Image Reconstruction from Non-Uniformly Sampled Spectral Data

Alfredo Nava-Tudela

May 14, 2009

Abstract

In this project we study and implement software to reconstruct images in the spatial domain from non-uniformly sampled data in the spectral domain. We study existing non-uniform fast Fourier methods and understand their use in image reconstruction. Also, we fit one of these methods to a technique previously developed to reconstruct images from uniformly sampled spectral data. In this project we also create a database of synthetic non-uniform spectral data to test the algorithms developed herein. We deliver Matlab code that implements the techniques described in this document.

1 Problem statement

In this project we address the following problem. Given a discrete set $\mathcal{S}$ of non-uniformly distributed spectral data in the $\widehat{\mathbb{R}}^2$ plane, find a step-wise function in the class of functions $\mathcal{C}_N$ whose Fourier transform best approximates $\mathcal{S}$. The set $\mathcal{C}_N$ is defined as follows

$$\mathcal{C}_N = \left\{ f : \mathbb{R}^2 \to \mathbb{R} \;\middle|\; f(x,y) = \sum_{k,l=0}^{N-1} f_{k,l}\Pi(x-k)\Pi(y-l), f_{k,l} \in \mathbb{R} \right\}, \tag{1}$$

where

$$\Pi(z) = \begin{cases} 1 & \text{if } |z| < \frac{1}{2}, \\ \frac{1}{2} & \text{if } |z| = \frac{1}{2}, \\ 0 & \text{if } |z| > \frac{1}{2}. \end{cases} \tag{2}$$

2 Introduction

In many scientific and engineering disciplines, the use of Fourier techniques to process data is pervasive and fundamental. With the advent of computers in the 1960s, the algorithm nowadays known as the Fast Fourier Transform (FFT), first discovered by Karl Friedrich Gauss in 1805, and later rediscovered by James W. Cooley and John W. Tukey in 1965 [4, 5], has become a workhorse for all sorts of applications.

The FFT, which computes discrete Fourier transforms (DFTs), is notably used in areas of spectral analysis and signal processing. However, numerous applications involve unevenly spaced data, whereas the FFT requires that input data be tabulated on a uniform grid [6].

For example, the Magnetic Resonance Imaging (MRI) technique generates spectral data in a non-uniform grid; and from this data, an image is to be produced. For this project, we are interested in studying the techniques that make this possible, and study how they are related to the FFT. In particular, we are interested in techniques to reconstruct images from non-uniformly sampled spectral data.

3 Images and their Fourier transform

As we mentioned in the introduction, we are interested in algorithms that reconstruct images from non-uniform spectral data. For example, some MRI machines produce spectral data mapped onto interleaving spirals [3]. As mentioned above briefly, the standard FFT algorithm requires evenly spaced data as its input. How is one to reconstruct spatial data from non-uniform spectral data?

3.1 What is an image?

We will adopt the following model for a black and white image. Suppose you have a square image of N by N pixels, and we will think of it as a patchwork of square tiles of unit area 1, each colored with a different tone of gray. See, for example, figure 1.

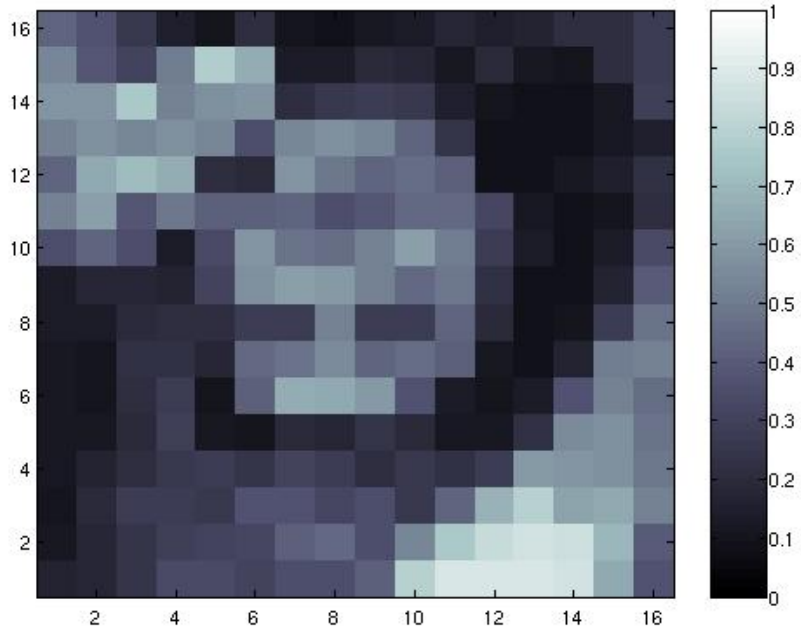

Figure 1: An image represented by the sum of step wise functions.

In this case, our image is a 16 by 16 square with a grayscale of 256 total different values graded from 0, the blackest; to 1, the brightest.

We have then that an image can be represented by a function $f = f(x, y)$ in the Cartesian plane $\mathbb{R}^2$ as:

$$f(x, y) = \sum_{k,l} f_{k,l}\chi_{k,l}(x, y), \tag{3}$$

where $k, l \in \{0, \ldots, N-1\}$, $f_{k,l} \in \mathbb{R}$, and $\chi_{k,l}(x, y)$ is the characteristic function of the square $\square_{k,l} = [k - 1/2, k + 1/2) \times [l - 1/2, l + 1/2)$; that is,

$$\chi_{k,l} = \begin{cases} 1 & \text{if } (x, y) \in \square_{k,l}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Note that the square $\square_{k,l}$ is the unit square centered around the point with integer coordinates (k, l) [1].

3.2 Fourier transform of an image

We can define the Fourier transform of an image $f = f(x, y)$ in the following way:

$$\hat{f}(\sigma, \gamma) = \int_{\mathbb{R}^2} f(x, y)e^{-2\pi i(x\sigma+y\gamma)}dxdy. \tag{5}$$

If we consider functions f as defined in 3.1, we then obtain

$$\begin{aligned} \hat{f}(\sigma, \gamma) &= \int_{\mathbb{R}^2} \sum_{k,l} f_{k,l}\chi_{k,l}(x, y)e^{-2\pi i(x\sigma+y\gamma)}dxdy \\ &= \sum_{k,l} \int_{\mathbb{R}^2} f_{k,l}\chi_{k,l}(x, y)e^{-2\pi i(x\sigma+y\gamma)}dxdy \\ &= \sum_{k,l} \int_{l-1/2}^{l+1/2} \int_{k-1/2}^{k+1/2} f_{k,l}e^{-2\pi i(x\sigma+y\gamma)}dxdy \\ &= \sum_{k,l} f_{k,l} \int_{k-1/2}^{k+1/2} e^{-2\pi ix\sigma}dx \int_{l-1/2}^{l+1/2} e^{-2\pi iy\gamma}dy, \end{aligned} \tag{6}$$

with $k, l \in \{0, \ldots, N-1\}$, for an N by N image. We pause for a moment, and introduce the rectangle function Π defined as follows,

$$\Pi(x) = \begin{cases} 1 & \text{if } |x| < \frac{1}{2}, \\ \frac{1}{2} & \text{if } |x| = \frac{1}{2}, \\ 0 & \text{if } |x| > \frac{1}{2}. \end{cases} \tag{7}$$

Lets compute $\widehat{\Pi}$ as this will help simplify further equation 6:

[1] This represents a change from the original setup where $\square_{k,l} = [k, k + 1) \times [l, l + 1)$, which simplifies the formulas some.

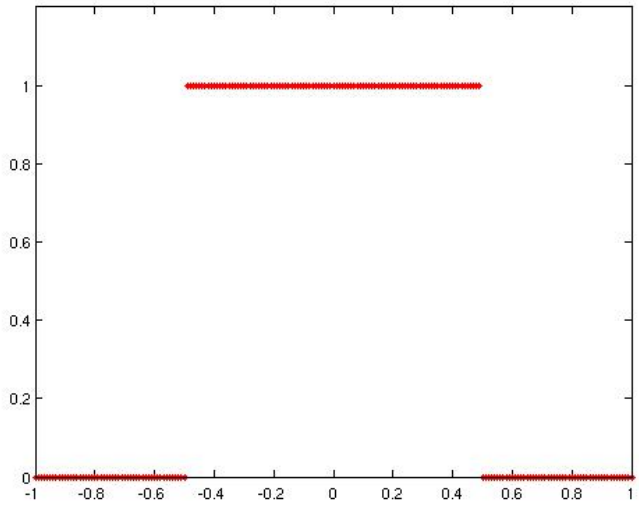

Figure 2: Graph of the rectangle function Π.

$$\begin{aligned}
\widehat{\Pi}(\gamma) &= \int_{-\infty}^{\infty} \Pi(x) e^{-2\pi i x\gamma} dx \\
&= \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i x\gamma} dx \qquad (8) \\
&= \frac{1}{-2\pi i\gamma} e^{-2\pi i x\gamma}\Big|_{x=-\frac{1}{2}}^{\frac{1}{2}} \\
&= \frac{1}{-2\pi i\gamma}\left(e^{-\pi i\gamma} - e^{\pi i\gamma}\right) \\
&= \frac{\sin(\pi\gamma)}{\pi\gamma} := \mathrm{sinc}_\pi(\gamma). \qquad (9)
\end{aligned}$$

Note that the right hand side of equation 8 is remarkably similar to both integrals in the right hand side of equation 6. In fact they are identical modulo a translation by k and l in x and y respectively. This is not surprising as we are dealing with the multiplication of translates of the rectangle function.

Here is where we can introduce and verify the following well known fact of the Fourier transform. If $f(x) \leftrightarrow \hat{f}(\gamma)$, then $f(x-a) \leftrightarrow e^{-2\pi i a\gamma}\hat{f}(\gamma)$. Coming back to equation 6 we obtain:

$$\begin{aligned}
\hat{f}(\sigma,\gamma) &= \sum_{k,l} f_{k,l} \int_{k-1/2}^{k+1/2} e^{-2\pi i x\sigma} dx \int_{l-1/2}^{l+1/2} e^{-2\pi i y\gamma} dy \\
&= \sum_{k,l} f_{k,l} \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i (x'+k)\sigma} dx' \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i (y'+l)\gamma} dy' \\
&= \sum_{k,l} f_{k,l} \Big(e^{-2\pi i k\sigma} \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i x'\sigma} dx' \Big) \Big(e^{-2\pi i l\gamma} \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i y'\gamma} dy' \Big) \\
&= \sum_{k,l} f_{k,l} e^{-2\pi i k\sigma} \mathrm{sinc}_\pi(\sigma) e^{-2\pi i l\gamma} \mathrm{sinc}_\pi(\gamma). \qquad (10)
\end{aligned}$$

Summarizing our results from this section, we then have that if

$$f(x,y) = \sum_{k,l} f_{k,l} \chi_{k,l}(x,y), \qquad (11)$$

then

$$\begin{aligned}
\hat{f}(\sigma,\gamma) &= \sum_{k,l} f_{k,l} e^{-2\pi i k\sigma} \mathrm{sinc}_\pi(\sigma) e^{-2\pi i l\gamma} \mathrm{sinc}_\pi(\gamma) \\
&= \mathrm{sinc}_\pi(\sigma) \mathrm{sinc}_\pi(\gamma) \Big(\sum_{k,l} f_{k,l} e^{-2\pi i k\sigma} e^{-2\pi i l\gamma} \Big), \qquad (12)
\end{aligned}$$

where $k,l \in \{0,\ldots,N-1\}$, $f_{k,l} \in \mathbb{R}$, $\chi_{k,l}(x,y)$ is the characteristic function of the square $\square_{k,l} = [k-1/2, k+1/2) \times [l-1/2, l+1/2)$, and sinc_π is as defined in equation 9.

