VLBA SCIENTIFIC MEMO No. 11

High Dynamic Range Imaging with the VLBA

T.J. Cornwell, A.J. Kemball and J.M. Benson

July 5, 1995

Abstract

We show high dynamic range noise-limited VLBA images of DA193 (peak/rms $\sim$ 110,000) and CTD93 (peak/rms $\sim$ 7,000) at λ6cm. We describe the procedures for editing, calibration and imaging that are required for such high dynamic range. We also describe a number of causes of closure error in the VLBA correlator and how these are corrected in postprocessing.

1 Introduction

In VLBA memo 697A, Briggs *et al.* showed a noise-limited VLBA image of DA193 at λ6cm. The aggregrate bandwidth was 1.75 MHz, resulting in a noise level of 264 μJy/beam.

Since this limited bandwidth image was noise-limited, the obvious next step was to expand the aggregrate bandwidth to the current limit (64MHz) and see if a noise limited image could be obtained at this deeper sensitivity (about 45-50 μJy/beam for a full track). This memo gives the answer to that question (Yes, eventually). It also describes how to acheive such high dynamic range and so it is intended to act as a cookbook for high dynamic range imaging with the VLBA. It also documents improvements made in the $\mathcal{AIPS}$ calibration and SDE imaging tasks required for high dynamic range imaging.

2 Observations of DA193

Two different sets of observations of DA193 were made, both having aggregrate bandwidth of 64 MHz but the first split into 4 channels of 16 MHz and the latter into 8 channels of 8 MHz. Since it is known from both design and experiment that the 16 MHz filters have substantially higher unevenness than the 8 MHz filters, a comparison of the two observations is a good test.

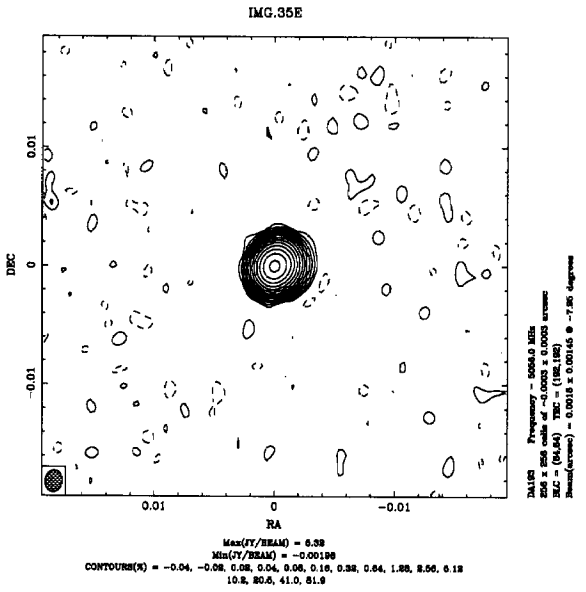

Figure 1: Final 2MHz image from Briggs *et al.*

2.1 4 16MHZ channels

The observations (called **4x16**) took place from 09:00 to 00:25 UT on July 9 and 10, 1994. The source was tracked from horizon to horizon, although in imaging only elevations greater than 10 degrees were used. All VLBA antennas functioned without a significant problem, except for Mauna Kea, where tape drive problems caused occasional intermittent loss of recording until 23:21 when the recorder failed completely and was out for the rest of the observation (about 1 hour).

The weather was good at all sites.

The data was recorded in VLBA 1:4 mode on two thin tapes.

The data were correlated once, on July 20 and 21 1994, using clocks determined from GPS measurements. This correlation run was called **4x16C1**. We note in passing that we find it useful to name separately observation and correlation runs and will use that convention below.

By default at that time, the samples input to the FFT stage were subject to Hanning windowing (see below). No overlapping of segments was possible because the required FFT rate was the maximum possible.

The playback quality was good at all stations, except for tape 1 from OV (a known capstan problem) and tape 2 from FD (cause unknown).

The integration time was 3.9s and 16 spectral channels were output

for each IF channel.

2.2 8 8MHZ channels

The observations (called **8x8**) took place from 02:00 to 17:25 UT on November 12, 1994. The source was tracked from horizon to horizon, although in imaging only elevations greater than 10 degrees were used. All VLBA antennas functioned with only one significant problem, the loss of BBC 5 at Mauna Kea.

The weather was astoundingly bad at the central three Southwest VLBA antennas: Pie Town, Kitt Peak and Los Alamos. All suffered heavy rainfall for at least 50% of the time DA193 was above the horizon. Kitt Peak received about 3cm of rain in 3 hours. The system temperature at all three sites fluctuated by up to 50% at times. The weather at the other sites was good.

The data were correlated three times. The first (called **8x8C1**), run on November 28 and 29 1994, used the clock values determined from GPS measurements as is standard. The second (called **8x8C2**), run on January 3 and 4 1995, used corrections to the delays determined from fringe fits to IF 1 from the first data set. The corrections were spread by up to 250 ns (max-min), and in **8x8C2**, the spread in the delays of the other 7 IFs reduced by about an order of magnitude. Both **8x8C1** and **8x8C2** used Hanning windowing of the samples input to the FFT in the VLBA correlator. The third correlation, **8x8C3**, has the same parameters as **8x8C2** except that uniform windowing was used. No overlapping of segments was possible.

For **8x8C1**, the playback quality was good for all stations except for KP, which suffered poor quality in reverse on tape 1. For **8x8C2** and **8x8C3**, the playback quality was uniformly good.

For all three correlations, the integration time was 3.9s and 16 spectral channels were output for each IF channel.

3 Calibration

Calibration proceeded as described in the $\mathcal{AIPS}$ cookbook chapter on VLBI data processing.

3.1 Amplitude calibration

For all datasets, the amplitude calibration was determined using the VLBA calibration file and the $\mathcal{AIPS}$ program ANCAL.

3.2 Fringe fitting

For all dataset, fringe fitting was performed using a two minute solution interval. No attempt to fit multi-band delays was made.

