

# Image Reconstruction from Non-Uniformly Sampled Spectral Data

Alfredo Nava-Tudela

November 3, 2008

## 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, and explore C and/or C++ code that parallelizes these algorithms.

## 1 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 [3, 4], 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 [5].

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.

## 2 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 [2]. 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?

## 2.1 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.

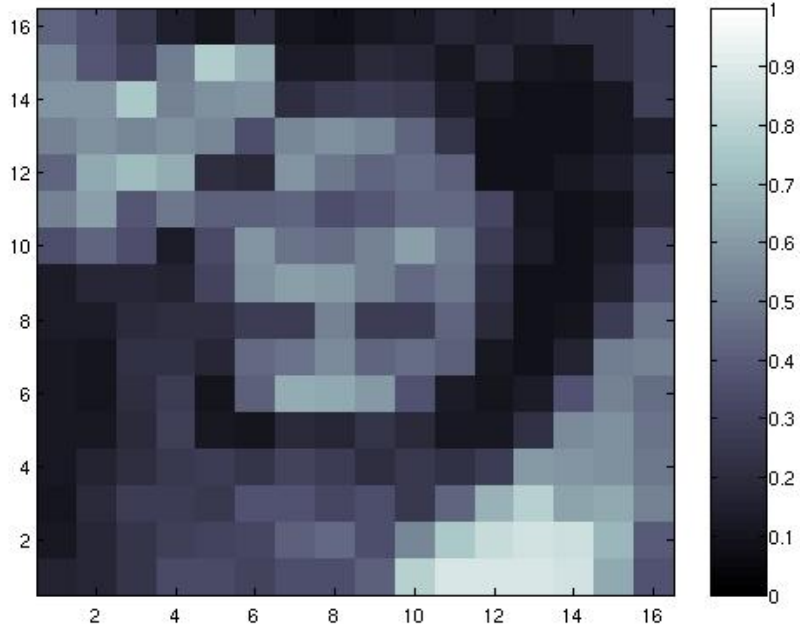


Figure 1: An image represented by the sum of step wise functions.

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:

$$f(x, y) = \sum_{k,l} f_{k,l} \chi_{k,l}(x, y), \quad (1)$$

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

$$\chi_{k,l} = \begin{cases} 1 & \text{if } (x, y) \in \square_{k,l}, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

## 2.2 Fourier transform of an image

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

$$\hat{f}(\sigma, \gamma) = \int_{\mathbb{R}^2} f(x, y) e^{-2\pi i(x\sigma + y\gamma)} dx dy. \quad (3)$$

If we consider functions  $f$  as defined in 2.1, we then obtain

$$\begin{aligned}
\hat{f}(\sigma, \gamma) &= \int_{\mathbb{R}^2} \sum_{k,l} f_{k,l} \chi_{k,l}(x, y) e^{-2\pi i(x\sigma + y\gamma)} dx dy \\
&= \sum_{k,l} \int_{\mathbb{R}^2} f_{k,l} \chi_{k,l}(x, y) e^{-2\pi i(x\sigma + y\gamma)} dx dy \\
&= \sum_{k,l} \int_l^{l+1} \int_k^{k+1} f_{k,l} e^{-2\pi i(x\sigma + y\gamma)} dx dy \\
&= \sum_{k,l} f_{k,l} \int_k^{k+1} e^{-2\pi i x \sigma} dx \int_l^{l+1} e^{-2\pi i y \gamma} dy,
\end{aligned} \tag{4}$$

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

$$\Pi(x) = \begin{cases} 1 & \text{if } |x| < \frac{1}{2}, \\ \frac{1}{2} & \text{if } |x| = \frac{1}{2}, \\ 0 & \text{if } |x| > \frac{1}{2}. \end{cases} \tag{5}$$

