EVLA Memo #198
MSUVBIN: A Way to Combine, Average, Flag Visibility Data

Kumar Golap and Emmanuel Momjian
NRAO

July 6, 2016

Abstract

This memo presents the details of a technique to average large projects onto some fixed size visibility data set. The size of the output is determined by the largest field of view and finest (spatial and spectral) resolutions needed in imaging. We discuss the advantage of binning onto a fixed grid and demonstrate its usefulness for some observations.

1 Introduction

Radio interferometric observations under some conditions require the need to average the visibility data first before imaging. The need for such a functionality is demonstrated by the usage of baseline based averaging as implemented in the *AIPS* task *UBAVG[1]* for instance. The usage is typically for data that will not need self-calibration (i.e., there are no bright sources in the field), spans multiple observing epochs, and may or may not be in the same array configuration.

2 Description

For multi-year observations of a given target an astronomer may wish to edit and calibrate each data set as best as possible, and average and accumulate each onto a single output visibility data set and then image this averaged output. As new data get acquired a new piece can then be added to the existing averaged dataset and re-imaged to produce a new combined-data image without the need to retouch the previous epochs' original data sets.

2.1 Averaging in Image Domain

For signal-to-noise reasons it is always preferable to average prior to deconvolution. A simple way of averaging data over days of observations of a given target source is to make a dirty image from each epoch of observation and average the images. To optimize this, a weighted averaging scheme has to be devised, otherwise images from days of low sensitivity will degrade the contribution of the images from days with high sensitivity observations. Therefore a weight image, which one can think of as the point spread function (PSF) image + sensitivity information for that observation, along with the image itself from each session has to be stored. Also, at the stage of making individual images per session a weighting scheme (natural, uniform or in-between) has to be chosen and schemes that require weight density in the neighborhood of a uv point (e.g., Brigg's weighting scheme [2]) cannot be controlled properly because one session's dirty image is completely independent of the weight density of another day's data set.

Assume that the highest resolution image ever needed for a given data set consists of $n_x \times n_y$ pixels along the RA, DEC axes and n_{pol} and n_{chan} pixels along the polarization and spectral axes, respectively. This requires storing two images of float numbers at the highest resolution for future usage or further averaging, with these two images being the dirty image and the PSF image. The total volume of these two images is $(2 \times n_x \times n_y \times n_{pol} \times n_{chan} \times sizeOfFloat)$ bytes along with some meta-data information about the images which are very small compared to the stored images themselves. Once the images of each data set are made, an average of the dirty images and an average of the PSF images can be made. Then the average PSF image is used to deconvolve the average dirty image.

Issues to note on averaging in the image domain are:

1. Errors made in previous imaging are frozen-in unless the original visibilities are re-used. Such errors may be due to having a bad antenna in some epochs, or Radio Frequency Interference (RFI) that has not been properly flagged.

2. Image weighting, sensitivity and final resolution control has to be decided at the very beginning of the process and is difficult to control and revisit while averaging.

2.2 Averaging in the visibility domain on a uniform grid

A solution to the above noted issues with averaging in the image domain is to average in the uv domain as done in the task *UBAVG* (and the procedure *stuffr*). However, instead of keeping track of the baseline length and the time allowed for averaging in order to avoid time smearing or, if averaging over frequency, to avoid bandwidth smearing, one can average (or bin) on a uv-grid that corresponds to the largest field of view at the highest resolution to be ever imaged from the data of interest. If the cell size, along the spatial axes, in the image plane is chosen so that it is smaller than $\lambda_{min}/2B_{max}$ (where B_{max} is the maximum baseline and λ_{min} is the smallest wavelength ever going to be binned on that uv-grid), then time smearing is prevented. This is effectively performing the gridding step before fast-Fourier transforming (FFT) to make an image, and equivalently creating a data set with uniform steps in the uv domain along with storing the weights in each uv cell or uv point. One then can take care of bandwidth decorrelation for example by accounting for the frequency effect in the uv distance properly (one term multi frequency synthesis). The compression achieved can be very high; highest being the case where the desired image products are going to be for continuum or involve several spectral channels averaged together.