Observing that when $m,n \in \mathbb{Z}$

$$e^{-2\pi i k(\sigma+m)} e^{-2\pi i l(\gamma+n)} = e^{-2\pi i k\sigma} e^{-2\pi i l\gamma}, \qquad (13)$$

we obtain from equation 12 that the quotient

$$\frac{\hat{f}(\sigma+m,\gamma+n)}{\hat{f}(\sigma,\gamma)} = \frac{\mathrm{sinc}_\pi(\sigma+m)\mathrm{sinc}_\pi(\gamma+n)}{\mathrm{sinc}_\pi(\sigma)\mathrm{sinc}_\pi(\gamma)}, \qquad (14)$$

or, equivalently,

$$\hat{f}(\sigma+m,\gamma+n) = \frac{\mathrm{sinc}_\pi(\sigma+m)\mathrm{sinc}_\pi(\gamma+n)}{\mathrm{sinc}_\pi(\sigma)\mathrm{sinc}_\pi(\gamma)} \hat{f}(\sigma,\gamma). \qquad (15)$$

Therefore, $\hat{f}(\sigma+m,\gamma+n)/\hat{f}(\sigma,\gamma)$ is independent of the values $\{f_{k,l}\}$, and knowing $\hat{f}$ in the unit square $[0,1)\times[0,1) \subset \widehat{\mathbb{R}}^2$ determines the value of $\hat{f}$ in all of $\widehat{\mathbb{R}}^2$, i.e., we only need to know what goes on in the unit square to know what goes on in all the plane.

4 Theory for solution to the image reconstruction problem

Let us revisit the definition of an image from section 3.1. We had established that we could think of a black and white image as the sum of piecewise constant functions, as described in equation 3.

Lets relabel the square tiles that make the image in a new alphabetical order, and rewrite f using this new index, say

$$f(x,y) = \sum_j f_j \chi_j(x,y), \tag{16}$$

where this new alphabetical ordering imposes a natural bijection $j \mapsto (k_j, l_j) = (k,l)$, where $f_j = f_{k,l}$ and $\chi_j = \chi_{k,l}$, for some $k,l \in \{0,\dots,N-1\}$ as before; and $j \in \{0,\dots N^2-1\}$. Using this notation, we can also rewrite the Fourier transform of f as

$$\hat{f}(\sigma,\gamma) = \sum_j f_j e^{-2\pi i k_j \sigma} \mathrm{sinc}_\pi(\sigma) e^{-2\pi i l_j \gamma} \mathrm{sinc}_\pi(\gamma), \tag{17}$$

or, equivalently,

$$\hat{f}(\sigma,\gamma) = \mathrm{sinc}_\pi(\sigma)\mathrm{sinc}_\pi(\gamma) \left(\sum_j f_j e^{-2\pi i k_j \sigma} e^{-2\pi i l_j \gamma} \right). \tag{18}$$

If we call for simplicity

$$a_j(\sigma,\gamma) = \mathrm{sinc}_\pi(\sigma)\mathrm{sinc}_\pi(\gamma) e^{-2\pi i k_j \sigma} e^{-2\pi i l_j \gamma}, \tag{19}$$

then we can write equation 17 as

$$\hat{f}(\sigma,\gamma) = \sum_j f_j a_j(\sigma,\gamma). \tag{20}$$

Now assume that we have an image $g \in L(\mathbb{R}^2) = \{f : \mathbb{R}^2 \to \mathbb{R} : \int_{\mathbb{R}^2} |f(x,y)|dxdy < \infty\}$[2] with Fourier transform $\hat{g}$. Assume that we have sampled in the spectral domain M values of $\hat{g}$ at $\{(\sigma_i,\gamma_i) \in \widehat{\mathbb{R}^2} : i = 0,\dots,M-1\}$, say $\hat{g}(\sigma_i,\gamma_i) = b_i$.

If we would like to reconstruct g from this data with a piecewise constant function f at a resolution of N by N pixels, we should satisfy —using equation 20— the following equality for all $i \in \{0,\dots,M-1\}$:

$$\hat{f}(\sigma_i,\gamma_i) = \sum_{j=0}^{N^2-1} f_j a_j(\sigma_i,\gamma_i) = b_i. \tag{21}$$

In matrix notation, this is equivalent to writing

[2]This condition guarantees that $\hat{g}(\sigma,\gamma) = \int\int g(x,y)e^{-2\pi i(x\sigma+y\gamma)}dxdy$ exists.

$$\begin{pmatrix} a_0(\sigma_0,\gamma_0) & a_1(\sigma_0,\gamma_0) & \dots & a_{N^2-1}(\sigma_0,\gamma_0) \\ a_0(\sigma_1,\gamma_1) & a_1(\sigma_1,\gamma_1) & \dots & a_{N^2-1}(\sigma_1,\gamma_1) \\ \vdots & \vdots & & \vdots \\ a_0(\sigma_{M-1},\gamma_{M-1}) & a_1(\sigma_{M-1},\gamma_{M-1}) & \dots & a_{N^2-1}(\sigma_{M-1},\gamma_{M-1}) \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N^2-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{M-1} \end{pmatrix} \tag{22}$$

or, equivalently,

$$\boldsymbol{A}\boldsymbol{f} = \boldsymbol{b}, \tag{23}$$

where $\boldsymbol{A}$ is the M by N^2 matrix that captures the geometry of the sampled points in the spectral domain, $\boldsymbol{f}$ is the N^2 by 1 vector of the reconstruction image coefficients, and $\boldsymbol{b}$ is the M by 1 vector of spectral data values sampled at the geometric locations coded into $\boldsymbol{A}$.

If we had $M = N^2$, the image reconstruction problem would boil down to solving the linear system of equations 23. However, we will most likely have $M \geq N^2$, and system 23 will be overdetermined.

In this case, we proceed by premultiplying equation 23 by $\boldsymbol{A}^*$, the conjugate transpose of $\boldsymbol{A}$. The linear system of equations then becomes

$$(\boldsymbol{A}^*\boldsymbol{A})\boldsymbol{f} = \boldsymbol{A}^*\boldsymbol{b}. \tag{24}$$

Since $\boldsymbol{A}^*\boldsymbol{A}$ is clearly Hermitian, and every Hermitian matrix is also normal, the finite-dimensional spectral theorem applies. This gives that $\boldsymbol{A}^*\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^*$ for some unitary matrix $\boldsymbol{Q}$, and some real diagonal matrix $\boldsymbol{D} = \text{diag}(\lambda_1, \dots, \lambda_{N^2})$. Since $\boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^*$ is a similarity transformation of $\boldsymbol{D}$, and the eigenvalues of $\boldsymbol{D}$ are clearly $\{\lambda_1, \dots, \lambda_{N^2}\}$, the eigenvalues of $\boldsymbol{A}^*\boldsymbol{A}$ are also $\{\lambda_1, \dots, \lambda_{N^2}\}$. Hence, $\boldsymbol{A}^*\boldsymbol{A}$ has real eigenvalues, and provided non of them is zero, we then have that $\boldsymbol{A}^*\boldsymbol{A}$ is invertible, giving the following solution to our image inversion problem

$$\boldsymbol{f} = (\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*\boldsymbol{b}. \tag{25}$$

It turns out that if the columns of $\boldsymbol{A}$ are linearly independent, then $\boldsymbol{A}^*\boldsymbol{A}$ is invertible [2]. Moreover, if the sampling occurs on a random pattern, it can be shown that with probability one $\boldsymbol{A}^*\boldsymbol{A}$ is invertible [1].

The matrix $(\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*$ is called the Moore-Penrose pseudoinverse of $\boldsymbol{A}$, and it will give the solution to equation 23 in the least squares sense [7, 9], that is

$$\|\boldsymbol{A}\boldsymbol{f} - \boldsymbol{b}\|^2 = \min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2. \tag{26}$$

5 Algorithms for image reconstruction from spectral data

If we wanted to reconstruct an image at a resolution of 32 by 32 pixels, for example, we would need a matrix $\boldsymbol{A}$ of at least dimensions 32^2 by 32^2. That is, we would need storage for 32^4 or 1,048,576 floating point numbers. In general, for an image resolution of N by N, we would need $\mathcal{O}(N^4)$ storage. This means that the algorithm we choose should be able to do its job without storing $\boldsymbol{A}$, or $\boldsymbol{A}^*\boldsymbol{A}$, explicitly.

5.1 The Singular Value Decomposition, insights and results

Recall from equation 26 that $\boldsymbol{f}$ is the solution of minimum norm to the least squares problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2. \tag{27}$$

It can be proven that a matrix $\boldsymbol{A}$ of dimensions M by N^2 can be written as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$, where $\boldsymbol{U}$ has dimensions M by M, $\boldsymbol{V}$ has dimensions N^2 by N^2, both $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices, and $\boldsymbol{\Sigma}$ is an M by N^2 matrix having only non-zero elements in the main diagonal and they are non-negative real numbers $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_{N^2} \geq 0$, called the singular values of $\boldsymbol{A}$ [8].

Observe that

$$\begin{aligned} \boldsymbol{A}^*\boldsymbol{A} &= (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*)^*\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* \\ &= (\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^*)\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^*. \end{aligned} \tag{28}$$

Since $\boldsymbol{V}$ is a unitary matrix, the right hand side of equation 28 is a similarity transformation, therefore the eigenvalues of $\boldsymbol{A}^*\boldsymbol{A}$ coincide with those of $\boldsymbol{\Sigma}^2$, which are exactly $\sigma_1^2 \geq \sigma_2^2 \ldots \geq \sigma_{N^2}^2 \geq 0$. Hence, $\boldsymbol{A}^*\boldsymbol{A}$ is positive semidefinite.

Now, back to our minimization problem given by equation 27. If we call $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$, then

$$\begin{aligned} \|\boldsymbol{r}\|^2 &= \boldsymbol{r}^*\boldsymbol{r} \\ &= (\boldsymbol{U}^*\boldsymbol{r})^*(\boldsymbol{U}^*\boldsymbol{r}) \\ &= \|\boldsymbol{U}^*\boldsymbol{r}\|^2 \\ &= \|\boldsymbol{U}^*\boldsymbol{A}\boldsymbol{x} - \boldsymbol{U}^*\boldsymbol{b}\|^2 \\ &= \|\boldsymbol{\Sigma}\boldsymbol{V}^*\boldsymbol{x} - \boldsymbol{c}\|^2, \end{aligned} \tag{29}$$

where $c_j = \boldsymbol{u}_j^*\boldsymbol{b}$, $j = 1, \ldots, M$ and $\boldsymbol{u}_j$ is the j^{th} column of $\boldsymbol{U}$. Now, by letting $\boldsymbol{w} = \boldsymbol{V}^*\boldsymbol{x}$, equation 29 is equivalent to

$$\|\boldsymbol{r}\| =^2 (\sigma_1 w_1 - c_1)^2 + \ldots + (\sigma_n w_n - c_n)^2 + c_{n+1}^2 + \ldots + c_m^2. \tag{30}$$

Therefore, minimizing the norm of $\boldsymbol{r}$ is equivalent to minimizing the right hand side of 30. This gives the following algorithm to solve equation 27.

