Notes on reconstructing the data cube in hyperspectral image processing

Charles Byrne (Charles_Byrne@uml.edu),
Department of Mathematical Sciences,
University of Massachusetts Lowell, Lowell, MA 01854

May 3, 2004

Abstract

The hyperspectral imaging technique described in [12] leads to the interesting problem of reconstructing a three-dimensional *data cube* from measured data. This problem involves three separate steps in which we must estimate values of a function from values of its Fourier transform. Depending on which of the two functions involved at each step has bounded support, that is, is zero off of a bounded set, the estimation process can take one of two forms, which we call Type One and Type Two problems. We discuss these two types and suggest techniques for solving both of them. For Type Two problems there is a good opportunity to incorporate whatever prior information we may have about the shape of the function being reconstructed. Since the data sets are large we recommend the use of iterative methods, such as the *algebraic reconstruction technique* (ART).

1 Introduction

Hyperspectral image processing provides an excellent example of the need for estimating Fourier transform values from limited data. In these notes we describe one novel approach, due to Mooney *et al* [12]; the presentation here follows [1, 13, 10].

In this hyperspectral imaging problem the electromagnetic energy reflected or emitted by a point, such as light reflected from a location on the earth's surface, is passed through a prism to separate the components as to their wavelengths. Due to the dispersion of the different frequency components caused by the prism, these components are recorded in the image plane not at a single spatial location, but at distinct points along a line. Since the received energy comes from a region of points, not a single point, what is received in the image plane is a superposition of different wavelength components associated with different points within the object. The first

task is to reorganize the data so that each location in the image plane is associated with all the components of a single point of the object being imaged; this is a Fourier transform estimation problem, which we can solve using band-limited extrapolation. The coordinate systems are chosen so that the points of the image plane are in one-to-one correspondence with points of the object plane.

The image reconstruction problem we face involves the estimation of values of a function from values of its Fourier transform. At one stage of the reconstruction the function being estimated is zero outside a bounded set, while at another stage the function being sampled is zero outside a bounded set. For concreteness we let $F(\omega)$ be defined for $\omega \in [0, 2\pi]$, with

$$f(x) = \frac{1}{2\pi} \int_0^{2\pi} F(\omega) e^{-ix\omega} d\omega. \tag{1.1}$$

We shall say that estimating values of $f(x)$ from finitely many samples of the function $F(\omega)$ is a problem of Type One, while estimating $F(\omega)$ from finitely many samples of the function $f(x)$ is a problem of Type Two. These two types of estimation problems are quite distinct and, as we shall see, require different techniques for their solution.

2 Hyperspectral Imaging

The region of the object plane that we wish to image is described by the two-dimensional spatial coordinate $\mathbf{x} = (x_1, x_2)$. For simplicity, we take these coordinates to be continuous, leaving until the end the issue of discretization. We shall also denote by $\mathbf{x}$ the point in the image plane corresponding to the point $\mathbf{x}$ on the object plane; the units of distance between two such points in one plane and their corresponding points in the other plane may, of course, be quite different. For each $\mathbf{x}$ we let $F(\mathbf{x}, \lambda)$ denote the intensity of the component at wavelength λ of the electromagnetic energy that is reflected from or emitted by location $\mathbf{x}$. The function $F(\mathbf{x}, \lambda)$ (or, rather, a discretized version of it) is the *data cube* we wish to reconstruct. We shall assume that $F(\mathbf{x}, \lambda) = 0$ for $(\mathbf{x}, \lambda)$ outside some bounded portion of three-dimensional space.

Consider, for a moment, the case in which the energy sensed by the imaging system comes from a single point $\mathbf{x}$. If the dispersion axis of the prism is oriented according to the unit vector $\mathbf{p}_\theta$, for some $\theta \in [0, 2\pi)$, then the component at wavelength λ of the energy from $\mathbf{x}$ on the object is recorded not at $\mathbf{x}$ in the image plane but at the point $\mathbf{x} + \mu(\lambda - \lambda_0)\mathbf{p}_\theta$. Here $\mu > 0$ is a constant and λ_0 is the wavelength for which the component from point $\mathbf{x}$ of the object is recorded at $\mathbf{x}$ in the image plane.

Now imagine energy coming to the imaging system for all the points within the

imaged region of the object. Let $G(\mathbf{x}, \theta)$ be the intensity of the energy received at location $\mathbf{x}$ in the image plane when the prism orientation is θ. It follows from above that

$$G(\mathbf{x}, \theta) = \int_{-\infty}^{+\infty} F(\mathbf{x} - \mu(\lambda - \lambda_0)\mathbf{p}_\theta, \lambda) d\lambda. \tag{2.1}$$

The limits of integration are not really infinite due to the finiteness of the aperture and the focal plane of the imaging system. Our data will consist of finitely many values of $G(\mathbf{x}, \theta)$, as $\mathbf{x}$ varies over the grid points of the image plane and θ varies over some finite discretized set of angles.