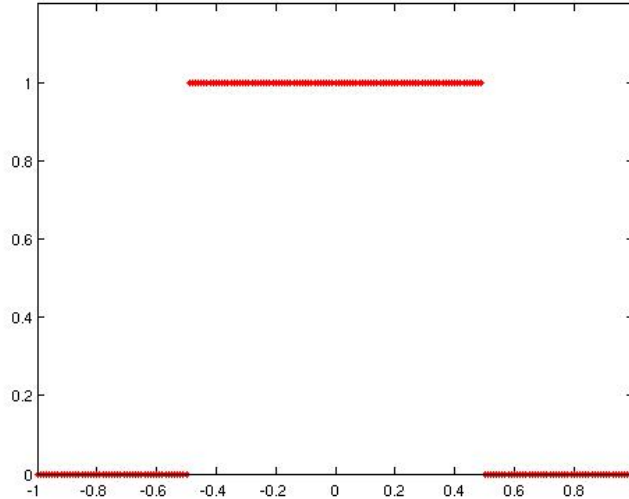


Figure 2: Graph of the rectangle function  $\Pi$ .

Lets compute  $\hat{\Pi}$  as this will help simplify further equation 4:

$$\begin{aligned}\widehat{\Pi}(\gamma) &= \int_{-\infty}^{\infty} \Pi(x) e^{-2\pi i x \gamma} dx \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i x \gamma} dx\end{aligned}\tag{6}$$

$$\begin{aligned}&= \frac{1}{-2\pi i \gamma} e^{-2\pi i x \gamma} \Big|_{x=-\frac{1}{2}}^{\frac{1}{2}} \\ &= \frac{1}{-2\pi i \gamma} (e^{-\pi i \gamma} - e^{\pi i \gamma}) \\ &= \frac{\sin(\pi \gamma)}{\pi \gamma} := \text{sinc}_{\pi}(\gamma).\end{aligned}\tag{7}$$

Note that the right hand side of equation 6 is remarkably similar to both integrals in the right hand side of equation 4. In fact they are identical modulo a translation by  $k + \frac{1}{2}$  and  $l + \frac{1}{2}$  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 4 we obtain:

$$\begin{aligned}\hat{f}(\sigma, \gamma) &= \sum_{k,l} f_{k,l} \int_k^{k+1} e^{-2\pi i x \sigma} dx \int_l^{l+1} e^{-2\pi i y \gamma} dy \\ &= \sum_{k,l} f_{k,l} \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i (x'+k+\frac{1}{2})\sigma} dx' \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i (y'+l+\frac{1}{2})\gamma} dy' \\ &= \sum_{k,l} f_{k,l} \left( e^{-2\pi i (k+\frac{1}{2})\sigma} \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i x' \sigma} dx' \right) \left( e^{-2\pi i (l+\frac{1}{2})\gamma} \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-2\pi i y' \gamma} dy' \right) \\ &= \sum_{k,l} f_{k,l} e^{-2\pi i (k+\frac{1}{2})\sigma} \text{sinc}_{\pi}(\sigma) e^{-2\pi i (l+\frac{1}{2})\gamma} \text{sinc}_{\pi}(\gamma).\end{aligned}\tag{8}$$

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

$$f(x, y) = \sum_{k,l} f_{k,l} \chi_{k,l}(x, y),\tag{9}$$

then

$$\begin{aligned}\hat{f}(\sigma, \gamma) &= \sum_{k,l} f_{k,l} e^{-2\pi i (k+\frac{1}{2})\sigma} \text{sinc}_{\pi}(\sigma) e^{-2\pi i (l+\frac{1}{2})\gamma} \text{sinc}_{\pi}(\gamma) \\ &= \left( \sum_{k,l} f_{k,l} e^{-2\pi i k \sigma} e^{-2\pi i l \gamma} \right) e^{-i\pi(\sigma+\gamma)} \text{sinc}_{\pi}(\sigma) \text{sinc}_{\pi}(\gamma),\end{aligned}\tag{10}$$

where  $k, l \in \{0, \dots, N-1\}$ ,  $f_{k,l} \in \mathbb{R}$ ,  $\chi_{k,l}(x, y)$  is the characteristic function of the square  $\square_{k,l} = [k, k+1) \times [l, l+1)$ , and  $\text{sinc}_{\pi}$  is as defined in equation 7.