1. Compute $\boldsymbol{c} = \boldsymbol{U}^*\boldsymbol{b}$
2. Let p be the number of nonzero singular values of $\boldsymbol{A}$
3. for $j = 1, \ldots, p$

   Set $w_j = c_j/\sigma_j$

   end

4. The minimum norm solution is $\boldsymbol{x} = \boldsymbol{V}(:, 1:p)\boldsymbol{w}$
   The norm of the residual is $(c_{p+1} + \ldots + c_m)^{1/2}$

Even though this algorithm would produce the right answer for our problem, it has the following caveat. It requires the computation of the SVD decomposition of $\boldsymbol{A}$ and storage of $\boldsymbol{U}$ and $\boldsymbol{V}$, which can be very large and dense. Therefore, we need a low storage algorithm.

5.2 Conjugate gradient method, a low storage solution

We say that a set $\mathcal{V} = \{\boldsymbol{v}_j \in \mathbb{C}^n : j = 1, \ldots, n\}$ is conjugate with respect to an n by n Hermitian positive definite matrix $\boldsymbol{B}$, or $\boldsymbol{B}$-conjugate, if

$$\boldsymbol{v}_j^* \boldsymbol{B} \boldsymbol{v}_k = 0 \quad \text{if and only if} \quad j \neq k. \tag{31}$$

Assume that we are given an $\boldsymbol{A}^*\boldsymbol{A}$-conjugate set $\mathcal{V}$, and that we want to solve equation 24. Consider $\boldsymbol{f}$ of the form

$$\boldsymbol{f} = \sum_{j=1}^{n} \alpha_j \boldsymbol{v}_j, \quad \text{with } \boldsymbol{v}_j \in \mathcal{V}, \tag{32}$$

then, substituting $\boldsymbol{f}$ into 24 and premultiplying by $\boldsymbol{v}_k^*$ both sides of that equation we obtain

$$\begin{aligned}
\boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{A} \boldsymbol{f} &= \boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{b} \\
\boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{A} \sum_{j=1}^{n} \alpha_j \boldsymbol{v}_j &= \boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{b} \\
\sum_{j=1}^{n} \alpha_j \boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{A} \boldsymbol{v}_j &= \boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{b} \\
\alpha_k \boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{A} \boldsymbol{v}_k &= \boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{b},
\end{aligned} \tag{33}$$

from which

$$\alpha_k = \frac{\boldsymbol{v}_k^* \boldsymbol{c}}{\boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{A} \boldsymbol{v}_k} \qquad \text{for } k = 1, \ldots, n, \tag{34}$$

where we have set $\boldsymbol{c} = \boldsymbol{A}^*\boldsymbol{b}$, and provided $\boldsymbol{v}_k^* \boldsymbol{A}^* \boldsymbol{A} \boldsymbol{v}_k$ is not zero. This means that if we obtain an $\boldsymbol{A}^*\boldsymbol{A}$-conjugate set $\mathcal{V}$ we can solve with 34 our least squares problem. This is the theoretical basis for the conjugate gradient algorithm [10, 12]. The algorithm basically minimizes $(1/2)\boldsymbol{x}^*\boldsymbol{B}\boldsymbol{x} - \boldsymbol{x}^*\boldsymbol{c}$, where $\boldsymbol{B}$ is a Hermitian positive definite matrix. In our case, based on the results of section 5.1, we have that $\boldsymbol{A}^*\boldsymbol{A}$ is at least positive semidefinite. However, it is known that with probability one $\boldsymbol{A}^*\boldsymbol{A}$ is positive definite for a randomly sampled spectral data set, see for example [1], and therefore we can assume that this is the case in general. The algorithm has the advantage that we do not have to explicitly store $\boldsymbol{A}^*\boldsymbol{A}$, but be able to compute $\boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x}$ for arbitrary vectors $\boldsymbol{x}$. Consider equation 24, then the conjugate gradient method algorithm translates into

1. Given $\boldsymbol{x}^{(0)}$, form $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x}^{(0)}$, $\boldsymbol{s}^{(0)} = \boldsymbol{r}^{(0)}$.

2. for $k = 0, 1, \ldots$ until convergence

   Let $\boldsymbol{z}^{(k)} = \boldsymbol{A}^* \boldsymbol{A} \boldsymbol{s}^{(k)}$

   Let the step length be $\alpha^{(k)} = (\boldsymbol{r}^{(k)*} \boldsymbol{s}^{(k)}) / (\boldsymbol{s}^{(k)*} \boldsymbol{z}^{(k)})$

   Let $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha^{(k)} \boldsymbol{s}^{(k)}$

   Update the negative gradient $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha^{(k)} \boldsymbol{z}^{(k)}$

   Let $\beta^{(k+1)} = (\boldsymbol{r}^{(k+1)*} \boldsymbol{r}^{(k+1)}) / (\boldsymbol{r}^{(k)*} \boldsymbol{r}^{(k)})$

   Let the new search direction be $\boldsymbol{s}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta^{(k+1)} \boldsymbol{s}^{(k)}$

6 Experiments and validation

To test and validate our model and algorithms, we conducted several experiments. We built code to implement both the direct solution of the system of equations 23 as well as the solution by Conjugate Gradient. The figures referred to in this section are in the appendices. We describe the different sets of experiments conducted next.

6.1 Proof of concept

The first experiment we conducted was the proof of concept. We took a 16 x 16 resolution image and set out to recreate a same resolution image from spectral data. To this effect, we built explicitly the matrices in equation 25 to find the image from the full spectral data, that is, we sampled on a regular grid in the spectral domain 256 points. Figure 3 shows the original image. On figure 4 we can see that the reconstruction by this method reproduces exactly the original. The error was within machine precision.

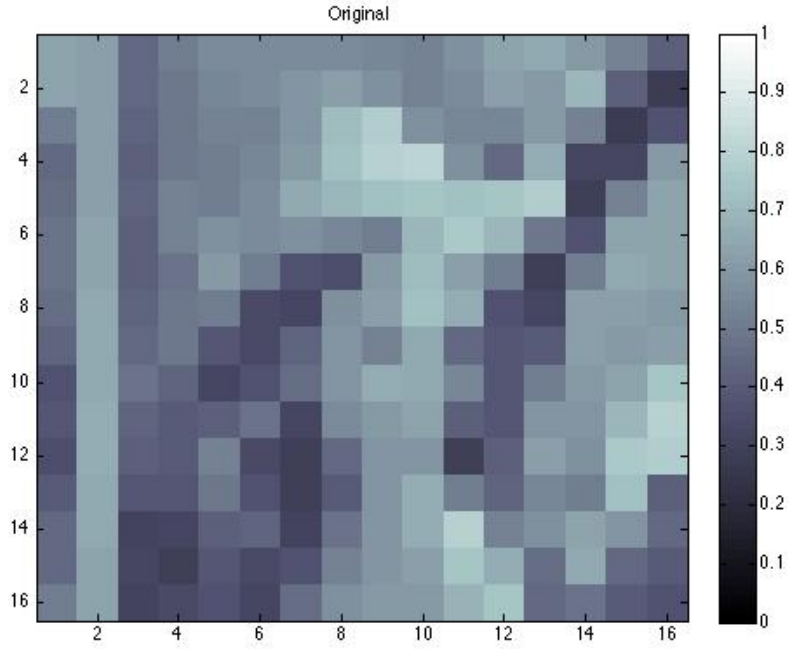

Figure 3: Proof of concept: original 16 x 16 image.

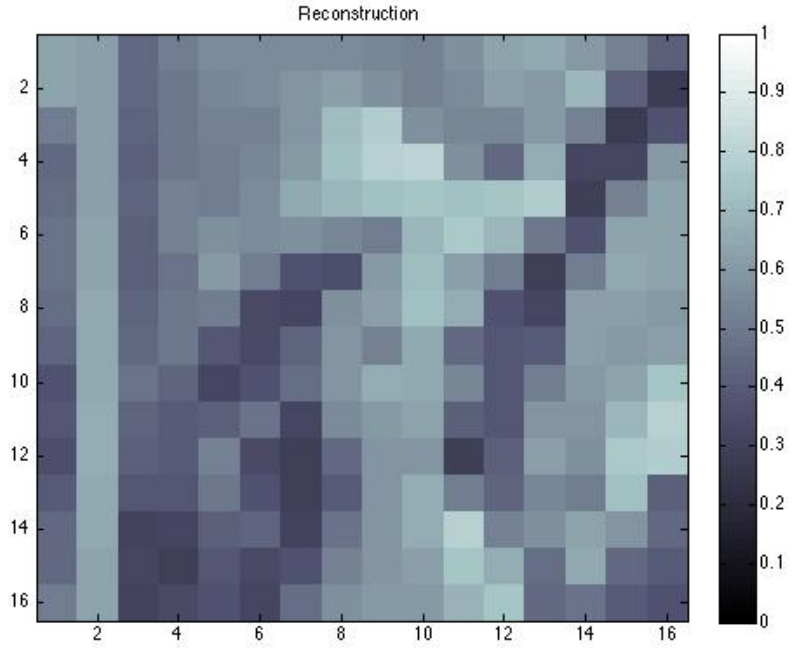

Figure 4: Proof of concept: reconstruction at 16 x16 resolution.

Another set of experiments we conducted, still using the direct solution of the full

system of equations explicitly, consisted of the following. We took a high resolution image from our image database, conducted a series of closest-neighbor-averaging to down sample in the spatial domain the original high resolution image, and then we used spectral information from the high resolution image to obtain an image reconstruction at the same low resolution than that of the down-sampled original, and compared the results.

In figure 5, we show one of the three 512 x 512 high resolution images that we used for this second set of experiments. In figure 6 we can see the result of down-sampling in the spatial domain the original high resolution image four times. Then, using the high resolution image to obtain the spectral data, we sampled the frequency information on a regular grid in the spectral domain. The area of the spectral domain used was the unit square, given that all the information is contained therein, as concluded from equation 15.

The reconstruction can be seen in figure 7. We note that the agreement is not within machine precision error, as corroborated in figure 8. This is because the effects of closest-neighbor-averaging on the Fourier transform are not the same as the spectral sampling from the high quality image but for a small correction.

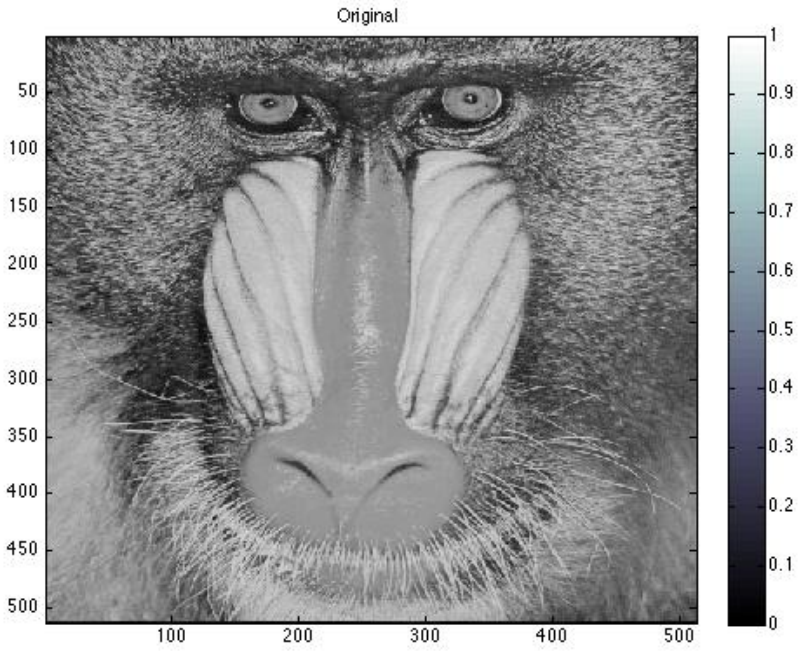

Figure 5: One of the 512 x 512 high resolution images used in the second set of experiments.

