RICA - Radio Imaging Combination Analyzer

Miles Lucas[1, 2]

[1]*Iowa State University, Ames, Iowa*

[2]*National Radio Astronomy Observatory, Socorro, New Mexico*

ABSTRACT

1. INTRODUCTION

In radio synthesis imaging a common problem arising from interferometry is the lack of zero-spacing data. Without this data, images lack total power information. Single dish radio telescopes retain this total power imformation but lack the angular resolution capabilities of radio interferometers. To solve this problem, astronomers combine the interferometry data with the total power data. This report seeks to characterize the effectiveness of different combination methods.

There are four methods of combination that were tested. The first is CASA[1]'s *feather* task. This takes the total power image and gets the Fourier components of the total power image. These components are then added to the interferometer data. The addition is weighted such that the total power components have all of the weight near the zero-spacing and taper off so that at around a third of the max UV distance the interferometer has all the weighting. This smooth combination helps to avoid weird artifacting from either component of the combination.

The next method is using the total power image as the starting model in CASA's *tclean* task. In the Cotton-Schwab CLEAN algorithm (CSCLEAN), each major-cycle begins with a blank model image and constructs an image on it by using minor-cycle iterations. By using a starting model, this model is no longer blank, but is passed as a parameter. By using the total power image as the starting model, we can hope to retain the total power information through the CLEAN algorithm.
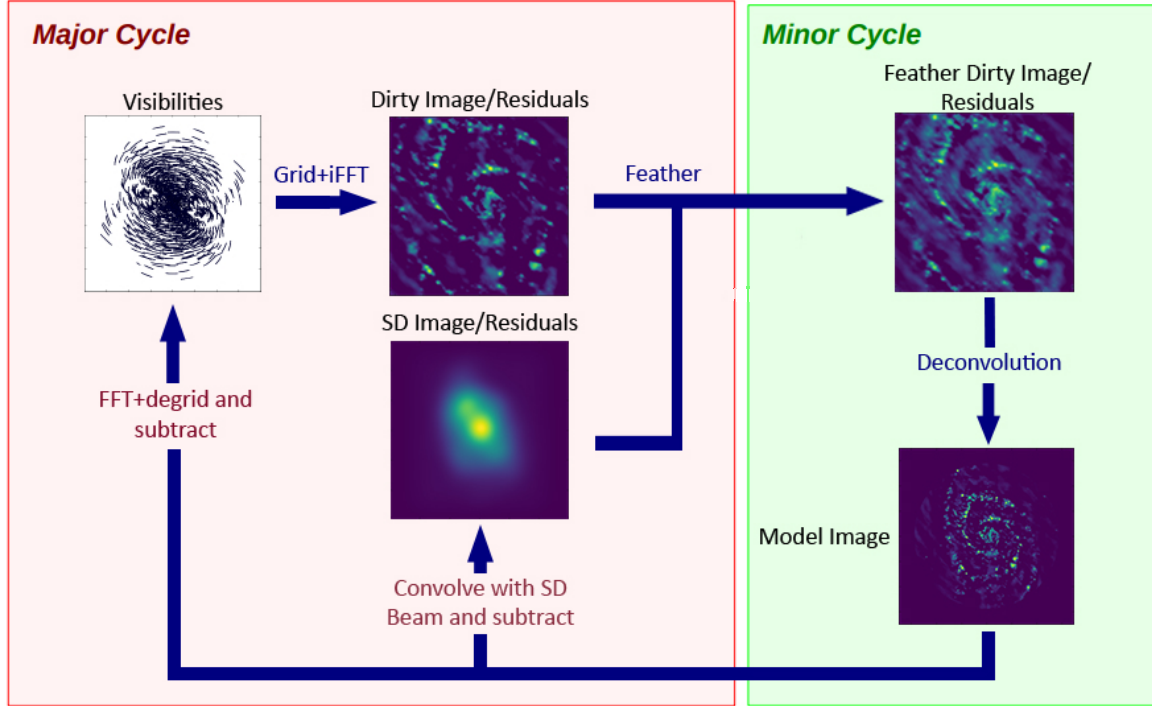
---

[1] https://casa.nrao.edu

Figure 1. A flow chart describing the joint-deconvolution approach to combining interferomoter and single dish data. Adapted from Rau, U. & Naik, N. 2018 (in prep).

