

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/240633535>

Fourier methods of improving reconstruction speed for CTIS imaging spectrometers

Article in *Proceedings of SPIE - The International Society for Optical Engineering* · September 2007

DOI: 10.1117/12.732669

CITATIONS

4

READS

49

2 authors:



Nathan Hagen

Utsunomiya University

51 PUBLICATIONS 572 CITATIONS

[SEE PROFILE](#)



Eustace L. Dereniak

The University of Arizona

283 PUBLICATIONS 2,208 CITATIONS

[SEE PROFILE](#)

Fourier methods of improving reconstruction speed for CTIS imaging spectrometers

Nathan Hagen^a, Eustace L. Dereniak^a, David T. Sass^b

^aOptical Sciences Center, University of Arizona, Tucson, AZ 85721

^bU. S. Army TACOM, Warren, MI 48397

ABSTRACT

A persistent barrier to the wider use of the Computed Tomographic Imaging Spectrometer (CTIS) has been the extraordinary demands it places on computational resources. Raw images can be obtained at snapshot speeds, but reconstructed datacubes typically require minutes of reconstruction time each. We present a new approach to the CTIS reconstruction problem which makes use of the spatial shift-invariance in a CTIS system to greatly reduce the dimensionality of the matrix inversion process performed during reconstruction. Preliminary results indicate that a speedup by a factor of 4000 is possible.

Keywords: spectrometry, imaging spectrometry, CTIS, computed tomography

1. COMPUTED TOMOGRAPHIC IMAGING SPECTROMETRY (CTIS)

Computed tomographic imaging spectrometry is a snapshot technique for acquiring a spectrally-resolved scene — a “datacube”. This is accomplished by placing a 2D transmission grating (a “disperser”) in the pupil of the system and using a field stop to limit the FOV, which prevents excessive overlap between multiple diffraction orders (see Fig. 1). The resulting raw measured image consists of a set of prismatically splayed streaks, each of which contain the multiplexed spatial and spectral signals present in the scene.

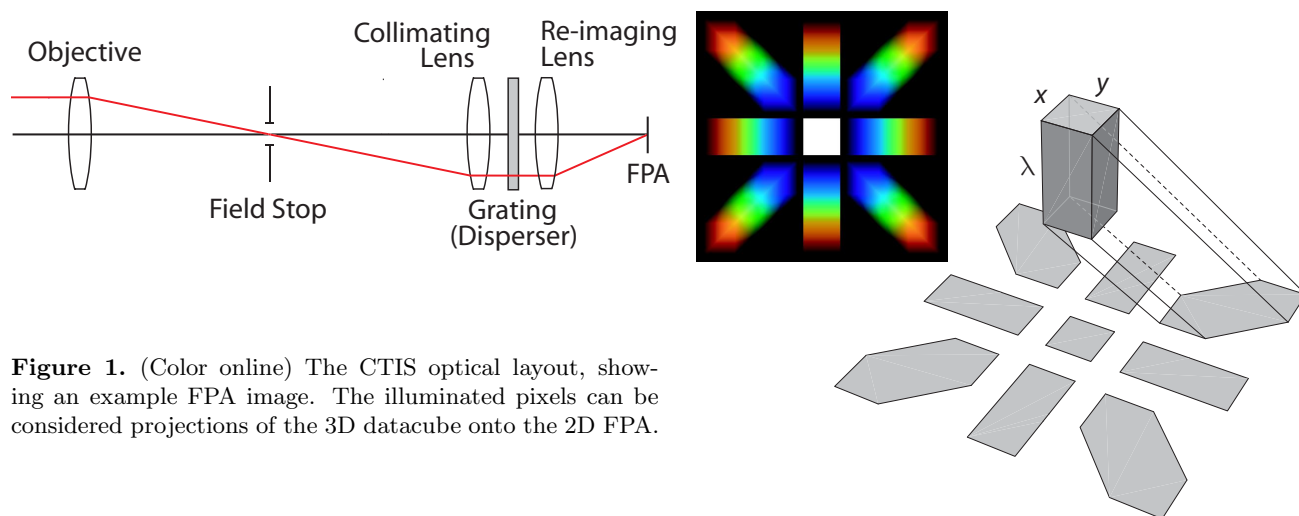


Figure 1. (Color online) The CTIS optical layout, showing an example FPA image. The illuminated pixels can be considered projections of the 3D datacube onto the 2D FPA.

In order to reconstruct the the observed scene’s datacube from the measured image, the typical approach is to start with the linear imaging equation¹

$$\underset{\substack{\text{image} \\ \text{(output)}}}{\mathbf{g}} = \underset{\substack{\text{system} \\ \text{operator}}}{\mathbf{H}} \underset{\substack{\text{object} \\ \text{(input)}}}{\mathbf{f}(x, y, \lambda)}$$

If we lexicographically order the pixel information of our detected image into a long 1D vector (indexed by m) and do the same for each voxel in the datacube (indexed by n), then each element H_{mn} of the system matrix maps the sensitivity of a voxel f_n in the datacube to a pixel g_m in the image. The full system matrix \mathbf{H} can then be calibrated using a series of detected images \mathbf{g} of a quasi-monochromatic point-source f_n (*i.e.* objects consisting of a single illuminated voxel). Experimentally, this is achieved by using a small-diameter optical fiber placed into the field stop and illuminated by a monochromator. Taking an image of the fiber tip at each wavelength band and each spatial position fills in the matrix elements of \mathbf{H} . In general, this calibration process requires complex mechanically-driven positioners for the fiber and it can take a considerable amount of time to acquire the data. To avoid these problems, spatial shift-invariance is assumed, such that shifting spatially by a pixel within the datacube produces an image at the focal plane array (FPA) which is shifted by a pixel in the same direction. Thus, if we take advantage of this property, we need measure calibration images at only one spatial position. (The validity of CTIS' shift-invariance is discussed at length in ref. 2.)