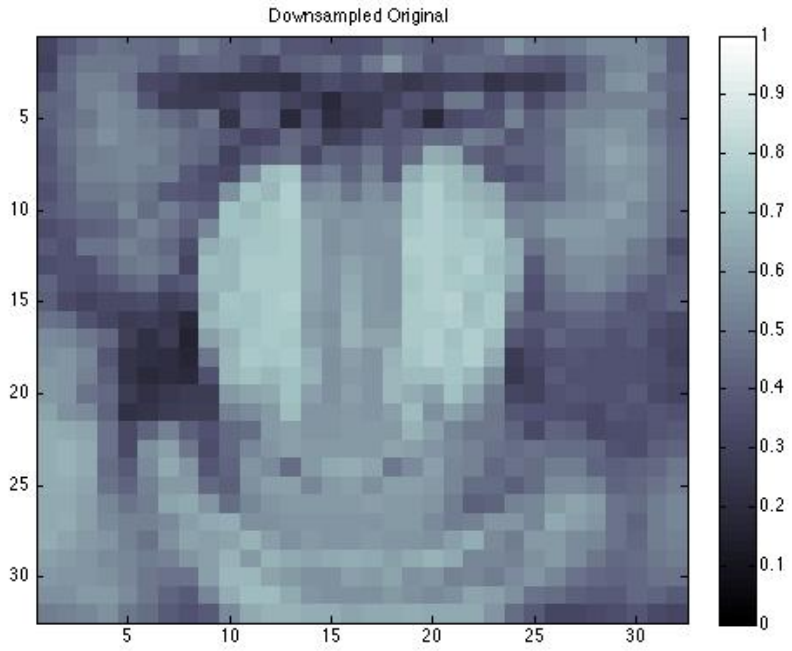

Figure 6: Image down-sampled in the spatial domain to 32 x 32 pixels from a 512 x 512 resolution.

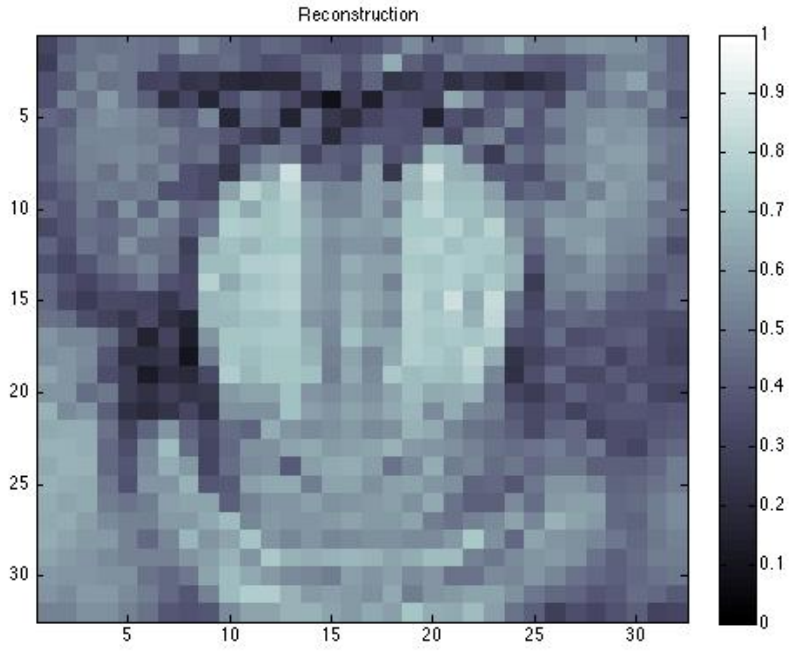

Figure 7: Reconstruction from high resolution image spectral data on a regular grid.

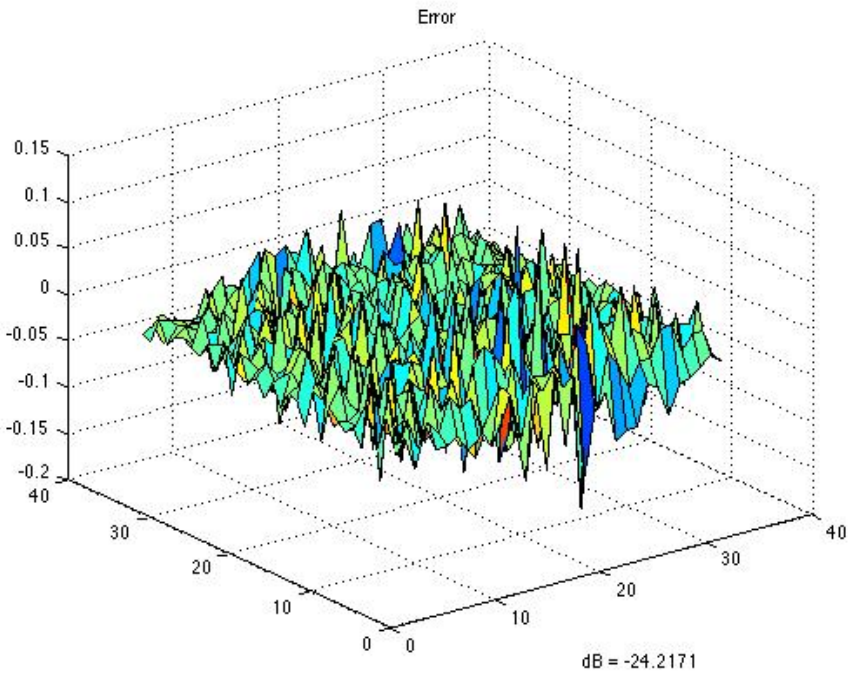

Figure 8: Error between the down-sampled image and the reconstructed image.

These experiments prove that the theory developed so far works. However, the direct solution of the system of equations 23 has the caveat that it is memory intensive. For example, consider trying to reconstruct an image at 64 x 64 bits of resolution. This would require a storage of at least 64^4 complex numbers when forming $\boldsymbol{A}^*\boldsymbol{A}$. This leads us to the implementation of the Conjugate Gradient method.

6.2 The Conjugate Gradient method experiments and results

Our implementation of the Conjugate Gradient method (CG) is taken from modifying the one described in [10], p. 50, but without the restart step. To implement this algorithm we had to create two functions, A_times() and A_star_times() in Matlab, that return –given the proper input– the products of $\boldsymbol{A}$ or $\boldsymbol{A}^*$, respectively, with a given vector.

The third set of experiments that we carried over were geared towards implementing and understanding the behavior of the CG method when the reconstructed image had the same resolution as the original. This would validate the implementation of the CG algorithm.

We selected an image of size 16 x 16 and reconstructed a 16 x 16 image using data taken on a 16 x 16 grid in the spectral domain at regular intervals in the unit spectral square. Figures 9 through 14 show the progressive approximation to the original image as the tolerance parameter in the CG code went from 10^{-1} to 10^{-6}. The images are represented in linear form as vectors thus plotted. The original image is painted in blue, the reconstruction in red. We repeated this experiment for images of sizes 32 x

32, 64 x 64, and 128 x 128 with identical results.

The spectral data for the runs in this experiment was produced using $\boldsymbol{A}$ and the fact that $\boldsymbol{Af} = \widehat{\boldsymbol{f}}$. To guarantee that the method worked by providing spectral data in a way other than by this artifact, we ran the next experiment.

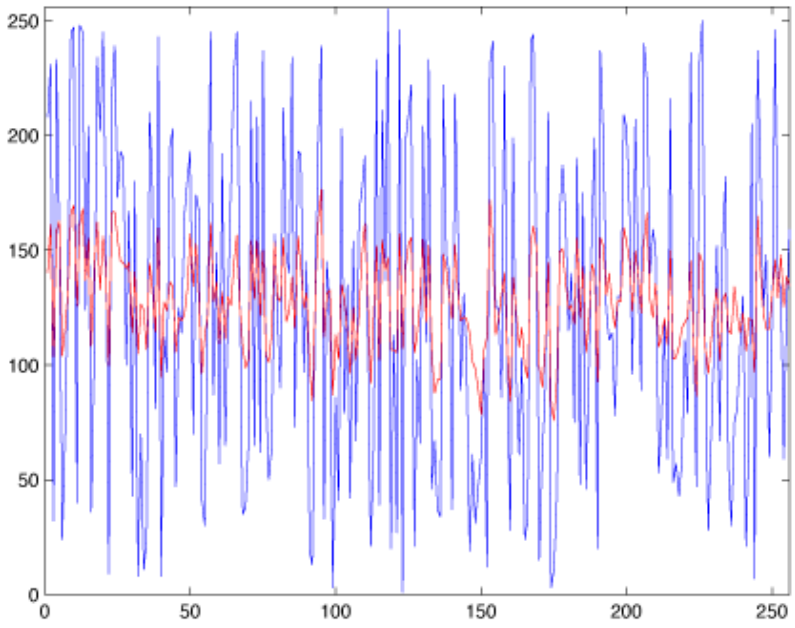

Figure 9: Approximation of the CG solution, in red, to the original image, in blue. Tolerance 1e-1, N=16.

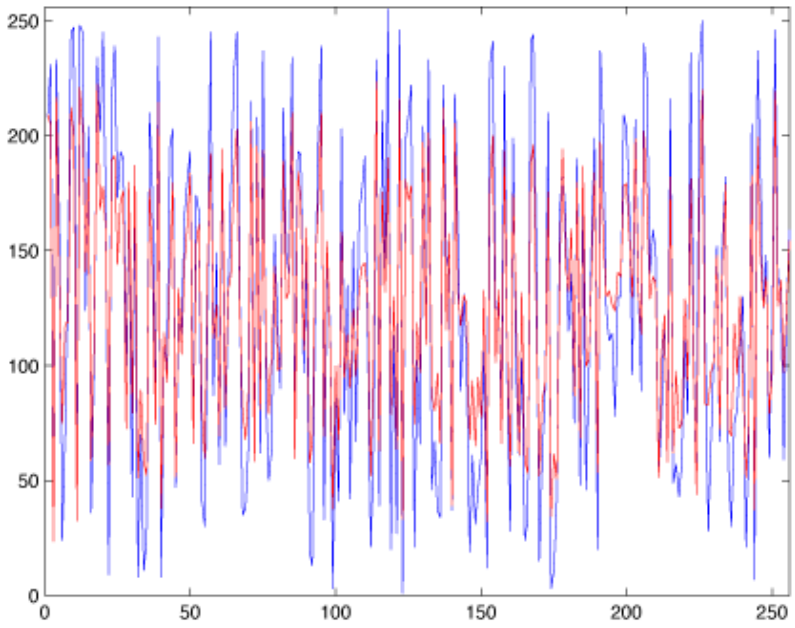

Figure 10: Approximation of the CG solution, in red, to the original image, in blue. Tolerance 1e-2, N=16.

