be done using an inverse DFT (due to the signs of the exponents).

C. Clustering of Nonzero z_n

The problem with using MUSIC is that since z_n is a sparsified image, nonzero values of z_n tend to cluster (see Fig. 3 below). And if there are zeros, say,

$$\{e^{j5/N}, e^{j6/N}, e^{j8/N}, e^{j9/N}\}$$

then the DFT will be very close to zero at $\frac{7}{N}$, so

$$s_5 = s_6 = s_8 = s_9 = 0; \qquad s_7 \approx 0. \tag{8}$$

The zeros of $S(z)$ can still be computed using Prony's method, but this is impractical for large K. And roundoff error in computing the null vector will affect locations of the zeros of $S(z)$.

D. Declustering Using DFT Scaling Property

Let L be any integer $1 < L < N$ relatively prime to N. Then L has a multiplicative inverse $\tilde{L}$ mod(N):

$$L\tilde{L} \equiv 1 \mod(N) \tag{9}$$

$\tilde{L}$ may be computed from L and N by using the Euclidian algorithm to solve in integers the equation

$$L\tilde{L} + NQ = 1 \tag{10}$$

for $\tilde{L}$ and Q (and then discarding Q afterwards).

Then the N-point DFT can be rewritten as

$$Z_k = \sum_{n=0}^{N-1} z_n \exp[-j2\pi \underbrace{(n\tilde{L})}_{n'} \underbrace{(Lk)}_{k'}/N] \tag{11}$$

where we define $n' \equiv n\tilde{L}$ and $k' \equiv Lk$ mod(N).

Changing variables and reordering the sum gives

$$Z_{(\tilde{L}k'\mathrm{mod}(N))} = \sum_{n'=0}^{N-1} z_{(Ln'\mathrm{mod}(N))} e^{-j2\pi n'k'/N}. \tag{12}$$

The point is that $z_{(Ln'\mathrm{mod}(N))}$ is a reordering of z_n that unclusters the nonzero values of z_n, since z_n has been stretched by a factor of L with indices mod(N):

$$n' = \{0, 1, 2, 3 \ldots\} \to \{z_0, z_L, z_{2L}, z_{3L} \ldots\}$$

Similarly, X_k has been compressed by a factor of L. So the original data should be obtained at indices

$$k' = \{0, 1, 2, 3 \ldots\} \to \{X_0, X_{\tilde{L}}, X_{2\tilde{L}}, X_{3\tilde{L}} \ldots\}$$

So the original data is measured at widely spaced frequencies. This may make data acquisition easier.

E. New Algorithm: Procedure

1. Measure these DFT values X_k of the original x_n: $\{X_0, X_{\tilde{L}}, X_{2\tilde{L}}, X_{3\tilde{L}} \ldots X_{K\tilde{L}\,\mathrm{mod}(N)}\}$
2. Compute the DFT Z_k of the sparsified signal z_n: Z_k=$X_k\Psi_k$, where Ψ_k is the DFT of the wavelet ψ_n.
3. Form the Hermitian Toeplitz matrix from $\{Z_k\}$. These are then $\{Z_0, Z_1 \ldots Z_K\}$ for unclustered z_n.
4. Compute inverse N-point DFT of the null vector of the Hermitian Toeplitz matrix. Its elements: s_n.
5. The locations n_i of the zero values of s_n are the locations of the nonzero values of the unclustered z_n.
6. Reorder the unclustered z_n to the original z_n.
7. Compute X_k=Z_k/Ψ_k from the DFT Z_k of z_n.
8. For k such that Ψ_k=0, repeat with other scalings of the wavelet ψ_n. Note that X_0 is known.

F. Modifications: Complex Signals and Images

For complex-valued signals x_n, the same procedure is followed. The only difference is that since conjugate symmetry no longer holds, twice as many DFT values (N–k_i as well as k_i) are required to form the Toeplitz matrix, which is no longer Hermitian.