The reconstruction process used to estimate the object \mathbf{f} from a measured image \mathbf{g} requires some kind of inversion of the system matrix. Past work in our group has concentrated on two iterative reconstruction algorithms, based on the Expectation Maximization (EM) method or the Multiplicative Algebraic Reconstruction Technique (MART):³

$$\text{EM: } \hat{\mathbf{f}}^{(k+1)} = \frac{\hat{\mathbf{f}}^{(k)}}{\sum_{m=1}^M H_{mn}} \left(\mathbf{H}^\top \frac{\mathbf{g}}{\mathbf{H}\hat{\mathbf{f}}^{(k)}} \right) \quad \text{MART: } \hat{\mathbf{f}}^{(k+1)} = \hat{\mathbf{f}}^{(k)} \left(\frac{\mathbf{H}^\top \mathbf{g}}{\mathbf{H}^\top \mathbf{H} \hat{\mathbf{f}}^{(k)}} \right)$$

where $\hat{\mathbf{f}}$ indicates an *estimate* of the object's datacube, and the $[\cdot]^{(k)}$ superscript describes the iteration number.

Although the \mathbf{H} -matrix is sparse (typically containing a 1:1000 ratio of nonzero to zero-valued elements), the sparse matrix multiplications used within the reconstruction algorithms require appreciable computing time to complete. For a CTIS using a 2048×2048 FPA, a datacube of dimensions $(x, y, \lambda) = (100, 100, 600)$ are typical, so that the \mathbf{H} -matrix has $2048^2 \times 100^2 \times 600 \approx 2.5 \times 10^{13}$ elements. Storing the matrix sparsely reduces this to 2.5×10^{10} elements, so that a matrix-vector multiplication requires $\sim 5 \times 10^{10}$ floating-point operations (*i.e.* two operations per element). On a typical desktop computer, this can take between 10 minutes and an hour to complete. Since a number of potential applications for CTIS' snapshot advantage require that it achieve datacube estimates at near-video rates ($10 \sim 60$ Hz), these reconstruction speeds are $30\,000 \sim 180\,000$ times too slow.

There are a number of well-known methods for speeding up the calculations and rate of convergence for the maximum likelihood algorithms, among which are multiresolution sampling,^{4,5} point-successive overrelaxation,^{6,7} and use of conjugate gradient methods instead of the EM and MART algorithms.⁸ For CTIS reconstructions, these and almost all other approaches are disappointing from the standpoint that they promise only modest gains in calculation speed; it is necessary to use supercomputing hardware to achieve anything near the desired reconstruction times.

There is an alternative approach, however. We can take advantage of our knowledge that the CTIS system matrix is spatially shift-invariant to improve the reconstruction algorithms. It is well-known that shift-invariant systems, represented by circulant matrices, are diagonalized by Fourier transformation. If this idea can be utilized to diagonalize the \mathbf{H} -matrix along its spatial dimensions, we could expect to reduce the time required for the matrix-vector multiplications by up to three orders of magnitude.

2. USING CTIS' SHIFT-INVARIANCE TO DIAGONALIZE THE H-MATRIX

To show how shift-invariance can be used to diagonalize much of the \mathbf{H} -matrix, we first show how moving the operation to the Fourier transform domain can convert a full matrix with linear shift-invariance into a diagonal matrix. For the moment, we ignore the dependence on wavelength in the system and treat only a monochromatic objects. Once the principle is established, we can go on to show how it can be used to advantage for a complete CTIS reconstruction.

The standard imaging equation is $\mathbf{g} = \mathcal{H}\mathbf{f}$, where $\mathbf{g} = g(\mathbf{r}_d)$ and $\mathbf{f} = f(\mathbf{r})$ are Hilbert-space vectors (in this case, continuous functions in 2D). The system operator \mathcal{H} maps a continuous function \mathbf{f} (in object space) onto

another, \mathbf{g} , in image space. (Image space coordinates are represented with a subscript “d” in order to indicate their placement at the detector.) The object can be represented in terms of its inverse Fourier transform as

$$\mathbf{f} = \mathcal{F}_2^{-1} \mathbf{F} \quad \text{or} \quad f(\mathbf{r}) = \int_{\infty} F(\boldsymbol{\rho}) e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}} d^2 \rho .$$

The image is likewise $\mathbf{g} = \mathcal{F}_2^{-1} \mathbf{G}$, so that the imaging equation can be rewritten as

$$\mathcal{F}_2^{-1} \mathbf{G} = \mathcal{H} \mathcal{F}_2^{-1} \mathbf{F} \quad \rightarrow \quad \mathbf{G} = \mathcal{F}_2 \mathcal{H} \mathcal{F}_2^{-1} \mathbf{F} .$$

Thus, if the object and image vectors are represented in a Fourier basis, then the system operator becomes $\mathcal{F}_2 \mathcal{H} \mathcal{F}_2^{-1}$ rather than \mathcal{H} . If we call this new operator \mathcal{K} then we can write out the operator explicitly as

$$\mathcal{K} = [\mathcal{F}_2 \mathcal{H} \mathcal{F}_2^{-1}](\boldsymbol{\rho}_d, \boldsymbol{\rho}) = \int_{\infty} d^2 r_d \int_{\infty} d^2 r \, e^{-i2\pi \boldsymbol{\rho}_d \cdot \mathbf{r}_d} h(\mathbf{r}_d, \mathbf{r}) e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}} .$$

If the operator \mathcal{H} is LSIV then the kernel $h(\mathbf{r}_d, \mathbf{r}) = h(\mathbf{r}_d - \mathbf{r})$. In the equation for \mathcal{K} , if we make the substitution $\mathbf{s} = \mathbf{r} - \mathbf{r}_d$ then

$$\begin{aligned} \mathcal{K}(\boldsymbol{\rho}_d, \boldsymbol{\rho}) &= \iint d^2 r_d \, d^2 s \, e^{-i2\pi \boldsymbol{\rho}_d \cdot \mathbf{r}_d} h(\mathbf{s}) e^{i2\pi \boldsymbol{\rho} \cdot (\mathbf{s} + \mathbf{r}_d)} \\ &= \int d^2 s \, h(\mathbf{s}) \int d^2 r_d \, \exp \left[-i2\pi \boldsymbol{\rho}_d \cdot \mathbf{r}_d + i2\pi \boldsymbol{\rho} \cdot \mathbf{r}_d + i2\pi \boldsymbol{\rho} \cdot \mathbf{s} \right] \\ &= \int d^2 s \, h(\mathbf{s}) e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{s}} \underbrace{\int d^2 r_d \, e^{-i2\pi (\boldsymbol{\rho}_d - \boldsymbol{\rho}) \cdot \mathbf{r}_d}}_{\delta(\boldsymbol{\rho}_d - \boldsymbol{\rho})} \\ &= \delta(\boldsymbol{\rho}_d - \boldsymbol{\rho}) \underbrace{\int d^2 s \, h(\mathbf{s}) e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{s}}}_{H(\boldsymbol{\rho})} \\ &= H(\boldsymbol{\rho}) \delta(\boldsymbol{\rho}_d - \boldsymbol{\rho}) . \end{aligned} \tag{1}$$