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

$$e^{-2\pi i k(\sigma+m)} e^{-2\pi i l(\gamma+n)} = e^{-2\pi i k \sigma} e^{-2\pi i l \gamma},\tag{11}$$

and

$$e^{-\pi i(\sigma+m+\gamma+n)} = e^{-\pi i(\sigma+\gamma)} (-1)^{m+n},\tag{12}$$

we obtain from equation 10 that the quotient

$$\frac{\hat{f}(\sigma + m, \gamma + n)}{\hat{f}(\sigma, \gamma)} = (-1)^{m+n} \frac{\text{sinc}_\pi(\sigma + m) \text{sinc}_\pi(\gamma + n)}{\text{sinc}_\pi(\sigma) \text{sinc}_\pi(\gamma)}, \quad (13)$$

or, equivalently,

$$\hat{f}(\sigma + m, \gamma + n) = (-1)^{m+n} \frac{\text{sinc}_\pi(\sigma + m) \text{sinc}_\pi(\gamma + n)}{\text{sinc}_\pi(\sigma) \text{sinc}_\pi(\gamma)} \hat{f}(\sigma, \gamma). \quad (14)$$

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.

### 3 Theory for solution to the image reconstruction problem

Let us revisit the definition of an image from section 2.1. We had established that we could think of a black and white image as the sum of piecewise constant functions, as described in equation 1.

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

$$f(x, y) = \sum_j f_j \chi_j(x, y), \quad (15)$$

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

$$\hat{f}(\sigma, \gamma) = \sum_j f_j e^{-2\pi i(k_j + \frac{1}{2})\sigma} \text{sinc}_\pi(\sigma) e^{-2\pi i(l_j + \frac{1}{2})\gamma} \text{sinc}_\pi(\gamma), \quad (16)$$

or, equivalently,

$$\hat{f}(\sigma, \gamma) = \left( \sum_j f_j e^{-2\pi i k_j \sigma} e^{-2\pi i l_j \gamma} \right) e^{-i\pi(\sigma + \gamma)} \text{sinc}_\pi(\sigma) \text{sinc}_\pi(\gamma). \quad (17)$$

If we call for simplicity

$$a_j(\sigma, \gamma) = e^{-2\pi i k_j \sigma} e^{-2\pi i l_j \gamma} e^{-i\pi(\sigma + \gamma)} \text{sinc}_\pi(\sigma) \text{sinc}_\pi(\gamma), \quad (18)$$

then we can write equation 16 as

$$\hat{f}(\sigma, \gamma) = \sum_j f_j a_j(\sigma, \gamma). \quad (19)$$

Now assume that we have an image  $g \in L(\mathbb{R}^2) = \{f : \mathbb{R}^2 \rightarrow \mathbb{R} : \int_{\mathbb{R}^2} |f(x, y)| dx dy < \infty\}^1$  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 19— the following equality for all  $i \in \{0, \dots, M-1\}$ :

$$\hat{f}(\sigma_i, \gamma_i) = \sum_{j=0}^{N^2-1} f_j a_j(\sigma_i, \gamma_i) = b_i. \quad (20)$$

In matrix notation, this is equivalent to writing

---

<sup>1</sup>This condition guarantees that  $\hat{g}(\sigma, \gamma) = \int \int g(x, y) e^{-2\pi i(x\sigma + y\gamma)} dx dy$  exists.

$$\begin{pmatrix} a_0(\sigma_0, \gamma_0) & a_1(\sigma_0, \gamma_0) & \dots & a_{N^2-1}(\sigma_0, \gamma_0) \\ a_0(\sigma_1, \gamma_1) & a_1(\sigma_1, \gamma_1) & \dots & a_{N^2-1}(\sigma_1, \gamma_1) \\ \vdots & \vdots & & \vdots \\ a_0(\sigma_{M-1}, \gamma_{M-1}) & a_1(\sigma_{M-1}, \gamma_{M-1}) & \dots & a_{N^2-1}(\sigma_{M-1}, \gamma_{M-1}) \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N^2-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{M-1} \end{pmatrix} \quad (21)$$

or, equivalently,

$$\mathbf{A}\mathbf{f} = \mathbf{b}, \quad (22)$$

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

If we had  $M = N^2$ , the image reconstruction problem would boil down to solving the linear system of equations 22. However, we will most likely have  $M \geq N^2$ , and system 22 will be overdetermined.

