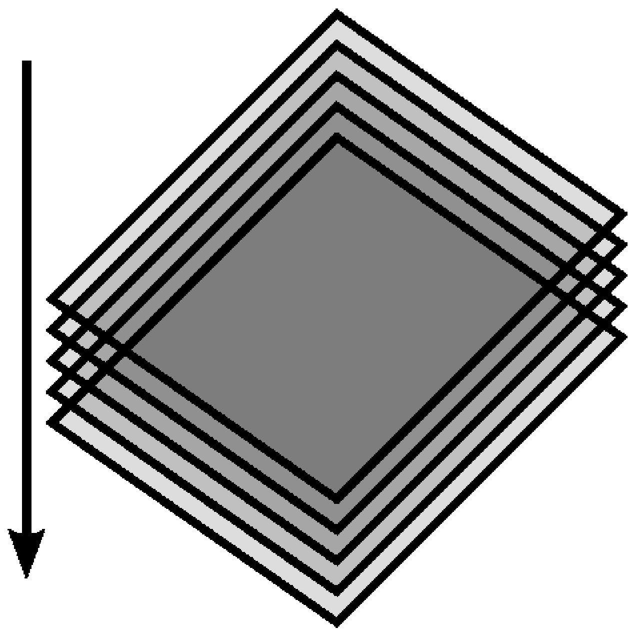
Python Astronomical Stacking Tool Array (PASTA) Manual

BW Keller
Astronomisches Rechen-Institut
Universität Heidelberg
kellerbw@mcmaster.ca

Jeroen Stil
University of Calgary
jstil@ucalgary.ca

September 17, 2018



Version 1.0
September 17, 2018

Contents

List of Figures

List of Tables

1 Introduction

PASTA is a software package written in the Python programming language for median stacking of astronomical sources. It includes a number of features for filtering sources, outputting stack statistics, generating Karma annotations, formatting sourcelists, and reading information from stacked Flexible Image Transport System (FITS) images. PASTA was originally written to examine polarization stack properties, and it includes a Monte Carlo modeller for obtaining true polarized intensity from the observed polarization of a stack. PASTA is also useful as a generic stacking tool, even if polarization properties are not being examined.

The basic operation of PASTA is to read in a sourcelist containing positions of sources to be stacked, as well as one or more FITS images. PASTA then generates an output list of stacked sources and their properties, and a pair of FITS files, one containing the median pixels of the stack, and the other containing the mean pixels.

Stacking allows the reduction of noise levels in the examination of sets of images. It produces a pseudo-source consisting of the median of all the stacked sources, with reduced background noise level. For more information on the performance of stacking, see [1, Stil et al. 2014].

1.1 Where To Obtain PASTA

PASTA is hosted on github (https://github.com/bwkeller/PASTA).

2 Software Requirements

PASTA uses a number of libraries to be installed prior to running. Most of these packages are available in the software repositories of the main Linux distributions.

2.1 Required Packages

- Python $\geq$ 2.4 *http://www.python.org*
- Numpy $\geq$ 1.2.0 *http://numpy.scipy.org*
- Scipy $\geq$ 0.7.0 *http://www.scipy.org*
- astropy $\geq$ 3.3.0 *http://astropy.org*

2.2 Recommended Packages

The following packages are not needed to run PASTA for stacking, but help for examining the results and using the plotting utilities.

- KARMA *http://www.atnf.csiro.au/computing/software/karma*
- matplotlib 0.98.6 or higher *matplotlib.sourceforge.net*

3 Basic Usage

3.1 Input Files

In order to run PASTA, you must first have one or more FITS images and a sourcelist. The sourcelist must be a flat text file, and contain 1 line for each source, with each column separated by white space. The values for each source must include a right ascension, and declination. You then simply run genstack.py, the stack generator, on the sourcelist and images, while specifying a stamp size (in pixels). i.e. (for an sourcelist storing the right ascension in the first column, declination in the second, and the intensity in the third column):

```
genstack.py -l "0 1 2" source_list source_image.fits 30
```

That command will take every source in `source_list` that can be found in `source_image.fits` and generate a set of output files, including FITS images of the median and mean stacked images.

