

# Towards a Wide-band Spectral Imaging Technique in Obit

W. D. Cotton, October 3, 2008

**Abstract**—This memo describes the work in progress on a technique for imaging wide-band radio interferometric data. The basic technique is to replace the traditional concept of an image as an array of scalar brightness values which that in which a pixel is the representation of a spectrum. Several approaches along this line are investigated. Many of the traditional robust imaging techniques have analogues in this paradigm. This memo describes an investigation using a simulated wide-band ( $\sim 50\%$  bandwidth) dataset. The technique of imaging and deconvolving narrow subsections of the total bandwidth and then deriving joint spectra in each pixel shows a lot of promise, producing dynamic ranges of well in excess of  $10^5$ .

**Index Terms**—Wide-band Imaging, Interferometry

## I. INTRODUCTION

ONE of the principle ways in which radio interferometer arrays currently under construction or in planning increase their sensitivities over current arrays is by a large increase in the bandwidth being sampled. This presents challenges to imaging the data from the new arrays as the large bandwidths violate one of the implicit assumption used in imaging, that the sky looks the same at all frequencies in the observed bandpass. This assumption can be relaxed somewhat by assuming that all the emission in the image has the same spectral shape but with a sufficiently wide bandpass and a range of spectral shapes in the primary antenna pattern, this assumption breaks down.

To reach the sensitivity which the new instruments are in theory capable, all of the sky in which there are detectable sources must be imaged in order to remove the side-lobes of these sources. At lower frequencies this essentially means that the entire primary beam must be imaged along with selected areas in the side-lobes. Thus, especially at lower frequencies, a wide-band imaging problem is also a wide-field imaging problem. This increases the potential for a range of spectral shapes in the image.

This memo explores the possibilities of replacing the “pixel as scalar” paradigm with “pixel as spectrum”. This exploration makes use of the rapid prototyping capabilities in the Obit ([1], <http://www.cv.nrao.edu/~bcotton/Obit.html>) package.

## II. IMAGING AND BANDWIDTH

One of the simplifying assumptions made in traditional imaging techniques is that the sky has a constant brightness across the bandpass. In general, the fractional bandpasses are so small that this is a reasonable approximation. This

assumption can be relaxed slightly by allowing a constant spectral shape (usually a spectral index) by scaling the data in a frequency dependent fashion.

With larger bandwidths, the problem becomes acute when a high dynamic range (ratio of peak brightness to off-source “noise” in the image) is also needed. This is the case when studying weak emission in the presence of much brighter emission. The ability to study fainter emission is the driving motivation behind increasing the sensitivity of interferometric arrays.

The techniques for obtaining high dynamic range are mostly variation on the visibility-based CLEAN (“Cotton-Schwab” CLEAN) of [2], [3] in which the model of the sky is iteratively refined by removing the current best estimate of the sky from the data and obtaining a better estimate of the remaining emission. This general technique requires accurately subtracting the sky model from the data. In the wideband-variable spectral index case, this requires knowing the spectral shape as well as the brightness of the emission.

Astrophysical sources of broadband emission generally emit by either thermal or synchrotron mechanisms. In either case, the spectral shape over large areas of the radio spectrum are close to a power law. This suggests the form of the spectrum to be used. The technique presented here uses the following adaptation of the traditional representation of a continuum source spectrum:

$$s_\nu = s_{\nu_0} e^{\alpha \log(\nu/\nu_0) + \beta \log(\nu/\nu_0)^2 + \dots} \quad eq.1$$

where  $s$  is the spectral flux density,  $\nu$  is frequency,  $\nu_0$  a reference frequency and  $\alpha, \beta, \dots$  are the “spectral index” and one or more curvature terms. As many terms can be used in the exponent as are needed to accurately represent the observed emission. This is the same as the traditional scalar pixel representation with the addition of the exponential term to give the spectral shape

In this model, a broadband continuum image becomes an cube in which the planes are the parameters of *eq. 1* for each pixel. With this changed definition of the meaning of a “pixel”, we can proceed with the redefining of the steps in image reconstruction.

## III. THE PROBLEM

Let’s explore the problem space by first generating a simulated wideband dataset with known properties and then apply the traditional techniques.

### A. Model Data

In order to test various techniques, a model wideband data set was created. This dataset was derived from a  $15 \times 390$

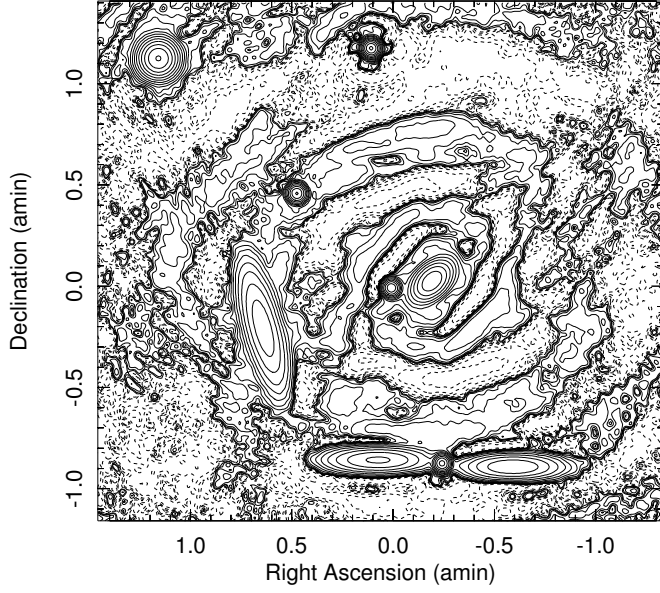


Fig. 1. Traditional imaging of all frequencies in the model noiseless wideband dataset. Contour levels are powers of 2 from 0.1 mJy/beam, the peak in the image is 3.7 Jy/beam. Negative contours are dashed