In this case, we proceed by premultiplying equation 22 by  $\mathbf{A}^*$ , the conjugate transpose of  $\mathbf{A}$ . The linear system of equations then becomes

$$(\mathbf{A}^*\mathbf{A})\mathbf{f} = \mathbf{A}^*\mathbf{b}. \quad (23)$$

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

$$\mathbf{f} = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{b}. \quad (24)$$

It turns out that if the columns of  $\mathbf{A}$  are linearly independent, then  $\mathbf{A}^*\mathbf{A}$  is invertible [1].

The matrix  $(\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*$  is called the Moore-Penrose pseudoinverse of  $\mathbf{A}$ , and it will give the solution to equation 22 in the least squares sense [6, 8], that is

$$\|\mathbf{A}\mathbf{f} - \mathbf{b}\|^2 = \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2. \quad (25)$$

## 4 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  $\mathbf{A}$  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  $\mathbf{A}$ , or  $\mathbf{A}^*\mathbf{A}$ , explicitly.

### 4.1 The Singular Value Decomposition, insights and results

Recall from equation 25 that  $\mathbf{f}$  is the solution of minimum norm to the least squares problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2. \quad (26)$$

It can be proven that a matrix  $\mathbf{A}$  of dimensions  $M$  by  $N^2$  can be written as  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ , where  $\mathbf{U}$  has dimensions  $M$  by  $M$ ,  $\mathbf{V}$  has dimensions  $N^2$  by  $N^2$ , both  $\mathbf{U}$  and  $\mathbf{V}$  are unitary matrices, and  $\mathbf{\Sigma}$  is an  $M$  by  $N^2$  matrix having only non-zero elements in the main diagonal and they are non-negative real numbers  $\sigma_1 \geq \sigma_2 \dots \geq \sigma_{N^2} \geq 0$ , called the singular values of  $\mathbf{A}$  [7].

Observe that

$$\begin{aligned}
\mathbf{A}^* \mathbf{A} &= (\mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^*)^* \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^* \\
&= (\mathbf{V} \boldsymbol{\Sigma} \mathbf{U}^*) \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^* \\
&= \mathbf{V} \boldsymbol{\Sigma}^2 \mathbf{V}^*.
\end{aligned} \tag{27}$$

Since  $\mathbf{V}$  is a unitary matrix, the right hand side of equation 27 is a similarity transformation, therefore the eigenvalues of  $\mathbf{A}^* \mathbf{A}$  coincide with those of  $\boldsymbol{\Sigma}^2$ , which are exactly  $\sigma_1^2 \geq \sigma_2^2 \dots \geq \sigma_{N^2}^2 \geq 0$ . Hence,  $\mathbf{A}^* \mathbf{A}$  is positive semidefinite.

Now, back to our minimization problem given by equation 26. If we call  $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$ , then

$$\begin{aligned}
\|\mathbf{r}\|^2 &= \mathbf{r}^* \mathbf{r} \\
&= (\mathbf{U}^* \mathbf{r})^* (\mathbf{U}^* \mathbf{r}) \\
&= \|\mathbf{U}^* \mathbf{r}\|^2 \\
&= \|\mathbf{U}^* \mathbf{A}\mathbf{x} - \mathbf{U}^* \mathbf{b}\|^2 \\
&= \|\boldsymbol{\Sigma} \mathbf{V}^* \mathbf{x} - \mathbf{c}\|^2,
\end{aligned} \tag{28}$$

where  $c_j = \mathbf{u}_j^* \mathbf{b}$ ,  $j = 1, \dots, M$  and  $\mathbf{u}_j$  is the  $j^{\text{th}}$  column of  $\mathbf{U}$ . Now, by letting  $\mathbf{w} = \mathbf{V}^* \mathbf{x}$ , equation 28 is equivalent to

$$\|\mathbf{r}\|^2 = (\sigma_1 w_1 - c_1)^2 + \dots + (\sigma_n w_n - c_n)^2 + c_{n+1}^2 + \dots + c_m^2. \tag{29}$$

Therefore, minimizing the norm of  $\mathbf{r}$  is equivalent to minimizing the right hand side of 29. This gives the following algorithm to solve equation 26.