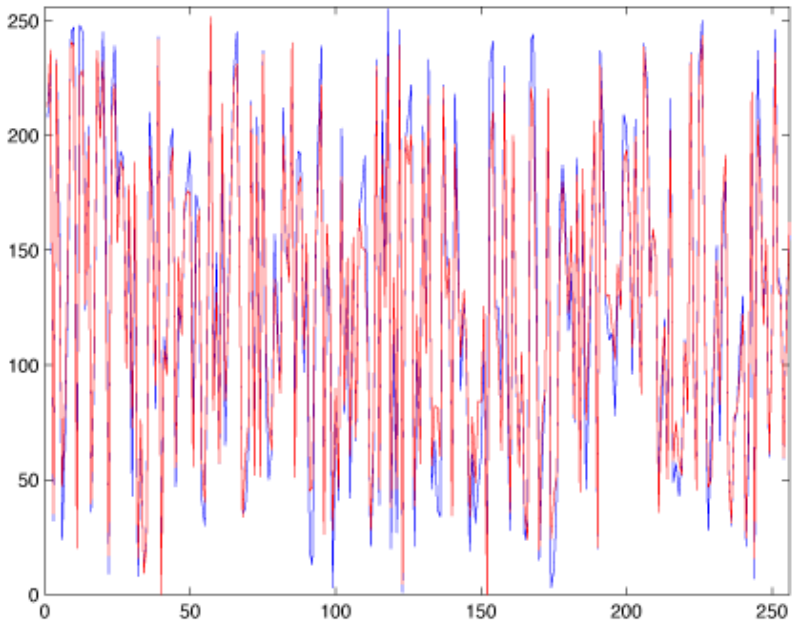

Figure 11: Approximation of the CG solution, in red, to the original image, in blue. Tolerance 1e-3, N=16.

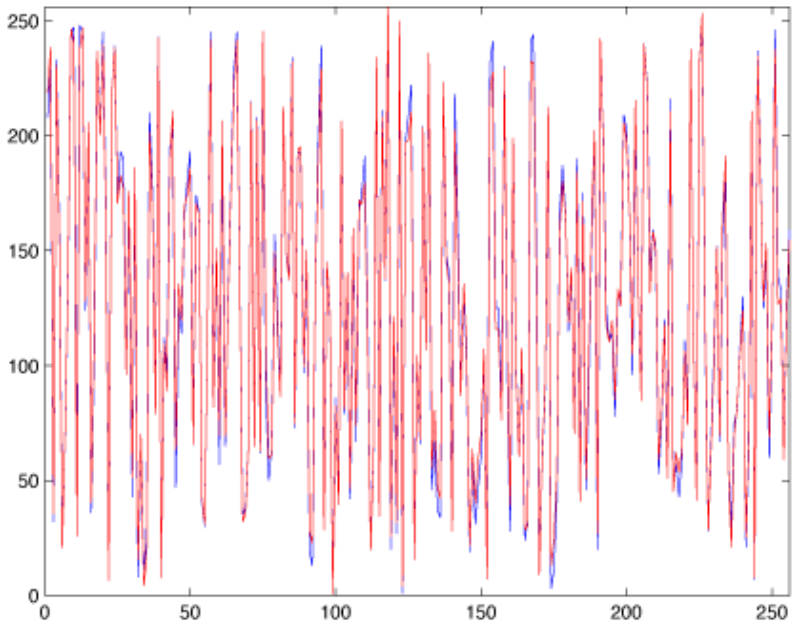

Figure 12: Approximation of the CG solution, in red, to the original image, in blue. Tolerance 1e-4, N=16.

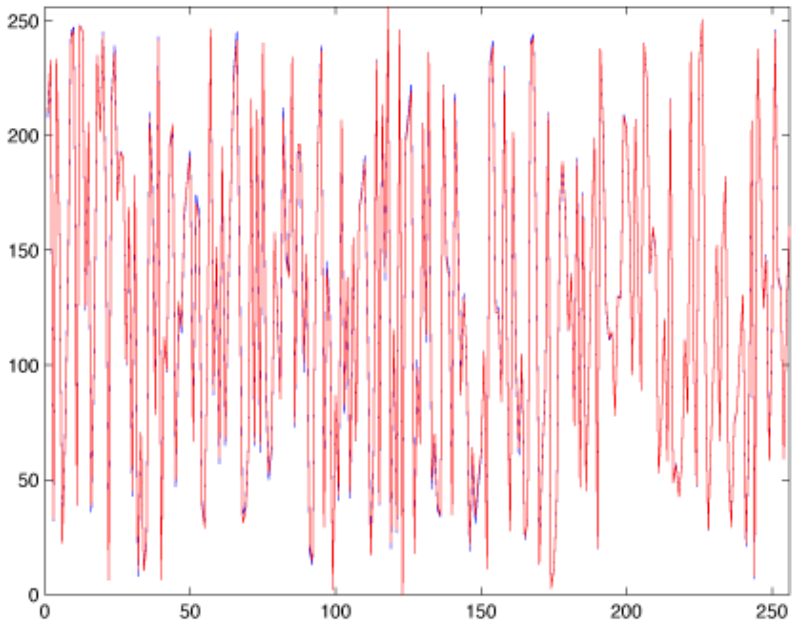

Figure 13: Approximation of the CG solution, in red, to the original image, in blue. Tolerance 1e-5, N=16.

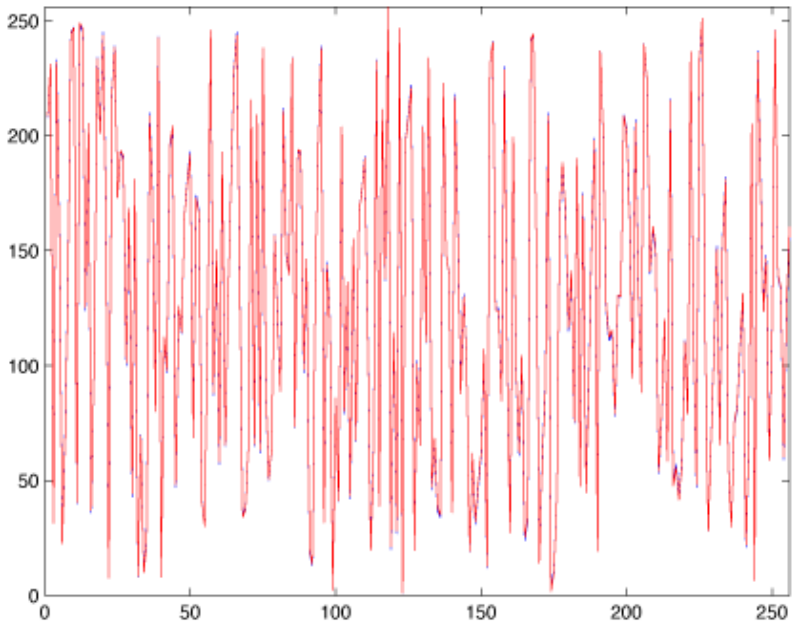

Figure 14: Approximation of the CG solution, in red, to the original image, in blue. Tolerance 1e-6, N=16.

The fourth set of experiments consisted of a similar setup as in the previous set, but this time the spectral data was obtain directly from the Fourier transform of the original image by virtue of the FFT and sinc_π functions as follows.

From equation 18, and setting $\sigma = l/N$ and $\gamma = m/N$, we get

$$\hat{f}\left(\frac{l}{N}, \frac{m}{N}\right) = \text{sinc}_\pi\left(\frac{l}{N}\right)\text{sinc}_\pi\left(\frac{m}{N}\right)\left(\sum_j f_j e^{-2\pi i k_j \frac{l}{N}} e^{-2\pi i l_j \frac{m}{N}}\right), \tag{35}$$

where N is the image resolution, assuming the image is square, and $l, m = 0, \ldots, N-1$. It is easy to identify the right hand side of equation 35 as an entry in the matrix that results from the entry-wise product of the FFT2 matrix of the original image and a sinc_π matrix $S_\pi(N)$ defined as follows:

$$S_\pi(N) = (\text{sinc}_\pi(0), \ldots, \text{sinc}_\pi(N-1))^T \times (\text{sinc}_\pi(0), \ldots, (\text{sinc}_\pi(N-1)), \tag{36}$$

then

$$\hat{f}\left(\frac{l}{N}, \frac{m}{N}\right) = \big(S_\pi(N) \odot \text{FFT2}(\text{Matrix form of } f)\big)_{m,l}, \tag{37}$$

where the subscript m, l in the right hand side denotes the element in row m and column l of the entry-wise matrix product. Note that m and l switched places. This is due to the definition of FFT2. The advantage of computing $\hat{f}$ this way is the considerable speedup that the FFT2 brings, compared to obtaining it by doing the product $\boldsymbol{Af}$. A one-dimensional example can be seen in the following figure

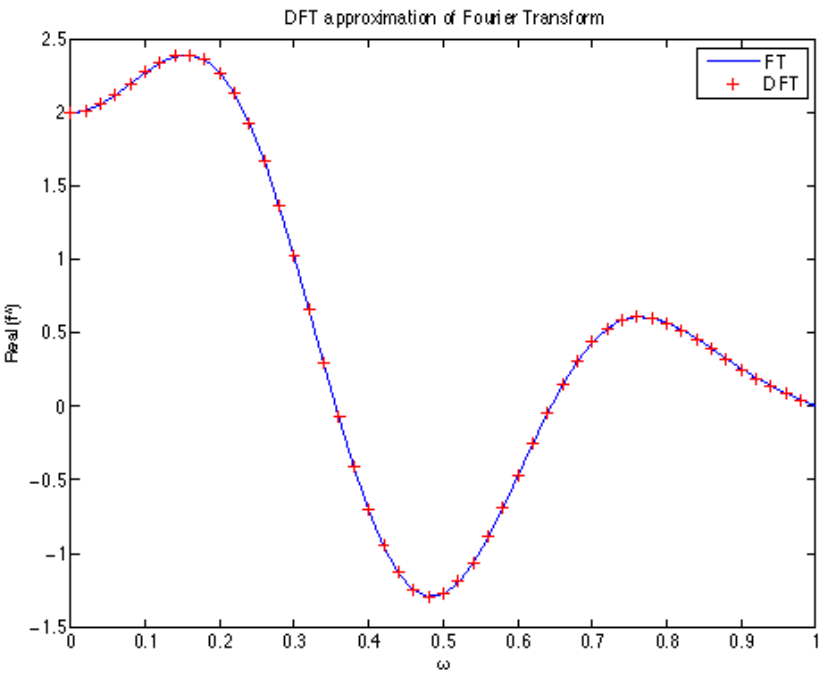

Figure 15: Fourier transform approximation by the DFT as implemented in the FFT of a piece-wise constant function. Note that even though the title and legend in the picture just mention the DFT, there is the entry-wise product by the respective sinc_π matrix that needs to be performed to compare with the analytic Fourier transform (FT), which was done here.

Coming back to the description of the results of the fourth set of experiments we repeated the set of experiments described in the third set of experiments, but this time, we used the results above to generate from the original image the spectral data set. Figures 16 to 19 show that the algorithms work in this case too.

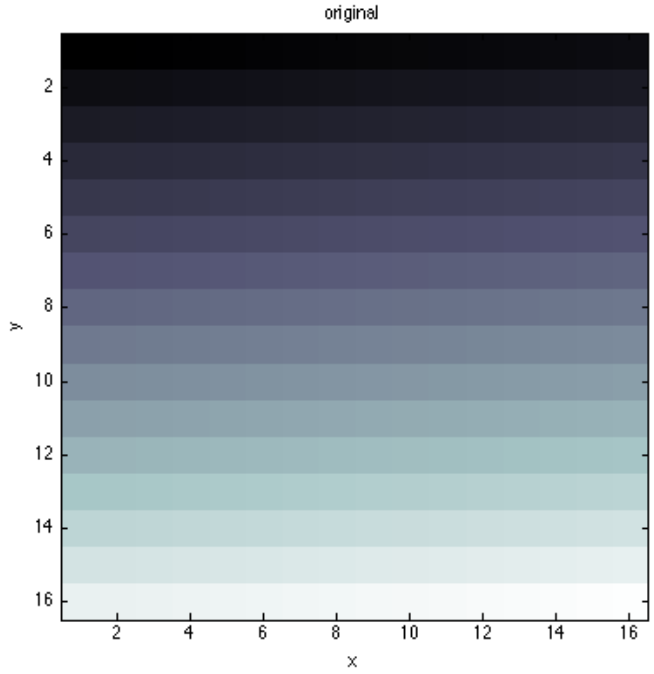

Figure 16: Original image to test the approximation of the Fourier transform by the $S_\pi \odot FFT2$ matrix. The resolution is 16 x 16.

