Recommendations for the AIPS++ Imaging Model

T.J. Cornwell, NRAO

May 29, 2015

1 Introduction

The Imaging Model as recommended for AIPS++ in the report of the Green Bank Meeting (Cornwell and Shone, 1992) is somewhat briefly and incompletely described in that document. This document proposes a more formal definition of the Imaging Model and of the various terms used in the GB report and subsequently in AIPS++ design documents.

The purpose of the Imaging Model was to act as an abstraction of how Telescopes convert Sky Brightness into Data. All Telescopes which conform to this abstraction would then be suitable for processing with AIPS++ with relatively little coding. All that would be needed in many cases would be a specialized form of the Imaging Model class. Actually the Calibration model would be needed as well but I totally ignore all issues connected with calibration in this document. To the extent that this abstraction is correct, we can therefore avoid duplication of effort in writing deconvolution tasks.

On reading the literature of image deconvolution, especially in radio astronomy, it is notable that many seemingly different approachs have been tried over the years. This is discouraging for a project like AIPS++ in which we hope to be able to apply many different types of deconvolution algorithm to data from many different telescopes. Despite this variety, I demonstrate below that one scheme can account for many different types of telescope. This scheme is based upon one particular abstraction of a telescope: namely, the way it converts images into data. The scheme is supported by two things: first, a reading of image processing literature outside of radio astronomy, and second, experience with SDE in which some of these ideas have been tried out. The implications of this scheme should be most apparent in the complicated imaging methods investigated by Holdaway and Bhatnagar (1992). I believe that some simplification of their suggested procedures may result.

2 A linear algebra model for Imaging

Many instruments are linear and can be represented by an operator A operating on an input sky I, to produce an data set D. If we use pixellated images then both I and D can be represented by vectors and the operator A can be represented by a matrix. We then have the matrix equation:

$$AI = D \tag{1}$$

The linearity means that images I^a and I^b which separately give rise to data D^a and D^b, together give rise to data $D^a + D^b$. The measurement matrix A is nearly always non-square and singular. Examples of A and D are as follows:

Convolution A represents convolution by a shift-invariant Point Spread Function. Optical Imaging and the standard interferometry convolution equation differ in the statistics of the noise: in the former, the noise is independent from pixel to pixel, whereas in the latter the noise is correlated with the shape of the dirty beam.

$$A_{i,j} = B(\mathbf{x}_i - \mathbf{x}_j) \tag{2}$$

In both cases, A is a Toeplitz matrix (i.e. matrix element $A_{i,j}$ is a function of $\mathbf{x}_i - \mathbf{x}_j$ only) and D is a vector representing the image.

Interferometer A is a matrix performing a Fourier transform, and D represents the visibility data. If the u, v plane vector is $\mathbf{u}$ and the image plane vector is $\mathbf{x}$, then the i, j element of A is given by:

$$A_{i,j} = e^{j2\pi \mathbf{u}_i . \mathbf{x}_j} \tag{3}$$

(actually one should separate into real and imaginary components but that is not important here).

Total Power A represents convolution of the brightness by the primary beam B, and D is the vector of total power samples:

$$A_{i,j} = B(\mathbf{x}_i, \mathbf{x}_j) \tag{4}$$

Beam Switched Total Power A is PSF(on)-PSF(off), and D is the vector of beam switched total power samples (Emerson *et al.*, 1979):

$$A_{i,j} = B(\mathbf{x}_i - \mathbf{x}_j) - B(\mathbf{x}_i - \mathbf{x}_j - \Delta\mathbf{x}) \tag{5}$$

where $\Delta\mathbf{x}$ is the beam throw.

Mosaic A is a matrix representing multiplication by a primary beam centered at a specific fixed pointing position $\mathbf{x}^p$, followed by Fourier transformation:

$$A_{i,j} = B(\mathbf{x}_j - \mathbf{x}^p) e^{j2\pi \mathbf{u}_i . \mathbf{x}_j} \tag{6}$$

Mosaic with variable pointing A is a matrix representing multiplication by a primary beam centered at a variable pointing position $\mathbf{x}_i^p$, followed by Fourier transformation:

$$A_{i,j} = B(\mathbf{x}_j - \mathbf{x}_i^p)e^{j2\pi \mathbf{u}_i . \mathbf{x}_j} \tag{7}$$

Wide field transform A is a matrix like that for the interferometer but including the w phase term.

Primary beam tapering is a strange case which is throws up some interesting points. The idea is to represent only the primary beam tapering of an interferometric array. The synthesized beam is ignored and so D is actually a collection of dirty images. We return to this case below in the discussion of *linear* mosaic.