kHz channel VLA dataset at 1.4 GHz with 1 “IF” involving multiple snapshots. This dataset was expanded to 10 IFs spaced at 100 MHz intervals which (sparsely) covers the range 1.4 to 2.3 GHz. The data values were replaced by the Fourier transform of the models given in Table I. The columns in Table I are: component label, RA offset, declination offset, flux density, spectral index, curvature, Gaussian major axis FWHM, Gaussian minor axis, and Gaussian position angle. The flux density of each model was calculated from the spectral parameters for each channel. The models include a range of flux densities, sizes, shapes and spectra. For some tests, a known amount of zero mean Gaussian pseudo noise was added.

### B. Traditional Imaging

The simplest way of imaging this data is to grid all visibility data onto a single uv-plane grid and then Fourier transform to derive the dirty image. This was done using Obit Task Imager using all data in the (noiseless) model dataset and deconvolved using the “DFT” (more accurate) model calculation technique. The result of this imaging is shown in Figure 1. The off-source RMS is  $7.3 \times 10^{-4}$  and the “dynamic range” (ratio of peak to off-source RMS) in this image is about 5000; the image clearly has strong artifacts. On the other hand, if only a single IF is imaged (see Figure 2) the result is far better. Here the off-source RMS is  $1.0 \times 10^{-4}$  Jy/beam or a dynamic range of 35,000. Clearly making a traditional image using a wideband dataset is the wrong thing to do.

## IV. SPECTRAL IMAGING

This is the most extreme in a range of techniques explored in which spectral images are derived by fitting spectral parameters in each pixel of a set of dirty images and then performing a CLEAN using the flux density at the reference

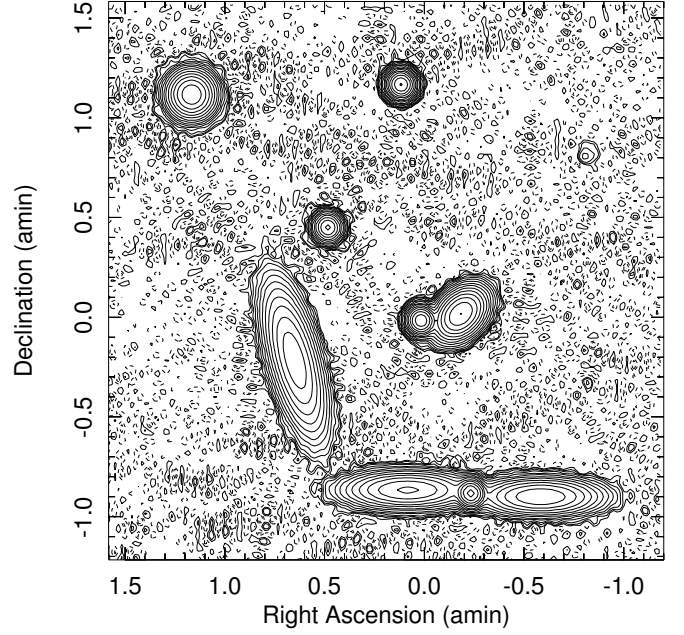


Fig. 2. Traditional imaging of one “IF” of the model noiseless wideband dataset. Contour levels are powers of 2 from 0.1 mJy/beam, the peak in the image is 3.5 Jy/beam. Negative contours are dashed

frequency. Various components of this technique are explored in the following.

### A. Forming the Dirty Image/Residual

The most straightforward path from a wideband visibility data set to a cube of spectra of the form given in eq. 1 is to first grid and transform the visibility data into an image cube. Since only broadband spectra are considered here, this image cube can be at fairly low spectral resolution but high enough that the constant spectrum over the frequency range approximation is not grossly violated. A spectral shape of the form in eq. 1 can then be fitted to the frequency samples in each cell of the image. Unfortunately, this fitting is non linear. In the implementation described here, a Levenberg-Marquardt least squares fitter from the Gnu Scientific Library was used.

As always, nonlinear least squares fitting must be approached with caution, fitting a high order function to noise has unpleasant consequences. In the technique described here, a variable number of terms from eq. 1 are fitted depending on the SNR and the results of a fit with fewer parameters. In order to evaluate the statistical significance of fitting results, “noise” estimates are made by a robust estimation in each of the frequency channel images; these RMS estimates are taken as the expectation of the scatter from a spectral fit and used to form weights for the spectral parameter estimation.

The initial parameter estimated in each pixel is  $s_{\nu_0}$  which is derived by a weighted average over all frequency images. If the  $\chi^2$  value of this fit is below a specified threshold (of order a few), and the SNR of the data is below another limit, then this fit is deemed acceptable and the terms in the exponential ( $\alpha, \beta, \dots$ ) are set to zero.