We explore the relationship between the measurements and the desired data cube by taking the two-dimensional inverse Fourier transform of $G(\mathbf{x}, \theta)$ with respect to the spatial variable $\mathbf{x}$ to get

$$g(\mathbf{y}, \theta) = \frac{1}{(2\pi)^2} \int G(\mathbf{x}, \theta) \exp(-i\mathbf{x} \cdot \mathbf{y}) d\mathbf{x}. \tag{2.2}$$

Inserting the expression for G in equation (2.1) into equation (2.2) we obtain

$$g(\mathbf{y}, \theta) = \exp(i\mu\lambda_0 \mathbf{p}_\theta \cdot \mathbf{y}) \int \exp(-i\mu\lambda \mathbf{p}_\theta \cdot \mathbf{y}) f(\mathbf{y}, \lambda) d\lambda, \tag{2.3}$$

where $f(\mathbf{y}, \lambda)$ is the two-dimensional inverse Fourier transform of $F(\mathbf{x}, \lambda)$ with respect to the spatial variable $\mathbf{x}$. Therefore

$$g(\mathbf{y}, \theta) = \exp(i\mu\lambda_0 \mathbf{p}_\theta \cdot \mathbf{y}) \mathcal{F}(\mathbf{y}, \gamma_\theta), \tag{2.4}$$

where $\mathcal{F}(\mathbf{y}, \gamma)$ denotes the three-dimensional inverse Fourier transform of $F(\mathbf{x}, \lambda)$ and $\gamma_\theta = \mu \mathbf{p}_\theta \cdot \mathbf{y}$. We see then that each value of $g(\mathbf{y}, \theta)$ that we estimate from our measurements provides us with a single estimated value of $\mathcal{F}$.

Our reconstruction involves three distinct Fourier transform estimation steps:

Step 1: We use the measured values of $G(\mathbf{x}, \theta)$ to estimate values of $g(\mathbf{y}, \theta)$ given by equation (2.2). Since the variable $\mathbf{x}$ is restricted to a bounded set this is a problem of Type One.

Step 2: For each fixed $\mathbf{y}$ we use the estimated values of $g(\mathbf{y}, \theta)$ to estimate values of the function $f(\mathbf{y}, \lambda)$, according to equation (2.3). Since the variable λ is restricted to a bounded set this is a problem of Type Two.

Step 3: From the estimated values of $f(\mathbf{y}, \lambda)$ we estimate its spatial Fourier transform, which is $F(\mathbf{x}, \lambda)$. The variable $\mathbf{x}$ is restricted to a bounded set, so this is a problem of Type Two.

In the next section we consider the two types of Fourier transform estimation problems.

3 Fourier transform estimation

It is often the case in remote sensing that what we want and what we can measure are related by Fourier transformation. Frequently one of the two functions has bounded support, so that the other one is band-limited. If our measurements are samples of a function of bounded support we shall say that we are solving a problem of Type One, while if the sampled function is band-limited we say the problem is of Type Two. As we shall see, these two types of problems are distinct and different techniques are required to solve them.

Throughout this section we let $F(\omega)$ be defined for $\omega \in [0, 2\pi]$, with

$$f(x) = \frac{1}{2\pi} \int_0^{2\pi} F(\omega) e^{-ix\omega} d\omega. \tag{3.1}$$

In applications $F(\omega)$ usually represents some physical object of limited extent. In problems of Type Two remote sensing has provided (usually noisy) values of $f(x)$ for finitely many x.

When algorithms are being developed and tested one often works with simulations. If the $F(\omega)$ to be simulated is specified analytically we may be able to compute values of $f(x)$ by performing the integrals in equation (3.1). It may be the case, however, that the integrals cannot be performed exactly or even that $F(\omega)$ is represented by a finite vector of samples. Estimating values of $f(x)$ in such cases becomes a problem of Type One. In the hyperspectral imaging problem discussed here problems of both types must be solved.

When discussing problems of Type One in this section we shall assume that we have the values $F_n = F(2\pi n/N)$, $n = 0, 1, ..., N-1$ and wish to estimate $f(x)$ for certain values of x. When discussing problems of Type Two in this section we shall assume, at first, that we have the values $f(m)$, $m = 0, ..., M-1$ and wish to estimate values of $F(\omega)$ and then allow the data to be $f(x_m)$, $m = 1, ..., M$, where the x_m are arbitrary.