Another modification to the Cotton-Schwab cycle in order to combine images is by doing a modified joint deconvolution. In each major Cotton-Schwab cycle, the visibilites are gridded and inverse Fourier transformed to create an image residual, which is then feathered with the total power residual before deconvolving. The deconvolved model is then convolved with the single-dish beam and subtracted from the previous single-dish residual. The deconvolved model is also Fourier transformed, degridded, and subtracted from the visibilites to create a new residual image These residuals are then degridded and transformed to finish the major cycle. See Figure 1 for a visual flowchart of the process.

The last method of combination we use is *tp2vis*[2] , which takes the total power image and spoofs a measurement set. The way *tp2vis* accomplishes this is by taking Fourier data from the total power image and sparsely sampling the Fourier data at close spacings (see Koda et al. 2011). This data
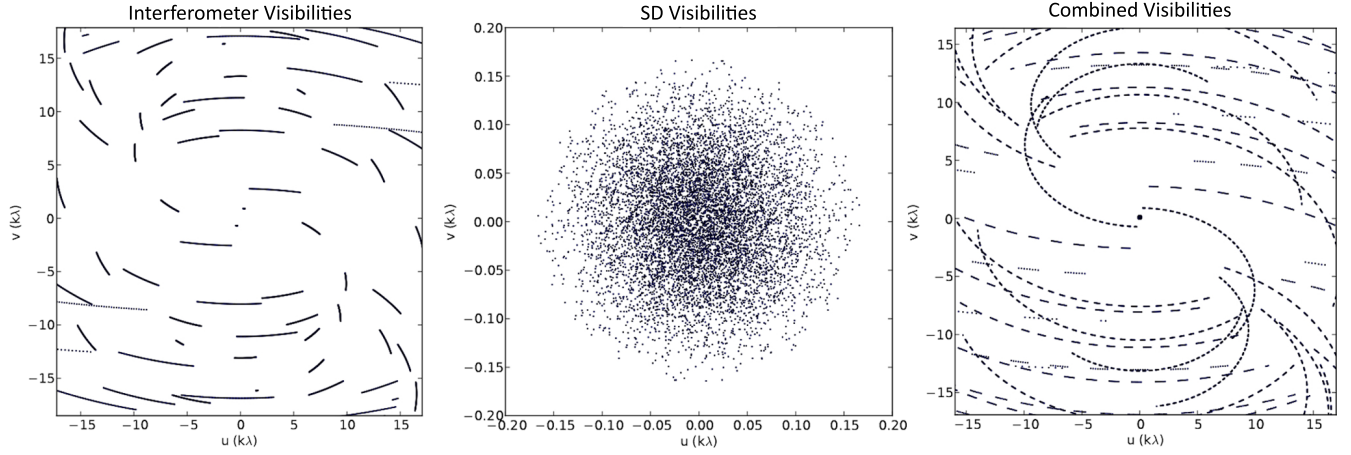
---

[2] https://github.com/tp2vis/distribute

Figure 2. A look at how the visibilites are added through using tp2vis for a model simulated on the VLA A configuration. In the center of the interferometer image in the first plot shows sparsity that is filled in on the third plot. The second plot shows the approximation tp2vis makes of the single dish.

can be concatanated with the interferometer data and deconvolved in *tclean*. Figure 2 shows how the single-dish is simulated and concatanated with the interferometer data. This should help fill the close-spacing gap created by the interferometer data and subsequently help increase total power information.