3.3 Editing

For high dynamic range imaging, editing of the data is crucial. Zero or very low data points are especially damaging to amplitude self-calibration and considerable care must be taken to expunge any such samples. Editing of VLBA data can be time-consuming since a large number of IFs are usually present. We found that the following steps usually suffice to edit the data satisfactorily:

Cal file The VLBA calibration file contains information on flagging of data due to various observing time events such as tape reversals. This is applied using *AIPS* task UVFLG. Since it is based upon record time information, it tends to miss flagging the initial sample in a scan which may often be of low weight.

***AIPS* task IBLED** IBLED allows interactive editing of data baseline by baseline. Running through a complete set of baselines with 60s integration time takes about 1 hour of real time. To speed up the process, we sometimes edited data based upon the average across all (calibrated) IFs. This is adequate for catching short dropouts that affect all IFs simultaneously but should be used carefully if substantial editing is apparently required.

Rejecting low weight samples The weight in VLBA visibility measures playback quality, a weight of 1.0 denoting that all expected sample points were passed. A somewhat draconian but effective way to trap poor playback is to flag all points with weight less than 0.95, say. We did this using UVCOP.

Examination of bandpasses Careful examination of the BP table using *AIPS* task POSSM helps with diagnosis of data problems. Either the average BP shows extra noisy bandpasses, or plots of the bandpasses for each scan can localize bad data.

***AIPS* CLIP** It is worth running *AIPS* task CLIP to reject low points, usually zeroes. We recommend that this be used mainly as a check and that if substantially flagging is performed by CLIP, the bad data should be chased down more carefully using IBLED.

In practice, it is often necessary to go through the calibration and editing more than once since IBLED is best run on calibrated data.

We found one important anomaly in the **4x16C1** fringe fits. Delay solutions for LA and KP showed a ripple of one hour period and 1 ns

amplitude. Since this also showed up in the cable calibration data, we believe it to be related to mild temparature cycling.

4 Imaging

4.1 NNLS and visamat

Briggs *et al.* argued that imaging DA193 at dynamic range exceeding 10000 requires use of the NNLS algorithm rather than CLEAN. In their work, this was possible using SDE programs **closure**, **uvmap**, and **svdconv**. For convenience, we combined the steps in these programs into one interactive SDE program **visamat**. All the images shown here were made using **visamat**.

In the SDE program **visamat**, the deconvolution problem is expressed as a matrix equation: $AX = B$. In this formulation X is a vector of the unknown pixel values in the deconvolved image and B represents the observed data that constrains X via the measurement matrix A. In **visamat**, the formulation can proceed in either the image or the Fourier plane. In the image plane, B is the set of pixel values of the dirty image and A contains the dirty PSF elements connecting pixels in X and B. In the Fourier plane, B represents the real and imaginary parts of the visibility samples, and A contains cos and sin terms that express the Fourier transform relation between X and B. The choice between the two formulations is mainly a practical matter dependent upon minimizing the size of A. Most of the time, the image plane formulation requires substantially less memory.

The three algorithms in **visamat** can be expressed in terms of the matrix equation:

Briggs' NNLS algorithm The matrix equation $AX = B$ is solved subject to the constraint that all elements of X are non-negative. See Briggs' Ph.D. thesis for further details at URL:

http://www.aoc.nrao.edu/ftp/dissertations/dbriggs/diss.html

Hogbom's CLEAN algorithm The normal equation $[A^T A]X = A^T B$ is solved iteratively in the well-known way. [1]

MEM A maximum entropy algorithm is also included. It solves $AX = B$ subject to an rms fit constraint and a constraint on the total flux of X. This is a high-accuracy algorithm which should

[1] A subtle point is that if the formulation proceeds in the image plane, then the distinction between A and $A^T A$ is usually glossed over. This is valid only if A is square (*i.e.* the number of constraint pixels equals the number of free pixels) and is reasonable if the data is uniformly weighted.

be capable of acheiving substantially higher dynamic range than VTESS or vm.

In **visamat**, the operations are actually performed in terms of the actual A matrices. Consequently the memory usage can be very large and for our largest memory IBMs (anasazi and ringtail) limits the number of non-zero pixels to about 6000-8000. For the NNLS algorithm, the large memory usage is unavoidable. For CLEAN and MEM, this is a silly way to write the algorithms except if the PSF is spatially variant. However, accepting the large memory usage, both algorithms run quickly compared to NNLS.

To demonstrate the quality of imaging possible (after the closure errors discussed in the next section have been corrected), we show in figures 2 and 3 the final Stokes I and Stokes V images from the **8x8C3** dataset. The noise level in the I image is 49 μJy/beam, and that in the Stokes V image is 45 μJy/beam. These are consistent with the theoretical noise for uniform windowing of the samples input to the FFT stage of the correlator. The corresponding visibility data is shown in figure 4.

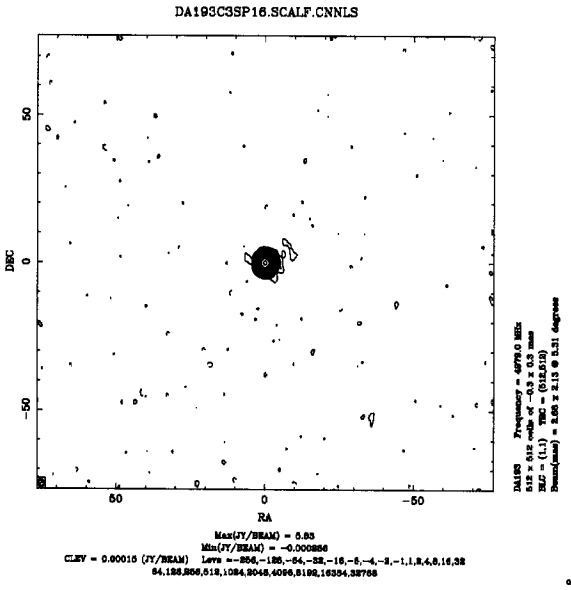

Figure 2: Final Stokes I image from **8x8C3** dataset. Lowest contour level is 150 μJy/beam ($3\sigma_I$)