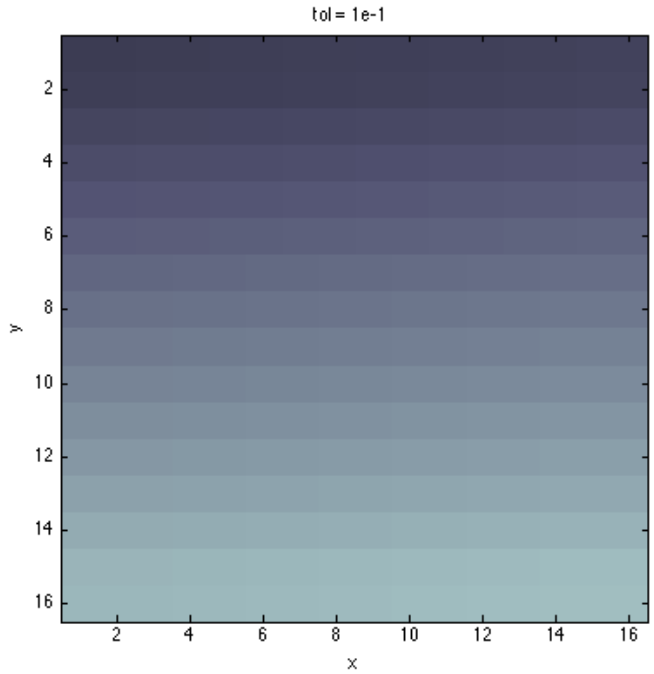

Figure 17: Reconstruction via Conjugate Gradient with a tolerance of 1e-1, N=16.

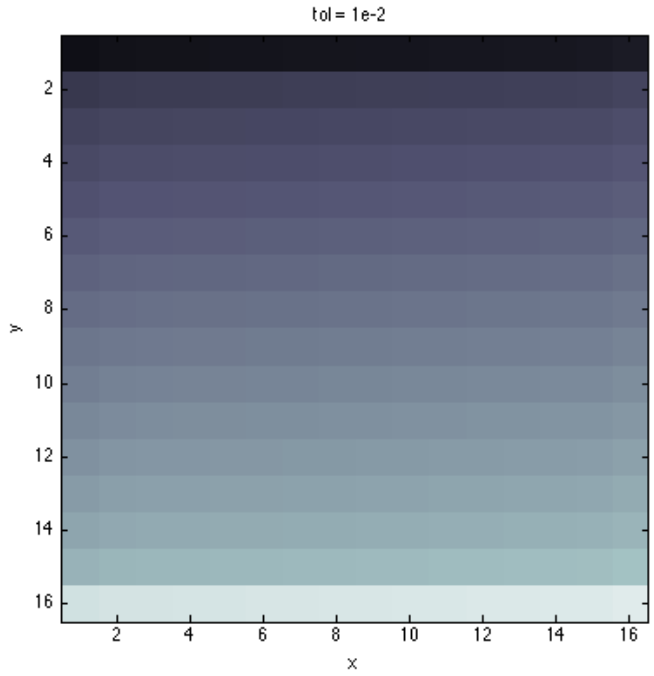

Figure 18: Reconstruction via Conjugate Gradient with a tolerance of 1e-2, N=16.

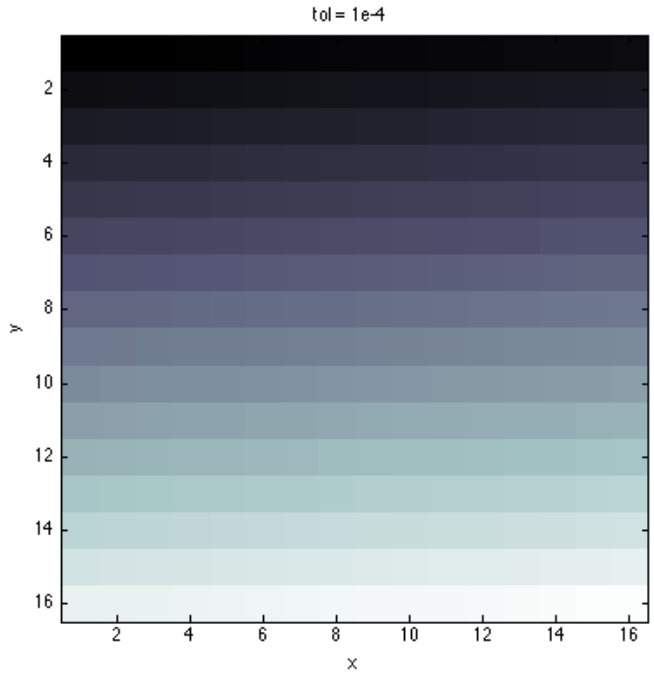

Figure 19: Reconstruction via Conjugate Gradient with a tolerance of 1e-4, N=16.

Finally, the fifth and last experiment we conducted takes a 16 x 16 image, pads it with zeros to the right and to the bottom to obtain a 32 x 32 image. Using this higher resolution image, we obtained its Fourier transform and sampled spectral data to form a 256 x 1 spectral data vector to feed into our CG algorithm and reconstruct a 16 by 16 image with it. The tolerance used for these experiments in the CG algorithm was 10^{-3}.

The original image was a "delta" function, that is, it is a function that is zero except at a point, where it takes the maximum value possible in our color map which is 255, in this case, given that the color map values range from 0 to 255. Figure 20 is the original image.

Figure 21 is the result of sampling the top left 16 x 16 sub-matrix of the spectral domain. This would correspond to all the lowest frequencies, or a low pass filter.

Figure 22 is the result of sampling the bottom right 16 x 16 sub-matrix of the spectral domain. This would correspond to all the highest frequencies, or a high pass filter.

Figure 23 this is similar to the previous two cases except that in this case we sampled every other frequency.

Finally, figure 24 is the result of random sampling on the spectral domain of the available frequency points.

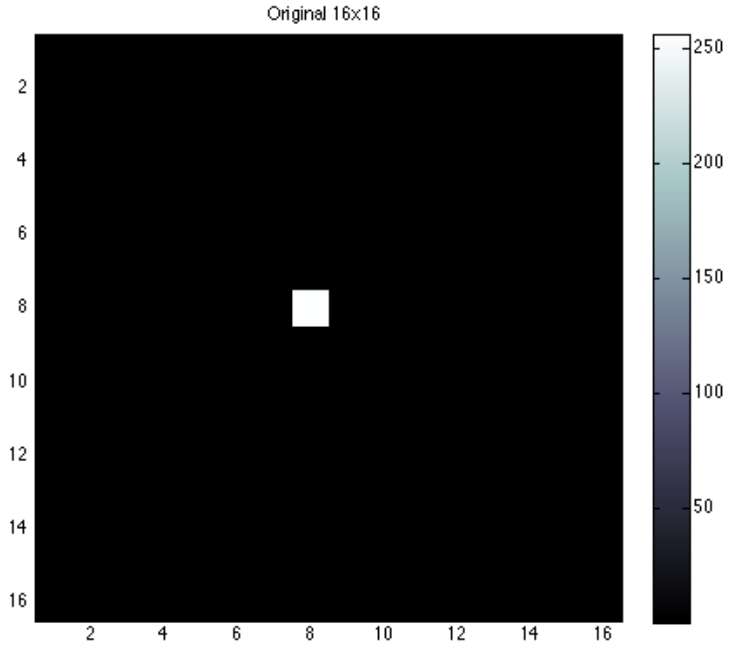

Figure 20: Original "delta" function for a 16 x 16 grid.

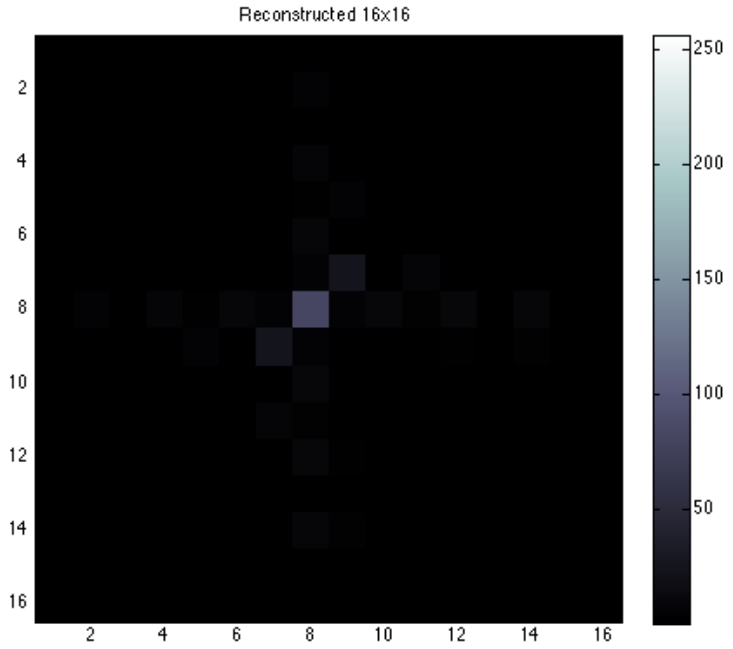

Figure 21: Reconstruction via Conjugate Gradient with a tolerance of 10^{-3}, N=16, low pass filter.

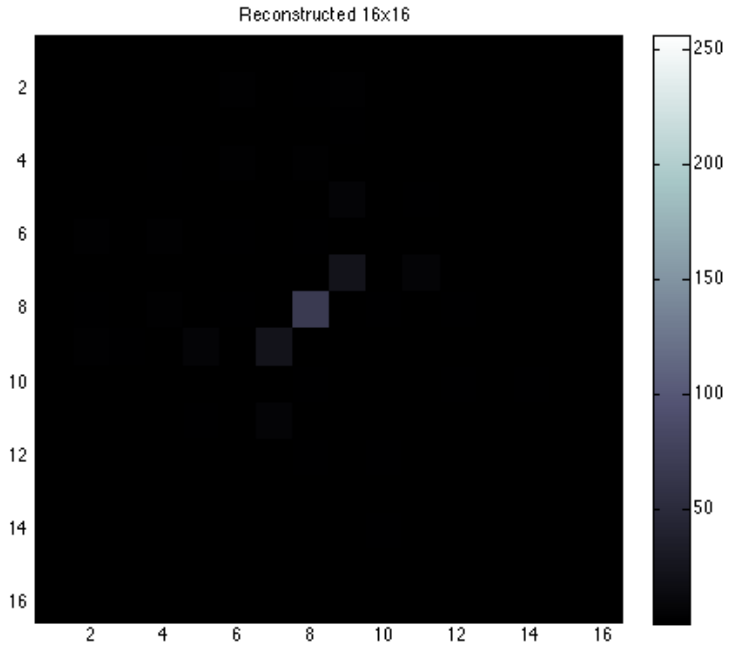

Figure 22: Reconstruction via Conjugate Gradient with a tolerance of 10^{-3}, N=16, high pass filter.