If the  $\chi^2$  value of this fit exceeds the cutoff, or the SNR is sufficiently high, a fit is attempted for  $s_{\nu_0}$  and  $\alpha$  using

weighted flux as the initial guess of  $s_{\nu_0}$  and 0 for  $\alpha$ . If the reduced  $\chi^2$  of this fit is lower than the single parameter fit, it is accepted, otherwise the  $\alpha$  and higher order terms are set to zero. If the reduced  $\chi^2$  of this fit is lower than the threshold and the SNR is below another threshold, no higher order fitting is done. In the case that the  $\chi^2$  exceeds the threshold, the next higher order term is attempted. This continues until either the  $\chi^2$  is acceptable or a maximum number of terms is fitted. This results in an image cube where the values in the planes correspond to terms in *eq. 1*. Error estimates can also be saved in further planes as well as the final  $\chi^2$  of the fit. The cube of fitted spectra becomes the analog of the dirty image in the traditional technique, hence should be referred to as the “dirty spectrum”.

The threshold effects when the order of the fit changes across the image introduce out-of-band artifacts that can drive CLEAN into serious oscillations if the CLEAN reaches that level. A Gaussian spatial frequency filter was used to reduce these out of band artifacts.

### B. Primary Beam Effects

With wide bandpasses, the power pattern of the antennas is quite variable across the bandpass, especially far from the pointing center. In the technique described here, correction for this effect for a homogeneous array can be deferred until after imaging is complete as the primary beam will steepen the spectrum and add extra curvature. The effect of the primary antenna pattern will be included in the spectral fitting.

### C. Synthesized Beam Effects

The synthesized beam also varies across the bandpass. A frequency variable beam causes trouble as in the process described here, it will be mapped into physically incorrect variations in spectral shape across a source, even across an unresolved source. Tapering of the higher frequencies can be used to force the beams to be (approximately) the same at all frequencies. An unfortunate consequence of this is that it reduces the sensitivity at the higher frequencies and reduces the resolution to those of the lower frequencies.

In the technique implemented here, the lowest frequency channel in the cube was imaged first and a 2D Gaussian fitted to the dirty beam. The expected beam in each of the higher frequency channels was then estimated from a simple frequency scaling. On the basis of this estimated beam, the equivalent image plane Gaussian convolution size was determined and then converted into a taper (including orientation of the ellipse) to be applied to each channel image. The UV coverage is not uniform, especially in the outer parts of the UV plane which determine the resolution of an image so the tapering approach does not precisely result in the desired beam size (as determined by Gaussian fitting); but with a fudge factor, seems to be adequate in the tests described below. The parameters of the taper used were:

$$taper_{major} = 0.95 \times 0.001 / (CB_{major} * const)$$

$$taper_{minor} = 0.95 \times 0.001 / (CB_{minor} * const)$$

$$taper_{posAngle} = (CB_{posAng})$$

where  $const = \left(\frac{2\pi}{360}\right) \frac{\sqrt{0.5}}{1.17741022}$ , and  $CB$  is the equivalent image plane convolving Gaussian. The fudge factor to correct for the sparseness of the UV coverage is 0.95.

The fudge factor depends of the weights, when all the weights were set to 1, the fudge factor became 1.05 or a bit larger; these need to be determined empirically.

The cell spacing must be the same for all frequency planes. The tapering to give constant synthesized beam size with frequency should keep the higher frequency images from being under sampled.

### D. Redefinition of CLEAN component

The flux density in the traditional CLEAN component needs to be replaced by a spectrum. This is easily accomplished by adding terms for the coefficients of the powers of  $\log(\nu/\nu_0)$  terms in *eq. 1*.

### E. CLEAN Using Spectra

The term  $s_{\nu_0}$  from *eq. 1* can be used as a simple replacement in CLEAN for the flux density of the traditional narrow band image. The only modification required to CLEAN is that it must keep track of the spectral terms from each pixel in the image and record them in the CLEAN component. The “dirty beam” used for the CLEAN is the average of the dirty beams derived for the planes in the spectral cube.

Variations in the number of terms fitted can cause low level discontinuities in the  $s_{\nu_0}$  image. This introduces “out-of-band” noise which can cause CLEAN to go into oscillations. The out-of band spatial frequencies can be filtered, or the CLEAN can be prohibited from going deep enough in each major cycle to get into trouble.

### F. Calculating the Sky Model

There are several techniques used to compute the instrumental response to a sky model, the “Direct Fourier Transform” (“DFT” in AIPS and Obit) is the simplest and the only implementation to date. This technique calculates the response of the interferometer to each CLEAN component in each visibility/frequency/IF. The necessary information to calculate the exponential term is carried in the CLEAN component. A simple frequency dependent factor applied to the flux density of each CLEAN component is sufficient to make the correction.

### G. Spectral Imaging Technique

The general spectral imaging method is illustrated in Figure 3, the blocks of which are described in the following list:

- Form Coarse Spectral Cube  
The multi-channel continuum data set is imaged averaging multiple channels but with enough spectral resolution that the images are not seriously degraded by the variation in spectral shape across the field. Frequency dependent UV tapering should be applied to obtain an (approximately) constant resolution for all planes of the

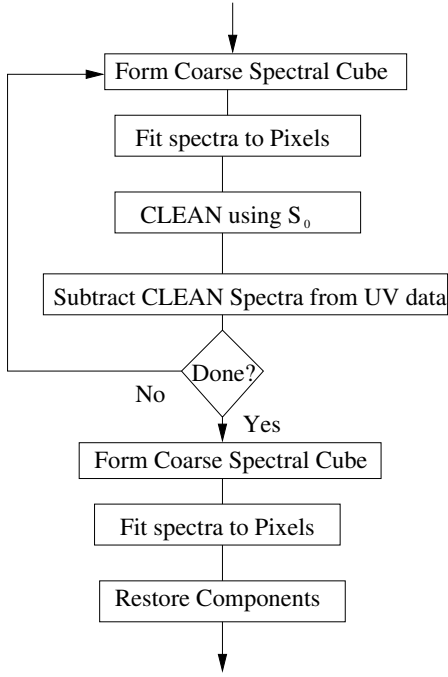


Fig. 3. Block diagram of the spectral imaging technique.

output cube. The same image grid should be used for all frequencies.