1. Compute  $\mathbf{c} = \mathbf{U}^* \mathbf{b}$
2. Let  $p$  be the number of nonzero singular values of  $\mathbf{A}$
3. for  $j = 1, \dots, p$ 
  - Set  $w_j = c_j / \sigma_j$
  - end
4. The minimum norm solution is  $\mathbf{x} = \mathbf{V}(:, 1 : p) \mathbf{w}$   
The norm of the residual is  $(c_{p+1} + \dots + c_m)^{1/2}$

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  $\mathbf{A}$  and storage of  $\mathbf{U}$  and  $\mathbf{V}$ , which can be very large and dense. Therefore, we need a low storage algorithm.

## 4.2 Conjugate gradient method, a low storage solution

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

$$\mathbf{v}_j^* \mathbf{B} \mathbf{v}_k = 0 \quad \text{if and only if} \quad j \neq k. \tag{30}$$

Assume that we are given an  $\mathbf{A}^* \mathbf{A}$ -conjugate set  $\mathcal{V}$ , and that we want to solve equation 23. Consider  $\mathbf{f}$  of the form

$$\mathbf{f} = \sum_{j=1}^n \alpha_j \mathbf{v}_j, \quad \text{with } \mathbf{v}_j \in \mathcal{V}, \tag{31}$$

then, substituting  $\mathbf{f}$  into 23 and premultiplying by  $\mathbf{v}_k^*$  both sides of that equation we obtain

$$\begin{aligned}
\mathbf{v}_k^* \mathbf{A}^* \mathbf{A} \mathbf{f} &= \mathbf{v}_k^* \mathbf{A}^* \mathbf{b} \\
\mathbf{v}_k^* \mathbf{A}^* \mathbf{A} \sum_{j=1}^n \alpha_j \mathbf{v}_j &= \mathbf{v}_k^* \mathbf{A}^* \mathbf{b} \\
\sum_{j=1}^n \alpha_j \mathbf{v}_k^* \mathbf{A}^* \mathbf{A} \mathbf{v}_j &= \mathbf{v}_k^* \mathbf{A}^* \mathbf{b} \\
\alpha_k \mathbf{v}_k^* \mathbf{A}^* \mathbf{A} \mathbf{v}_k &= \mathbf{v}_k^* \mathbf{A}^* \mathbf{b},
\end{aligned} \tag{32}$$

from which

$$\alpha_k = \frac{\mathbf{v}_k^* \mathbf{c}}{\mathbf{v}_k^* \mathbf{A}^* \mathbf{A} \mathbf{v}_k} \quad \text{for } k = 1, \dots, n, \tag{33}$$

where we have set  $\mathbf{c} = \mathbf{A}^* \mathbf{b}$ , and provided  $\mathbf{v}_k^* \mathbf{A}^* \mathbf{A} \mathbf{v}_k$  is not zero. This means that if we obtain an  $\mathbf{A}^* \mathbf{A}$ -conjugate set  $\mathcal{V}$  we can solve with 33 our least squares problem. This is the theoretical basis for the conjugate gradient algorithm. The algorithm basically minimizes  $(1/2) \mathbf{x}^* \mathbf{B} \mathbf{x} - \mathbf{x}^* \mathbf{c}$ , where  $\mathbf{B}$  is a Hermitian positive definite matrix. In our case, based on the results of section 4.1, we have that  $\mathbf{A}^* \mathbf{A}$  is at least positive semidefinite. We will eventually prove definiteness. The algorithm has the advantage that we do not have to explicitly store  $\mathbf{A}^* \mathbf{A}$ , but be able to compute  $\mathbf{A}^* \mathbf{A} \mathbf{x}$  for arbitrary vectors  $\mathbf{x}$ . Consider equation 23, then the conjugate gradient method algorithm translates into