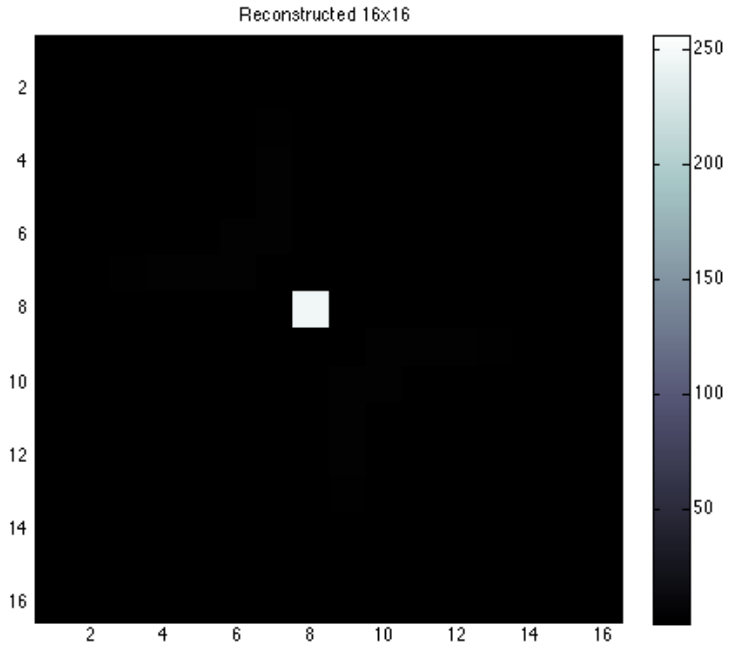

Figure 23: Reconstruction via Conjugate Gradient with a tolerance of 10^{-3}, N=16, every other frequency sampled.

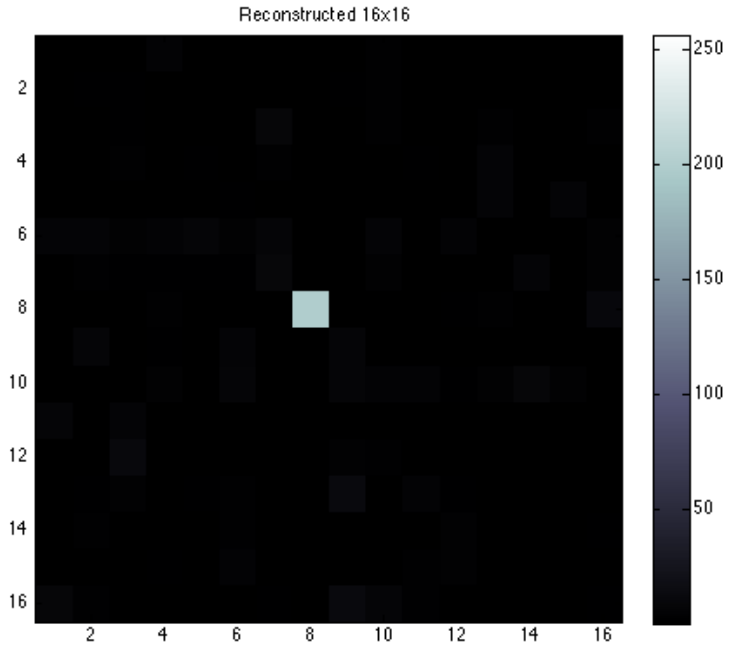

Figure 24: Reconstruction via Conjugate Gradient with a tolerance of 10^{-3}, N=16, random frequency sampling.

We observe that the best reconstruction is given by sampling regularly every other frequency, but that the second best reconstruction of the image is obtained by sampling randomly the spectral domain grid.

7 Performance results

We now present some performance and memory usage results for the experiments conducted with the Conjugate Gradient method.

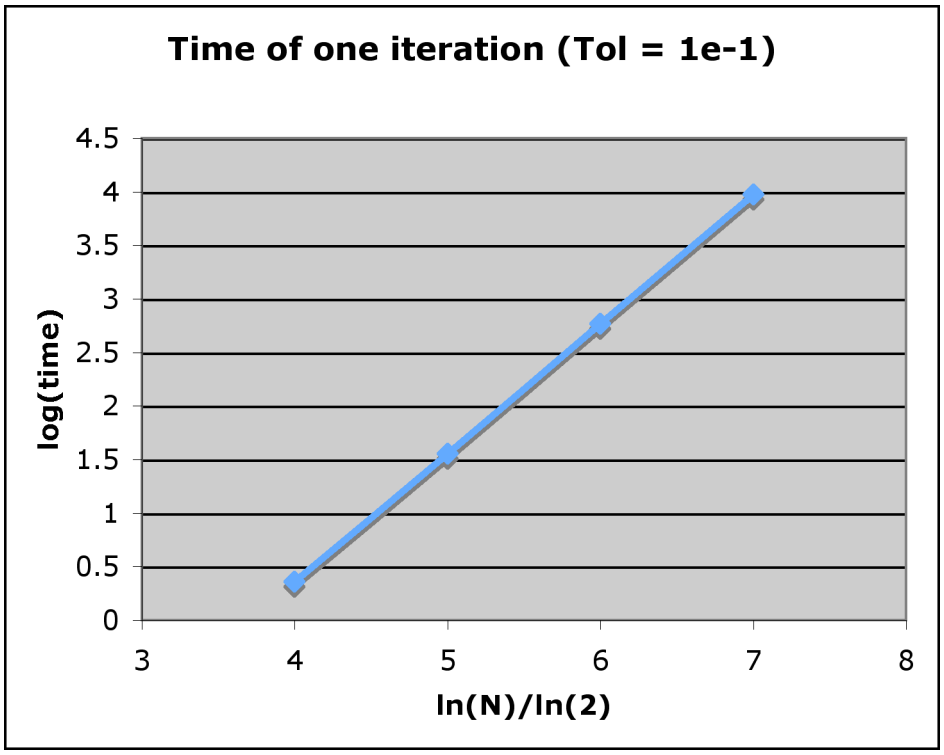

Figure 25: Time of one iteration vs image size.

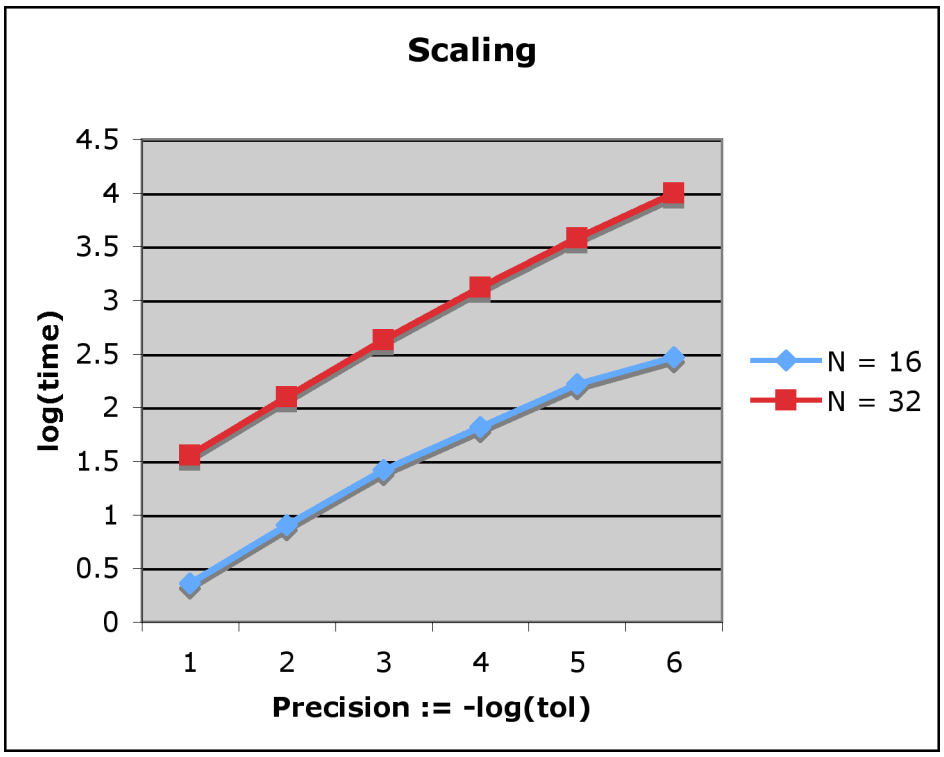

Figure 26: Runtime vs precision.

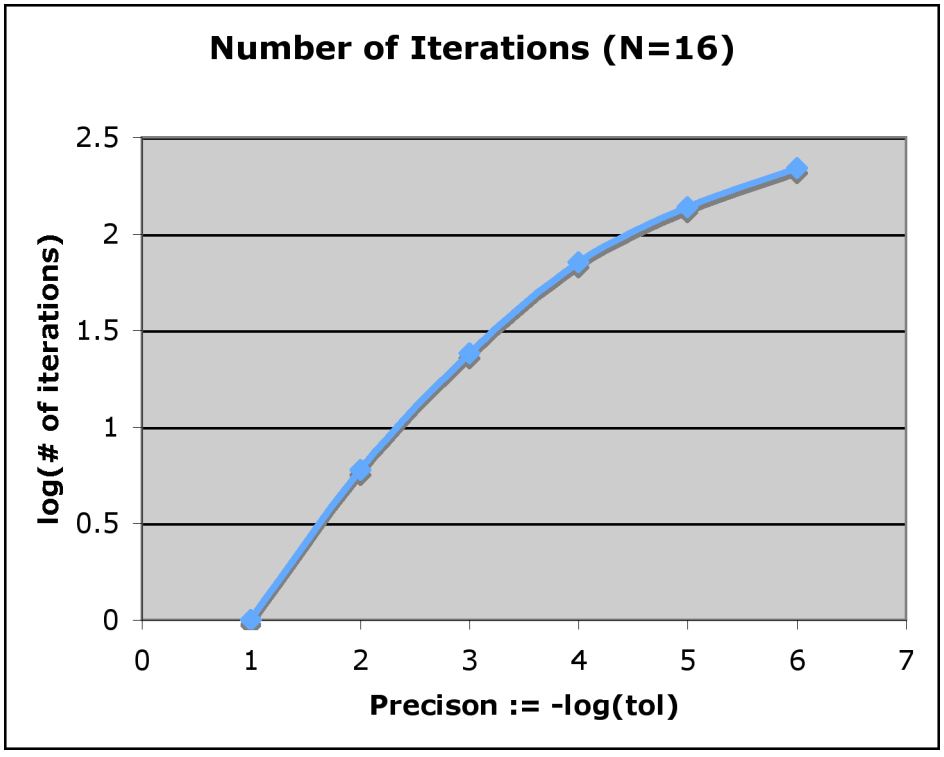

Figure 27: Number of iterations vs precision.