2. METHODS

In order to evaluate the different combination methods, a metric needed to be created and tested on a suite of models. The metrics used were CLEAN residuals, fidelity images, and a ratio of the power spectrum densities (PSD). The fidelities are given Equation 1

$$\text{Fidelity}(i,j) = \frac{|\text{Model}(i,j)|}{\max\left(|\text{Difference}(i,j)|, 0.7 \times \text{rms(Difference)}\right)} \tag{1}$$

where model is the reference image and diff is the difference between the test and reference images (Pety et al. 2011, p.19). The exigence for using PSDs is that the zero-spacing power is readily visible for every image. By using these ratios, the closer the ratio is to 1.0 at short spacings (when compared to the true model), the more accurate the combination. In addition, when compared with the total
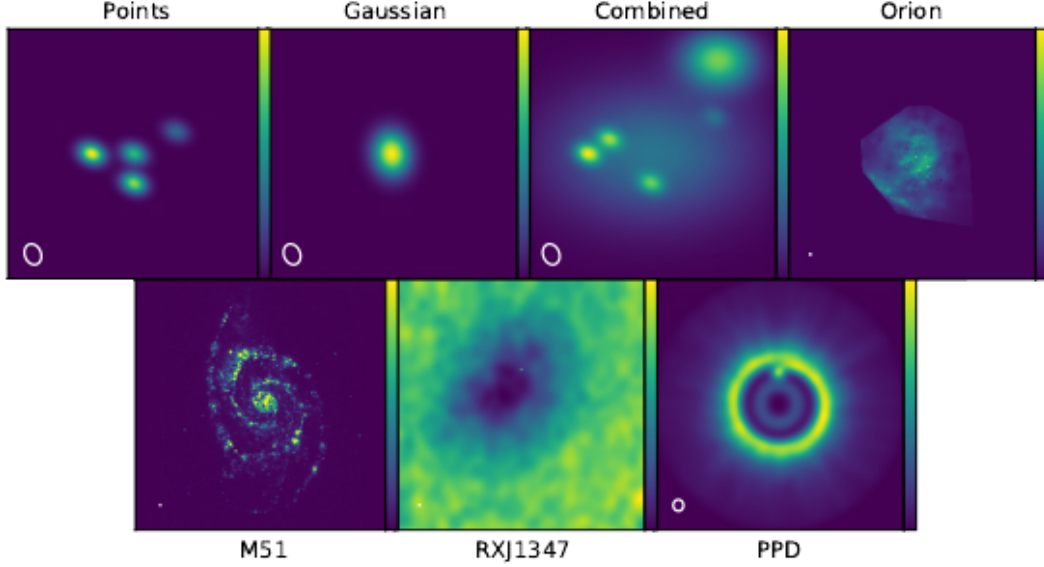
Figure 3. The models used for testing combination methods convolved with some restoring beam to show structure better.

power image, it shows how much effective weight is given to the total power image in the combination. The ratio of the PSDs is given by Equation 2

$$\text{Ratio}(UV) = \frac{\text{Power}_{test}(UV)}{\text{Power}_{ref}(UV)} \cdot \frac{\text{BA}_{ref}}{\text{BA}_{test}} \tag{2}$$

where Power is the power from the PSD, BA is the beam area, ref refers to the reference image (model or SD), and $test$ refers to whichever image is being tested.

Each combination was compared to both the true model and the single dish, total power image. Many models were tested with various extra parameters. Three models were generated from component lists. One has 4 point sources, one has a single Gaussian source, and one has a mixture of 4 point sources, one very broad Gaussian, and on off-center, stronger Gaussian. There were alos various models based off real structure, including M51 (based off an H-α image), Orion, RXJ1347, and a protoplanetary disk (PPD) simulation. Figure 3 shows all of the models. It is important to note that these models have been regridded onto a common coordinate system that is not representative of the true astronomical targets. This was in effort to simplify the simulation process for creating measurement sets.

The code base for testing the effectiveness is hosted publicly for anyone to use[3]. In the source code there are many scripts and methods to facilitate simulating, combining, and comparing. Every model is described in *src/_models.py* with a dictionary defining the simulation and *tclean* parameters. The pipeline for testing these models was as follows:

1. Simulate measurement set (MS) based on VLA configurations

2. Simulate single dish image by convolving model with Gaussian

3. Image the MS

4. Do the combinations (feather, startmodel, joint deconvolution, and tp2vis)

5. For each combination, do a comparison with the true model (convolved with the restoring beam) and the single dish image

and is implemented in *src/pipeline.py*.