- Fit spectra to pixels

A spectrum is fitted using non-linear least squares to the spectral samples in each pixel producing a cube of the parameters in *eq. 1*. The number of terms actually fitted depends on the SNR of the samples in each pixel. At least the  $s_{\nu_0}$  term is fitted for each pixel. The resultant images are filtered to reduce the “out-of-band” noise.

- CLEAN using  $s_{\nu_0}$

The  $s_{\nu_0}$  image is used as in the traditional CLEAN except that the resultant CLEAN components include the spectral terms from the image pixel from which they were derived.

- Subtract CLEAN spectra from UV data

The CLEAN components contain a spectrum which is used to compute the instrumental response to the sky model in each frequency channel. These frequency dependent model values are subtracted from the visibility samples in the corresponding frequency channels to produce a residual UV dataset.

- Restore Components

After the CLEANing has converged, a final spectral image cube is formed and the  $s_{\nu_0}$  values of the CLEAN components removed in the CLEANing process are restored to the residual  $s_{\nu_0}$  image.

#### H. Evaluation of Spectral Imaging

In order to evaluate the results of the spectral imaging technique, a model data set with 0.013 Jy/visibility Gaussian noise added was processed as in the previous discussion. This noise level corresponds to  $\sim 1.0 \times 10^{-5}$  Jy/beam in images

TABLE I  
Model Parameters

	X "	Y "	S Jy	$\alpha$	$\beta$	Maj "	min "	PA °
A	-50.0	50.0	0.001	0.0	0.0	0.0	0.0	
B	-50.0	-20.0	0.0002	0.0	0.0	0.0	0.0	
C	-36.0	-53.0	2.0	-0.75	-0.01	15.0	3.0	90.0
D	-15.0	-52.0	1.0	0.25	0.0	0.0	0.0	
E	-12.0	2.0	5.0	-1.25	-0.033	7.0	4.0	135.0
F	0.0	0.0	1.5	0.0	0.0	0.0	0.0	
G	4.0	-51.0	2.5	-0.65	-0.05	16.0	3.0	90.0
H	6.0	71.0	3.5	0.25	-0.15	0.0	0.0	
I	28.0	28.0	1.0	0.0	-0.0	1.0	1.0	
J	38.0	-12.0	3.0	-0.55	-0.025	20.0	5.0	15.0
K	69.0	68.0	4.0	-0.05	-0.1	5.0	5.0	

TABLE II  
Fitted Parameters (Spectral Fitting)

	X "	Y "	S Jy	$\alpha$	$\beta$	Maj "	min "	PA °
A	-50.0	50.0	0.00077	-0.39	0.94	0.0	0.0	
B	-49.0	-20.0	0.00005			0.0	0.0	
C	-36.0	-53.0	2.01	-0.81	0.00	15.0	3.0	90.0
D	-15.0	-52.0	1.01	0.17	0.10	0.0	0.0	
E	-12.0	2.0	5.02	-1.28	0.00	7.0	4.0	135.0
F	0.0	0.0	1.51	-0.06	0.08	0.0	0.0	
G	4.0	-51.0	2.51	-0.70	0.00	16.0	3.0	89.9
H	6.0	71.0	3.51	0.20	-0.08	0.0	0.0	
I	28.0	28.0	1.00	-0.08	0.12	1.0	1.0	
J	38.0	-12.0	3.01	-0.57	0.07	20.0	5.0	15.0
K	69.0	68.0	4.02	-0.12	0.00	5.0	5.0	

derived from a single IF. The processing proceeded with the spectral fits including up to 3 spectral terms ( $s_{\nu_0}$ ,  $\alpha$ ,  $\beta$ ).

Once the strongest emission was CLEANed, the spectral fitting became less robust so an additional imaging was performed with only the first two spectral terms fitted. The results of spectral imaging using a two or three term spectral fit are shown in Figure 4 which gives the  $s_{\nu_0}$  images. At high flux density levels the derived images show excellent agreement with the model.

At lower flux levels, the results were disappointing; the off-source RMS levels were well above the expected level and only marginally better than the single IF traditional image shown in Figure 2. The RMS levels were  $6.9 \times 10^5$  and  $9.4 \times 10^5$  Jy/beam for the 2 and 3 term fits; this gives dynamic ranges of 50,800 and 37,200 respectively. The 1 mJy faint source is visible in the images in Figure 4 but not the 200  $\mu$ Jy source.