And now we can see that the operator \mathcal{K} has a diagonal form: it is zero for any $\boldsymbol{\rho} \neq \boldsymbol{\rho}_d$.

2.1. Generalizing the technique for polychromatic images

The next step in the analysis is to generalize this approach for CTIS such that we can retain the benefits of diagonalizing the spatial dimensions of the system operator. Rather than using the continuous variable λ for the spectral characteristic of the object, it helps if we make use of the “deck of cards” representation of a CTIS system (see Fig. 2). In this discrete approach, we construct L discrete planes in the datacube, indexing by ℓ each of the monochromatic images ($1 \leq \ell \leq L$); then we can write the object as $\mathbf{f}_\ell = \sum_l f_\ell(\mathbf{r})$ for $f_\ell(\mathbf{r}) \equiv f(\mathbf{r}, \lambda_\ell)$. Each of these planes, monochromatic in the resolution of the system, maps via a PSF-like operator, to L images at the detector. A single *measurement* image comprises a sum of all L monochromatic images. Prior to performing this sum operation, we are dealing with a set of independent LSIV mappings.

The system operator for this setup is

$$h(\mathbf{r}, \mathbf{r}_d, \lambda_\ell) \equiv h_\ell(\mathbf{r} - \mathbf{r}_d) ,$$

indicating that each wavelength λ_ℓ has a convolution kernel h_ℓ describing the object→image mapping. The imaging equation in this form can be represented by

$$\mathbf{g} = \mathcal{H} \mathbf{f} = \sum_{\ell=1}^L [\mathcal{H}_\ell \mathbf{f}_\ell] .$$

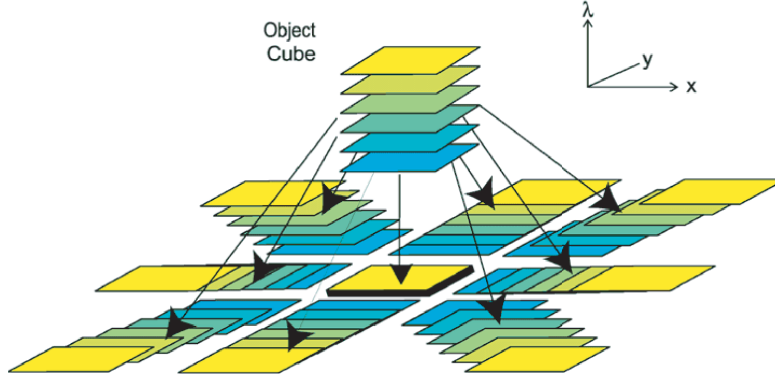


Figure 2. (Color online) The operation of a CTIS can be illustrated by imagining the datacube as a series of stacked planes, much like a deck of cards. The disperser spreads out the deck, shearing the planes, and the measurement at a given pixel involves summing across all planes overlaying the pixel.

In practical terms, \mathbf{H}_ℓ is just the regular \mathbf{H} -matrix with the columns corresponding to $\lambda_{\ell'} \neq \lambda_\ell$ set to zero. The components of the sum have the form

$$g_\ell(\mathbf{r}_d) = [\mathcal{H}_\ell \mathbf{f}_\ell](\mathbf{r}_d) = \int_{\mathcal{S}_f} h_\ell(\mathbf{r} - \mathbf{r}_d) f_\ell(\mathbf{r}) d^2r ,$$

where \mathcal{S}_f is the support region of the object. If, using (1), we transfer this problem into the frequency domain, then we can rewrite this as

$$G_\ell(\boldsymbol{\rho}_d) = \mathcal{K}_\ell(\boldsymbol{\rho}_d) F_\ell(\boldsymbol{\rho}_d) ,$$

which reduces the full CTIS matrix-multiplication problem into L sub-multiplications, each of which is *diagonal*.

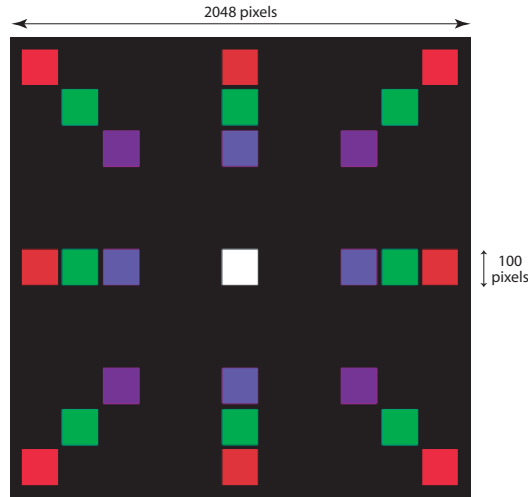


Figure 3. (Color online) An example CTIS image corresponding to a datacube composed of a uniform object with only three monochromatic bands. This can also be considered as a view of three images \mathbf{g}_ℓ for three values of ℓ , or as a view of 27 subimages $\mathbf{g}_{\ell,p}$ for three values of ℓ and nine projections p .