For problems of Type One it is tempting to take as our estimate of $f(x)$ what is perhaps the obvious choice, the function

$$\hat{f}(x) = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi nx/N}. \tag{3.2}$$

and for problems of Type Two the estimate

$$\hat{F}(\omega) = \sum_{m=0}^{M-1} f(m) e^{im\omega}. \tag{3.3}$$

If, in the first case, we decide to estimate $f(x)$ only for the integer values $j = 0, ..., N-1$ then we get

$$\hat{f}(j) = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi nj/N}, \tag{3.4}$$

which can be calculated using the Fast Fourier Transform. Similarly, if, in the second case, we decide to estimate $F(\omega)$ only for the values $\omega = \omega_k = 2\pi k/M$, $k = 0, ..., M-1$, we get

$$\hat{F}(\omega_k) = \sum_{m=0}^{M-1} f(m) e^{2\pi km/M}, \tag{3.5}$$

The main theme of this section is that while these estimates may be obvious, they are not necessarily good choices.

3.1 Problems of Type One:

Let us assume that $F(\omega)$ is Riemann integrable. For each x we can approximate the integral in equation (3.1) by the Riemann sum

$$rs(x; N) = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi inx/N}, \tag{3.6}$$

which is our estimate in equation (3.2). The problem is that how good an approximation $rs(x; N)$ is of $f(x)$ will depend on x; as $|x|$ gets large the integrand becomes ever more oscillatory and a larger value of N will be needed to obtain a good approximation of the integral.

The basic idea here is to use the measured values $F(2\pi n/N)$, $n = 0, ..., N-1$ to find an approximation of the function $F(\omega)$ and then to take the inverse Fourier transform of this approximation as our estimate of the function $f(x)$. One particular approximation we study in detail is a step function, although other approximations can be considered. It is helpful to remember that the estimate in equation (3.2) is N-periodic and is based on the unrealistic approximation of $F(\omega)$ by finitely many delta functions supported on the points $2\pi n/N$, $n = 0, ..., N-1$.

Consider the step function approximation of $F(\omega)$ given by

$$S(\omega) = \sum_{n=0}^{N-1} F_n \chi_{\pi/N}(\omega - \frac{2n+1}{N}\pi) \tag{3.7}$$

with

$$s(x) = \frac{1}{2\pi} \int_0^{2\pi} S(\omega) e^{-2\pi ix\omega} d\omega. \tag{3.8}$$

Performing the integrations we find that

$$s(x) = e^{-ix\pi/N} rs(x; N) \frac{\sin(\pi x/N)}{\pi x/N}. \tag{3.9}$$

If N is large enough for $S(\omega)$ to provide a reasonable approximation of $F(\omega)$ then $s(x)$ should be a good estimate of $f(x)$, at least for smaller values of x. Of course, since the rate of decay of $f(x)$ as $|x|$ approaches infinity depends on the smoothness of $F(\omega)$ we must not expect $s(x)$ to approximate $f(x)$ well for larger values of x. Before leaving our discussion of problems of Type One we want to investigate to what extent the function $rs(x; N)$ provides a good estimate of $f(x)$.

Notice that the first positive zero of $\sin(\pi x/N)$ occurs at $x = N$, which suggests that $rs(x; N)$ provides a reasonable estimate of $f(x)$ for $|x|$ not larger than, say, $N/2$; therefore we may use f_k to estimate $f(k)$ for $0 \leq k \leq N/2$. To be safe, we may wish to use a smaller upper bound on k. Note also that $rs(-x; N) = rs(-x + N; N)$, which means that we may use f_{N-k} to approximate $f(-k)$ for $0 < k \leq N/2$.

To summarize, the N samples of $F(\omega)$ provide useful estimates $rs(k; N)$ of $f(k)$ for $-N/2 < k \leq N/2$. For $N = 2K$ we have $-K < k \leq K$, so that the N samples of $F(\omega)$ provide $2K = N$ useful estimates of $f(k)$ through the use of $rs(k; N)$.

There is yet another way to look at this problem. If $F(\omega)$ is twice continuously differentiable then

$$F(\omega) = \sum_{m=-\infty}^{\infty} f(m) e^{im\omega} \tag{3.10}$$

with uniform convergence of this Fourier series for $0 \leq \omega \leq 2\pi$. Therefore, for M large enough, we can estimate $F(\omega)$ using the truncated Fourier series

$$T(\omega; M) = \sum_{m=-M}^{M} f(m) e^{im\omega}. \tag{3.11}$$

Let $N = 2M + 1$ now.

Substituting $\omega = 2\pi n/N$ into equation (3.11) we obtain

$$T(2\pi n/N; M) = \sum_{m=-M}^{M} f(m) e^{2\pi imn/N}. \tag{3.12}$$

For $j = -M, ..., M$ multiply both sides of equation (3.12) by $e^{-2\pi inj/N}$, sum over $n = 0, ..., N - 1$ and use orthogonality to get $f(j)$ on the right side and

$$\frac{1}{N} \sum_{n=0}^{N-1} T(2\pi n/N; M) e^{-2\pi inj/N} \tag{3.13}$$

on the left. Viewing $T(2\pi n/N; M)$ as an estimate of $F(2\pi n/N)$ and replacing the former by the latter in equation (3.13), we conclude once again that $f(k)$ is well approximated by f_k for $0 \leq k \leq M$ and $f(-k)$ by f_{N-k} for $1 \leq k \leq M$.