In this approach an output visibility data set is made with $n_x \times n_y$ visibilities or (u, v) points. Each visibility point at position (u, v) has n_{pol} and n_{chan} data points. The volume of the output data set is $(n_x \times n_y \times n_{pol} \times n_{chan} \times sizeOfComplex)^1$ for the visibilities and $(n_x \times n_y \times n_{pol} \times n_{chan} \times sizeOfFloat)$ for the weights. Thus the complete gridded visibility data set is $(3 \times n_x \times n_y \times n_{pol} \times n_{chan} \times sizeOfFloat^2)$ plus some extra meta data info. We will refer to this approach of binning the uv data henceforth as *msuvbin*. If individual parallel hand polarization products (e.g., RR and LL) are not needed then we can even reduce the volume of the output data of *msuvbin* by combining these two products into one (i.e., Stokes I).

In a simple implementation of *msuvbin* the input visibility data points are summed (weighted averaging to keep track of sensitivity) to the nearest grid point of its u, v value. The visibility of a cell, in the output grid, is the weighted sum of all the visibility data that came within $(\pm\triangle\frac{u}{2}, \pm\triangle\frac{v}{2})$ where $(\triangle u, \triangle v)$ is the pixel size in the visibility dimension. This loss in accuracy in the visibility position results in the fact that no sources outside the field of view of $(\frac{1}{\triangle u}, \frac{1}{\triangle v})$ from what is

[1] SizeOfComplex is twice the SizeOfFloat

[2] SizeOfFloat is usually 4 bytes in single precision

chosen as phase center can ever be imaged. In fact if such sources exist they will cause aliasing issues in the resulting image. Therefore the field of view which is related to the grid cell size in u, v should be decided with care to encompass all bright sources known. Otherwise they have to be subtracted from the visibilities of each session's data (e.g by *uvsub* [3]) prior to uv-gridding.

Worth noting that combining uv-data from different resolution interferometers onto the same uv-grid is a none issue through *msuvbin.* In fact this might be a good way to combine Single Dish data along with interferometer data for joint deconvolution. Care has to be taken though to ensure that the weight contributed to a given uv cell by the single dish is representing the sensitivity of the single-dish observation according to the radiometer equation.

2.3 Wprojection

If the best field of view ever to be imaged is large enough, then in some cases correction for the w-term problem need to be addressed [4]. This can be corrected in the gridding process by using the *wprojection* algorithm [4].

To correct for the effect of the w-term the baseline visibility ($V(u, v, w_i)$) is convolved in (u, v) with the function $\tilde{G}(u, v, w_i)$, where:

$$\tilde{G}(u, v, w_i) = \int e^{-2\pi i[w_i(\sqrt{1-\ell^2-m^2}-1)]}\ e^{-2\pi i[u\ell+vm]}d\ell dm \tag{1}$$

The process of correcting for the w-term therefore integrates nicely in the process of *msuvbin* as it involves gridding in (u, v) with an extra convolution. However there are some caveats which we discuss below.

2.3.1 Issue with Cotton-Schwab style major cycle

In the Cotton-Schwab [5] major cycle the clean model obtained in the deconvolution stage is Fourier transformed and the visibility values at the observed (u,v) location is predicted and subtracted from the observed visibility to produce residual visibilities. If we do not use *wprojection* to grid the data then it is straight forward to compare the predicted visibilities directly with the *msuvbin* visibilities at the location (u, v). This is not true now for data that have been gridded with *wprojection.* Most of the points spread on the grid by the function $\tilde{G}(u, v, w_i)$ around location (u_o, v_o) are corrections for the visibility $V(u_o, v_o, w_i)$, therefore they cannot be compared directly to the predicted visibilities in those cells. Hence, for now imaging should only involve a deconvolution rather Cotton-Schwab major cycles when using a *msuvbin* gridded data made with *wprojection.* We are actively investigating some ideas in order to achieve a *msuvbin* data set produced with *wprojection* that can be used with Cotton-Schwab major cycles but the volume of the output gridded data set will increased by a factor of two or more.

2.4 Self calibration possibility

The self-calibration capability is lost using the gridded data as many antenna pairs, at quite different times, may contribute to a given uv point in the *msuvbin* data set. However, we may go past the impossibility of self calibration if the data distribution is sparse and not many different antenna pairs visit the same uv-cell. We could relabel such cells that have baselines consisting of different antenna pairs to be as if from a new, single, baseline and they effectively do not get added into the self-calibration solution. Note that this is not a generic solution.