#### V. SPECTRAL FITTING

The spectral imaging process consists of two nonlinear components, the spectral fitting and the CLEAN. Of these, the CLEAN seems more robust and doing the spectral fitting to

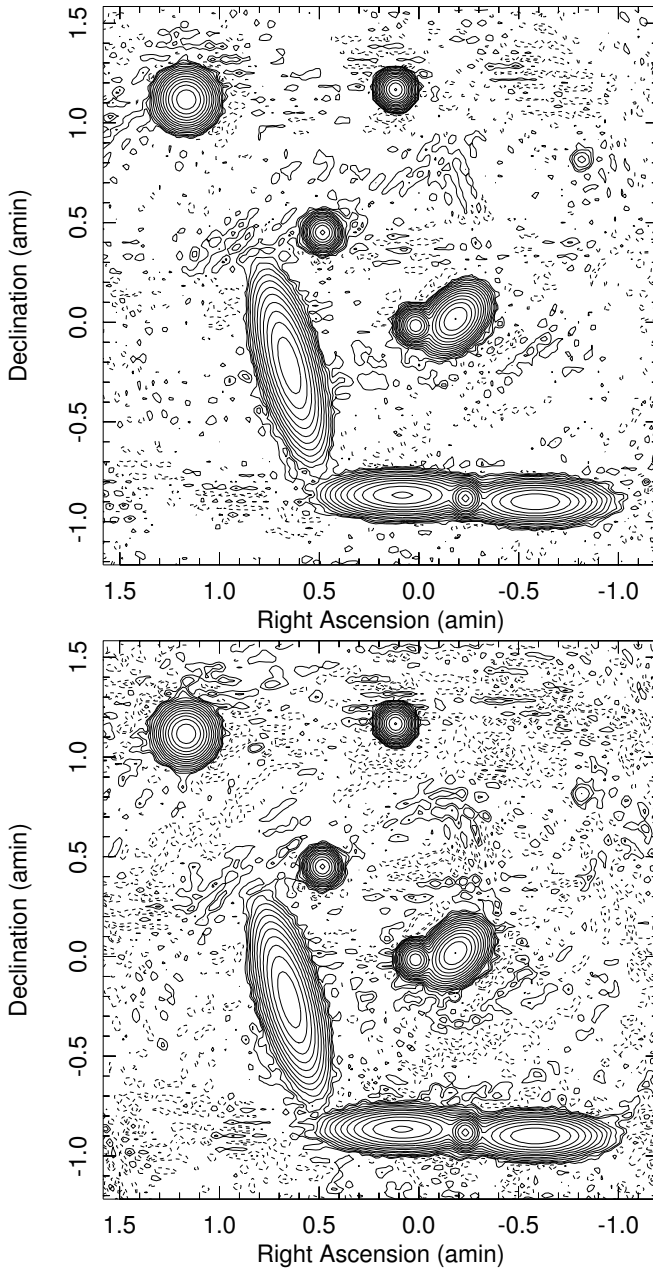


Fig. 4. Spectral imaging  $s_{\nu_0}$  results using a model set with 0.013 Jy/visibility added noise. On the top is the result using a two term spectrum (flux density and spectral index), and on the bottom using a three term (also curvature) spectral fit. Both plots use the same contouring with levels spaced by powers of 2.0 from  $1.0 \times 10^{-4}$  Jy/beam.

the dirty/residual images seems to be limited by artifacts well above the expected level. Reversing the order of the nonlinear operations may produce superior results. This process will be referred to in the following as “Spectral Fitting” to distinguish it from the “Spectral Imaging” described in the previous.

In the spectral fitting process, each IF is imaged independently and deconvolved using a visibility-based CLEAN using the “DFT” model calculation. Then, the CLEAN images are convolved to a common resolution, accumulated into a spectral cube and spectra fitted in each pixel in the same way as was done for the spectral imaging case on dirty/residual cubes. The results for two different noise levels are given in Figure

TABLE III  
COMPARISON OF METHODS AND TESTS

Method	RMS (mJy)	DR
Imager 1 IF <sup>1</sup>	0.100	35400
Imager all IFs <sup>1</sup>	0.730	5100
Sp. Image 2 term <sup>1</sup>	0.069	50800
Sp. Image 3 term <sup>1</sup>	0.094	37200
Sp. Fit <sup>1</sup>	0.014	283000
Sp. Fit <sup>2</sup>	0.034	104000

Notes:

<sup>1</sup> The data had added noise in the amount of 0.013 Jy/visibility.

<sup>2</sup> The data had added noise in the amount of 0.13 Jy/visibility.

5. The python function used for this operation is given in the appendix.

For the data with 0.013 Jy/visibility added, the expected noise level per IF is  $\sim 1.0 \times 10^{-5}$  Jy/beam and the measured off-source RMS was  $1.0 \times 10^{-4}$  in a single IF image and  $1.4 \times 10^{-5}$  Jy/beam for the full band value; this corresponds to a dynamic range of 253,000. These images appear to be dynamic range rather than noise limited; possibly due to the limited uv coverage of the data set. The reduction in RMS faster than  $\sqrt{n}$  also suggests the presence of strong systematics. Gaussians fitted to the images are given in Table II and can be compared to the input parameters in Table I. Spectral parameters in Table II are derived from the value in the appropriate plane corresponding to the pixel of maximum flux density.

The other test shown in Figure 5 used data with 0.13 Jy/visibility added which should give a single IF noise level of  $1.0 \times 10^{-4}$ . In this case, the measured single IF RMS was  $1.4 \times 10^{-4}$  and for the full band fit,  $3.4 \times 10^{-5}$ . Here, the RMS values are closer to what was expected due to the noise added although the single IF images may still be suffering from some dynamic range limitation. The full band dynamic range achieved was 104,000. Both the 1 mJy and 200  $\mu$ Jy faint sources are visible in the images in Figure 5. A comparison of the various methods used is summarized in Table III.

## VI. DISCUSSION

The preceding section has demonstrated that the spectral fitting technique (CLEAN narrow band images then do spectral fitting) produces significantly better results than “spectral imaging” in which the order of these operations is reversed. Dynamic ranges of more than 200,000 were obtained using a relatively complex source model which allowed the easy recovery of a source 1/17500 th of the brightest in the field. This is basically a proof of concept of the method which now needs further development and conversion into production quality software for further evaluation and use. The following list discusses areas that need further investigation or development.