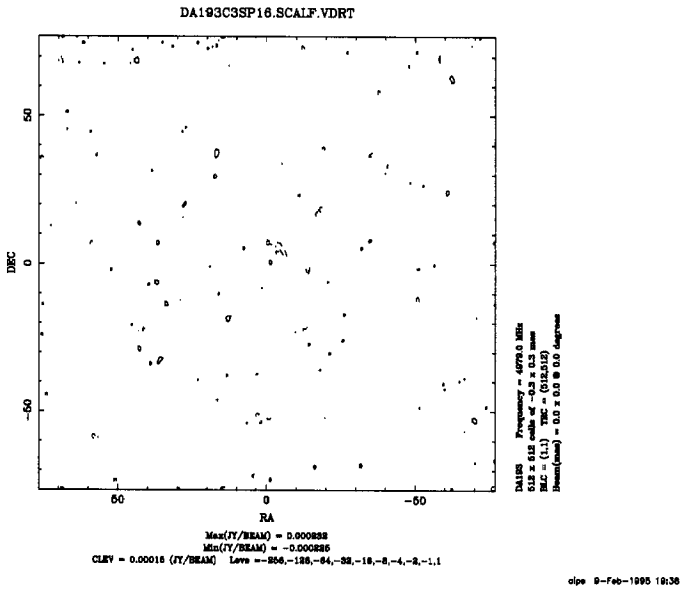

Figure 3: Final Stokes V image from **8x8C3** dataset. Lowest contour level is 150 μJy/beam ($3\sigma_V$)

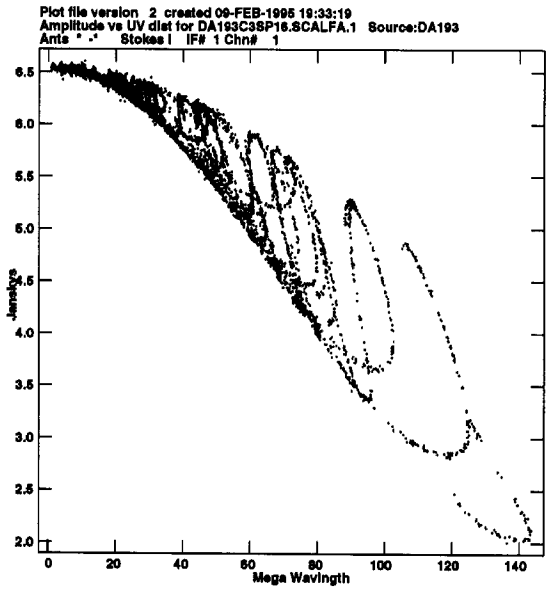

Figure 4: Final Stokes I visibility data from **8x8C3** dataset, averaged to 60s. Only every tenth point is plotted.

4.2 Comparison with CLEAN

As Briggs *et al.* demonstrated, the NNLS algorithm in general performs better than the CLEAN algorithm in deconvolving VLBA images at this high dynamic range. Rather than re-iterating this point, we want to show that CLEAN can actually do a reasonable job deconvolving the DA193 data, but that driving the algorithm in conjunction with self-calibration is very difficult. In figures 5 and 6, we show the results of CLEANing down to 100 and 10 μ Jy/beam. The higher level is about 1.4 σ. This illustrates a basic point discovered by Briggs (1995): even with a very well-calibrated dataset CLEAN can leave errors in the final image that resemble calibration errors. CLEANing deeper will remove these effects. However, when self-calibration is also being performed, correct driving of the CLEAN algorithm is difficult. By comparison, Briggs' NNLS algorithm is much easier to drive, there being essentially no controlling parameters apart from the windows. The main disadvantage of the NNLS algorithm is the considerable extra computational time required.

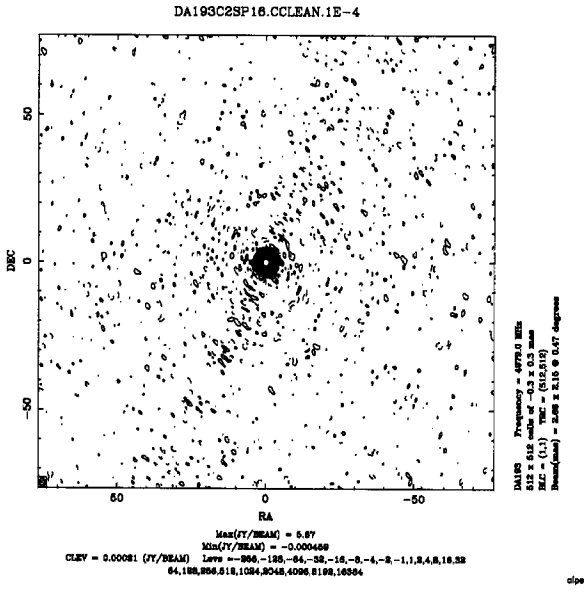

Figure 5: Final CLEAN image from **8x8C2** dataset, cleaned down to 100 μJy/beam

Apart from these preceding images, all the images shown in this memo were made using the NNLS algorithm.

The convergence of NNLS and CLEAN when used in conjunction

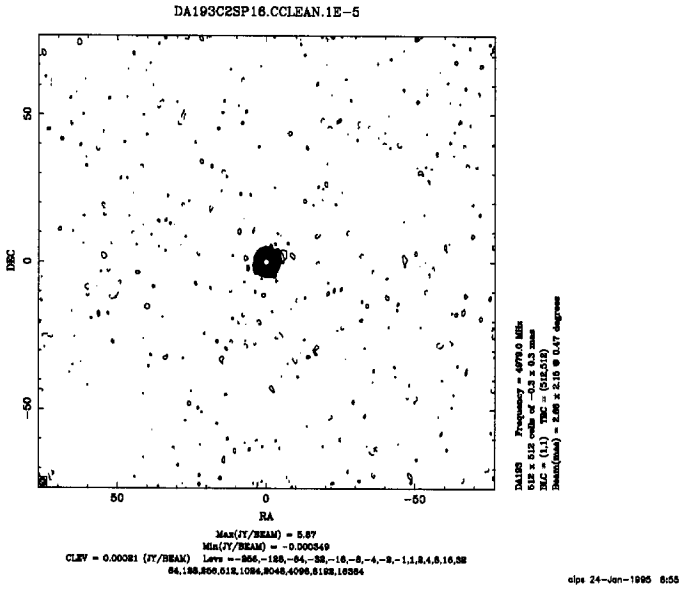

Figure 6: Final CLEAN image from **8x8C2** dataset, cleaned down to 10 μJy/beam

with self-calibration is shown in figures 7 and 8. For the first iteration, only the antenna phases were estimated. Thereafter both amplitude and phase were estimated. For both deconvolution algorithms, this rate of convergence is slower than many people seem to expect. However, poor convergence is expected for this type of algorithm: essentially steepest descent (*i.e.* the outer selfcalibration loop is steepest descent). Other algorithms converge more rapidly but at the cost of considerable extra complexity.