2.5 Some results of *msuvbin*

Figure 1 shows the HI 21 cm line profile of a galaxy from two different image cubes. One image cube was made by directly combining six data sets in the CASA task *clean* using *wprojection*. The second cube was made after binning the six data sets into a single *msuvbin* data set with *wprojection* then imaging with *clean*. The six data sets are of VLA B-configuration observations at L-band from the COSMOS HI Large Extragalactic Survey (CHILES; PI J. van Gorkom). The grid size is equivalent to a field of view of 2^o with a pixel size of $2''$. The noise and spectral profiles are quasi identical.

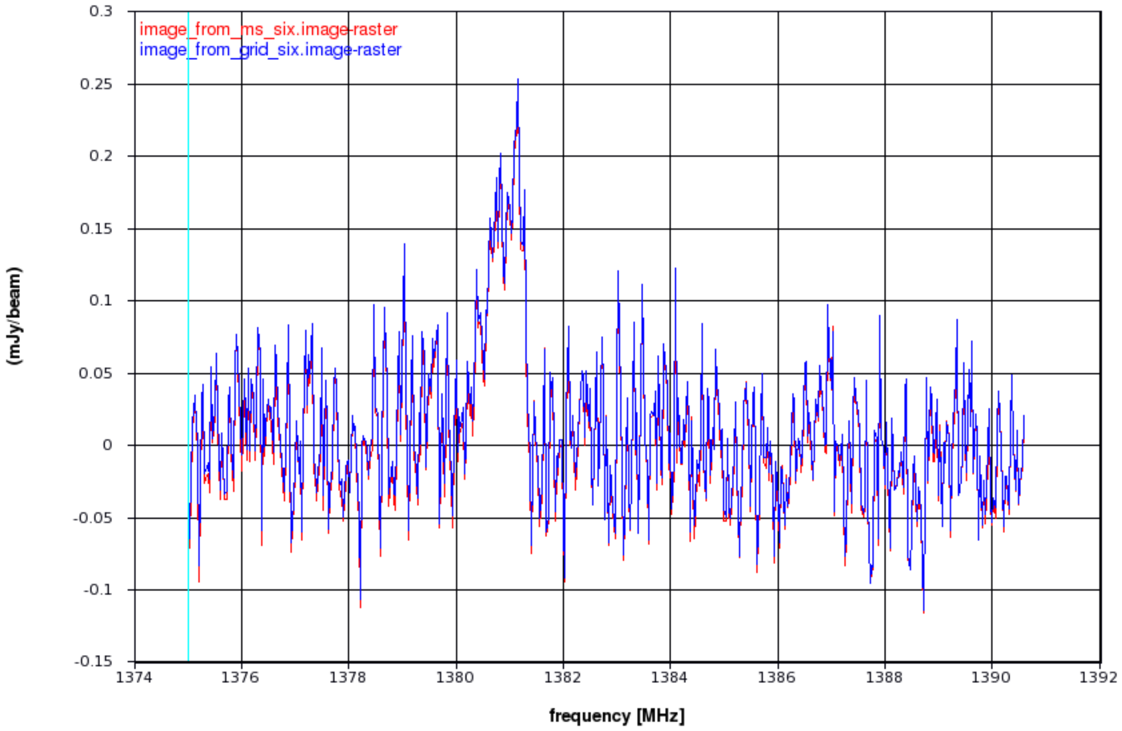

Figure 1: Spectral profiles showing the HI 21 cm emission line from a galaxy within the CHILES field. The spectra are extracted from image cubes made directly from six data sets (*red*) and from the *msuvbin* output of the same six data sets (*blue*).

Figure 2 shows the variation of the rms noise with frequency obtained from two small spatial regions, one near the center (*left*) and another near the edge (*right*), of three image cubes. The three curves in each plot are for (i) the rms noise estimations for the image cube made directly from six data sets, (ii) an image cube made from a *msuvbin* data set generated from the six data sets, and (iii) the same image cube as that made with the *msuvbin* output except that it has been corrected for a *sinc* function effect due to convolution function sampling (see appendix A for an explanation). This correction is extremely negligible near the center of the image but noticeable near the edge.