When $F(\omega)$ is real-valued $f(x)$ is conjugate-symmetric, that is, $f(-x) = \overline{f(x)}$ for each x. If we view f_j as an estimate of $f(j)$ for $j = 1, ..., M$, then we should view f_{N-j} as an estimate of $f(-j)$. It does not make good sense to view f_{N-j} as an estimate of $f(N-j)$ since there need be no relation between $f(j)$ and $f(N-j)$, while f_j and f_{N-j} are complex conjugates of each other.

3.2 Problems of Type Two:

In problems of Type Two we want to estimate the function $F(\omega)$ having bounded support and have samples of its Fourier transform, $f(x)$. As we shall see, this type of problem presents difficulties that are quite different from those presented by problems of Type One.

According to Shannon's sampling theorem we can recover $F(\omega)$ completely from the infinite sequence of samples $\{f(k\Delta)\}$, where k runs over all the integers, for any sampling rate $\Delta \leq 1$. Unfortunately, we do not have infinitely many samples. In most applications there is a bounded set of x variables within which we select our sampling points. We may take as many sampling points as we desire, but must remain within the bounded set. We need not take the samples equispaced one unit apart; in fact, we may take irregularly spaced sample points. Let us assume now that we have the samples $\{f(x_m),\, m = 1, ..., M\}$, from which to estimate the function $F(\omega)$. We have several options now. One method, which we shall not discuss here (see [4, 5, 2, 3]) estimates $F(\omega)$ for all ω, using a certain finite parameter model. Only after this is done is the estimated function discretized. A second method, the one we shall present here, is closely related to the first method, but begins with a discretization of the function $F(\omega)$.

We replace the function $F(\omega)$ with the vector $\mathbf{F} = (F_1, F_2, ..., F_N)^T$, where the entry F_n can be viewed as $F_n = F(2\pi(n-1)/N)$. Our data is

$$f(x_m) = \frac{1}{2\pi} \int_0^{2\pi} F(\omega) e^{-ix_m\omega} d\omega,$$

for $m = 1, ..., M$. We approximate the integrals with finite sums, obtaining

$$f(x_m) = \frac{1}{2\pi} \sum_{n=1}^{N} F_n e^{-2\pi i x_m n}, \tag{3.14}$$

which we write in matrix form as $\mathbf{f} = A\mathbf{F}$, with A the M by N matrix with entries $A_{mn} = \frac{1}{2\pi} \exp(-ix_m n)$. For reasons to be discussed below we want to take $N > M$.

Since $M < N$ the systems of equations $A\mathbf{F} = \mathbf{f}$ will typically have infinitely many solutions. Our goal is to incorporate our prior knowledge of the function $F(\omega)$ in the choice of solution.

A common choice in such underdetermined problems is to select the *minimum norm* solution, given by

$$\mathbf{F}_{minnorm} = A^\dagger (AA^\dagger)^{-1}\mathbf{f},$$

where the superscript $\dagger$ indicates conjugate transpose and we assume, reasonably, that the matrix $AA^\dagger$ is invertible. However, suppose we have some prior information about the shape of the function $F(\omega)$, such as it is zero outside some interval $[a, b]$ contained within $[0, 2\pi]$, or, more generally, $|F(\omega)|$ can be approximated by some nonnegative function $P(\omega) \geq 0$. We then let $P_n = P(2\pi(n-1)/N)$ and $W_n = P_n^{-1/2}$ whenever $P_n > 0$; let $W_n = \alpha > 0$ for some small $\alpha > 0$ otherwise. Let W be the diagonal matrix with entries W_n. The minimum weighted norm solution of $\mathbf{f} = A\mathbf{F}$ is

$$\mathbf{F}_{mwn} = W^{-1}A^\dagger (AW^{-1}A^\dagger)^{-1}\mathbf{f}.$$

This minimum weighted norm solution can be obtained from the minimum norm solution of a related system of linear equations. Let $B = AW^{-1/2}$ and $\mathbf{G} = W^{1/2}\mathbf{F}$. Then $\mathbf{f} = A\mathbf{F} = B\mathbf{G}$. The minimum norm solution of $\mathbf{f} = B\mathbf{G}$ is

$$\mathbf{G}_{minnorm} = B^\dagger (BB^\dagger)^{-1}\mathbf{f} = W^{-1/2}A^\dagger (AW^{-1}A^\dagger)^{-1}\mathbf{f}$$

and

$$\mathbf{F}_{mwn} = W^{-1/2}\mathbf{G}_{minnorm}.$$

We calculate $\mathbf{F}_{mwn}$ iteratively, either by applying the *algebraic reconstruction technique* ART directly to the system $\mathbf{f} = B\mathbf{G}$ or rewriting the ART iterative step for this system in terms of the original system $\mathbf{f} = A\mathbf{F}$.

