Exotic Imaging Methods:
A Guide For Masochistic AIPS++ Designers

M.A. Holdaway
S. Bhatnagar
April 7 1992

This document examines mosaic imaging algorithms and non-isoplanatic calibration and imaging algorithms in the light of the ImagingModel and TelescopeModel objects in AIPS++.

1 Mosaicing

A mosaic image is produced from interferometer data taken from several different, adjacent pointings on the celestial sphere. A single-pointing of data can be treated as a mosaic. One does not want to force the treatment of all single-pointing data as mosaics, ie, one does not always want to image the entire primary beam (PB) or correct for the PB. The mosaic machinery also allows for combining data from telescopes with different PB's and different imaging models, an interferometer and a single dish, for example.

1.1 Operations on Mosaic Datasets (Visibilities)

One will need to perform "bulk" operations on data from all pointings from one telescope (ie, calibration). One will need to perform "piecemeal" operations on all data from a single pointing from a single telescope (ie, gridding/FFTing). For convenience of "bulk" operations, all the data from one telescope should be stored together in one dataset. Then the "piecemeal" operations could be performed by selecting out subsets of data. In addition to all the generic data selection possible with single-pointing interferometers, one must be able to select on sky pointing position (a single dish selector) and Telescope. Something like the YegSet Iterator which would step through all pointings one at a time would be nice. It is unclear whether the data from multiple telescopes should be stored in one dataset or in one dataset for each telescope. The number of different telescopes used will usually be one or two, so association of mosaic datasets from different telescopes *by hand* will not be too cumbersome.

1.2 Images

The final product of the MOSAIC imaging algorithm is a plain old image. The main difference between a MOSAIC image and an image produced by CLEAN, say, is that it is inappropriate to apply a PB to the final MOSAIC image: the parameters OBSRA, OBSDEC, and TELESCOP are not uniquely specified. While it is unclear how they would be used, if these parameters are present at all, they should be in arrays attached to the image indicating ALL pointings of ALL telescopes. There may or may not be an associated SENSITIVITY image, which is essentially a generalized primary beam, and can be used to determine the theoretical thermal noise in each pixel.

The fundamental equation at work in mosaicing is derived by making a pixel by pixel least square fit to the true sky brightness from a number of overlapping images weighted by the PB:

$$I_{mosaic}(\mathbf{x}) = \frac{\sum_{i=1}^{Npointings} I_i A_i(\mathbf{x} - \mathbf{x}_i)}{\sum_{i=1}^{Npointings} (A_i(\mathbf{x} - \mathbf{x}_i))^2} \tag{1}$$

where $I_i(\mathbf{x})$ is an image made at pointing i, and $A_i(\mathbf{x} - \mathbf{x}_i)$ is the primary beam of the telescope used for pointing i with pointing center $\mathbf{x}_i$. The square root of the denominator is the sensitivity image.

It has not been specified what kind of images the $I_i(\mathbf{x})$ are. In some cases, I_i will be independently deconvolved images. In this case, they may have some permanence outside the mosaicing method. More common, at least when the MMA comes on line, will be the case where I_i are dirty images produced for the sole purpose of making a mosaiced image. The mosaiced dirty image may then be deconvolved with a single effective point spread function. In this case, the I_i have no permanence. And finally, the most complicated case:

$$I_i(\mathbf{x}) = I_i^D(\mathbf{x}) - PSF_i * (A_i(\mathbf{x} - \mathbf{x}_i)\hat{I}(\mathbf{x})) \tag{2}$$

where $I_i^D(\mathbf{x})$ is the dirty image made from data from pointing i, PSF_i is the point spread function for pointing i, and $\hat{I}(\mathbf{x})$ is a model image of the entire mosaic field. An unnormalized mosaic (ie, not divided by the denominator in Equation 1) of this quantity is proportional to the gradient of χ^2 with respect to the model image, and indicates how the model image must be altered in the next iteration of a maximum entropy mosaicing scheme. In this case, the $I_i^D(\mathbf{x})$ and the PSF_i (or more conveniently, the Fourier transform of the PSF_i) should persist from iteration to iteration for efficiency sake, though they could be recreated on each iteration if memory or disk space were in short supply. The intermediate images $I_i(\mathbf{x})$ can disappear as soon as they are accumulated into the gradient mosaic. The I_i^D and PSF_i do not need to persist after the final mosaic image is formed. While it may be convenient to place each of the I_i^D and PSF_i into *collections* of images, they are not really *images* of images. The details of how each of the I_i are combined should be implicit in the coordinate headers of the I_i and the coordinate header of some template image (ie, the final mosaic image or the gradient image).