In this process, we have so far assumed that we have each g_ℓ and can perform its Fourier transform to get \mathbf{G}_ℓ . In reality, what we measure is not the set of \mathbf{g}_ℓ but rather $\mathbf{g} = \sum_\ell \mathbf{g}_\ell$, and so it may not be clear how to perform the above multiplication step in the Fourier domain. Section 3 below shows how the reconstruction can be modified to accommodate the change. In principle, the theory is complete here: we can use iterative

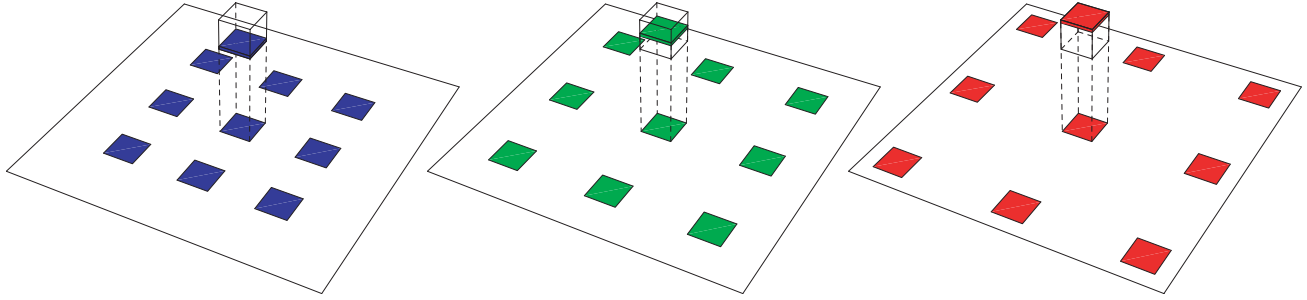


Figure 4. (Color online) Three example views of a single plane \mathbf{f}_ℓ of the datacube and their associated partial images, $\mathbf{g}_{\ell,p}$.

algorithms with the Fourier-domain versions of \mathbf{f} , \mathbf{g} , and \mathbf{H} to estimate the datacube,

$$\begin{aligned} \text{EM:} \quad \hat{\mathbf{F}}^{(k+1)} &= \frac{\hat{\mathbf{F}}^{(k)}}{\sum_{m=1}^M K_{mn}} \left(\mathbf{K}^\top \frac{\mathbf{G}}{\mathbf{K} \hat{\mathbf{F}}^{(k)}} \right) \\ \text{MART :} \quad \hat{\mathbf{F}}^{(k+1)} &= \hat{\mathbf{F}}^{(k)} \left(\frac{\mathbf{K}^\top \mathbf{G}}{\mathbf{K}^\top \mathbf{K} \hat{\mathbf{F}}^{(k)}} \right) \end{aligned}$$

After the final iteration, an inverse Fourier transform along the spatial dimensions of $\hat{\mathbf{F}}$ gives the pixel-domain representation of the estimated datacube. (Note that the assumption of Poisson statistics underlying the EM algorithm are hard to justify for the above formula. Nevertheless, the algorithm can still be useful for producing good reconstructions.⁹)

Whereas a sparse matrix-vector multiplication will require 5×10^{10} floating-point operations, if we represent the object and image in a Fourier basis then the operation $\mathbf{G}_\ell = \mathbf{K}_\ell \mathbf{F}_\ell$ involves $100 \times 100 = 10\,000$ multiplications (and the same number of additions) at each of the 600 wavelength bands, for a total of $\sim 12 \times 10^6$ flops, a reduction by a factor of ~ 4000 from the sparse-matrix approach. This understates the actual benefit because the non-sparse multiplication operation is significantly faster than its sparse counterpart. And, the speedup benefit increases for a CTIS design which chooses a larger spatial dimension at the expense of a smaller spectral.

2.2. Implementing the Fourier-domain approach

Having a workable theory is of course a great help, but we quickly find that there are obstacles to direct implementation. For example, what we measure during calibration is \mathbf{H} , and what we need to use in the Fourier domain is \mathbf{K} . When we constructed \mathbf{K}_ℓ to be a diagonal operator, it was diagonal in the sense that each spatial frequency $\boldsymbol{\rho}$ in the object maps directly to a single spatial frequency $\boldsymbol{\rho}_d$ in the image. But the image \mathbf{g} and its partial images \mathbf{g}_ℓ have dimension 2048×2048 whereas the datacube will have something more like 100×100 . The spatial frequencies of the one will not correspond to the correct ones of the other. Thus, an additional step is necessary: dividing the 2048×2048 images \mathbf{g}_ℓ into 100×100 *subimages* for each projection p . These subimages will each have the same dimensionality as the corresponding plane of the datacube, and thus their spatial frequencies will match properly.

3. A GEOMETRIC VIEW OF THE RECONSTRUCTION PROBLEM

The system matrix \mathbf{H} describing the mapping between the datacube \mathbf{f} and image \mathbf{g} is determined by a step-by-step calibration which fills out the elements of a huge matrix. This pixel-voxel mapping approach is a very abstract formulation of the system which is quite flexible but which unnecessarily hinders understanding of what is going on. There is a more intuitive way to think about the measurement process going on inside a CTIS, based on known geometrical and physical relationships.

As indicated in Fig. 4, for each monochromatic plane in the datacube, there are a set of P associated square regions $\mathbf{g}_{\ell,p}$ (*i.e.* subimages) on the FPA. We can visualize the action of the \mathbf{H} -matrix as the following sequence:

1. Take the datacube slice \mathbf{f}_ℓ and place it above the appropriate location on the FPA (this is the location of $\mathbf{g}_{\ell,p}$ for projection p).
2. Apply to the subimage $\mathbf{g}_{\ell,p}$ (still floating just above the FPA) the PSF for wavelength λ_ℓ and projection p . Note that the PSF need not be a normalized function, so that depending on the choice of units for signal in the datacube, the PSF peak can taken on any value.
3. The subimage pixel grid will not in general match exactly to the pixel grid of the FPA, so that the subimage next needs to be interpolated onto the FPA grid (see Fig. 5).
4. Iterate over all P projections.
5. Iterate and sum over all L wavelengths

This prescription gives us a means of performing the $\mathbf{g} = \mathbf{H}\mathbf{f}$ “projection” mapping without using the \mathbf{H} -matrix directly. To differentiate it from the matrix approach, we can call this the *geometric projection* approach.