When the data is noisy we often do not want an exact solution of $\mathbf{f} = A\mathbf{F}$. In that case we *regularize* by taking as our approximate solution the vector

$$\mathbf{F}_{rmwn} = W^{-1}A^\dagger (AW^{-1}A^\dagger + \epsilon^2 I)^{-1}\mathbf{f},$$

where $\epsilon > 0$ is small and I is the identity matrix. This solution can also be found iteratively, using ART, without having to calculate the matrix $AW^{-1}A^\dagger$.

4 The ART

In applied mathematics it is often the case that the solution to our problem cannot be written in closed form, nor can it be calculated exactly in a finite number of steps.

In such cases we are forced to find approximate solutions using iterative algorithms; the Newton-Raphson method for solving $f(x) = 0$ is an example of an iterative method. There are also situations in which, in theory, the solution can be found exactly, assuming infinitely precise calculations, but to do so would be impractical: solving large systems of linear equations is an example of such a problem. We know that, in theory, Gauss elimination will find the solution in a finite number of steps, if there is a unique solution. But, when there are thousands of equations in thousands of unknowns, as is commonly the case in image processing, Gauss elimination is not practical. The iterative *algebraic reconstruction technique* (ART) was devised to solve just such large systems of linear equations.

Finding a solution to the complex system of M linear equations in N unknowns given in matrix form by $A\mathbf{x} = \mathbf{b}$ is equivalent to finding a complex N-dimensional vector $\mathbf{x}$ that is in all of the sets

$$H_m = \{x | (A\mathbf{x})_m = b_m\},$$

for $m = 1, ..., M$. The sets H_m are *hyperplanes* in C^N. One way to find such an $\mathbf{x}$ is to use the ART method.

In ART we begin with an arbitrary starting vector $\mathbf{x}^0$. We then let $\mathbf{x}^1$ be the vector in H_1 closest to $\mathbf{x}^0$, then $\mathbf{x}^2$ the vector in H_2 closest to $\mathbf{x}^1$, and so on. When we have found vector $\mathbf{x}^M$ in H_M closest to $\mathbf{x}^{M-1}$, we then let $\mathbf{x}^{M+1}$ be the vector in H_1 closest to $\mathbf{x}^M$, etc.; that is, we cycle once again through each of the M hyperplanes. This process is known to converge to the vector closest to $\mathbf{x}^0$ that is in all of the H_m.

Given any vector $\mathbf{x}$ and hyperplane H_m, the vector $\mathbf{z}$ in H_m closest to $\mathbf{x}$ can be written explicitly. We have

$$z_n = x_n + \overline{A_{mn}}(b_m - (A\mathbf{x})_m) / \sum_{j=1}^{N} |A_{mj}|^2.$$

Therefore, the ART algorithm can be written explicitly as follows: for $k = 0, 1, ...$ and $m = k(\mathrm{mod}\, M) + 1$ we have

$$x_n^{k+1} = x_n^k + \overline{A_{mn}}(b_m - (A\mathbf{x}^k)_m) / \sum_{j=1}^{N} |A_{mj}|^2.$$

It is known that the ART can be slow to converge if the equations that make up $A\mathbf{x} = \mathbf{b}$ are ordered so that successive rows of A are not significantly different. To avoid this it is highly recommended that the equations be reordered according to some random selection prior to using ART.

If we know that the solution being sought is real-valued we can rewrite the system of equations so that the matrix A and the vector $\mathbf{b}$ are real. If, in addition, we know

that the solution being sought has nonnegative entries then we can use *clipped* ART on the real system $A\mathbf{x} = \mathbf{b}$, which has the following iterative step:

$$x_n^{k+1} = [x_n^k + A_{mn}(b_m - (A\mathbf{x}^k)_m)/\sum_{j=1}^N A_{mj}^2]_+,$$

where r_+ is the maximum of the real number r and 0.

When $M \leq N$ the matrix $AA^\dagger$ is typically invertible, the system $A\mathbf{x} = \mathbf{b}$ typically has solutions and the ART algorithm converges to the solution closest to the starting vector $\mathbf{x}^0$. If $\mathbf{x}^0 = 0$ then the ART limit is then the minimum norm solution, $\mathbf{x} = A^\dagger(AA^\dagger)^{-1}\mathbf{b}$. Sometimes, however, we do not want an exact solution. In such cases regularization is required.