5 Closure errors in an FX correlator

We know of three principal sources of closure error in the VLBA correlator: time-averaging decorrelation, spectral averaging decorrelation, and segmentation decorrelation. We discuss each of these in turn and describe their effects on our data.

5.1 Time averaging decorrelation

If the baseline (residual) fringe rate $\dot{\phi}$ is non-zero then averaging over an integration time Δt will cause decorrelation. The formula for time-

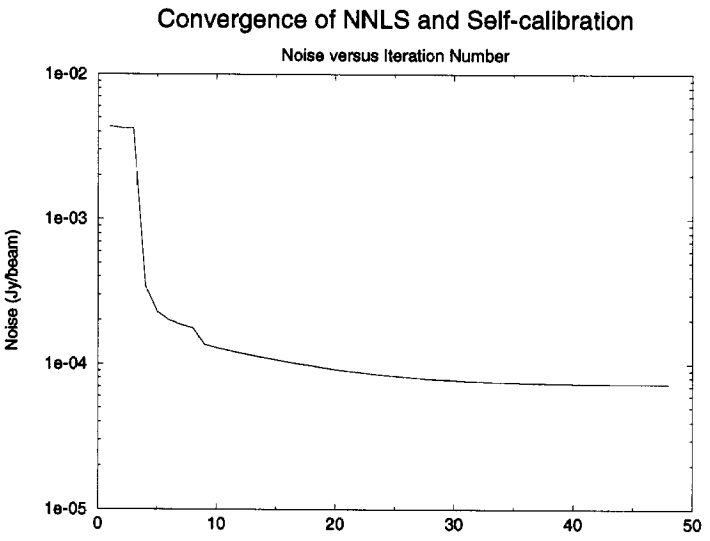

Figure 7: Convergence of NNLS and self-calibration for the **8x8C2** dataset.

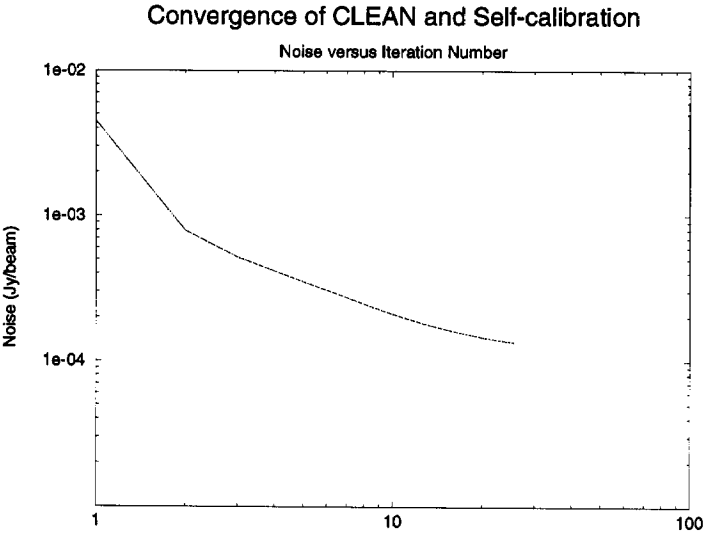

Figure 8: Convergence of CLEAN and self-calibration for the **8x8C2** dataset.

averaging decorrelation is:

$$D_{i,j} = \mathrm{sinc}\left(\frac{1}{2}\Delta t \dot{\phi}_{i,j}\right) \quad (1)$$

The correction for time-averaging decorrelation is automatically applied in $\mathcal{AIPS}$.

5.2 Spectral averaging decorrelation

The VLBA correlator optionally averages over N spectral channels before writing the correlation coefficients. This is used for continuum observations since it reduces the data rate by a large factor. The penalty is a decorrelation due to any residual delay error τ across the bandpass. The form of the spectral averaging decorrelation is:

$$D_{i,j} = \frac{\sin\left(\frac{1}{2}\Delta\nu\tau\right)}{N\sin\left(\frac{1}{2N}\Delta\nu\tau\right)} \quad (2)$$

where $\Delta\nu$ is the channel bandwidth. N is known as the *spectral averaging factor.*

For large N, this reduces to a sinc-function:

$$D_{i,j} \sim \mathrm{sinc}\left(\frac{1}{2}\Delta\nu\tau\right) \quad (3)$$

In $\mathcal{AIPS}$, this correction and the segmentation decorrelation correction discussed in the following section, are implemented via a new CQ table; see the EXPLAIN file FXVLB for details. Further information will be given in a forthcoming memo.

For **8x8C1** and **4x16C1**, the best images obtained after correction for spectral averaging decorrelation are shown in figures 9 and 10. It is important to recount some history at this point. After obtaining these images, and believing that we had corrected the only known source of delay-related closure errors: spectral averaging decorrelation, we saw no alternative but to recorrelate **8x8** with the delays set to zero to determine if another delay-related effect could be present. This recorrelation (**8x8C2**) led to the excellent images shown in figure 11. From the significant decrease in noise level, we knew that another delay-related effect must still be present and afflicting the **8x8C1** dataset. This next level effect, *segmentation decorrelation* is intimately related to the windowing of the samples input to the FFT stage. During the discussions of this effect, it was discovered that Hanning windowing was being applied by default to all observations. Another correlation **8x8C3** was therefore performed using Uniform windowing.