For images, the Hermitian Toeplitz matrix becomes a Hermitian Block Toeplitz matrix with Toeplitz blocks (BTTB) matrix. Since conjugate symmetry does not hold within each block, again twice as many DFT values are required to form the matrix. The indices are now reordered separately horizontally and vertically, and 2-D DFTs are used. The polynomial zeros are now 2-D polynomial zero curves, sampled only at the 2-D DFT frequencies.

For images, let the sparsified image be K-sparse, and let ceil(x) be the smallest integer exceeding x, e.g., ceil(3.2)=4. Then the BTTB matrix is $K_c^2 \times K_c^2$; it has $K_c \times K_c$ blocks, each of size $K_c \times K_c$, where

$$K_c = \mathrm{ceil}(\sqrt{K}). \tag{13}$$

The number M of 2-D DFT values required is

$$M = ((2K_c - 1)^2 + 1)/2 \tag{14}$$

excluding their complex conjugates.

For the following example, K_c=ceil($\sqrt{1064}$)=33 and M=((2(33)–1)^2+1)/2=2113. The BTTB matrix is $33^2 \times 33^2$=1089×1089.

III. NUMERICAL EXAMPLE

We demonstrate this procedure by reconstructing the 256×256 Shepp-Logan phantom shown in Fig. 1 from the 2113 2-D DFT values and their complex conjugates (4225 total values) at locations shown in Fig. 2. This is significantly fewer than the number required for ℓ_1 norm minimization. The regular pattern of the frequency locations used may make data acquisition of 2-D DFTs at these locations easier.

A. Clustered Sparsified Image

The sparsifying function used is ψ_{n_1,n_2}=$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$. This is the Haar wavelet in the vertical direction. The *unknown* sparsified image is shown in Fig. 3. This image is 1064-sparse, so K=1064 (also unknown).

Note the sparsified image has clustered nonzero values. Attempts to reconstruct the sparsified image in Fig. 3 using DFT values failed, due to this clustering, and are not shown. There was no clear threshold to determine the locations of sparsifed values, and even using the actual value of K=1064 yielded incorrect locations, so the procedure failed.

B. Unclustered Sparsified Image

The DFT scaling property was used with L=17 to uncluster the sparsified image. This gave $\tilde{L}$=241:

$$(17)(241) \equiv 1 \quad \mathrm{mod}(256). \tag{15}$$

The 2-D DFT values Z_{k_1,k_2} are required for $k_1, k_2 \in$

$$\{0, \pm1(241), \pm2(241), \ldots \pm 32(241)\ \mathrm{mod}(256)\} \tag{16}$$

where reduction mod(256) gives an integer between 0 and 255. Note half of these 2-D DFT values (excluding $Z_{0,0}$) are complex conjugates of the other half. The total number of 2-D DFT values used is then

$$(1 + 32 + 32)^2 = 4225; \quad (4225 + 1)/2 = 2113. \tag{17}$$

The unclustered sparsified image (also unknown) is shown in Fig. 4. Note the locations of nonzero values now appears to be random. This greatly improved the conditioning of the problem.

The problem of reconstructing the sparsified image thus has 2113 observations in 65536 unknowns of a 1064-sparse image. Note that these are independent of L; different values of L can be used for declustering, although this will lead to different 2-D DFT frequencies at which the 2-D DFT must be measured.

C. Null Vector of BTTB Matrix

The BTTB matrix had 33×33 blocks of size 33×33 each, so the matrix was $33^2\times33^2$, i.e., 1089×1089. This required the sparsified image have K=1088 or smaller, or the BTTB matrix would not be singular. Fortunately, the sparsity K=1064 < 1089.

The reconstructed reclustered sparsified image computed by finding a null vector of the BTTB matrix and then computing its inverse 2-D DFT is shown in Fig. 5, which should be compared to Fig. 3. The sorted values of s_n showed a sharp threshold:

$$s_{1064}=27\cdot10^{-10}\quad\text{and}\quad s_{1065}=619\cdot10^{-10}\qquad(18)$$

indicating clearly that there are 1064 nonzero values at locations indicated by 1064 smallest values of s_n.