4.1 Regularizing ART:

It is often the case that the entries of the vector $\mathbf{b}$ in the system $A\mathbf{x} = \mathbf{b}$ come from measurements, so are usually noisy. If the entries of $\mathbf{b}$ are noisy but the system $A\mathbf{x} = \mathbf{b}$ remains consistent (which can easily happen in the underdetermined case, with $N > M$) the ART begun at $\mathbf{x}^0 = 0$ converges to the solution having minimum norm, but this norm can be quite large. The problem is that the matrix $AA^\dagger$, while invertible, can be *ill-conditioned*, which means that slight changes in $\mathbf{b}$ can lead to larger changes in the minimum norm solution. As a result, the minimum norm solution is probably useless. Instead of solving $A\mathbf{x} = \mathbf{b}$ we can *regularize* by minimizing, for example, the function

$$||A\mathbf{x} - \mathbf{b}||^2 + \epsilon^2||\mathbf{x}||^2, \tag{4.1}$$

for some small ϵ^2. The solution to this problem is the vector

$$\mathbf{x} = (A^\dagger A + \epsilon^2 I)^{-1}A^\dagger\mathbf{b}. \tag{4.2}$$

It can be shown that this regularized solution is also

$$\mathbf{x} = A^\dagger(AA^\dagger + \epsilon^2 I)^{-1}\mathbf{b},$$

where the possibly ill-conditioned matrix $AA^\dagger$ used in the minimum norm solution has been replaced by the better conditioned matrix $AA^\dagger + \epsilon^2 I$.

Although we want the regularized solution we do not want to have to calculate $A^\dagger A$ or $AA^\dagger$, particularly when the matrix A is large. Fortunately the ART can find the regularized solution using only the matrix A.

We discuss two methods for using ART to obtain regularized solutions of $A\mathbf{x} = \mathbf{b}$. The first one is new, the second one is due to Gabor Herman, in unpublished notes.

In our first method we use ART to solve the system of equations given in matrix form by

$$[\, A^\dagger \quad \epsilon I \,] \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = 0.$$

We begin with $\mathbf{u}^0 = \mathbf{b}$ and $\mathbf{v}^0 = 0$. The lower component of the limit vector is then $\mathbf{v}^\infty = -\epsilon\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ minimizes the function in (4.1).

Gabor Herman's method is similar. In his we use ART to solve the system of equations given in matrix form by

$$[\, A \quad \epsilon I \,] \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix} = \mathbf{b}.$$

We begin at $\mathbf{x}^0 = 0$ and $\mathbf{v}^0 = 0$. Then the limit vector has for its upper component the desired regularized solution $\mathbf{x}^\infty = \hat{\mathbf{x}}$. Also $\epsilon\mathbf{v}^\infty = \mathbf{b} - A\hat{\mathbf{x}}$.

As Herman and Meyer have shown [9], the order in which the equations are accessed in ART, as well as the use of relaxation parameters, can greatly affect the speed of convergence. The main consideration is to avoid taking the equations in an order such that each equation substantially repeats the information about the image present in the previous equation. To avoid such a situation we could employ a random ordering of the equations, although more carefully designed ordering may achieve somewhat faster convergence.

5 The data cube reconstruction again

In this section we return to the three-step data cube reconstruction procedure, to see what is to be done at each step, in light of our previous discussion.

5.1 Step 1:

Our measured data we take to be $G(\mathbf{x}_j, \theta_k)$, for $j = 1, ..., J$ and $k = 1, ..., K$, with J the number of pixels in the image plane grid and K the number of orientation angles for the prism. Since Step 1 is a Type One problem, we should take as fine a grid as is practical. Since $G(\mathbf{x}, \theta) \geq 0$ we want to make sure the estimates $g(\mathbf{y}, \theta)$ are conjugate symmetric in the variable $\mathbf{y}$; that is, $g(-\mathbf{y}, \theta) = \overline{g(\mathbf{y}, \theta)}$. As discussed above, we may estimate as many values of $g(\mathbf{y}, \theta)$ as desired, using the approximation of $G(\mathbf{x}, \theta)$ by a two-dimensional step function analogous to $S(\omega)$ earlier.

To be concrete, let us suppose that $J = J_1 J_2$ and that the pixel grid has J_1 rows, J_2 columns. Let the pixels have length Δ_1 in the first variable and Δ_2 in the second variable. Let the j-th sample point be $\mathbf{x}_j$ with coordinates $((j_1 - 1)\Delta_1, (j_2 - 1)\Delta_2)$,

where $j = (j_1-1)J_2+j_2$, so that the j-th pixel consists of those points in the rectangle R_j, that is, those $\mathbf{x} = (x_1, x_2)$ with $(j_1 - 1)\Delta_1 \leq x_1 < j_1\Delta_1$ and $(j_2 - 1)\Delta_2 \leq x_2 < j_2\Delta_2$.