3.2 Output Files

After genstack has completed running, it will create 4 files. Two of these files, `noise_source_list.dat` and `source_list.ann` are flat text files. The former contains a list of values for each stamp. This file is useful for looking at properties of the stack beyond the median/mean intensity, such as the distribution of background noise, the ratio of accepted/rejected sources, or for using as a sourcelist for running another stack. These columns are defined as:

1. Right ascension (Degrees)
2. Declination (Degrees)
3. Sourcelist intensity (Original list units)
4. Image intensity (FITS units)
5. Maximum pixel intensity (FITS units)
6. X location of maximum pixel
7. Y location of maximum pixel
8. Minimum pixel intensity (FITS units)
9. X location of minimum pixel
10. Y location of minimum pixel
11. Absolute Median of outer border (FITS units)
12. RMS of outer border (FITS units)
13. Rejection flag
14. Source name (Standard IAU format)

The rejection flags are described as follows:

a Stamp accepted for stack

d Stamp dimensions do not match specified dimensions, rejected (This can occur if the source is closer to the edge of the image than would allow a full stamp to be extracted)

n Stamp border absolute median above given noise threshold, rejected (This occurs when the stamp is pulled from an area of the image with a higher noise than specified. The absolute median is used to prevent rejection due to interloping sources, and instead only reject on more uniformly high background noise.)

b Stamp contains NaN pixels, rejected (This can occur if a source is in or near a blank area of the FITS image.)

α	δ	List Flux	Stack Flux	Max Flux	X	Y	Min Flux	X	Y	Background	RMS Noise	Flag	Name
357.533	-33.69944	2.14000E+00	8.16045E-01	1.19337E+00	22	3	3.96306E-02	11	19	4.69556E-01	5.82890E-01	a	J235008-331800
1.96625	0.71725	2.04000E+00	6.91129E-01	1.72867E+00	1	6	1.86304E-02	20	21	4.22984E-01	5.36032E-01	a	J000751+004300
1.98504	-0.12561	2.14000E+00	3.62772E-01	2.76332E+00	19	7	4.38120E-02	14	27	5.03270E-01	6.14033E-01	a	J000756-005200
2.05488	0.18464	2.05000E+00	5.54100E-01	1.75255E+00	8	12	5.52007E-02	28	11	4.51224E-01	6.07346E-01	a	J000813+001100
2.05758	-0.18597	2.21000E+00	1.10080E-00	1.62729E+00	17	15	4.46124E-02	1	18	4.69224E-01	5.62914E-01	a	J000813-004800
2.09183	0.11883	2.06000E+00	4.09074E-01	1.74689E+00	23	21	2.83552E-02	7	21	4.10038E-01	5.08445E-01	a	J000822+000700
2.13729	-1.48322	2.05000E+00	4.24715E-01	1.02757E+00	4	4	7.50757E-03	12	4	3.58331E-01	4.52300E-01	a	J000832-013100
2.15954	-1.52417	2.10000E+00	1.02308E-01	1.08270E+00	10	2	1.70393E-02	22	10	3.19294E-01	4.33960E-01	a	J000838-012800
2.19300	0.62006	2.03000E+00	4.23197E-01	1.68701E+00	21	14	4.63659E-02	12	27	6.06562E-01	7.57554E-01	a	J000846+003700
2.23992	0.03683	2.11000E+00	5.23646E-01	4.70614E+00	20	25	3.76965E-02	1	8	3.64927E-01	7.66693E-01	a	J000857+000200

Table 1: Example Noise File: NRAO VLA Sky Survey (NVSS) Stack, Sources with 2.00-2.24 mJy flux, 30 pixel stamps

The second text file contains a set of KARMA annotations that display circles over the sources used, and colour codes them. Accepted sources are green, sources rejected due to blank pixels in the stamp are yellow, and sources rejected due to high background level are coloured blue. Sources rejected due to dimension errors are not plotted (as they are dominated by off-image sources).