- Integration into Obit library/tasks

The current implementation of the spectral fitting is in

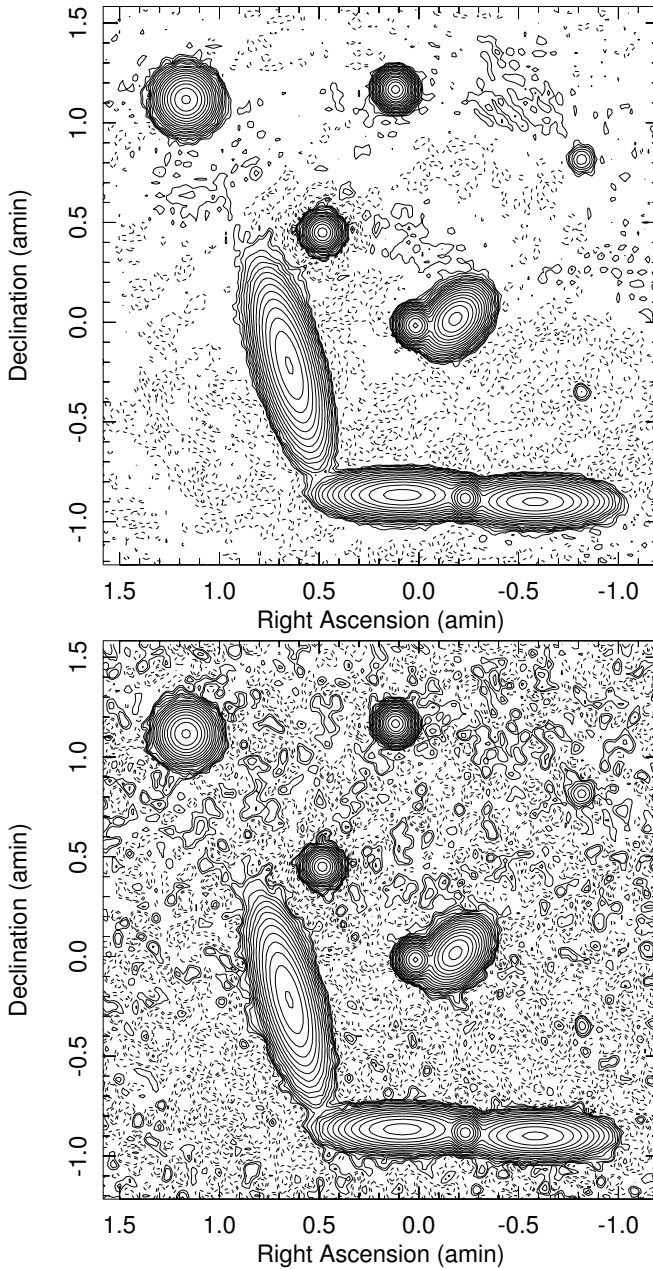


Fig. 5. Spectral fitting  $s_{\nu_0}$  results. On the top using a model set with 0.013 Jy/visibility added noise, and on the bottom 0.13 Jy/visibility. Both plots use the same contouring with levels spaced by powers of 2.0 from  $2.0 \times 10^{-5}$  Jy/beam.

the form of a python function that should be translated to c allowing it to be used by tasks.

- Remove typical spectral index

The majority of sources in a field observed at GHz frequencies are optically thin synchrotron sources with spectral indexes ( $\alpha$ ) of around -0.7. The data could be scaled in frequency to remove this spectral index which would then become the effective default as opposed to the current default of 0.0. Any spectral index removed during the processing would need to be restored at the end.

- Adapting to use with self-calibration

The spectral image produced by spectral fitting can be

used in the self calibration process. The spectral image can be decomposed into a set of CLEAN components with attached spectra by “CLEANing” the final image as was done in spectral imaging although using the restoring beam instead of the dirty beam. These components can then be used in routines developed for spectral imaging to compute the frequency dependent sky model for self calibration.

- Primary beam corrections

Since the  $s_{\nu_0}$  image is at the reference frequency, the primary beam correction for it is the same as for a traditional image. The effect of the primary beam on the other spectral terms is to steepen and add curvature to the spectrum. Primary beam corrections to the spectral terms are needed.

- Heterogeneous Arrays

The techniques described here were designed for and tested on simulated from from a heterogeneous array. The technique needs to be expanded for arrays like eMerlin which have a variety of antenna sizes. This could possibly be implemented by explicitly making primary beam corrections in the imaging and deconvolution stages.

- Control fitting threshold effects

The spectral fitting was adequate for the tests done here but need improvement; especially the threshold effects of variations from pixel to pixel of the number of terms added needs work.

## VII. CONCLUSION

This memo has described the work in progress on the spectral fitting technique to produce high dynamic range images of data with a wide bandpass. The technique is basically to replace the concept of a pixel as a scalar brightness with one of the pixel as a spectrum. The spectral fitting technique applied to simulated wideband data gave a dynamic range significantly better than the traditional technique when limited to a narrow-band section of the same data. Further investigation and development of this technique appears warranted.

## ACKNOWLEDGMENT

The author would like to thank Ed Fomalont and Juan Uson for discussions on this topic.

## APPENDIX

The following python scripts show the processing used for spectral fitting.