D. Reconstruction of Original Image

Once the locations of nonzero values of the sparsified image were found, reconstruction of z_n required solution of an 4225×1064 linear system of equations. Then using X_k=Z_k/Ψ_k recovered the missing X_k.

However, since the 2-D DFT of ψ_{n_1,n_2}=$\begin{bmatrix}1\\-1\end{bmatrix}$ is

$$\psi_{n_1,n_2}=\begin{bmatrix}1\\-1\end{bmatrix}\rightarrow\Psi_{k_1,k_2}=e^{j2\pi k_2/256}-1,\qquad(19)$$

which is zero for all k_1 when k_2=0, the 2113 2-D DFT values had to be augmented with the additional values $X_{k_1,0}$ not already specified. Alternatively, the entire procedure could be repeated using ψ_{n_1,n_2}=$[1,-1]$, the Haar wavelet in the horizontal direction, but this sparsfied image is only 1482-sparse, so even more 2-D DFT values would be needed.

The final reconstructed image is shown in Fig. 6. It matches the original image in Fig. 1, so the original image was successfully reconstructed.

E. Matlab Program

```
clear;N=256;X=phantom(N);FX=fft2(X);
figure,imagesc(X),colormap(gray)
E=exp(-2j*pi*[0:N-1]/N);O=ones(N,1);
FY=FX.*(O*E).'-FX;%DFT of sparsified image
Y=real(ifft2(FY));%(Unknown) sparsified image
figure,imagesc(Y),colormap(gray)
Y=Y(rem(17*[0:255],N)+1,rem(17*[0:255],N)+1);
figure,imagesc(Y),colormap(gray)
%(Unknown) reordered sparsified image.
%Y not used below-only given to illustrate
Ka=rem(241*[0:32],N)+1;L=length(Ka);L2=2*L-1;
K=[Ka N+2-fliplr(Ka(2:L))];%241*17=1(mod 256)
FI(N,N)=0;FI(K,K)=1;%Locations of known DFT
figure,imagesc(FI),colormap(gray)
FF=FY(K,K);UU(N,N)=0;
%GOAL: Compute Y, then X, from given DFT FF.
%Form Toeplitz-Block-Toeplitz matrix:
for I=1:L;IL=[1 L2:-1:L+1];
T=toeplitz(FF(I,IL),FF(I,1:L));
T=toeplitz(FF(I,IL),FF(I,1:L));
for J=I:L;J1=(J-1)*L;IJ=(J-I)*L;
TT([1:L]+J1,[1:L]+IJ)=T';
TT([1:L]+IJ,[1:L]+J1)=T;
end;end;%Matrix assembly complete
[U,E1]=eig(TT);%Minimum not last one
[Q,EMIN]=min(log10(abs(diag(E1))));
UU(Ka,Ka)=reshape(U(:,EMIN),L,L);
FU=fft2(UU,N,N)';
figure,imagesc(log10(abs(FU))),colormap(gray)
S=sort(abs(FU(:)));S(1062:1067)%sparsity=1064:
%10^-6[.0014,.0027,.0027,.0619,.0628,.0776]
[I J]=find(abs(FU)<S(1065));%We now know 1064
KK=[kron(ones(1,L2),K-1);kron(K-1,ones(1,L2))];
A=exp(2j*pi/N*([I-1 J-1]*KK));%System matrix:
ZY=A'\FF(:);ZZ(N,N)=0;%Sparsified image
for II=1:length(I);ZZ(I(II),J(II))=real(ZY(II));end;
H=(O*E).'-1;FXHAT=fft2(ZZ)./H;%Deconvolve H
FXHAT(N,N)=0;FXHAT(1,:)=FX(1,:);%Known X(0,k)
figure,imagesc(real(ifft2(FXHAT))),colormap(gray)
```