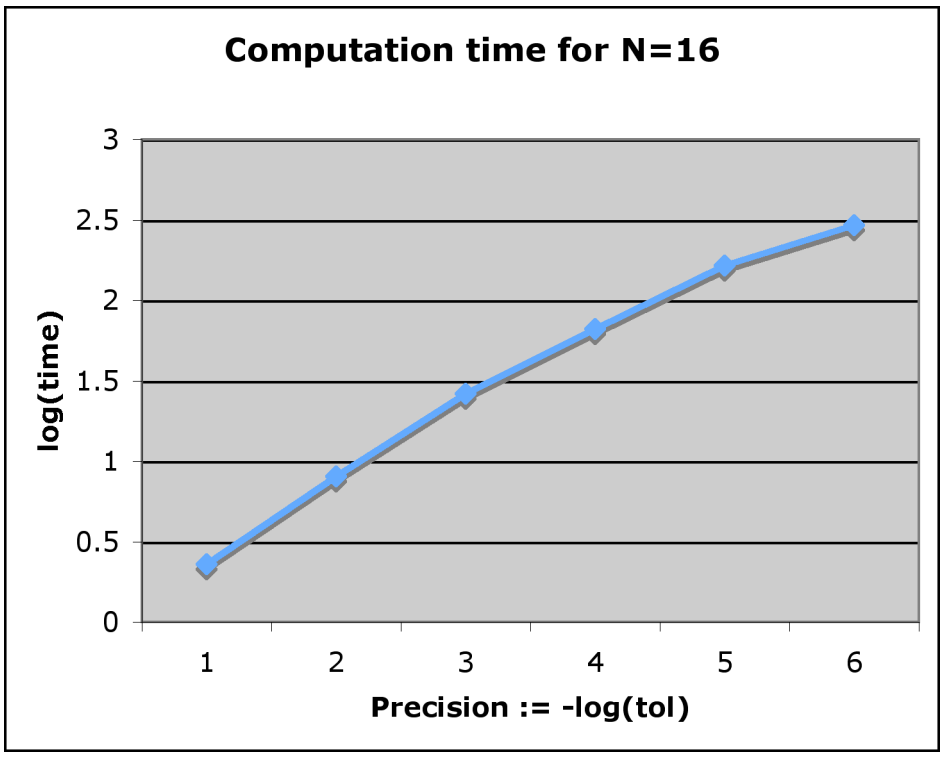

Figure 28: Computation time vs precision.

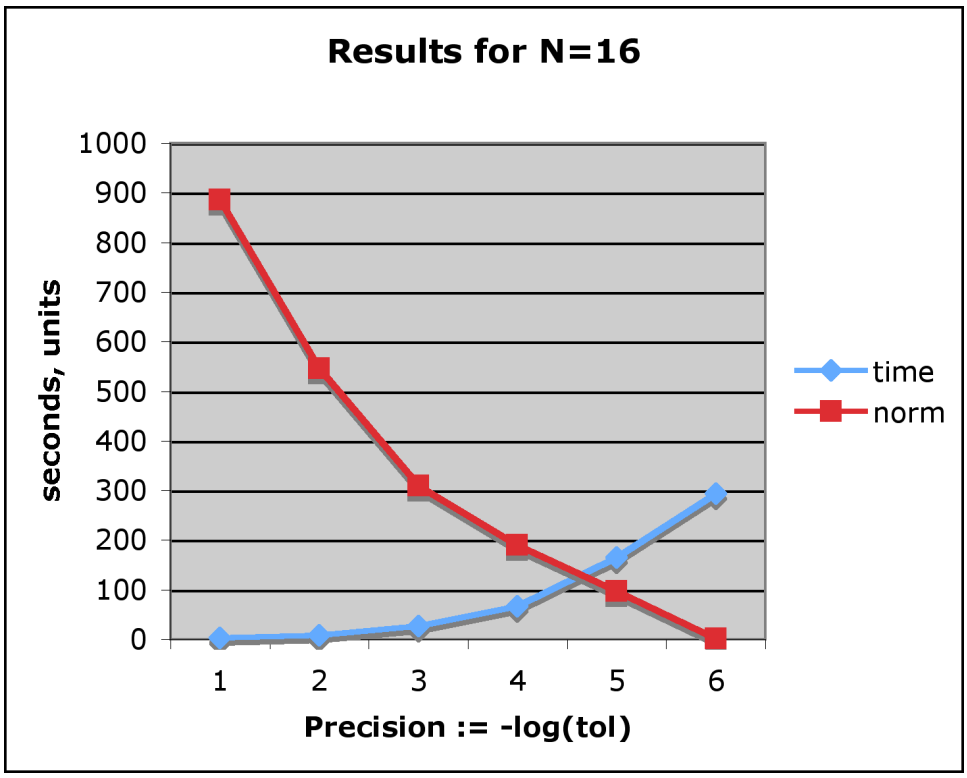

Figure 29: Combined convergence and time results.

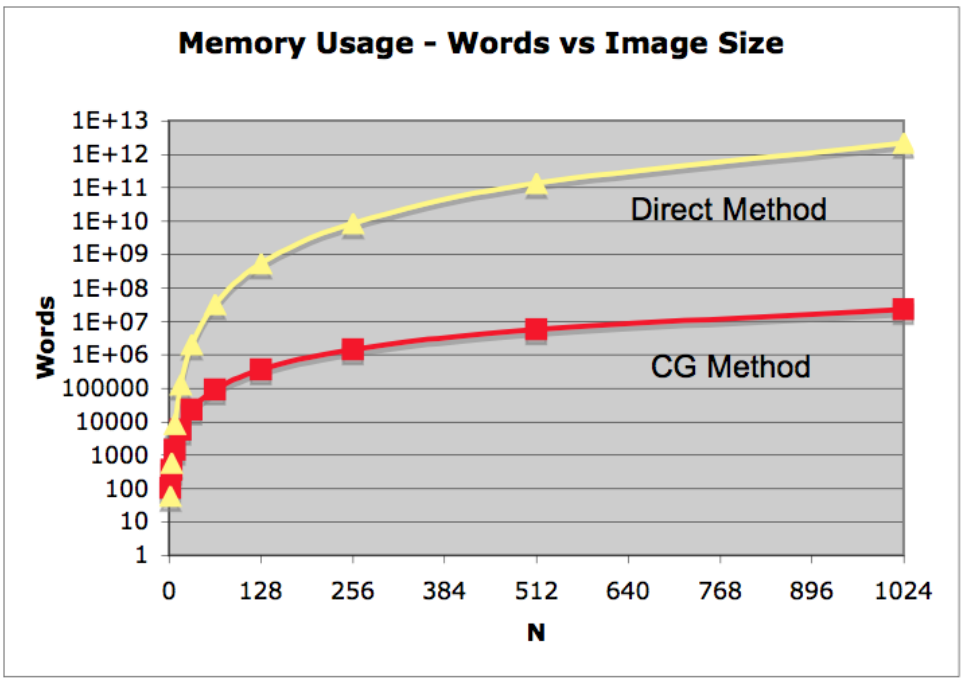

Figure 30: Memory usage.

8 Project debriefing

In this section we compare the objectives that we had set to achieve during the early planning and goal setting stages of this project with what was actually achieved.

The objective of this project was to implement the algorithms described in the previous sections, carry out the experiments for their validation, and test performance on a database.

We list the original points that needed to be tested or validated in *italics*, and we comment on the status after completion of the project shortly afterwards.

- *For the conjugate gradient method to work, we need to guarantee that* $\boldsymbol{A}^*\boldsymbol{A}$ *is positive definite. We need to research this fact.*

  This important theoretical requirement necessary for the existence of a solution to equation 24 was addressed in [1]. We would like to thank Radu Balan for pointing out this article to us. This article basically says that this system of equations has a solution with probability one for a large enough random sampling of the spectral domain.

- *We need to be able to compute in an efficient way the Fourier transform of the high resolution images that we'll use to create the data to run our experiments on. We need to study how to do this.*

  This is also something that we accomplished to do. The result was derived from studying the form of equation 18 and reflected in the comments surrounding equations 35, 36, and 37

- *We need to find a metric that measures how good our reconstruction is compared to the downsampled versions of the high resolution images. We will use the* ℓ^1

and ℓ^2 norms to measure the reconstruction error.

I have to thank Aleksey Zimin who suggested I talk to Christopher Miller about this issue given that Chris had done work regarding image processing. Chris suggested to take a look at [11]. Unfortunately we did not have a chance to implement the measures described in that reference to compare the goodness of the reconstructions described in our project. This will be in the list of things to do. We simply performed ℓ^2 norm of the residues and visual inspections for goodness of the reconstruction rating.

Our list of goals continued: "After this theoretical and practical details are worked through, we have to implement the conjugate gradient algorithm proposed. We will use Matlab for that purpose, and if time permits, port our code to C/C++."

We indeed produced Matlab code that implemented both the direct and Conjugate Gradient methods to test our theoretical framework. Unfortunately we ran out of time and the C/C++ implementations were not realized.

We further stated the following objectives for the code:

- *Show that the theoretical framework does work by writing code that can generate low resolution reconstructions*
- *Implement code that can generate higher resolution reconstructions*

  The experiments described in this paper worked for both low and high resolution images as expected for both the direct and Conjugate Gradient methods. However, the code as currently implemented is not efficient and more detail experiments for higher resolution images were very limited due to the computation times involved. The methodology described around equations 35, 36, and 37 will probably hold the key to efficiency.
- *Write code to measure how good the reconstructions are compared to the originals*

  This issue was addressed above.

9 Conclusions and future work

We successfully implemented and tested our methodology. We showed that both the direct and the Conjugate Gradient methods recreate adequately images from spectral data provided we use an adequate number of sample points in the spectral domain and that these points are either relatively uniformly-spaced, or sampled randomly.

We had success in recreating images at the same resolution as the original and using the full spectral information, or using partial spectral information on a finer grid, but we encountered some difficulties when sampling directly a high resolution image and then trying to obtain a lower resolution image from that information. We don't understand yet why this is so, and it is one of the issues to explore in the future.

We also managed to reconstruct an image of 128 x 128 pixels at a tolerance of 10^{-1}. The most important thing to mention is that the memory requirements for the CG gradient method made this possible, and this was one of the initial motivations of this project.

However, we also established that unless we do something to speed up the multiplication of $\boldsymbol{A}$ and a vector, or $\boldsymbol{A}^*$ and a vector, both used in our CG algorithm, the practical limitations that this imposes on our methodology would limit its usefulness

to the reconstruction of images at 64 x 64 pixels of resolution.This would be one of the areas to explore that would dramatically improve the results presented here.

Finally, we want to refer back to the items in the previous section that were not completed as paths to future work.

The experiments described herein were conducted on a MacBook Pro with a single 2.16 GHz Intel Core 2 Duo processor, with 2 GB 667 MHz DDR2 SDRAM, and running Matlab version 7.4.0 (R2007a).

I would like to express thanks to the instructors of AMSC 663/664 Radu Balan and Aleksey Zimin for their thoughtful and useful comments reflected in the contents of this work, and to my advisor John J. Benedetto.

References

[1] R. F. Bass and K. Gröchenig, *Random sampling of multivariate trigonometric polynomials*, SIAM Journal on Mathematical Analysis, 36 (2005), pp. 773–795.

[2] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses*, Springer-Verlag, 2003.

[3] J. J. Benedetto and P. J. S. G. Ferreira, *Modern Sampling Theory: Mathematics and Applications*, Birkhäuser, 2001.

[4] J. W. Cooley, *The rediscovery of the Fast Fourier Transform algorithm*, Mikrochim. Acta [Wien], III (1987), pp. 33–45.

[5] J. W. Cooley and J. W. Tukey, *An algorithm for the machine computation of complex fourier series*, Math. Comp., 19 (1965), pp. 297–301.

[6] A. Dutt and V. Rokhlin, *Fast Fourier Transforms for nonequispaced data, II*, Applied and Computational Harmonic Analysis, 2 (1995), pp. 85–100.

[7] E. H. Moore, *On the reciprocal of the general algebraic matrix*, Bulletin of the American Mathematical Society, 26 (1920), pp. 394–395.

[8] D. P. O'Leary, *Scientific computing with case studies.* Book in preparation for publication, 2008.

[9] R. Penrose, *On best approximate solution of linear matrix equations*, Proceedings of the Cambridge Philosophical Society, 52 (1956), pp. 17–19.

[10] J. R. Shewchuk, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, 1/4 ed., August 1994.

[11] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: From error measurement to structural similarity*, IEEE Transactions on Image Processing, 13 (2004), pp. 1–14.

[12] Wikipedia. http://en.wikipedia.org/wiki/Conjugate_gradient_method.