## REFERENCES

- [1] W. D. Cotton, “Obit: A Development Environment for Astronomical Algorithms,” *PASP*, vol. 120, pp. 439–448, 2008.
- [2] F. R. Schwab, “Optimal Gridding of Visibility Data in Radio Interferometry,” in *Indirect Imaging. Measurement and Processing for Indirect Imaging. Australia, Cambridge University Press, Cambridge, England, New York, NY, L C # QB51.3.E43 153 1984. ISBN # 0-521-26282-8. P.333, 1983, 1983*, pp. 333–.
- [3] W. D. Cotton, ““Special problems in imaging,”” in *Synthesis Imaging in Radio Astronomy*, ser. ASP Conference Series, R. A. Perley, F. R. Schwab, and A. H. Bridle, Eds., no. 6. Astronomical Society of the Pacific, 1989, pp. 243–246.

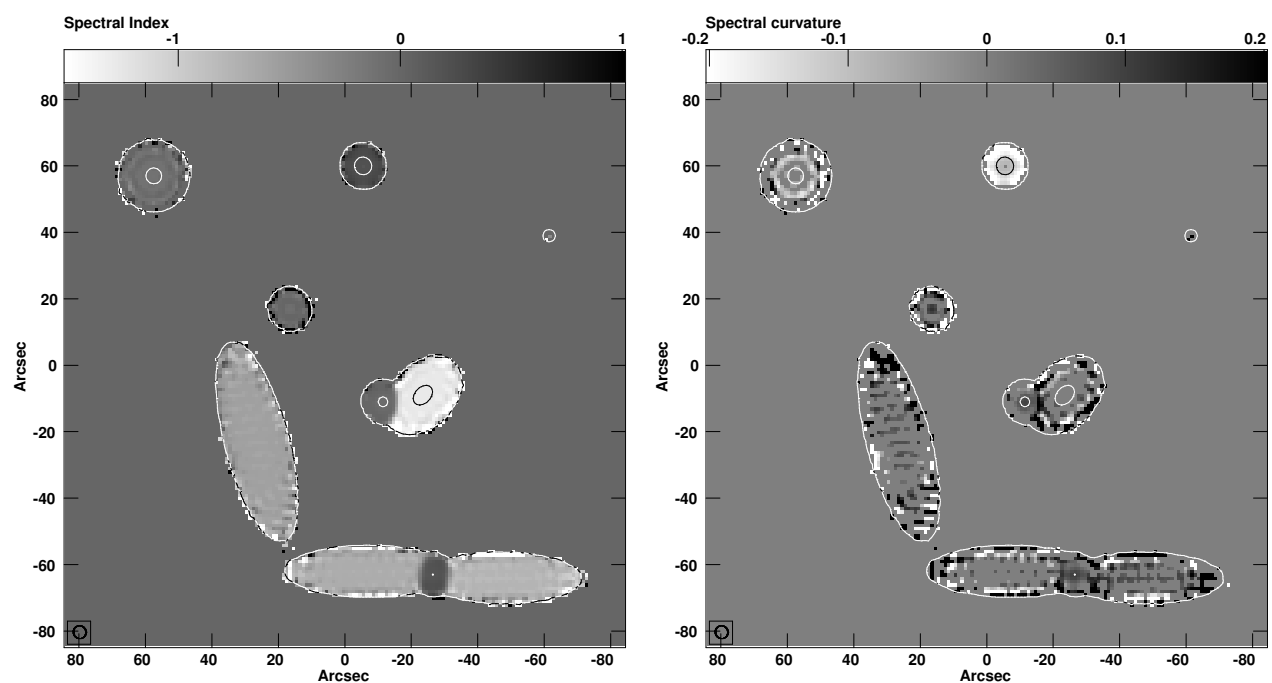


Fig. 6. Spectral fitting spectral results for the model set with 0.013 Jy/visibility added noise. Contours of  $s_{\nu_0}$  are shown at 1 mJy/beam and 1 Jy/beam. Values are given by the scale bar at the top, unfitted values are shown as 0.0. On the left is spectral index and on the right is spectral curvature.

# Python routine to make a wideband image mosaic by spectral fitting

```
import UVImager, ImageMosaic, Image, UV, Table, FArray, SpectrumFit, OErr
import CleanVis, ConvUtil
import math
```

```
def MakeWBClean (inUV, err, \
                 outName='CleanWB', outCClass='Cube', outSClass='Spect', outDisk=1, outSeq=1,
                 FOV=3.0/60., cells=1.0, robust=0.0, gain=0.03, niter=25000, minFlux=0.0, \
                 winds=[], maxChi2=5.0, nterm=3, dispURL="None"):
    """ Create a wideband image with spectrum from an ObitUV
```

```
    returns (spectral_cube, spectral_fit)
```

Each IF is CLEANed, convolved to the lowest resolution, formed into a cube  
and Spectrum fitted in each pixel to produce the output mosaic.

```
    inUV      Data to be imaged
    err       message/error object
    outName   Name of output AIPS image
    outCClass Class of output AIPS spectral cube image
    outSClass Class of output AIPS fitted spectrum image
    outDisk   output disk
    outSeq    output sequence
    FOV       Field of view in deg
    cells     cellspacing in asec
    gain      CLEAN loop gain
    niter     max Number of CLEAN iterations
    minFlux   min. level to CLEAN to
    winds     CLEAN windows
    robust    Briggs robust weighting factor
    nterm     number of terms in spectrum including flux
    """
```