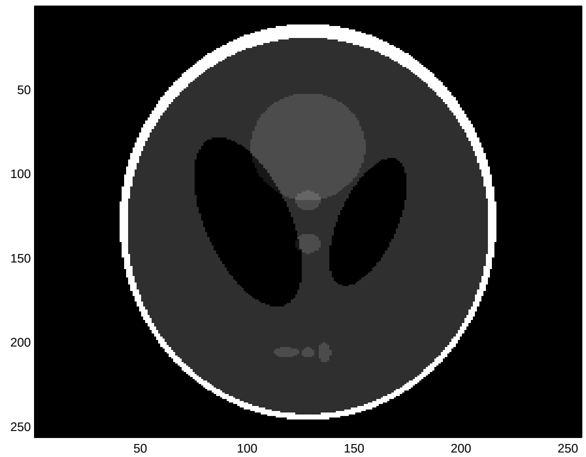

Fig. 1. Original 256×256 Shepp-Logan Phantom.

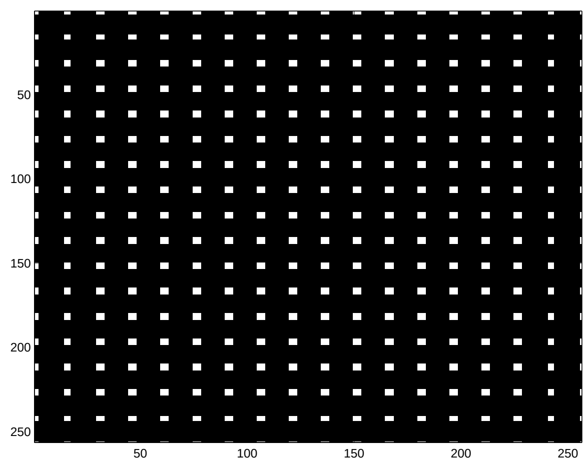

Fig. 2. Locations of the 4225 2-D DFT Values Used.

Fig. 3. Sparsified Image. Note the Clustering.

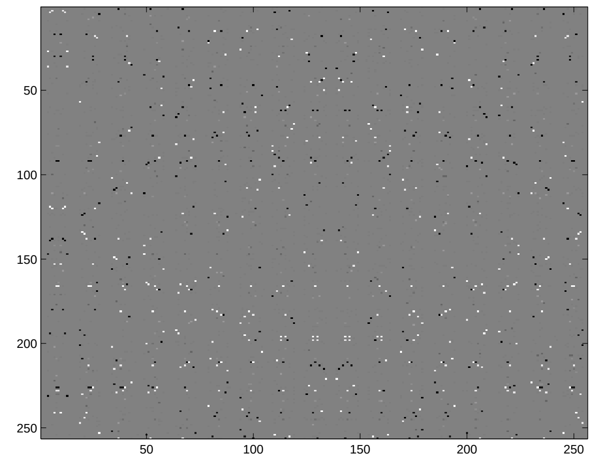

Fig. 4. Declustered Sparsified Image After DFT Scaling. Note the Lack of Clustering of Nonzero Values.

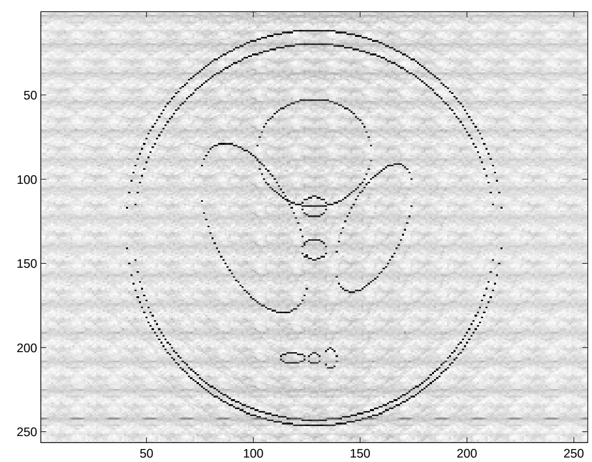

Fig. 5. Reconstructed Sparsified Image.

Fig. 6. Reconstructed Original Image.