For each fixed θ_k define $S_k(\mathbf{x})$, the step function approximation of $G(\mathbf{x}, \theta_k)$ to be

$$S_k(\mathbf{x}) = \sum_{j=1}^{J} G(\mathbf{x}_j, \theta_k)\chi_j(\mathbf{x}),$$

where $\chi_j(\mathbf{x}) = 1$ if $\mathbf{x} \in R_j$ and zero otherwise. The inverse Fourier transform of $\chi_j(\mathbf{x})$ is

$$[\frac{1}{\pi y_1} e^{-i(j_1+\frac{1}{2})\Delta_1 y_1} \sin(y_1\Delta_1/2)][\frac{1}{\pi y_2} e^{-i(j_2+\frac{1}{2})\Delta_2 y_2} \sin(y_2\Delta_2/2)],$$

so the inverse Fourier transform of $S_k(\mathbf{x})$ is

$$s_k(\mathbf{y}) = s_k(y_1, y_2) = \tag{5.1}$$

$$\sum_{j_1=1}^{J_1}\sum_{j_2=1}^{J_2} G(x_{j_1}, x_{j_2}, \theta_k)[\frac{1}{\pi y_1} e^{-i(j_1+\frac{1}{2})\Delta_1 y_1} \sin(y_1\Delta_1/2)][\frac{1}{\pi y_2} e^{-i(j_2+\frac{1}{2})\Delta_2 y_2} \sin(y_2\Delta_2/2)].$$

Use equation (5.1) to estimate values $g(\mathbf{y}_m, \theta_k)$ for $m = 1, ..., M$. Make sure that one of the $\mathbf{y}_m$ is the zero vector.

5.2 Step 2:

In Step Two we discretize in the λ domain to equispaced wavelengths $\lambda_1, ..., \lambda_N$, where $N > K$ and $\Delta_\lambda > 0$ is the spacing. Let

$$h(\mathbf{y}_m, \theta_k) = g(\mathbf{y}_m, \theta_k)e^{-i\mu\lambda_0 \mathbf{p}_k \cdot \mathbf{y}_m}$$

so that for each fixed $\mathbf{y}_m$ we have the approximation

$$h(\mathbf{y}_m, \theta_k) = \Delta_\lambda \sum_{n=1}^{N} f(\mathbf{y}_m, \lambda_n)e^{-i\mu\lambda_n \mathbf{p}_k \cdot \mathbf{y}_m}$$

for $k = 1, ..., K$. This system of equations is underdetermined. We solve for the minimum weighted norm solution using ART, with regularization if desired. The weights can be uniform over each of the λ_n involved or can be modulated to incorporate prior information as to the relative strengths of the contributions from each of the wavelengths.

5.3 Step 3:

In Step Three we fix each λ_n and use the values $f(\mathbf{y}_m, \lambda_n)$, $m = 1, ..., M$, to estimate $F(\mathbf{x}, \lambda_n)$. This is a Type Two problem again. We begin by discretizing the variable $\mathbf{x}$. Note that we need not use the same number of pixels employed in the discretizing of the function $G(\mathbf{x}, \theta)$.

Let $\mathbf{x}_i$, $i = 1, ..., I$ be the discretization of the variable $\mathbf{x}$. We want to estimate the nonnegative quantities $F(\mathbf{x}_i, \lambda_n)$ from the estimated values $f(\mathbf{y}_m, \lambda_n)$. We assume that $I > M$.

Approximating the Fourier transform integrals by finite sums we obtain the equations

$$f(\mathbf{y}_m, \lambda_n) = \frac{1}{(2\pi)^2} \Delta A_x \sum_{i=1}^{I} F(\mathbf{x}_i, \lambda_n) e^{-i\mathbf{x}_i \cdot \mathbf{y}_m}, \tag{5.2}$$

where ΔA_x is the area of each of the pixels created by the discretization. For each fixed n we have an underdetermined system of M complex equations in I nonnegative unknowns. Since the unknowns are real-valued we transform this complex system to a real system.

Since $F(\mathbf{x}, \lambda)$, the Fourier tranform of $f(\mathbf{y}, \lambda)$ with respect to the variable $\mathbf{y}$, is nonnegative, therefore real-valued, the function $f(\mathbf{y}, \lambda)$ is conjugate-symmetric:

$$f(-\mathbf{y}, \lambda) = \overline{f(\mathbf{y}, \lambda)}.$$

Substituting $-\mathbf{y}_m$ for $\mathbf{y}_m$ in equation (5.2) and averaging with the original equation (5.2) we obtain the real equations

$$\frac{1}{2}[f(\mathbf{y}_m, \lambda_n) + f(-\mathbf{y}_m, \lambda_n)] = \frac{1}{(2\pi)^2} \sum_{i=1}^{I} F(\mathbf{x}_i, \lambda_n) \cos(\mathbf{x}_i \cdot \mathbf{y}_m). \tag{5.3}$$