# define clean

```
CleanInput = CleanVis.CleanInput
CleanInput["Niter"]      = niter
CleanInput["Gain"]       = gain
CleanInput["minFlux"]    = minFlux
CleanInput["autoWindow"] = True
CleanInput["Stokes"]     = "I"
CleanInput["Mode"]       = 1
CleanInput["Robust"]     = 0.0
CleanInput["Type"]       = 1 # 'AIPS'
CleanInput["Name"]       = 'TempWB'
CleanInput["Class"]      = 'Temp'
CleanInput["Seq"]        = 0
CleanInput["Disk"]       = outDisk
CleanInput["NField"]     = 0
CleanInput["FOV"]        = FOV
CleanInput["doFull"]     = True
CleanInput["xCells"]     = cells
CleanInput["yCells"]     = cells
CleanInput["doRestore"]  = True
CleanInput["doFlatten"]  = True
CleanInput["BChan"]      = 1
CleanInput["EChan"]      = 0
CleanInput["PBCor"]      = False
CleanInput["dispURL"]    = dispURL
CleanInput["doCalSelect"] = True
```



```

# Get IF frequencies from FQ table
refFreq=inUV.Desc.Dict["crval"][inUV.Desc.Dict["jlocf"]]
FQtab=inUV.NewTable(Table.READONLY,"AIPS FQ",1,err)
FQtab.Open(Table.READONLY,err)
FQrow=FQtab.ReadRow(1,err)
FQtab.Close(err)
OErr.printErrMsg(err,"Error reading frequencies ")
numIF = len (FQrow["IF FREQ"])
freq = []
for f in FQrow["IF FREQ"]:
    freq.append(refFreq+f)

# Loop over IFs - image all frequencies in each
cube = None # No spectrum cube yet
beams = [] # Restoring beams
for IF in range(1,numIF+1):
    print "Start IF",IF

    # Create CLEAN object
    CleanInput["Seq"] = 10+IF
    CleanInput["BIF"] = IF
    CleanInput["EIF"] = IF
    clean = CleanVis.PCreate("Clean", inUV, err, input=CleanInput)

    # Any windows
    for win in winds:
        clean.AddWindow(1, win, err)

    # Image/Clean
    CleanVis.PClean(clean, err)
    OErr.printErrMsg(err,"Error imaging/cleaning IF "+str(IF))

    # Extract Flattened Clean plane from mosaic
    mosaic = CleanVis.PGetMosaic(clean)

    full = ImageMosaic.PGetFullImage(mosaic, err)
    if full.me==None:
        del full
        full = ImageMosaic.PGetImage(mosaic, 0, err)
    # output cube exist yet?
    if cube==None:
        cube = Image.newPAImage("cube",outName,outCClass,outDisk,outSeq,False,err)
        d = full.Desc.Dict
        d["inaxes"][d["jlocf"]] = numIF
        d["cdelt"][d["jlocf"]] = freq[1] - freq[0]
        cube.Desc.Dict = d
        cube.Open(Image.WRITEONLY, err)
        cube.Close(err)
        # Make scratch copys of full for convolution
        tmpImage1 = full.Scratch(err)
        tmpImage2 = full.Scratch(err)
        OErr.printErrMsg(err,"Error creating output cube ")

```

```

    # Collect into Cube
    full.Open(Image.READONLY,err)
    full.Read(err)
    # Set up output
    plane = [IF,1,1,1,1]
    cube.Open(Image.READWRITE, err)
    cube.PutPlane(full.FArray, plane,err)
    full.Close(err)
    cube.Close(err)
    OErr.printErrMsg(err,"Error copy to cube for IF "+str(IF))

    # Save beam information
    h=full.Desc.Dict
    Beam = [h["beamMaj"], h["beamMin"] ,h["beamPA"]]
    beams.append(Beam)

    # Cleanup
    ImageMosaic.PZapImage(mosaic, -1, err)
    OErr.printErrMsg(err,"Error cleaning up mosaic "+str(IF))
    del clean
    # End Image/CLEAN IF loop

# Convolve all to same resolution as IF 1 and collect into cube
for IF in range(2,numIF+1):
    plane = [IF,1,1,1,1]
    cube.Open(Image.READWRITE, err)
    # Get convolution structures
    refBeam = beams[0]
    IFBeam = beams[IF-1]
    Beam = ConvUtil.Deconv(refBeam, IFBeam)
    convFn = ConvUtil.PGaus(cube, Beam)
    # Normalize peak
    scale = (refBeam[0]/IFBeam[0])*(refBeam[1]/IFBeam[1])

    # Get plane from cube
    cube.GetPlane(cube.FArray, plane,err)
    tmpImage1.Open(Image.READWRITE, err)
    tmpImage1.WriteFA(cube.FArray,err)
    tmpImage1.Close(err)
    ConvUtil.PConv(tmpImage1, convFn, False, scale, tmpImage2 ,err)
    tmpImage2.Open(Image.READONLY, err)
    tmpImage2.ReadFA(cube.FArray,err)
    tmpImage2.Close(err)
    cube.PutPlane(cube.FArray, plane,err)
    cube.Close(err)
    OErr.printErrMsg(err,"Error convolving IF "+str(IF))
    del convFn
    # End convolution loop

# Spectral fitting
spec = Image.newPAImage("spectrum",outName,outSClass,outDisk,outSeq,False,err)
sf = SpectrumFit.PCreate("fitter", nterm)
sf.Cube(cube, spec, err)
OErr.printErrMsg(err,"Error fitting spectrum to cube ")

# Done
del tmpImage1, tmpImage2
return (cube, spec)
# end MakeWBClean

```