Overall, this is a clean way of writing down the operation of a Telescope, but it does look rather unfamiliar in some aspects. The key insight which connects it to most of the algorithms described in the literature is that in practice we hardly ever actually use linear algebra to calculate AI. Instead we use special symmetries of A to make shortcuts. For example, multiplication by a Toeplitz matrix is best performed using the circulant approximation relying upon zero-padding to twice the number of pixels on each axis, following by FFT convolution (see Andrews and Hunts, 1977 for a description of this trick). Similarly, we approximate AI for the interferometer by the familiar degridding step, and we use either the 3D or the polyhedron approach (Cornwell and Perley, 1992) for wide-field imaging. It is also fruitful to consider more complicated algorithms as using various approximations for A. For example, the Clark CLEAN algorithm (Clark 1980) can be regarded as the result of alternately using a sparse and a circulant approximation for the Toeplitz dirty beam matrix (more on this below).

This formulation becomes even more useful when applied to the inverse problem of determining the sky brightness from the data. I discuss this next.

3 The inverse problem

Since the matrix A is nearly always singular, it has a null space. Any vector I^z in the null space will vanish: $AI^z = 0$. The sky brightness corresponding to I^z is therefore invisible to the Telescope and must be determined by other means such as the non-linear part of a deconvolution algorithm. To see that a linear method will not recover any vectors in the null-space consider a matrix C operating on D. Since $D = AI$, we see immediately that $CD = CAI$ must also suffer from the same null-space as A. The two most prominent non-linear algorithms are the Maximum Entropy MEM (see e.g. Narayan and Nityananda, 1986) and CLEAN (Högbom, 1974). The MEM image can be defined as that solution to

$AI = D$ which has maximum entropy $-I^T(\ln I - \ln I^M - 1)$ where I^M is a default image. The requirement that the entropy be maximal generates vectors in the null-space. CLEAN can be viewed purely as a way to solve $AI = D$ which implicitly assigns values to the null space vectors I^z. Each identification of a clean component generates vectors in the null space.

4 The Imaging Model

We have seen how the relation $AI = D$ can describe the measurement equation of a telescope. This suggests that we think of $AI = D$ as the Imaging Model. How is the Imaging Model to be used in a deconvolution algorithm? Many algorithms perform a least squares fit of a model to the data. For example, CLEAN does least squares fitting of sinusoids to the visibility data, while MEM does least squares fitting augmented by an entropy term. The χ^2 term can be written as:

$$\chi^2 = (AI - D)^T w(AI - D) \tag{8}$$

where w is inverse of the covariance matrix of errors. Often the errors are independent between data points, in which case w is diagonal with elements:

$$w_{i,i} = \frac{1}{\sigma_i^2} \tag{9}$$

There are two closely related approachs to using χ^2. First, we consider an approach based upon the idea of optimization: many iterative algorithms update an estimate of I based upon χ^2 and its gradient with respect to I:

$$\frac{\partial\chi^2}{\partial I} = 2A^T w(AI - D) \tag{10}$$

We can define the services of the Imaging Model which are required evaluate this term:

1. Use a service **predict** to find AI,
2. Subtract D to get $AI - D$,
3. Use a service **invert** to get $2A^T w(AI - D)$

The second approach is based on the *normal equation.* At the optimum solution I, the gradient of χ^2 must be zero and so we have that:

$$A^T wAI = A^T wD \tag{11}$$

Note that in imaging, the normal equation sometimes becomes the convolution equation.

Only in rare cases can the normal equation be solved exactly and only in even rarer cases is it a good idea to do so (see Press *et al.*, 1989). Often a better strategy is to make corrections to a trial image based upon the residual in this equation $A^TwAI - A^TwD$.

The normal equation probably warrants a service: **residual**, returning χ^2 and $A^TwAI - A^TwD$ (which is also, conveniently, $\frac{1}{2}\frac{\partial\chi^2}{\partial I}$).

Some readers may feel that the distinction between these two approaches is subtle to the point of vanishing. Perhaps the best reason to make such a distinction is that the literature does. Both views are seen in published papers. The normal equation approach looks a bit simpler and avoids the worrying connotations of optimization theory. The optimization approach is more easily improved: for example, one could use conjugate gradients or variable metric algorithms for the optimization. In the rest of this document I will retain the distinction, principally to respect the original derivation of each algorithm.

Following Holdaway and Bhatnagar (1992), I will use the term *Imager* to denote any algorithm which estimates I from A and D and any extra information. Many existing Imagers fit one of these two approaches:

UVMAP raises up a couple of interesting points. First, it does nothing to estimate vectors in the null space, and in fact, these are usually absent altogether. Second, the image produced by uvmap actually corresponds to A^TwD normalized by the diagonal element of A^TwA. It is obvious that this is a more satisfactory estimate of I than A^TwD alone. If A^TwA is diagonal then this is inverse of A. This estimate is probably worth making generally available as a service of the Imaging Model. This suggests that we extend the definition of **solve**: returns A^TwD for given D, optionally normalized by the diagonal elements of A^TwT. Also **residual** could be augmented in a similar way. We would also benefit from a service **Hdiagonal** which returns the diagonal elements of the Hessian matrix A^TwA.