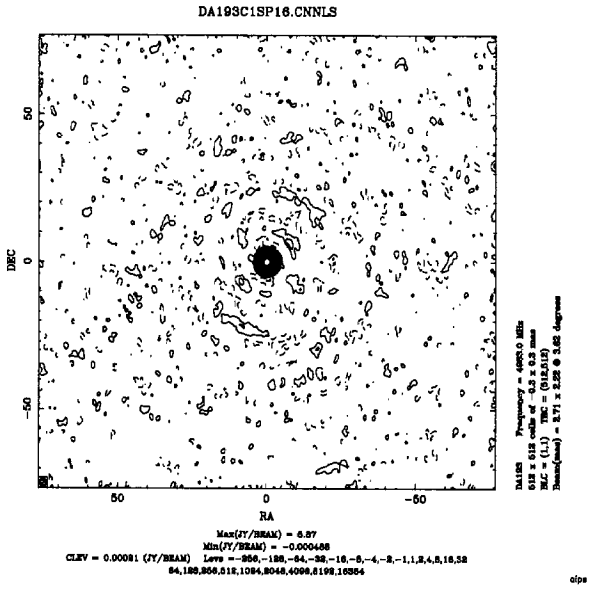

Figure 9: Final image from **8x8C1** dataset, after correction for spectral averaging decorrelation.

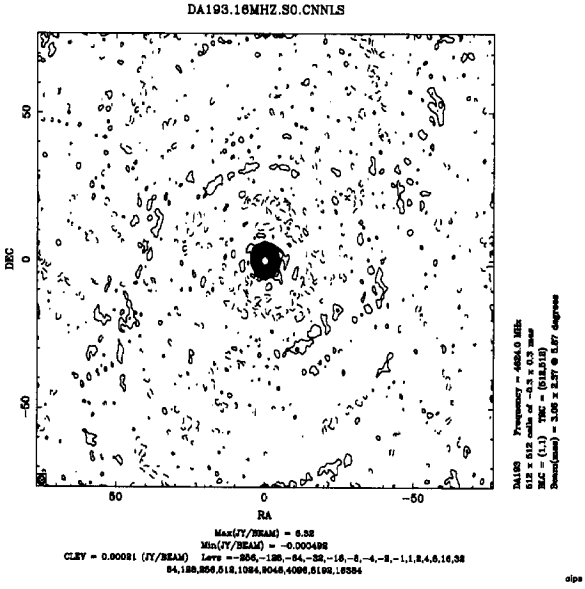

Figure 10: Final image from **4x16C1** dataset, after correction for spectral averaging decorrelation.

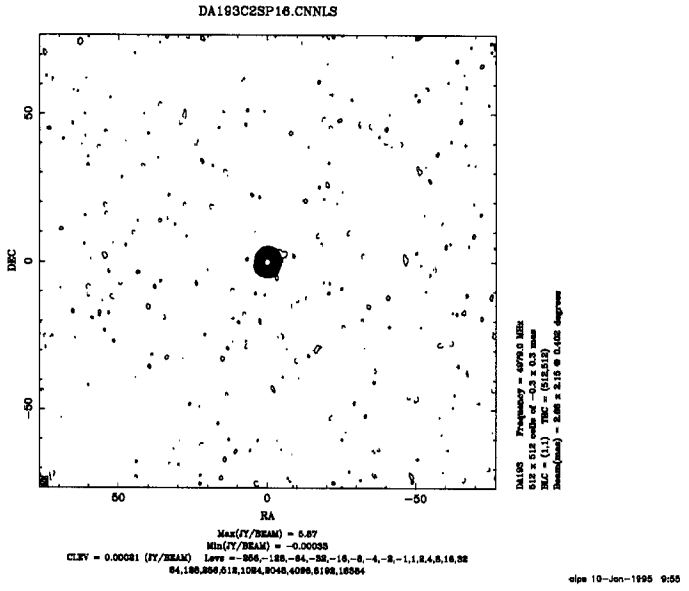

Figure 11: Final image from **8x8C2** dataset

5.3 Segmentation decorrelation

Delay errors at the various stations can lead to a misalignment of the segments fed to the FFT in an FX correlator. For the ranges of delays seen in our observations, this is typically a few samples but can range up to 10-12 samples. Hence this number of samples cannot participate in the correlation, leading to a loss of correlation. The degree of loss depends upon the window function used in the FFT. We call this effect *segmentation decorrelation.* For a continuum spectrum, the loss in correlation is given by the autocorrelation function of the windowing function.

This effect is particularly strong and easy to understand when the input samples are uniformly windowed. In this case, the autocorrelation function is a triangle function:

$$D_{i,j} = 1 - \left| \frac{\tau_i - \tau_j}{\tau_{\max}} \right| \tag{4}$$

where the maximum delay is $\tau_{\max} = N_{\text{seg}} \Delta\tau$.

The correlator can also perform an approximation to Hanning windowing. This, the windowing of the autocorrelation function and the corresponding spectral response function are portrayed in figures 12, 13 and 14. The effect of this windowing is to weight down the points near

the edge of the window with the result that segmentation decorrelation is much less important than for uniform windowing. A disadvantage is that Hanning windowing also loses signal to noise (see below).

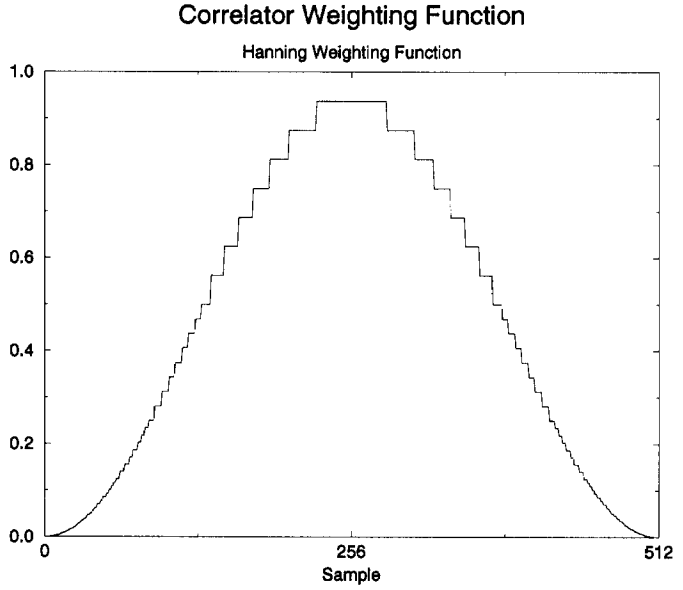

Figure 12: Discrete Hanning Windowing Function

In the case of a continuum spectrum, only the value of the auto-correlation function near zero lag is important. Thus it is useful to expand the decorrelation function $D_{i,j}$ as a power series around zero lag, the first two terms of which are most important.