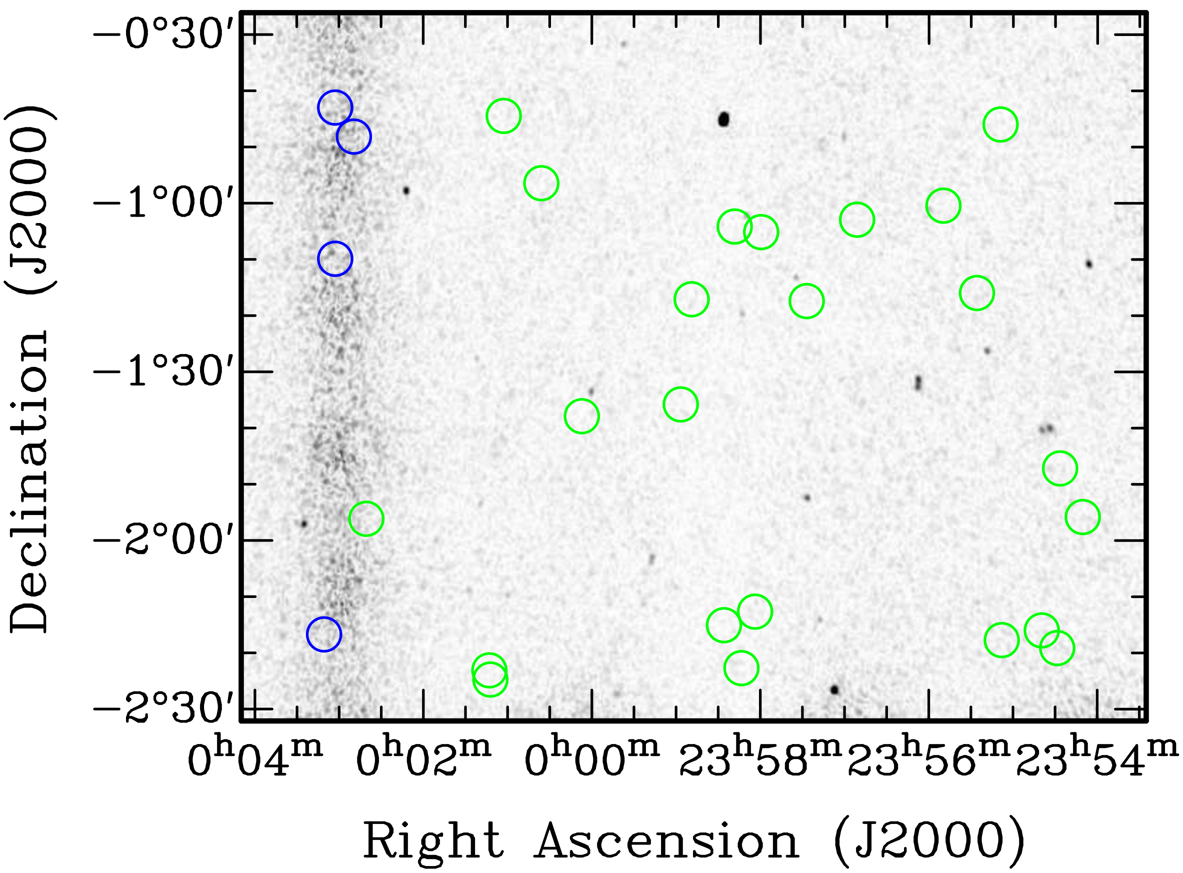


Figure 1: Sample of annotation file generated from NVSS stacking (polarized intensity)

The two other files created are both FITS files, `source_listsource_image.fits_median.fits` and `source_listsource_image.fits_mean.fits` one containing the median pixel values of the stack, and one containing the mean pixel values. The following example shows a stack with a peak brightness of 2.02 mJy and a root mean square deviation (RMS) background noise of 0.0465 mJy. This stack was generated using a sourcelist and images from the NVSS.[2, Condon et al. 1998] The "spokes" coming from the centre of the source are the sidelobes of the VLA beam. As these sidelobes lie in the same position within each stamp, they are preserved through the stacking process.