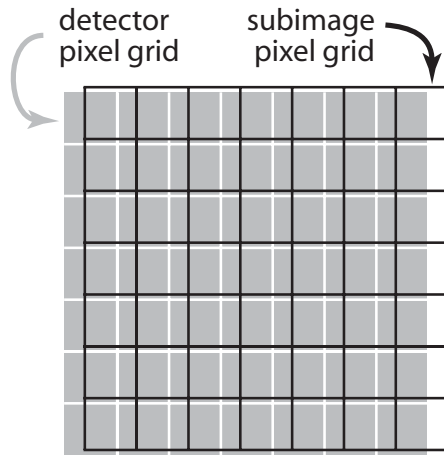


Figure 5. In general, the image of a datacube plane does match directly to the pixel grid of the FPA. In order to generate the projection, the image must be interpolated onto the detector pixel grid.

In order to make the geometric approach a valuable method, we also need an implementation of the back-projection mapping, $\tilde{\mathbf{f}} = \mathbf{H}^T \mathbf{g}$. Together with the forward projection, this would give us enough to replace the matrix-multiplication elements of the EM and MART algorithms with their geometric counterparts. The backprojection is less familiar, and somewhat less intuitive, than the forward projection operation, but we can develop a prescription for it as well. Figs 6 & 7 provide a visual guide, but the basic concept is straightforward: every voxel in the datacube which maps to a given pixel receives a signal equal to the pixel’s value times the mapping strength. The amount of signal thus allocated to the datacube is excessive, from the standpoint that a forward projection of the result will produce an image having gained a huge boost in overall signal. This by itself is not a problem — the backprojection is not meant to be an inverse projection. But if we want to preserve signal in the backprojection (*i.e.* perform a *normalized* backprojection) then we can divide the backprojected voxels by an amount such that the total signal added to the datacube is equal to the signal at the pixel being backprojected. The prescription for the complete process is as follows:

1. Locate a subimage on the focal plane array corresponding to $\mathbf{g}_{\ell,p}$, the image of a datacube plane, within projection p .
2. Calculate the wavelength λ corresponding to that subimage (in general, this wavelength will *not* correspond exactly to one of the planes of the datacube, but will lie somewhere between them).
3. Apply to the subimage (now floating just above the FPA) the PSF for wavelength λ and projection p .
4. The subimage pixel grid will generally not correspond to the spatial sampling grid of the datacube, so we need to interpolate.

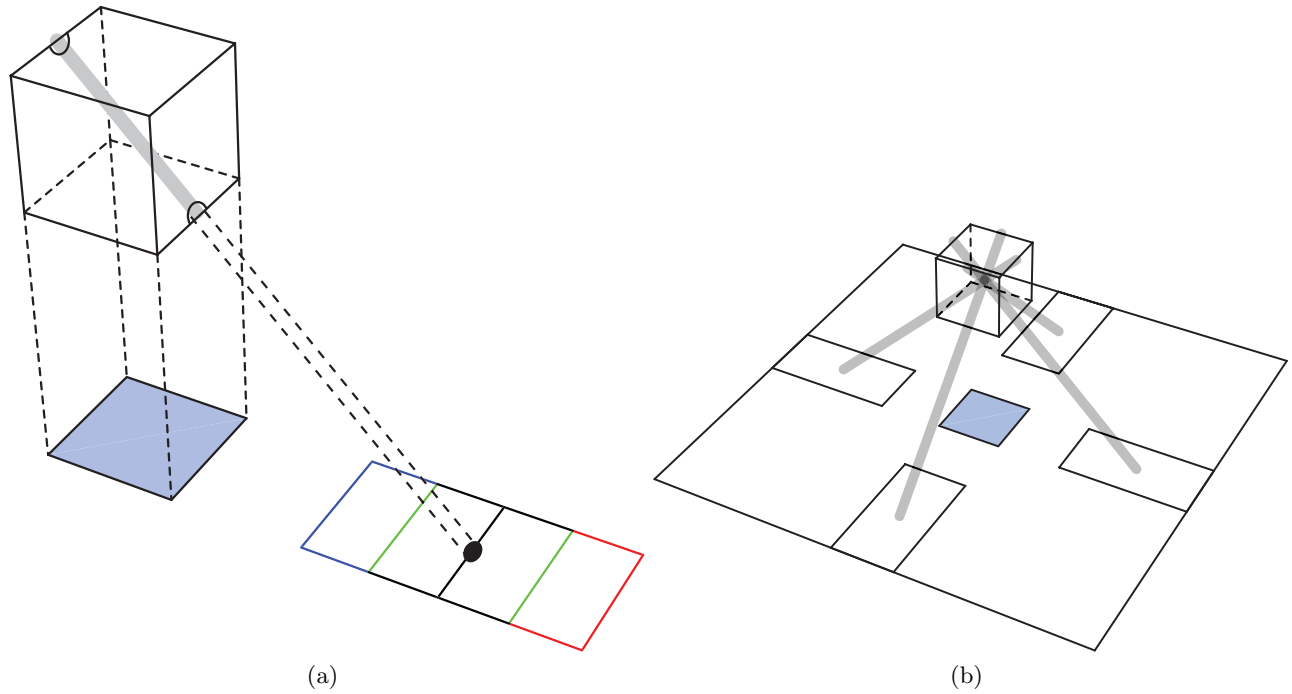


Figure 6. (Color online) A single pixel g_{xy} in the image backprojected into the datacube (a) for a single projection, (b) for a set of four projections. The angle of the backprojected column within the datacube changes depending upon the relative sizes of the spatial dimension of the datacube and spectral spread. The figures here are geometrically correct if we represent the datacube as a perfect cube.