$$D_{i,j} = 1 - a_1 \left| \frac{\tau_i - \tau_j}{\tau_{\max}} \right| - a_2 \left(\frac{\tau_i - \tau_j}{\tau_{\max}} \right)^2 \qquad (5)$$

For uniform windowing, $a_1 = 1$ and $a_2 = 0$. Expanding the ACF for Hanning windowing in a 10-th order Taylor series around zero lag, and ignoring odd-powered terms, we find that second order coefficient a_2 is about 6.4. This value of a_2 results in an image that is insignificantly improved. We can also determine the coefficient a_2 empirically by a variant of self-calibration where a_2 is a free variable (see Appendix B). We did this for **8x8C1** dataset using the best image from **8x8C2** as an initial model in the self-calibration. This was performed entirely in SDE using a program `decal` that applies the same number for all IFS. This is in error by typically 20-30 ns, which is probably negligible. The optimal value of a_2 is 29, about 5 times larger than the value for

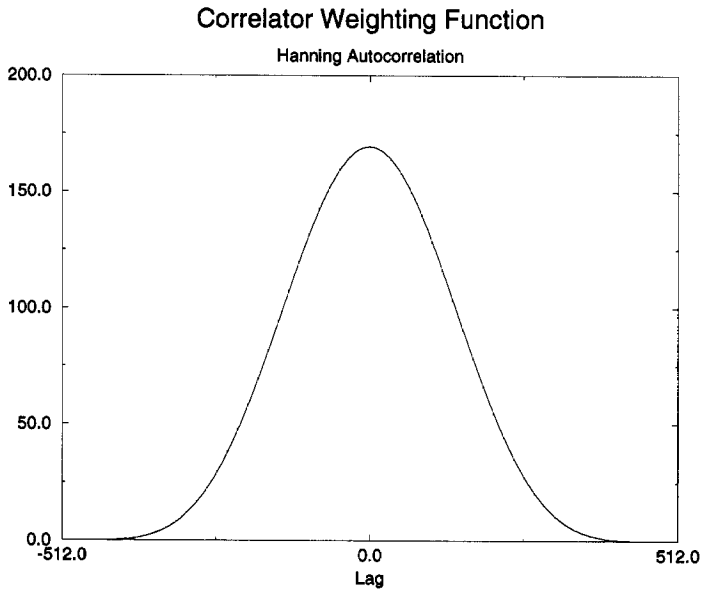

Figure 13: Autocorrelation of discrete Hanning Windowing Function

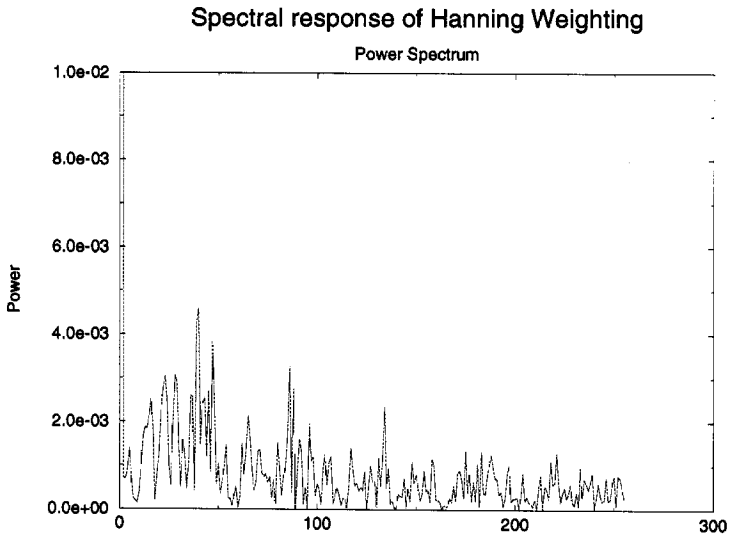

Figure 14: Spectral (power) response of discrete Hanning Windowing Function

Hanning windowing segmentation decorrelation. The corresponding image, shown in figure 15, is rather good. Thus we conclude that

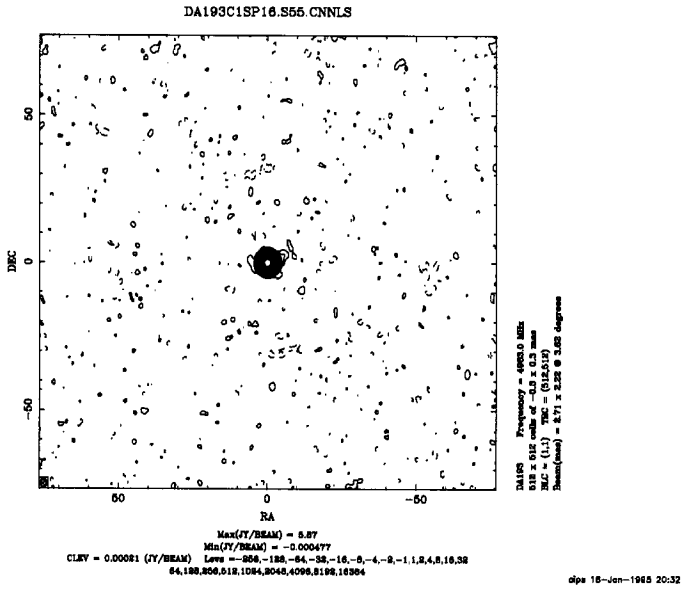

Figure 15: Final image from **8x8C1** dataset using the optimal value of $a_2 = 29$

another source of decorrelation must be present in the **8x8C1** data.

The corresponding estimation for **4x16C1** yields $a_2 = 7$ and the image shows improvement (figure 16) over the image with no correction (figure 10). Thus for **4x16C1** data, Hanning windowing segmentation decorrelation does explain the total decorrelation seen.

6 Observations of CTD93

DA193 is only mildly resolved and so one might worry that on a resolved source, the dynamic range acheived might be lower for any of a number of reasons. To check this, we observed the compact double CTD93 with the same observational setup as used in the DA193 observations.

NL was lost during much of the observation due to a problem with the VME computer. The weather was good with the exception of some clouds at KP.

