Note on Tuning the Middle autoWindow Loop: Adding Deferents to Epicycles

W. D. Cotton, December 12, 2017

***Abstract*—The use of the auto windowing technique in Obit allows CLEANing tasks to determine the CLEAN window as the CLEANing proceeds. The technique is very conservative in that it will add only one component to the window per cycle. This was initially per major cycle which led to poor performance for complex and/or crowded fields. A "middle" loop with an image based Clarke–like proper subtraction of the accumulated components was added allowing a new window component per facet per middle loop. This was optimized for problems for which the data volume greatly exceeded that of the derived images. This is not always the case and the middle loop may be slower than the major loop. This memo describes further tuning of this algorithm. In a test case using 16 antenna MeerKAT data, the improved version of the algorithm ran in less than half the time of the older version.**

***Index Terms*—imaging, interferometry**

I. Introduction

USE of a CLEAN "Window" (AKA mask) in CLEAN can improve its performance by limiting the support to regions of plausible emission. Traditionally this was specified by the user manually as the CLEAN progressed but in many cases human specification of the window becomes impractical and automated techniques are needed. The initial implementation of "auto windowing" in Obit is described in [1] in which a single round component could be added to the CLEAN window in each facet once per major cycle. This was done if the largest residual in the CLEANable region of the facet were 1) outside the current window and 2) larger than five times the facet RMS. If a new component was added, it was centered on the peak and had a radius determined from the structure function of the residual about the peak. The limitation of one component was successful in limiting the number of side-lobes being added to the window but led to many major cycles for complex and/or crowded fields.

To improve the performance, a "middle" loop was added [2] to allow adding multiple window components per major cycle. This middle loop was basically a Clark–like [3] proper image based subtraction of the accumulated CLEAN components to derive an improved residual free of the limitations of the inner CLEAN (also see [3]). In cases where the data volume greatly exceeded the volume of derived images, this provided a substantial performance enhancement.

Circumstances may be different in the current/next generation of new instruments such as MeerKAT and ngVLA. With smaller dishes, the field of view is much larger and the improved sensitivity means sources out into the primary antenna side-lobes will need to be deconvolved. The improved sensitivity means snapshot imaging or large areas becomes practical (e.g. VLASS survey with the EVLA). The assumption of the data volume vastly exceeding the image volume is not always valid. The outer, "major" loop is highly parallelized and for small data sets producing large images may run faster than image–based subtraction which has less efficient parallelization. Further tuning of the algorithm based on the knowledge of the sizes of the visibility data and derived images is discussed in the following. This memo evaluates this technique using the Obit package [4] [1].

National Radio Astronomy Observatory, 520 Edgemont Rd., Charlottesville, VA, 22903 USA email: bcotton@nrao.edu

TABLE I
MAXIMUM MIDDLE LOOP COUNT

Ratio uv/im	max. loop count
< 1.0	1
1.0 – 1.999	2
2.0 – 2.999	3
3.0 – 3.999	4
> 4.0	5

II. Tuning the "Middle" loop

The "middle" image–based component subtraction loop continues until one of the following is encountered:

- The inner loop terminates for a reason other than it has reached a flux density at which there are residuals in the CLEANable regions of the facets being processed but outside of the current CLEAN window and which are brighter than any inside the CLEAN window.
- No new components to the window were added
- A specified maximum loop count is reached, this was initially 5.

The "tunable" parameter in this procedure is the maximum loop count. The maximum loop count is now set according to the radio of the volume of the input UV data to the sum of the sizes of the images in the mosaic of facets. This is implemented in the ObitDConClean class function ObitDConCleanVisDeconvolve using the new routine autoWinLoopCount and the values given in Table I.

III. Testing

Testing was done on a workstation with 16×3.1 GHz cores with AVX and 256 GBytes of RAM of which 100 was in a RAM disk which was used for scratch files. Other data files are on a RAID-5 disk system. The workstation was otherwise unused. Testing of the the modification of the middle loop was

[1] http://www.cv.nrao.edu/∼bcotton/Obit.html

TABLE II
CLEANING COMPARISON

	5 loop max.	1 loop max.
Real time[1]	2.5 hr	1.1 hr
CPU time[2]	5.0 hr	5.8 hr
Total flux density[3]	0.679 Jy	0.679 Jy
Min. CLEAN[4]	0.367 mJy	0.350 mJy
Peak flux density[5]	0.0412 Jy	0.0411 Jy
RMS Residual[6]	62.3 μJy	62.4 μJy

Notes:
[1] Total wall clock time of run.
[2] Total CPU time.
[3] Sum of combined CLEAN flux densities.
[4] Minimum combined flux density of CLEAN.
[5] Peak flux density at the reference frequency in the derived image.
[6] RMS in flux density at the reference frequency.

done using a 16 antenna MeerKAT data set which included the 8 km baselines. The observations spanned 4 3/4 hours (including calibrators), was subjected to baseline dependent time averaging [5] and averaged to 256 $\times$ 418 kHz channels in each of 8 spectral windows. Imaging used a radius of 1.2° needing 85 facets and 15 frequency bins in ObitTask MFImage. Cleaning used 5000 CLEAN components and the UV/Image size ratio was 0.39; this results in a single pass through the middle loop per major cycle. Two identical runs were made, one using the a maximum middle loop count of 5 and the second using the new algorithm giving a single middle loop. Table II compares various results of the two runs.

CLEAN is a very nonlinear process and the two runs followed different trajectories through solution space. The various result values in Table II are comparable if not identical. A blinking of the two images showed only minor differences in the noise.

The older, maximum 5 middle loop, run took 2.3 times the real time but only 0.87 times the CPU time. Examination of the logs indicated that typically a middle loop doing the image–based component subtraction took about 5 times longer than the comparable outer visibility–based subtraction/reimage (which is then done anyway). In this case the new algorithm is a clear win.

ACKNOWLEDGMENT

I would like to thank the MeerKAT staff especially Simon Perkins, Kim McAlpine and Fernando Camilo for assistance and for providing the data used for this tests.

REFERENCES

[1] W. D. Cotton, "Automatic CLEAN windowing," *EVLA Memo series*, vol. 116, pp. 1–17, 2006. [Online]. Available: ftp://ftp.cv.nrao.edu/NRAO-staff/bcotton/Obit/autoWindow.pdf

[2] ——, "Performance Enhancement of the autoWindow technique," *Obit Development Memo Series*, vol. 9, pp. 1–3, 2009. [Online]. Available: ftp://ftp.cv.nrao.edu/NRAO-staff/bcotton/Obit/autoWin2.pdf

[3] B. G. Clark, "An efficient implementation of the algorithm 'CLEAN'," *A&A*, vol. 89, pp. 377–+, Sep. 1980.

[4] W. D. Cotton, "Obit: A Development Environment for Astronomical Algorithms," *PASP*, vol. 120, pp. 439–448, 2008.

[5] W. D. Cotton, "Effects of Baseline Dependent Time Averaging of UV Data," *Obit Development Memo Series*, vol. 13, pp. 1–9, 2009. [Online]. Available: ftp://ftp.cv.nrao.edu/NRAO-staff/bcotton/Obit/BLAverage.pdf