Hogbom CLEAN is appropriate for the interferometric Convolution Imaging Model. It operates on the normal equation residual by selecting the peak as an estimate of the strength and location of a point source increment to the current image. In this case, A is the full dirty beam and is therefore Toeplitz. We use the normalized **solve** to get the dirty image, and a new service, the normalized **observe** which returns A^TwAI, to get one row of the PSF where I^P is an appropriately centered point source.

Clark CLEAN is also appropriate for the interferometric Convolution Imaging Model. The algorithm proceeds in minor and major cycles. In a minor cycle, a non-Toeplitz subsection of the dirty beam is used in a tightly written inner loop to clean only the high points of the residual image. In the major loop, the full Toeplitz dirty beam is used via zero-padding and an FFT-convolution. As mentioned above, one can regard this algorithm as the logical result of alternately using a sparse and then a circulant approximation for A.

MX uses both the interferometric Convolution and the Interferometric Imaging Models. In the minor cycle, a sparse approximation for the Convolution A is used, while in the major cycle, multiplication by the Interferometric A is mimiced by using degridding and gridding together with an FFT.

Projection on Convex Sets uses a *projection* operator $\mathcal{T}$ to project the normal equation residual into the space of feasible solutions. The update formula can be written as follows:

$$I^{(n+1)} = I^{(n)} + \mathcal{T}\left(A^T w(D - AI^{(n)})\right) \tag{12}$$

where $\mathcal{T}$ is the projection operator. In some cases, it pays to apply the inverse of the diagonal terms of the Hessian:

$$I^{(n+1)} = I^{(n)} + \mathcal{T}\left(\left[\text{diag}(A^T wA)\right]^{-1} A^T w(D - AI^{(n)})\right) \tag{13}$$

A simple example would be to increment the current estimate of I by the positive parts of the normal equation residual. The operator $\mathcal{T}$ then just passes the positive parts of the image. CLEAN is a POCS algorithm in which the projection operator just passes the maximum pixel. You may think that POCS is a fancy term for something quite obvious but there is a convergence proof and other mathematical machinery which is quite useful: see Youla (1974).

MEM can be applied to any of the Imaging Models described above. It is usually implemented via an optimization algorithm. The algorithm in AIPS uses $\frac{\partial\chi^2}{\partial I}$ together with the entropy gradient $\ln I^M - \ln I$ to find the deconvolved image using an iterative scheme (see Cornwell and Evans, 1985), stopping when χ^2 reaches a set limit. The algorithm also requires the diagonal elements of $A^T wA$ provided by **Hdiagonal**. The update formula is:

$$I^{(n+1)} = I^{(n)} + \gamma\left[\frac{1}{I} + \alpha\text{diag}(A^T wA)\right]^{-1}\left(\ln I^M - \ln I^{(n)} + \alpha A^T w(D - AI^{(n)})\right) \tag{14}$$

where γ and α are carefully chosen constants.

VTESS applies to the Mosaic Imaging Model. It uses the sum of $\frac{\partial\chi^2}{\partial I}$ for all pointings together with the entropy gradient $\ln I^M - \ln I$ to find the mosaic image (see Cornwell 1988).

LTESS (also known as *linear mosaic*) is an interesting case where the normalized **solve** service produces the solution to the normal equation. For the **Primary Beam Tapering** Imaging Model described above, the normalized **solve** returns:

$$I(\mathbf{x}_i) = \frac{\Sigma_p B(\mathbf{x}_i - \mathbf{x}_p) I_p^D(\mathbf{x}_i)}{\Sigma_p B(\mathbf{x}_i - \mathbf{x}_p)^2} \tag{15}$$

where I_p^D is the dirty image for pointing position $\mathbf{x}_p$ and the summation is over all pointing centers. This algorithm has two uses, first for low SNR images where the sidelobes of the synthesized beam are unimportant, and second for adding residuals back to the result of *non-linear* mosaic.

In all these Imagers, the Imaging Model is separated out in a very clean way. For example, MEM needs only to call two services of the Imaging Model: **residual** and **Hdiagonal**. An interesting question is whether *all* Imagers call Imaging Models directly. The only possible exception is the Högbom CLEAN where the point spread function $A^T wAI^P$ is cached and used in an inner loop. This split is really a semantic one only since one could envisage a program which took a visibility dataset and made the dirty image $A^T wD$ and the dirty beam $A^T wAI^P$ directly and then proceeded to the CLEAN step without writing either to disk. We therefore conclude that in essence all Imagers call upon services of Imaging Models. In some case, many different Imaging Models can be invoked. For example, the *non-linear* mosaic algorithm can potentially call the **residual** service of many different Imaging Models: Interferometric, Total Power, Beam-Switched Total Power, etc.