The playback was poor for BR (first tape) and LA (second tape).

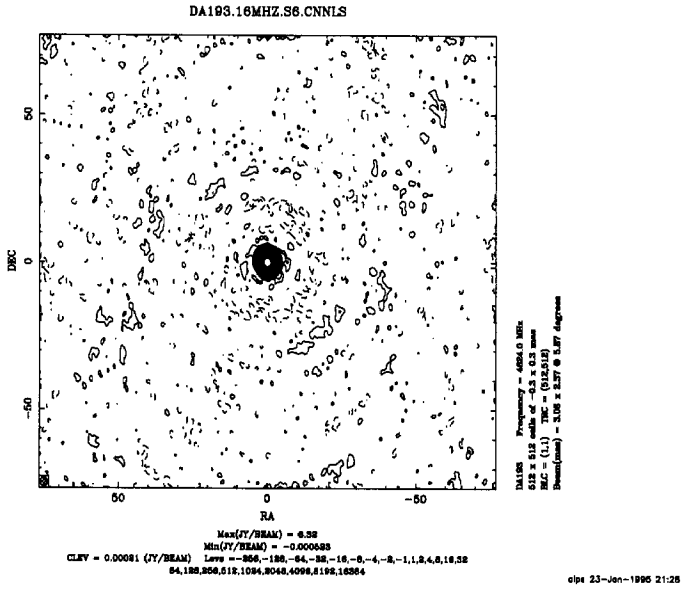

Figure 16: Final image from **4x16C1** dataset using the optimal value of $a_2 = 6$

The date were correlated once on Feb 12 1995 (**8x8CTD93C1**) with Hanning windowing and GPS clocks.

The integration time τ was 3.9s and 16 spectral channels were output for each IF channel.

Calibration and editing was as for DA193. No bandpass correction was applied.

The data were fringe-fitted using FRING and then imaged via **visamat** thus effectively using a point source model for the first phase self-calibration. After inspection of the initial dirty image, windows were set up and iteration proceeded. After three cycles of phase self-calibration, amplitude self-calibration was switched on. As the iteration proceeded, the windows were expanded slightly to include flux that was not initially apparent. About 20 self-calibration / imaging cycles were required to reach the thermal noise limit ($\sim 54\ \mu$Jy/beam). The resulting Stokes I and V images are shown in figure 17 and 18.

7 The use of Hanning windowing

In an XF correlator, Hanning smoothing is used to ameliorate the effects of the sharp truncation of the measured correlation function at

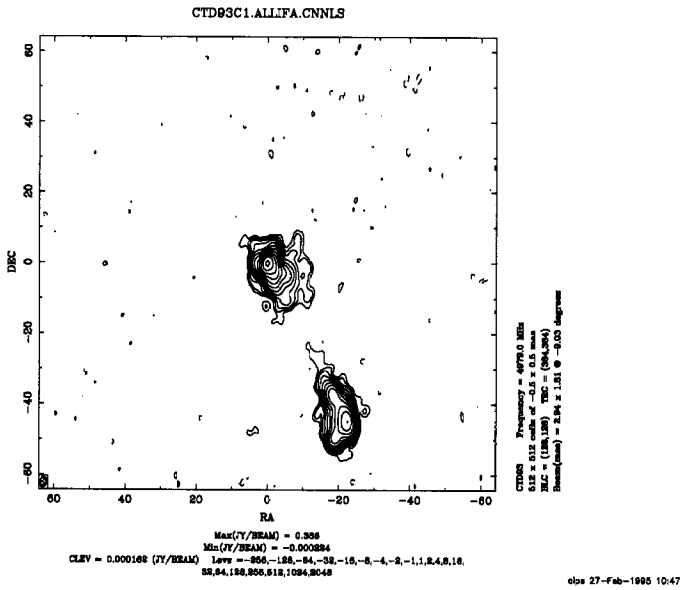

Figure 17: Final Stokes I image from **8x8CTD93C1** dataset. Lowest contour level is 162 μJy/beam ($3\sigma_V$)

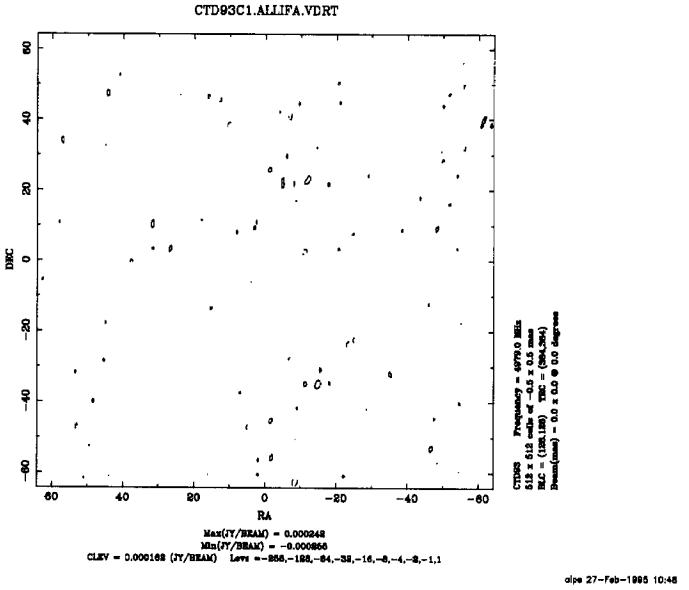

Figure 18: Final Stokes V image from **8x8CTD93C1** dataset. Lowest contour level is 162 μJy/beam ($3\sigma_V$)

the maximum lag $\tau_{\max}$. In the lag domain, the filter is $\frac{1}{2}(1 + cos\frac{\tau}{\tau_{\max}})$. In the spectral domain, where it is usually performed, it corresponds to a convolution with a 1/4, 1/2, 1/4 kernel. In the VLBA correlator, a limited precision approximation is used in windowing the *time domain* samples. Ignoring for the moment the effects of the approximation, in the spectral domain this is equivalent to convolution with a kernel proportional to 1/16, 1/4, 1/16. The approximation scatters power from sharp spectral features. For 512 sample points, the maximum sidelobe is about 1/2 %.