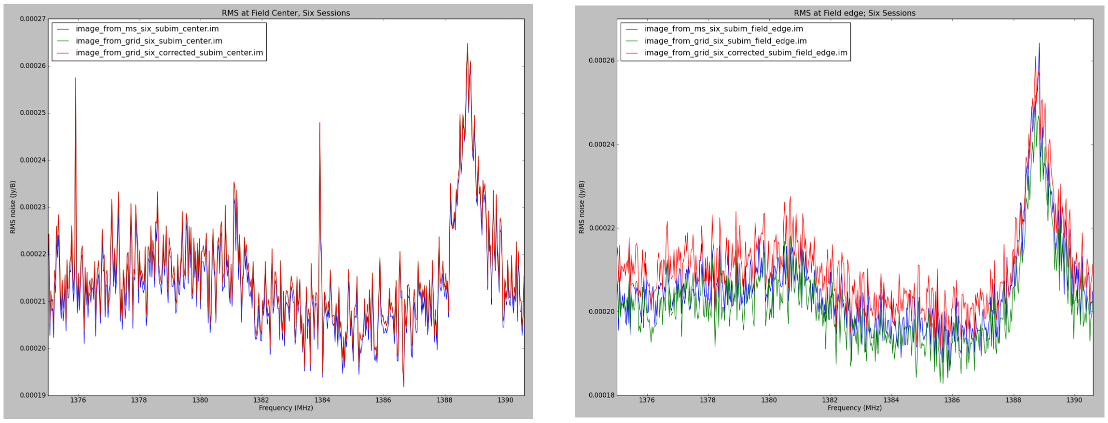

Figure 2: The rms noise variation in frequency obtained from a region near the center (*left*) and near the edge (*right*) of three image cubes. These cubes are: (i) made directly from six data sets (*blue*), (ii) made from a *msuvbin* data set generated from the six data sets (*green*), and (iii) the same image cube as that made with the *msuvbin* output but corrected for a *sinc* function effect due to convolution function sampling (*red*).

3 ***msuvbin*** as a flagging agent

It is a well known fact that any signal that appears widespread in a given domain is going to be compact in the Fourier domain. There is a large population of RFI sources that can appear as large scale structures (e.g., stripes) in images and therefore can be located and flagged out using the uv-grid data set made by *msuvbin*; even if binning and averaging is not needed.

To flag RFI, the algorithm locates compact high visibilities that deviate from a smooth variation on the msuvbin grid and flags all the baselines that contributed to that particular uv-cell in the original data set. The following steps demonstrate an example on how to use *msuvbin* as a flagging agent:

1. Make a continuum msuvbin grid across the full frequency span of the data set.
2. Find a radial average best fit function from the uv-grid made in step (1).
3. Make a spectral cube msuvbin grid with spectral channel bins corresponding to the desired spectral resolution to flag RFI.
4. For each channel in the msuvbin output of step (3) compare the grid with the radial average function from step (2) and locate sharp deviations.
5. Mark the uv-cells and the channels of deviant points from the comparison in step (4).
6. Using a reverse *msuvbin* process flag all the data points in the original data set that contribute to the marked uv-cells in step (5).

Using a simplified version of the above algorithm Figure 3 shows dirty images of the before (*left*) and after (*right*) *msuvbin* based RFI flagging using a VLA data set near 990 MHz. This example highlights the effectiveness this approach may have in flagging RFI.