Note that as well as being able to compute the effect of A, we also need to be able to compute the effect of the transpose of A. This is usually quite straightforward and computationally efficient.

Convolution A^T represents convolution by a shift-invariant spread function after inversion through the origin:

$$A^T_{j,i} = B(\mathbf{x}_j - x_i) \tag{16}$$

Interferometer A^T is a matrix performing a Fourier sum:

$$A^T_{j,i} = e^{-j2\pi \mathbf{u}_i . \mathbf{x}_j} \tag{17}$$

Total Power

$$A^T_{j,i} = B(\mathbf{x}_j, \mathbf{x}_i) \tag{18}$$

Beam Switched Total Power

$$A^T_{j,i} = B(\mathbf{x}_j - \mathbf{x}_i) - B(\mathbf{x}_j - \mathbf{x}_i - \Delta\mathbf{x}) \tag{19}$$

Mosaic A^T is a matrix representing Fourier transformation followed by multiplication by a primary beam centered at a specific fixed pointing position $\mathbf{x}^p$:

$$A^T_{j,i} = B(\mathbf{x}_j - \mathbf{x}^p)e^{-j2\pi \mathbf{u}_i . \mathbf{x}_j} \tag{20}$$

Mosaic with variable pointing A^T is a matrix representing Fourier transformation followed by multiplication by a primary beam centered at a variable pointing position $\mathbf{x}^p$:

$$A^T_{j,i} = B(\mathbf{x}_j - \mathbf{x}^p_i)e^{-j2\pi \mathbf{u}_i . \mathbf{x}_j} \tag{21}$$

Wide field transform A^T is a matrix like that for the interferometer but including the w phase term.

5 Discussion

This interpretation of Imaging Model is perhaps the one which many of us at Green Bank had in mind but did not have time to write down completely. I hope that the emphasis given here to linear algebra takes some of the mystery out of inverse methods and emphasizes the distinction between the **linear** part of the problem $AI = D$ and the **non-linear** part encoded in algorithms (or Imagers) like MEM and CLEAN. The key advantage of this abstraction is that in AIPS++ we should be able to mix and match deconvolution algorithms and Imaging Models as required.

I therefore propose that AIPS++ adopt the definition of Imaging Model as a class with the limited, fixed set of services summarized in Appendix A. I'm sure that someone will be able to think of instruments and deconvolution algorithms for which this abstraction does not work. However, the fact that it does work in so many different situations means that we have much to gain by adopting it as the definition of an Imaging Model.

6 References

Andrews, H.C., Hunt, B.R., 1977, *Digital Image Restoration*, Prentice-Hall.

Clark, B.G., 1980, A&A, **8**, 377.

Cornwell, T. J., Evans K. F., 1985, A&A 143, 77

Cornwell, T. J., 1988, A&A 202, 316

Cornwell, T. J. and Shone, D. L., 1992, ed. "Calibration, Imaging and Datasystems for AIPS++ - Report of the meeting held at Green Bank, West Virginia, 3-14 February 1992", AIPS++ Memo series.

Cornwell, T. J., Perley, R. A., 1992, A&A 261, 353

Emerson, D. T., Klein, U., Haslam, C. G. T., 1979, A&A 76, 92

Högbom, J., 1974, ApJS 15, 417

Holdaway, M.A., and Bhatnagar, S., 1992, AIPS Implementation note 132.
Narayan, R., Nityananda, R., 1986, ARA&A 24, 127
Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., 1989, *Numerical Recipes*, Cambridge University Press, Cambridge, U.K.
Schwarz, U.J., 1978, A&A 65, 345
Youla, D., 1974, IEEE Trans. Circults and Control Systems.

Appendix A: Summary of the proposed properties of the Imaging Model

The Imaging Model is a class with the following services:

predict returns AI for given I,

solve returns A^TwD for given D, optionally normalized by the diagonal elements of A^TwA (if A^TwA is diagonal, the normalized version is the solution to the normal equation),

residual returns $(AI - D)^Tw(AI - D)$ and $A^Tw(AI - D)$ for given I and D, optionally normalized by the diagonal elements of A^TwA,

observe returns A^TwAI for given I, optionally normalized by the diagonal elements of A^TwA,

Hdiagonal returns the diagonal elements of A^TwA.

One could imagine defining a special Imaging Model such that the matrix A is supplied explicitly in the constructor. These various services would then be realized by actually calling the relevant Math routines. It is anticipated that only in the rare simple cases will any of these operations be implemented using simple matrix algebra. In general, special properties of the matrix A must be used to make the computation feasible. Hence the Imaging Model could be regarded as related to the Matrix class but with only some operations allowed.

In current AIPS++ terms, D is a YegSet and I is an Image.