1. Given  $\mathbf{x}^{(0)}$ , form  $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}^* \mathbf{A} \mathbf{x}^{(0)}$ ,  $\mathbf{s}^{(0)} = \mathbf{r}^{(0)}$ .
2. for  $k = 0, 1, \dots$  until convergence
  - Let  $\mathbf{z}^{(k)} = \mathbf{A}^* \mathbf{A} \mathbf{s}^{(k)}$
  - Let the step length be  $\alpha^{(k)} = (\mathbf{r}^{(k)*} \mathbf{s}^{(k)}) / (\mathbf{s}^{(k)*} \mathbf{z}^{(k)})$
  - Let  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{s}^{(k)}$
  - Update the negative gradient  $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^{(k)} \mathbf{z}^{(k)}$
  - Let  $\beta^{(k+1)} = (\mathbf{r}^{(k+1)*} \mathbf{r}^{(k+1)}) / (\mathbf{r}^{(k)*} \mathbf{r}^{(k)})$
  - Let the new search direction be  $\mathbf{s}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta^{(k+1)} \mathbf{s}^{(k)}$

## 5 Experiments and validation

To test our model and algorithms, we will select ten different high resolution black and white images from which we will sample spectral data and try to reconstruct from that data the original images at lower resolution.

The sampling will be on the unit square  $[0, 1) \times [0, 1)$  in the spectral domain, and we will use different sampling patterns: regular sampling, uniformly distributed random sampling, and spiral sampling.

We will compare the reconstruction to the downsampled originals, and choose a metric that measures how well the reconstruction performs.

We also have to complete some theoretical details such as verifying that  $\mathbf{A}^* \mathbf{A}$  is not singular.

## 6 Project management and timeline

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

To achieve these goals, there are a series of tasks that we need to complete. We describe them next, and assign a timeline for their completion. Also, we define the different deliverables that will measure the progress done during this project.



We had some details in the previous exposition that were left unresolved and that need to be addressed:

- For the conjugate gradient method to work, we need to guarantee that  $\mathbf{A}^*\mathbf{A}$  is positive definite. We need to research this fact.
- 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.
- 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  $\ell^1$  and  $\ell^2$  norms to measure the reconstruction error.

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++. To this end we will:

- Show that the theoretical framework does work by writing code that can generate low resolution reconstructions
- Implement code that can generate higher resolution reconstructions
- Write code to measure how good the reconstructions are compared to the originals

## 6.1 Milestones

- Need to select images to run experiments on
  - Establish criteria that images must meet
  - Select images based on those criteria
- Work theoretical aspects mentioned above (extra)
- Write code to import images using Matlab
- Write code to compute Fourier transform of an image at  $(\sigma, \gamma)$  in the spectral domain
  - Write theory
  - Matlab implementation
- Create data set
  - Establish sample point sets
- Test theory with low resolution reconstruction using the SVD algorithm using Matlab
- Write conjugate gradient algorithm code for high resolution reconstructions in Matlab
- Validate the implementation
- Test on database for several densities of frequency sampling points

## 6.2 Deliverables

- Matlab code and documentation
- All theoretical work carried out to implement code
- Table and plot of results

## References

- [1] Adi Ben-Israel and Thomas N. E. Greville. *Generalized Inverses*. Springer-Verlag, 2003.
- [2] John J. Benedetto and Paulo J. S. G. Ferreira. *Modern Sampling Theory: Mathematics and Applications*. Birkhäuser, 2001.
- [3] J. W. Cooley and J. W. Tukey. An algorithm for the machine computation of complex fourier series. *Math. Comp.*, 19:297–301, 1965.
- [4] James W. Cooley. The rediscovery of the Fast Fourier Transform algorithm. *Mikrochim. Acta [Wien]*, III:33–45, 1987.
- [5] A. Dutt and V. Rokhlin. Fast Fourier Transforms for nonequispaced data, II. *Applied and Computational Harmonic Analysis*, 2:85–100, 1995.
- [6] E. H. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26:394–395, 1920.
- [7] Diane P. O’Leary. Scientific computing with case studies. Book in preparation for publication, 2008.
- [8] Roger Penrose. On best approximate solution of linear matrix equations. *Proceedings of the Cambridge Philosophical Society*, 52:17–19, 1956.