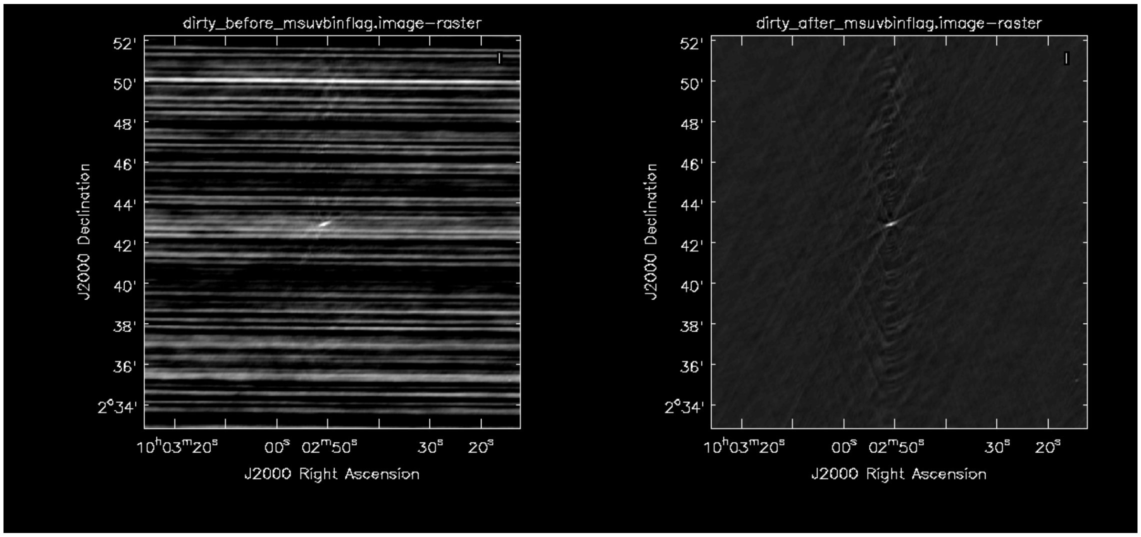

Figure 3: Dirty images made using data before (*left*) and after (*right*) *msuvbin* based flagging at around 990 MHz

4 Future improvements

In the following we list the items that need to be addressed and/or incorporated as enhancements into the current implementation of *msuvbin*.

- Achieve a *msuvbin* data set produced with wprojection that can be used with Cotton-Schwab major cycles.
- Add gridding and combining single dish image onto the uv-grid data of an interferometer.
- Add automatic poststamp imaging of well known bright continuum sources and *uvsub* them before running *msuvbin*. This is particularly important to minimize the side lobes of continuum sources that are outside the gridded field.
- Provide an easy way to flag visibilities without much user input using *msuvbin*.

5 Acknowledgments

We thank the CHILES team for giving us access to their edited and calibrated data to test and commission *msuvbin*.

References

[1] `http://www.aips.nrao.edu/cgi-bin/ZXHLP2.PL?UBAVG`

[2] Dan Brigg's thesis `http://www.aoc.nrao.edu/dissertations/dbriggs/`

[3] casa uvsub task `http://casa.nrao.edu/docs/TaskRef/uvsub-task.html`

[4] Cornwell, T.J., Golap, K., Bhatnagar, S.; The Noncoplanar Baselines Effect in Radio Interferometry: The W-Projection Algorithm; IEEE Journal Of Selected Topics In Signal Processing, vol. 2, No. 5, October 2008.

[5] Schwab, F. R; Relaxing the isoplanatism assumption in self-calibration; applications to low-frequency radio interferometry, AJ, vol. 89, 1984. 1076-1081 (The reference to Cotton-Schwab algorithm is to a paper "under preparation" on page 1078)

A Effect of sampling size of convolution function on the final image

When using a convolution function while gridding the visibilities, the resultant image afterwards may need to be corrected by a *sinc* function. This is the result of the finiteness of the sampling of the convolution function.

Let us consider a 1-Dimensional (1-D) convolution for simplicity. The convolution function, $C(u)$, used is sampled at $\triangle u_c$. Note that $\triangle u_c$ is smaller or equal to the $\triangle u$ cell size of the *msuvbin* grid cell size. Therefore the effective convolution function is:

$$C_{eff}(u) = \Pi(u) * (\sum_{n=-\infty}^{\infty} C(u)\delta(u - n\triangle u_c)) \tag{2}$$

Where $\Pi(u)$ is a rectangular function from $-\triangle u_c/2$ to $\triangle u_c/2$ and $C(u)$ is the true convolution function and δ is the impulse function.

The output image is the Fourier transform (FT)

$$I_o(l) = FT(C_{eff}(u) * V(u)) \tag{3}$$

Where $V(u)$ are the observed visibilities.

Therefore from equation 2 we have the observed image as

$$I_o(l) = FT(\Pi(u))FT((\sum_{n=-\infty}^{\infty} C(u)\delta(u - n\triangle u_c)) * V(u)) \tag{4}$$

$$= sinc(\triangle u_c l)FT(C(u))I_d(l) \tag{5}$$

where $FT(C(u))$ is the Fourier transform of the convolution function and $I_d(l)$ is the true dirty image. Therefore the more oversampled is the convolution function the smaller is the correction due to the $sinc(\triangle u_c l)$.