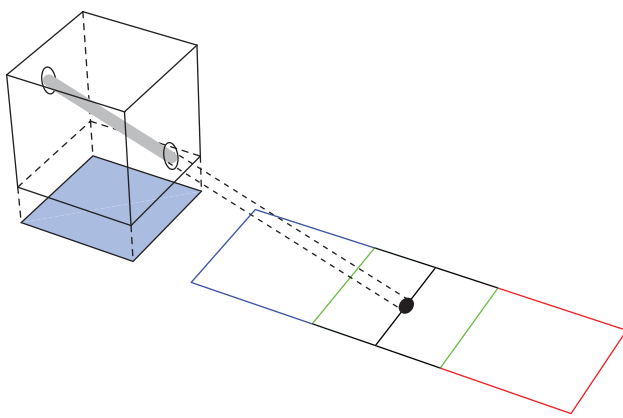


Figure 7. (Color online) The backprojection angle need not be 45° , as in Fig. 6. If the spectral dispersion length L on the FPA is longer than the spatial dimension D of the datacube, then the angle becomes something less than 45° . In the example shown here, $L = 2D$, so the angle becomes $\arctan(0.5) = 25^\circ$.

5. The subimage now corresponds to a plane of the datacube, but the plane in general lies *between* the appropriate sampling locations in wavelength, so there is a further 1D interpolation step along the wavelength axis needed here. Since the wavelength distance between two subimages $\mathbf{g}_{\ell,p}$ and $\mathbf{g}_{\ell+1,p}$ in general will not equal the wavelength distance between two planes of the datacube, this step will also involve a scale factor.
6. If we want to perform a normalized backprojection, then we need to multiply the subimage by an array of weights, determined during the system calibration. (These weights correspond to the inverse of the number of voxels a given pixel backprojects into — in practice, this is easily determined as the inverse of the uniform datacube's projected image, the uniform datacube being the object whose voxels all have signal = 1. See Fig. 8 for an example array of weights.)
7. Add the subimage signal to the appropriate interpolated plane/s in the backprojected datacube.

8. Iterate and sum over all P projections.
9. Iterate and sum over all L wavelengths.

As is probably obvious from the above sequence of steps, programming the geometric projection and backprojection operations is complicated, and needs to be done carefully in order to maintain pixel/voxel registration.

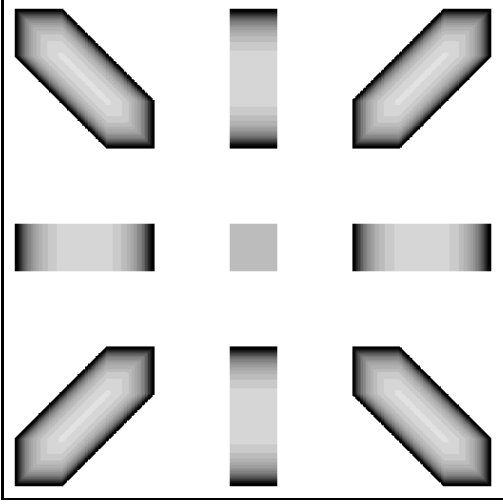


Figure 8. A greyscale image of the weights which are applied to the measured image prior to backprojection. (minimum weight, white = 0.00; maximum weight, black = 0.04)

3.1. Fourier-domain implementation

So far, the description of the geometric approach has said nothing about how we can in practice implement the Fourier approach mentioned in Section 2.2. However, if we operate in the Fourier domain, using $G_{\ell,p}(\boldsymbol{\rho})$ and $F(\boldsymbol{\rho})$ rather than $g_{\ell,p}(\mathbf{r})$ and $f(\mathbf{r})$, we can follow similar prescriptions for the forward- and back-projection operations. The spatial interpolation is easily handled in the Fourier domain by applying a complex exponential corresponding to the appropriate shift; the PSF blurring is likewise easily handled by a multiplication with the optical transfer function (OTF). Windowing of each of the subimages can be handled by multiplying by the appropriate sinc-function. Operations along the wavelength axis, lacking shift-invariance, are handled the same way as in the pixel domain approach.

During CTIS system calibration, the PSF is typically modelled as a symmetric 2D Gaussian, which means that the OTF at each wavelength band can be obtained analytically. For example, if we write (\bar{x}, \bar{y}) as the (not necessarily integer) center-location of the Gaussian peak, w as the Gaussian width parameter, and U as the volume under the Gaussian, then the PSF is given by

$$h_{\ell}(x, y) = \sum_{p=1}^P \frac{U_{\ell,p}}{\sqrt{2\pi}w_{\ell,p}^2} \exp \left(- [(x - \bar{x}_{\ell,p})^2 + (y - \bar{y}_{\ell,p})^2] / 2w_{\ell,p}^2 \right). \quad (2)$$

The Fourier transform of this function, using the reciprocal coordinates (ξ, η) , is

$$k_{\ell}(\xi, \eta) = \sum_{p=1}^P k_{\ell,p}(\xi, \eta) = \sum_{p=1}^P U_{\ell,p} \exp \left[- i2\pi(\bar{x}_{\ell,p}\xi + \bar{y}_{\ell,p}\eta) \right] \exp \left[- 2\pi^2 w_{\ell,p}(\bar{x}_{\ell,p}\xi^2 + \bar{y}_{\ell,p}\eta^2) \right]. \quad (3)$$

If the system satisfies strong shift-invariance, then the p subscripts on U and w can be dropped, but the equation must still be summed over p in order to accommodate the proper shift terms for each individual projection. The expression for the Fourier-domain blurring kernel, $k_{\ell,p}$, can be used in both the forward-projection and backprojection operations, replacing the PSF blurring step described in the prescriptions for the geometric approach.

4. USING SHIFT-INVARIANCE TO OBTAIN THE PSEUDOINVERSE

A long-sought goal for CTIS systems is the ability to use a pseudoinverse matrix for datacube reconstructions, due to its versatility, speed, and analytic properties. But achieving reconstructions through this route has been considered intractable due to several difficulties.

First, the CTIS \mathbf{H} -matrix is so large that conventional approaches to calculating the pseudoinverse requires unrealistic computational resources. Even if we could store the pseudoinverse matrix, the best supercomputers would not be able to complete the calculations required to obtain it. Forming the pseudoinverse \mathbf{H}^+ requires doing eigenvalue factorization of

$$\begin{pmatrix} \mathbf{0} & \mathbf{H}^\dagger \\ \mathbf{H} & \mathbf{0} \end{pmatrix},$$

an operation which requires $(\# \text{ of elements})^3 \sim 10^{40}$ flops. At 3 GHz speed, this would take 10^{23} years. Secondly, if these problems were not enough, there remains the vexation that the pseudoinverse of a sparse matrix is in general not sparse itself, and a non-sparse matrix multiplication for such a large matrix is extremely slow. The multiplication $\mathbf{H}^+ \mathbf{g}$ would take hundreds of times longer than the related sparse-matrix multiplication $\mathbf{H} \mathbf{f}$.