1.3 Pathological Cases

The following two ImagingModels require knowledge of some parameters which seem to reside in the TelModel. The extent of the dependence of the ImagingModel on the TelModel is analyzed for the case of varying antenna voltage patterns and varying antenna pointing position.

1.3.1 Different voltage patterns for each antenna

If the primary beams are different for each antenna, then a single visibility is related to the sky brightness distribution via a Fourier sum:

$$V_{j,k}(\mathbf{u}) = \int I(\mathbf{x}) E_j(\mathbf{x}' - \mathbf{x}_p) E_k^*(\mathbf{x}' - \mathbf{x}_p) e^{-i2\pi \mathbf{x}\cdot\mathbf{u}} d\mathbf{x} \tag{3}$$

where $E_j(\mathbf{x})$ is the complex voltage pattern for the j antenna ($PB(x) = |E(x)|^2$), and $\mathbf{x}'$ is related to $\mathbf{x}$ by rotation for azimuth-elevation mount antennas. For the general case of $E_j \neq E_k$ for $j \neq k$, it is more efficient to use a direct Fourier sum than an FFT to calculate a model visibility from a model sky brightness distribution since the calculation must be performed one visibility point at a time. Correction for different voltage patterns may be important for high dynamic range MMA mosaic observations. The voltage patterns would probably be determined holographically. An imaging algorithm would then adjust the image iteratively using a gradient of χ^2 with respect to $I(\mathbf{x})$ which looks something like

$$\Delta I(\mathbf{x}) = 2 \sum_{i=1}^{Npointings} \sum_{j,k;j<k} FT(V_{i;j,k} - \hat{V}_{i;j,k}) E_j(\mathbf{x} - \mathbf{x}_i) E_k(\mathbf{x} - \mathbf{x}_i) \tag{4}$$

where $\hat{V}_{i;j,k}$ is the model visibility calculated from the current model of the sky brightness distribution as per Equation 3. The basic requirement from Equations 3 and 4 is that the ImagingModel must have access to the complex voltage patterns for each antenna. Actually, "VoltagePattern.Apply" is the only method required external to ImagingModel. Under the current understanding, the voltage patterns would reside in the TelModel. If this is so, then the VPMosImagingModel requires very little of the TelModel.

1.3.2 Different pointing centers for each antenna

Pointing errors are thematically similar to different voltage patterns. A single visibility is related to the true sky brightness distribution as

$$V_{j,k}(\mathbf{u}) = \int I(\mathbf{x}) E(\mathbf{x}' - \mathbf{x}_{p_j}) E^*(\mathbf{x}' - \mathbf{x}_{p_k}) e^{-i2\pi \mathbf{x}\cdot\mathbf{u}} d\mathbf{x}, \tag{5}$$

where the differences between Equation 5 and Equation 3 are the inclusion of antenna-dependent pointing positions $\mathbf{x}_{p_j}$ and the omission of antenna dependent voltage patterns. Like the voltage pattern case, a direct Fourier sum must be used. The equation for the gradient image will be omitted to save on indices, but the basic idea is the same as for the antenna-dependent voltage pattern case. The PEMosImagingModel must have access to a method which supplies either a pointing correction or an absolute pointing position for a given antenna and at a given time. Again, this seems to require that the PEMosImagingModel have knowledge of the TelModel.

It is unclear whether these imaging methods for these two pathological cases will actually be implemented. However, it is known that *simulations* of pointing errors and voltage pattern errors (as caused by surface errors, for example) is required for the design of the MMA and other high frequency instruments.

2 Non-isoplanatic Imaging

At low frequencies, the imaging problems related to the ionosphere are aggravated. The field of view is large and the ionospheric phases cannot be considered to be a function of time alone - they become a function of direction in the sky as well. The phase delays goes as λ^2 at low frequencies and that makes it essential to correct the raw phases as a function of direction in the sky to get moderate dynamic range maps. Also the 2D approximation in inverting the visibilities to get the image breaks down. A 2D Fourier Transform is no longer valid.

The problem of 3D inversion is well understood and implemented (in SDE etc.). Here we attempt to describe the proposed algorithm for doing SelfCal for the non-isoplanatic case. Implementing these algorithms in AIPS++ will be a good test of the flexibility of the system and also a requirement.

2.1 The problem

In ordinary SelfCal, the gain corrections applied to the visibilities, are assumed to be antenna based and constant for the entire field of view of the individual antennas for the given time range (the SelfCal integration time interval). This means that a single gain is associated with a given antenna, at a given time. In other words, the properties of the ionosphere do not vary appreciably over each antenna's PB. However, when the field of view is large, the gain associated with the individual antennas is a function of the direction in the sky. Therefore, the gain solution needs to be a function of time as well as the direction in the sky.