Similarly, we obtain the real equations

$$\frac{1}{2i}[f(\mathbf{y}_m, \lambda_n) - f(-\mathbf{y}_m, \lambda_n)] = \frac{1}{(2\pi)^2} \sum_{i=1}^{I} F(\mathbf{x}_i, \lambda_n) \sin(\mathbf{x}_i \cdot \mathbf{y}_m). \tag{5.4}$$

We have the additional information that the desired values $F(\mathbf{x}_i, \lambda_n)$ are nonnegative. To obtain an estimate with this property we can employ clipping in the ART algorithm. To obtain the weights needed we can either use a uniform weight over each $\mathbf{x}_i$ or modulate the weights to incorporate any prior information we may have as to how the intensities $F(\mathbf{x}_i, \lambda_n)$ vary with $\mathbf{x}_i$, for each fixed λ_n.

There is another way to incorporate nonnegativity into the estimate of the values $F(\mathbf{x}_i, \lambda_n)$. There are iterative algorithms, such as the *rescaled block-iterative expectation maximization maximum likelihood* (RBI-EMML) [6, 7, 8], for solving systems

of equations of the form $\mathbf{y} = P\mathbf{x}$ where P is a matrix with nonnegative entries, $\mathbf{y}$ is a known vector with positive entries and a nonnegative solution $\mathbf{x}$ is sought. To apply these algorithms to the problem in Step 3 we need to perform one more transformation, to make everything nonnegative.

We do this by adding the equation

$$f(\mathbf{0}, \lambda_n) = \frac{1}{(2\pi)^2} \Delta A_x \sum_{i=1}^{I} F(\mathbf{x}_i, \lambda_n)$$

to both equations (5.3) and (5.4) above. Then the terms $\cos(\mathbf{x}_i \cdot \mathbf{y}_m)$ and $\sin(\mathbf{x}_i \cdot \mathbf{y}_m)$ are replaced by $1 + \cos(\mathbf{x}_i \cdot \mathbf{y}_m)$ and $1 + \sin(\mathbf{x}_i \cdot \mathbf{y}_m)$, respectively. The resulting system of linear equations then has the form $\mathbf{y} = P\mathbf{x}$ as described above and the RBI-EMML can then be applied. This algorithm has already been used successfully on related unmixing problems in hyperspectral image processing [11].

References

[1] Brodzik, A., and Mooney, J. (1999) Convex projections algorithm for restoration of limited-angle chromotomographic images, *Journal of the Optical Society of America, A*, **16 (2)**, pp. 246–257.

[2] Byrne, C. and Fiddy, M. (1987) Estimation of continuous object distributions from Fourier magnitude measurements, *JOSA A*, **4**, pp. 412–417.

[3] Byrne, C., and Fiddy, M. (1988) Images as power spectra; reconstruction as Wiener filter approximation, *Inverse Problems*, **4**, pp. 399–409.

[4] Byrne, C. and Fitzgerald, R. (1982) Reconstruction from partial information, with applications to tomography, *SIAM J. Applied Math.*, **42(4)**, pp. 933–940.

[5] Byrne, C., Fitzgerald, R., Fiddy, M., Hall, T. and Darling, A. (1983) Image restoration and resolution enhancement, *J. Opt. Soc. Amer.*, **73**, pp. 1481–1487.

[6] Byrne, C. (1996) Block-iterative methods for image reconstruction from projections, *IEEE Transactions on Image Processing*, **IP-5**, pp. 792–794.

[7] Byrne, C. (1997) Convergent block-iterative algorithms for image reconstruction from inconsistent data, *IEEE Transactions on Image Processing*, **IP-6**, pp. 1296–1304.

[8] Byrne, C. (1998) Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods, *IEEE Transactions on Image Processing*, **IP-7**, pp. 100-109.

[9] Herman, G. T. and Meyer, L. (1993) Algebraic reconstruction techniques can be made computationally efficient, *IEEE Transactions on Medical Imaging*, **12**, pp. 600-609.

[10] Higbee, S. (2004) *private communication.*

[11] Meidunas, E. (2001) *Re-scaled Block Iterative Expectation Maximization Maximum Likelihood (RBI-EMML) Abundance Estimation and Sub-pixel Material Identification in Hyperspectral Imagery*, MS thesis, Department of Electrical Engineering, University of Massachusetts Lowell, Lowell MA.

[12] Mooney, J., Vickers, V., An, M., and Brodzik, A. (1997) High-throughput hyperspectral infrared camera, *Journal of the Optical Society of America, A*, **14 (11)**, pp. 2951–2961.

[13] Nelson, R. (2001) Derivation of the Missing Cone, *unpublished notes.*