Because of these difficulties, it may seem a waste to look further into a pseudoinverse reconstructor, but there is one further idea which has not been considered: it may be that calculating \mathbf{H}^+ is infeasible, but what about calculating $\mathbf{H}^+ \mathbf{g}$ *together*, as a single algorithm? That is, if we never explicitly form the pseudoinverse matrix but rather use an expression for it in an operation on our measured image, then we may be able to get around all of these difficulties. The classic Landweber algorithm is an example of this type of approach, easily implemented as

$$\hat{\mathbf{f}}^{(k+1)} = \hat{\mathbf{f}}^{(k)} + \frac{\mathbf{H}^\dagger (\mathbf{g} - \mathbf{H} \hat{\mathbf{f}}^{(k)})}{\sum_{m=1}^M H_{mn}}.$$

On the other hand, we can use our knowledge of the instrument physics, in particular the shift-invariance of the spatial dimension, to derive an expression for the pseudoinverse operator acting on \mathbf{g} , $\hat{\mathbf{f}} = \mathbf{H}^+ \mathbf{g}$. As in Section 2, we approach this problem by first considering a monochromatic system and then generalize to a polychromatic system.

We start by taking a look at the result of applying the backprojection operator to the image:

$$\left. \begin{aligned} [\mathcal{H}^\dagger \mathbf{g}](\mathbf{r}) &= \int d^2 r_d h^*(\mathbf{r}, \mathbf{r}_d) g(\mathbf{r}_d) \\ &= \int d^2 r_d \int d^2 r' h^*(\mathbf{r}, \mathbf{r}_d) h(\mathbf{r}', \mathbf{r}_d) f(\mathbf{r}') \\ &= \int d^2 r' f(\mathbf{r}') \int d^2 r_d h^*(\mathbf{r}, \mathbf{r}_d) h(\mathbf{r}', \mathbf{r}_d) . \end{aligned} \right\} \quad (4)$$

If \mathcal{H} is LSIV then $h(\mathbf{r}, \mathbf{r}_d) = h(\mathbf{r} - \mathbf{r}_d)$. In contrast to the approach showing how to diagonalize the \mathbf{H} -matrix, for the SVD we start by postulating eigenvectors \mathbf{u} for the $\mathcal{H}^\dagger \mathcal{H}$ operator. Since we know that the system is shift-invariant, we can select planewaves: $u(\mathbf{r}) = e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}}$. Substituting this for $f(\mathbf{r})$ in (4), and also using the

substitutions $\mathbf{s} = \mathbf{r}' - \mathbf{r}_d$ and $\mathbf{t} = (\mathbf{r}_d - \mathbf{r})$, gives

$$\begin{aligned}
[\mathcal{H}^\dagger \mathcal{H} \mathbf{u}](\mathbf{r}) &= \int d^2 r' \int d^2 r_d e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}'} h^*(\mathbf{r} - \mathbf{r}_d) h(\mathbf{r}' - \mathbf{r}_d) \\
&= \int d^2 s \int d^2 r_d e^{i2\pi \boldsymbol{\rho} \cdot (\mathbf{s} + \mathbf{r}_d)} h^*(\mathbf{r} - \mathbf{r}_d) h(\mathbf{s}) \\
&= \int d^2 r_d e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}_d} h^*(\mathbf{r} - \mathbf{r}_d) \underbrace{\int d^2 s e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{s}} h(\mathbf{s})}_{H(\boldsymbol{\rho})} \\
&= H(\boldsymbol{\rho}) \int d^2 t e^{i2\pi \boldsymbol{\rho} \cdot (\mathbf{t} + \mathbf{r})} h^*(-\mathbf{t}) \\
&= H(\boldsymbol{\rho}) e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}} \underbrace{\int d^2 t e^{-i2\pi \boldsymbol{\rho} \cdot \mathbf{t}} h^*(\mathbf{t})}_{H^*(\boldsymbol{\rho})} \\
&= e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}} H(\boldsymbol{\rho}) H^*(\boldsymbol{\rho}) , \tag{5}
\end{aligned}$$

so that the eigenvector is $u(\boldsymbol{\rho}) = e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}}$ and the eigenvalue $\mu = |H(\boldsymbol{\rho})|^2$. Eigenvalues of $\mathcal{H}^\dagger \mathcal{H}$ are the singular values \mathbf{u} of \mathcal{H} .

Next, we know that the image-space singular vectors are given by $\mathbf{v} = \frac{1}{\sqrt{\mu}} \mathcal{H} \mathbf{u}$. Plugging in for the object-space singular vectors \mathbf{u} , and the known singular values μ , we get

$$\begin{aligned}
v(\mathbf{r}_d) &= \frac{1}{|H(\boldsymbol{\rho})|} \int d^2 r_d h^*(\mathbf{r} - \mathbf{r}_d) e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}} \\
&= \frac{1}{|H(\boldsymbol{\rho})|} \int d^2 t e^{i2\pi \boldsymbol{\rho} \cdot (\mathbf{r}_d - \mathbf{t})} h^*(-\mathbf{t}) \quad (\mathbf{t} = \mathbf{r}_d - \mathbf{r}) \\
&= \frac{1}{|H(\boldsymbol{\rho})|} e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}_d} \int d^2 t e^{-i2\pi \boldsymbol{\rho} \cdot \mathbf{t}} h^*(-\mathbf{t}) \\
&= e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}_d} . \tag{6}
\end{aligned}$$

Finally, we can calculate the pseudoinverse \mathcal{H}^+ from its singular value decomposition, $\mathcal{H}^+ = \sum_{k=1}^R \frac{1}{\sqrt{\mu_k}} \mathbf{u}_k \mathbf{v}_k^\dagger$:

$$\hat{\mathbf{f}} = [\mathcal{H}^+ \mathbf{g}](\mathbf{r}) = \int_{\infty} d^2 \rho \frac{1}{H(\boldsymbol{\rho})} e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}} \int d^2 r_d e^{-i2\pi \boldsymbol{\rho} \cdot \mathbf{r}_d} g(\mathbf{r}_d) = \int_{\infty} d^2 \rho \frac{G(\boldsymbol{\rho})}{H(\boldsymbol{\rho})} e^{i2\pi \boldsymbol{\rho} \cdot \mathbf{r}} .$$

This is the pseudoinverse estimate of the object, given an image $g(\mathbf{r}_d)$.