The loss of SNR for a continuum spectrum goes as the square root of the ratio of the peak of the autocorrelation functions of the two windowing functions (Hanning to uniform). This is about $\sqrt{\frac{180}{512}} = 0.59$ for a 512 point FFT. One can think of 180 as the effective number of samples used in the FFT. If overlapping over segments is performed (as is always the case when possible) then the SNR can be partially recovered. A fuller analysis of this has been given in a recent VLBA scientific memo (number 10) by Leonia Kogan.

Overall, the use of Hanning windowing in the correlator seems to have little to recommend it. If narrow spectral lines are of interest then the correct and fully accurate version of Hanning windowing can be applied to the spectra in subsequent processing via AIPS. If only the continuum is of interest, then Hanning serves only to destroy SNR. For these reasons, the VLBA has switched to a default of Uniform windowing.

Acknowledgements

We had very useful discussions with many people at the AOC but in particular Craig Walker, Jon Romney, and Vivek Dhawan made important points in discussion.

Appendix A: Use of `visamat`

The use of SDE is described in the SDE User's Guide available via the WWW from the SDE Home Page *http://www.aoc.nrao.edu/sde/sde.html*. At the AOC, it is usually sufficient to do `. /zia/u/sde/sdeini` or `source /zia/u/sde/sdeini.csh` to initialize the environment.

Typical inputs to **`visamat`** are as follows:

```
Imsize = 512, 512, 1
Cellsize = 0.3000000142E-03, 0.3000000142E-03, 1.000000000
Shift = 0.0000000000E+00, 0.0000000000E+00
```

```
FOV = 0.0000000000E+00
Stokes = IV
Uvlimits = 6000000.000, 0.1000000000E+11
Muvlimits = 6000000.000, 0.1000000000E+11
Filter = 0.0000000000E+00, 0.0000000000E+00, 0.0000000000E+00
Timerange = 0, 0, 0, 0, 0, 0, 0, 0
Vis = DA193C2SP16NB
NewVis = DA193C2SP16NB.SCAL
Model = DA193C2.ALLIFB.8MHZ.BIG.LOOSE.UV.NNLS
FluxWindow = bigsource
DataWindow = bigconstraint
AMatrix =
DoImage = T
Algorithm = NNLS
MaxIter = 0
Flux = 0.0000000000E+00
Niter = 1
Image = DA193C2SP16NB.NNLS
CImage = DA193C2SP16NB.CNNLS
Residual = DA193C2SP16NB.RESID
Beam = 0.0000000000E+00, 0.0000000000E+00, 0.0000000000E+00,
0.0000000000E+00
Threshold = 7.000000000
Tedit = 300.0000000
Tamp = 60.00000000
Tphase = 3.000000000
Mode = AMPNORMPHI
DisplayCommand = da193.plot # &
DisplayFile = DA193C2SP16NB.CNNLS
```

Some notes about the input parameters:

DoImage `visamat` can work with constraints as either dirty image pixels or as visibility samples. Since the former are usually fewer in number, it is probably best to work in the image plane (DoImage = T).

Algorithm `visamat` can perform deconvolution via either the CLEAN algorithm or via the NNLS algorithm. The latter is the default and is recommended.

MaxIter determines the number of iterations that the algorithm is to be run. This is not the same as the number of self-calibration and imaging cycles.

Flux Stopping flux for CLEAN.

Niter The number of self-calibration and imaging cycles.

Image Name of output image containing components.

CImage Name of output restored image.

Residual Name of output residual image.

Tamp and Tphase `visamat` allows separate time scales for amplitude and phase selfcalibration.

Mode The mode of selfcalibration can be:

' ' Phase only

AMPPHI Amplitude and phase

AMPNORMPHI Amplitude and phase, but with amplitude change renormalized back to unity.

The windows for the dirty image (DataWindow) and the output image (FluxWindow) can be set using saoimage as described in the User's guide. It is best to make these quite tight. The CLEAN algorithm requires that the A matrix be square so DataWindow and FluxWindow must be the same.

DisplayCommand This shell command is invoked by `go display` after substituting the hash mark with the name specified in `DisplayFile`. This gives a lot of flexibility in displaying images. One possibility is just to start `saoimage`, but an alternative is to use a script. For example, the script `da193.plot` makes a plot of the current image and prints to the default postscript printer.

```
imgplot<<EOF
 Image = $1
 BLC = 1, 1, 0
 TRC = 512, 512, 1
 Levs = -256, -128, -64, -32, -16, -8, -4, -2, -1, 1,2, 4, 8, 16, 32, 64, 128,
 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 0
 Plev = 0.
 Clev = 2.1e-4
 Pixrange = 0., 0., 0.
 SAOColor = bm.col
 Scale = LIN
 SPar = 0.
 Invert = T
 Plot = CO
 Title = \$Image
 XLabel = *
 YLabel = *
 Left = T
 Annotate =
 Linwidth = 1
```

```
 LabelLev = 2
 CLabel =
 ID = T
 CHeight = 0.600000
 PSFEllipse = T
 Beam = 0., 0., 0., 0.
 PSFSlice = F
 %Label = F
 VLBI = T
 NBotLab = 5
 Device = "\$im.ps"/ps
EOF
psprint \$1.ps
```

Appendix B: Estimation of a_2

We could choose that value of a_2 which minimizes the discrepancy between observed data and corrupted model data:

$$\mathcal{S} = \Sigma_{i,j} \left| V_{i,j} - D_{i,j} \widehat{V_{i,j}} \right|^2 w_{i,j} \tag{6}$$

However, since at this point the data have been self-calibrated, some part of the decorrelation D have been absorbed in the antenna gains. Hence, while in the original data, the effect of the decorrelation is that all baselines are diminished in amplitude, the effect in the self-calibrated data is that some baselines are decreased in amplitude and some are increased. Thus the characteristic signature of the decorrelation is lost. For a quadratic model, the appropriate modified form of the decorrelation function is therefore:

$$D_{i,j} = 1 + 2a_2 \frac{\delta\tau_i \delta\tau_j}{\tau^2_{\max}} \tag{7}$$

where $\delta\tau$ has been adjusted to be zero mean.

Since the modified D is linear in a_2, $\mathcal{S}$ can be minimized by a simple least squares fit.