One of the solution suggested (Schwab), divides the PB of each antenna into smaller patches, each of which can be assumed to be isoplanatic and then solve for the gains associated with each of these patches, for each antenna. Thus for an array with N antennas, and each PB divided into M patches, the number of unknowns to be solved for will be $(M * N) - 1$ (phases for all the patches in the reference antenna cannot be set to zero). The number of patches that one needs to have in the PB will depend on the size of the iso-planatic patches and that in turn will depend on the stability of the ionosphere, observing frequency, and PB size.

For the iso-planatic SeflCal, for an array with N antennas, the problem is overdetermined by a factor of N/2. If the source model is inaccurate or the

S/N ratio is poor, for SelfCal to suceed, this overdeterminacy is essential. For non-isoplanatic case, the limit on the maximum number of patches is set by this requirement of overdeterminacy and overdeterminacy decreases as the number of patches increase. For large M (M will be large when the ionosphere is bad and the iosoplanatic patches are small), accurate initial model and high S/N ratio will be required.

For arrays like GMRT and VLA, the primary beams of most of the antennas will overlap at the typical ionospheric height (300 Km) for low frequency (<300 MHz) observations. This means that in the above scheme, the gains associated with the patches where the PBs overlap, will be solved for more than once (for each of the PBs in the overlapping region). Since the total number of unknows should be less than the number of baselines, this restricts the total number of patches in the PB to less than $N(N-1)+2/2N$. This also adds the extra computation of solving for redundant unknowns in the overlapping regions.

In the scheme proposed by Subrahmanya, the overlap of PBs at the ionospheric height, is taken into account by dividing the ionosphere, intercepted by the entire array, into isoplanatic patches and solving for the associated gains of these patches. Thus, for an array with N antennas with the ionosphere divided into M isoplanatic patches, the total number of unknowns would be N+M-2. This will relax the limit on the maximum possible number of isoplanatic patches to $N(N-1)/2++2-N$. However, in general, the ionosphere cannot be regarded as a thin layer and the ionospheric structure in vertical direction has to be considered.

2.2 Formulation of the problem

For the case of isoplanatic ionosphere, the measured visibility can be written as

$$V_{pq}^{iso} = V^{mod} e^{i(\phi_q(t)-\phi_p(t))} \tag{6}$$

where ϕ_q is the antenna based phase. Since the the antenna based phases are a function of time alone, the assumption of iso-planaticity is implicit.

Let the model sky image be represented by

$$B(x,y) = \sum_k A_k \delta(x-x_k, y-y_k) \tag{7}$$

and the model visibility given by

$$V^{mod} = \sum_k \xi_k \tag{8}$$

$$\xi_k = A_k e^{i(ux_k + vy_k + w\sqrt{1-x_k^2-y_k^2})}. \tag{9}$$

In the non-isoplanatic case, the observed visibilities can be expressed as

$$V_{pq}^{non} = e^{i(\phi_q(t)-\phi_p(t))} \sum_k \xi_k e^{i(\psi_k(p)-\psi_k(q))} \tag{10}$$

where $\psi_k(p)$ is the total antenna based phase (apart from the phase due to the source) in the PB of antenna p in the direction (x_k, y_k) (the isoplanatic case is a special case of this equation where $\psi_k(p) = \psi_k(q)$). If $\theta_l(p)$ represent the phase associated with the l^{th} cell, the total phase $\psi_k(p)$ can be written as

$$\psi_k(p) = \sum_l w_l(p; x_k, y_k)\theta_l \tag{11}$$

where $w_l(p; x_k, y_k)\theta_l$ are weights associated with some interpolation scheme used for interpolation.

If the model visibility is represented as

$$V_{pq}^{mod} = e^{i(\phi_q-\phi_p)} \sum_k \xi e^{i(w_l^{qk}-w_l^{pk})\theta_l} \tag{12}$$

the unknowns are the ϕ's and θ's and can be solved for by minimizing the equation

$$S = \sum_p \sum_q |V_{pq}^{non} - V_{pq}^{mod}|^2. \tag{13}$$

2.3 The operations required

2.3.1 Non-Isoplanatic Selfcalibration

In the case of iso-planatic SelfCal, the algorithm walks through the time slices and generates the antenna gains as a function of time by solving for different time slices. For the non-isoplanatic case, the algorithm has to not only walk through the time slices, but also the space slices - the iso-planatic patches and generate antenna gain solutions as a function of direction in the sky for each time slice.