2.1. *Simulations and Deconvolution*

All simulations were done using the VLA for the telescope model except for the PPD simulation, which was done using *simalma* and was only simulated once. Each test model could define which of the VLA configurations to use, from A, B, BnA, C, CnB, D, and DnC. The spectral window for the simulations mimic the C-band at $4.5\,\mathrm{GHz}$ to $5.5\,\mathrm{GHz}$ (central $5\,\mathrm{GHz}$) with 101 channels at $10\,\mathrm{MHz}$ intervals. The field observed was J2000 21h28m31.0 45°0′0.0″. The observation date and time was 2018/06/01 at 12:00:00.0 UTC. The integration time was $10\,\mathrm{s}$ each with a total of $30\,000\,\mathrm{s}$. Finally, the data was corrupted with $1\,\mathrm{mJy}$ of simple noise.

The single dish images were created by taking the true model and convolving with a Gaussian equivalent to the single dish beam at the given frequency.

$$FWHM(\nu, D) = \frac{3.66 \times 10^8\,\mathrm{m\,s^{-1}}}{\nu \cdot D} \tag{3}$$

---

[3] https://gitlab.com/mileslucas/rica

⁷⁹ Equation 3 gives the Gaussian kernel full-width half-maximum (FWHM) in radians for a frequency

⁸⁰ in Hz and dish diameter in meters. For instance, to simulate data from the Green Bank Telescope

⁸¹ (GBT) with a dish diameter of $100\,\mathrm{m}$ at $5\,\mathrm{GHz}$ gives a FWHM of $2'31''$, which is in agreance with

⁸² values from the GBT proposer's guide[4]. Note that this is not the most accurate equation for other

⁸³ telescopes, because there is a constant involved in this equation based off the taper-length of the

⁸⁴ telescope's feedhorns. In addition, these single dish images have 5×10^{-6} uniform noise added.

⁸⁵ Each model defines its own paramaters for cleaning. As a default, a model simulated in VLA

⁸⁶ D configuration has an image size of 64 with a cell size of $7.0''$. The restoring beam for the D

⁸⁷ configuration is approximately $20.41''$ by $15.60''$ so the cell accounts for half to a third of the beam

⁸⁸ width. In general, deconvolutions were done with very high numbers of iterations, relying on a

⁸⁹ threshold to stop the algorithm for consistency. This differs for the models tested at explicitly

⁹⁰ different cleaned levels.

⁹¹ 2.2. *Comparisons*

⁹² Each comparison was ran with the same clean parameters for consistency. Because *feather* does

⁹³ not actively clean, it has its own set of parameters, but all of the models tested used the default

⁹⁴ parameters. After all the combination permutations were imaged, they were compared against the

⁹⁵ true model convolved with the restoring beam and the single dish image alongside the uncombined

⁹⁶ interferometer image. All of the ratio were saved into individual CSV files for further analysis. In

⁹⁷ all, each model has 10 comparisons (5 agains the models and 5 against the single-dish images) and 4

⁹⁸ separate cleans to perform.

⁹⁹ In order to run a new model to test comparison methods, there are two general methods. If there

¹⁰⁰ exists a true model, simply add an entry to the models dictionary defining the simulation and clean

¹⁰¹ parameters. It is also important to edit *src/simulate.py* to accomadate the new model. Follow the

¹⁰² existing code base for guidance. It is also important to create a copy of the model and regrid it to

---

⁴ p.9, https://science.nrao.edu/facilities/gbt/proposing/GBTpg.pdf

the common coordinate system (use an existing model to snag the coordinate system). The model can then be ran through the pipeline along with any other models using *src/pipeline.py*.

If there is no true model, it is still useful to test and compare against the total power image, alone. In this case, using a cleaned image, a measurement set, and a total power image with the *src/combine.py* script will produce all of the combinations. These can then be compared using *src/compare.py*. It is possible to edit *src/pipeline.py* to accomadate special models, see how the PPD model is handled in the pipeline.

3. RESULTS

4. CONCLUSION

REFERENCES

Koda, J., Sawada, T., Wright, M. C. H., et al. 2011, ApJS, 193, 19, doi: 10.1088/0067-0049/193/1/19

Pety, J., Gueth, F., & Guilloteau, S. 2011, ALMA Memo #398 Impact of ACA on the Wide-Field Imaging Capabilities of ALMA, Tech. rep.