Seeing that the generalization from monochromatic to broadband reconstructions involves merely adding subscript ℓ and summing over the discrete wavelength index, the pseudoinverse reconstruction estimate becomes simply

$$\hat{\mathbf{f}} = \sum_{\ell} \mathbf{f}_{\ell} = \sum_{\ell} \int_{\infty} \frac{G_{\ell}(\boldsymbol{\rho})}{K_{\ell}(\boldsymbol{\rho})} e^{-i2\pi \boldsymbol{\rho} \cdot \mathbf{r}} d^2 \rho . \tag{7}$$

The difficulty with implementing this approach, once again, lies in obtaining the set of \mathbf{G}_{ℓ} from \mathbf{g} . Here, too, it seems that the only recourse is to an iterative approach.

But looking at the form of the result, we see that the linear reconstructor is actually an inverse filter applied to each of the discrete wavelengths, one by one. Since this linear estimator doesn't directly treat the problem of crosstalk between the different monochromatic images \mathbf{G}_{ℓ} , we can easily generalize the expression to include the case of noise. For an inverse filter, in the presence of uniform noise, we know that the ideal linear estimator is a Wiener-Helstrom filter. For the problem $g(x) = f(x) * k(x) + n(x)$ with object f , convolution kernel k , noise n , and image g , the filter in the Fourier domain is given by

$$\mathbf{W}(\xi, \eta) = \frac{\mathbf{K}^*(\xi, \eta)}{|\mathbf{K}(\xi, \eta)|^2 + \frac{\langle |\mathbf{N}(\xi, \eta)| \rangle}{\langle |\mathbf{F}(\xi, \eta)| \rangle}}$$

so that the object estimate is

$$\mathbf{F}(\xi, \eta) = \mathbf{W}(\xi, \eta) \mathbf{G}(\xi, \eta) .$$

Applying this result to the linear estimator for CTIS modifies the linear estimator (7) to

$$\hat{\mathbf{F}}(\boldsymbol{\rho}) = \sum_{\ell} \frac{\mathbf{K}_{\ell}(\boldsymbol{\rho})}{|\mathbf{K}_{\ell}(\boldsymbol{\rho})|^2 + \frac{\langle |\mathbf{N}_{\ell}(\boldsymbol{\rho})| \rangle}{\langle |\mathbf{F}_{\ell}(\boldsymbol{\rho})| \rangle}} \quad (8)$$

where $\hat{\mathbf{f}}(\mathbf{r}) = \mathcal{F}^{-1}\{\hat{\mathbf{F}}(\boldsymbol{\rho})\}$ and $\langle \cdot \rangle$ represents an ensemble average. (Note that since the filter depends upon an unknown function, \mathbf{F} , it can only be applied approximately, by replacing \mathbf{F} with $\hat{\mathbf{F}}$, and using the iterative algorithm to modify the filter at each step.)

5. CONCLUSION

Spatial shift-invariance of the CTIS system matrix has been exploited in the past for making the calibration process tractable, but it has never been exploited for optimizing the reconstruction algorithms. The estimates provided here indicate that utilizing our knowledge of shift-invariance can bring the time required to achieve reconstructions close to what is needed for real-time response. At first, the fact that CTIS system calibrations measure the matrix \mathbf{H} rather than its Fourier-domain counterpart \mathbf{K} may seem to prevent implementation of this idea, but the geometric projection approach provides a means of connecting the two. With our knowledge of the PSF $h_{\ell,p}$ provided by the calibration, and following the described prescriptions for performing the projection and backprojection operations, we can assemble the elements of \mathbf{K} without measuring them directly.

Finally, utilizing shift-invariance also provides a guide to building a pseudoinverse reconstruction estimate. It appears that this too must take the form of an iterative algorithm, but the approach given here avoids all of the previously known difficulties with obtaining this linear estimator. Moreover, we find that its final form indicates that it is very closely related to Wiener deconvolution.

REFERENCES

1. H. H. Barrett and K. Myers, *Foundations of Image Science*, John Wiley & Sons, 2004.
2. N. Hagen, *Snapshot imaging spectropolarimetry*. PhD thesis, University of Arizona, August 2007.
3. C. E. Volin, *Portable snapshot infrared imaging spectrometer*. PhD thesis, University of Arizona, Tucson, Arizona, 2000.
4. M. V. Ranganath, A. P. Dhawan, and N. Mullani, "A multigrid expectation maximization reconstruction algorithm for positron emission tomography," *IEEE Trans. on Medical Imaging* **7**, pp. 273–278, 1988.
5. A. Raheja and A. P. Dhawan, "Wavelet based multiresolution expectation maximization image reconstruction algorithm for positron emission tomography," *Computerized Medical Imaging and Graphics* **24**, pp. 359–376, 2000.
6. P. A. Jansson, "Traditional linear deconvolution methods," in *Deconvolution of Images and Spectra*, ch. 3, pp. 76–106, Academic Press, 2 ed., 1997.
7. J. Qi and R. M. Leahy, "Iterative reconstruction techniques in emission computed tomography," *Physics in Medicine and Biology* **51**, pp. R541–R578, 2006.
8. D. S. Lalush and B. M. W. Tsui, "A fast and stable maximum *a posteriori* conjugate gradient reconstruction algorithm," *Medical Physics* **22**, pp. 1273–1284, 1995.
9. G. L. Zeng and G. T. Gullberg, "Unmatched projector/backprojector pairs in an iterative reconstruction algorithm," *IEEE Trans. on Medical Imaging* **19**, pp. 548–555, 2000.