The most basic additional requirement for handling non-isoplanatic selfcalibration is to be able to get the estimate of the visibilities as a function of the direction in the sky. This requirement is explicit in Equation 10.

To start the non-isoplanatic imaging cycle, a model image is needed. At very low frequencies, the phases will be stable for less than a minute. With many isoplanatic patches across the PB and source structure which is often somewhat homogeneous (ie, no dominant point source), getting a model image to begin the selfcalibration cycle is non-trivial. A point-source model is usually inappropriate. A reasonable solution would be to observe the region of interest at higher frequencies where phase stability is not a problem and use the higher frequency image as a starting model.

Once a model image exists and each patch of sky's contribution to the model visibilities can be calculated, a least-square minimization can be done on Equation 13 to get solutions of θ_l and ϕ_p. Since the visibilities are required as a function of the direction in the sky, Solve requires the ImagingModel's PredictYegs method and therefore couples the TelModel with the ImagingModel explicitly.

One requirement on TelModel is that the gains must always be iteratively adjusted; you cannot apply the gains to a visibility set and then throw the gains away since the visibility set will then be phase stable for only one patch of sky.

2.3.2 Non-Isoplanatic Imaging

Once direction dependent gains exist, we can go about the work of imaging. Two approaches seem fruitful: a direct method, which applies the "local" gains for each patch, inverts the visibilities, and cleans; and an inverse approach, which tries to find the image which reproduces the data (raw) visibilities after unapplying the gains.

The direct approach is similar in spirit to MX and the SDE task FLY. It will likely be built onto FLY since FLY correctly deals with the non-coplanar baseline problem already. FLY generates a wide-field image from a collection of overlapping 2-D facets, each tangent to the celectial sphere, reducing the computation required to solve the full 3-D problem. In general, the facet size will be much smaller than the isoplanatic patch size, and the facet size is resolution dependent while the isoplanatic patch size is not. So, a patch will probably cover many facets.

In an N facet FLY, the visibilities are gridded onto N different Fourier planes, each conjugate to a facet plane, and Fourier transformed. The brightest feature in all facets is found. CLEANING in each facet proceeds independently until the maximum residual in each facet reaches the global maximum times the synthesized beams peak sidelobe level. At this point, the clean components which have been found in each facet are Fourier transformed and subtracted from the data visibilities. This ends one round of the major cycle. The major cycle is repeated until the desired image residual is achieved.

FLY can easily be modified to treat nonisoplanatic imaging. I assume that the gains for the isoplanatic patches have been determined from selfcal. As noted above, an isoplanatic patch probably spans severl FLY facets. However, the gains from adjacent patches may be different enough to warrent some interpolation of the gains in sky position. Some form of error analysis must be carried out to find out how small the image sections must be to produce the desired image fidelity. Smaller chunks will lead to less efficiency. We will ignore the interpolation problem in this discussion.

To image, apply the gains for the i^{th} isoplanatic patch to the visibilities. Grid them onto the Fourier planes and transform to produce the facets within the i^{th} isoplanatic patch. Loop over all isoplanatic patches, applying the appropriate gains. Again, conservative CLEANING proceeds, stopping in each facet when the maximum residual gets close to the global image peak before CLEANING started times the maximum sidelobe level. At this point, the clean components from each facet are Fourier transformed back to their respective visibility planes, the gains from the appropriate isoplanatic patch are *un-applied* to the component visibilities just generated, and the newly corrupted component visibilities are subtracted from the raw, uncalibrated visibilities. This is the end of the major cycle, which is repeated until the desired image residual is achieved. Both the Gainapply and Gain-unapply methods of TelModel are required by ImagingModel,

The inverse nonisoplanatic imaging method will not be developed in detail since it is not clear to us how to solve the noncoplanar baseline problem within its single-image context. It may be useful for VLBI or linear arrays. The inverse method requires the same services of TelModel or ImagingModel which the calibration and the direct non-isoplanatic imaging algorithm require.

2.3.3 Summary of Required Operations

- For each time interval, there is a mapping for each antenna of the ionosphere's isoplanatic patches onto the celestial sphere. Both ImagingModel and TelModel must know about this mapping.

- TelModel's Solve method must use the method PredictYegs for each patch on the source.

- There can be no "split" to apply all the gains to the raw visibilities; rather, the gains must be updated and the raw data visibilities must be kept.

- ImagingModel must use the TelModel.Gain-unapply and TelModel.Gainapply. (The inverse method only requires Gain-unapply.)