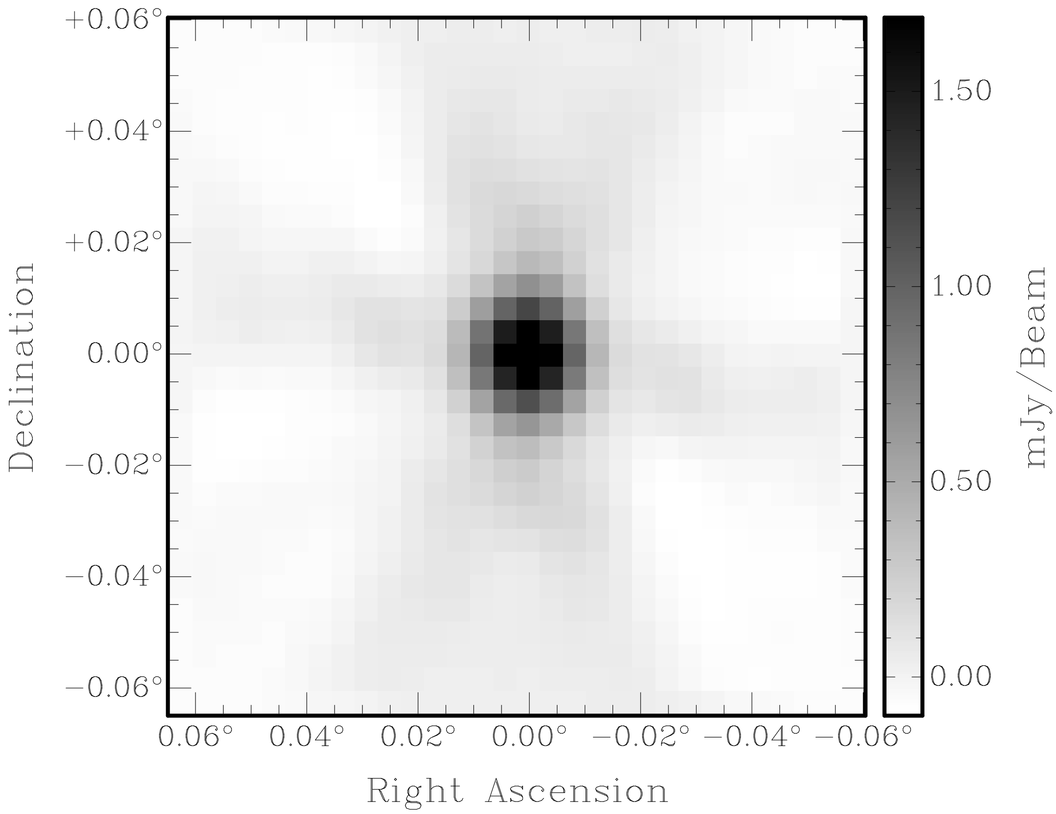


Figure 2: Stokes I Stack of NVSS sources from 2.000 to 2.244 mJy, stacked border rms of 46.5 μJy

Each of these FITS files contains an additional set of header values. These header values contain information about the stack not found in the pixel data.

STACKMIN: The minimum central pixel intensity in the stack (the 0^{th} percentile)

STACKMAX: The maximum central pixel intensity in the stack (the 100^{th} percentile)

STACK50: The median value of the central pixel column in the stacked image

STACK25: The 25^{th} percentile value of the central pixel column in the stacked image

STACK75: The 75^{th} percentile value of the central pixel column in the stacked image

STACKNUM: The number of stamps used to generate the stack

4 Features and Options

4.1 Genstack Command Line Options

genstack (The stack generator) includes a number of command line arguments. These arguments can be used to control the acceptance of stamps, transformations done to the individual stamps before stacking, and the what files to output.

–**help** Print out a summary of command line options

-n The input list is a genstack noise file, use only accepted sources.

-g n Regrid by this factor n

-l "a b c" In the input list, the RA, Dec, and intensity are stored in columns a, b, c

-m n The maximum median background value to accept is n

-s Save to disk each stamp used in the stack WARNING: Using regridding will cause the astrometry to be incorrect on these saved stamps

-a Do not save the annotation file

-i imagelist Read in an ASCII file 'imagelist' containing the filenames of the images to be stacked (one file per line). This is useful if you have a large number of files to read, and the command line can't handle them all. With this, you don't need `source_image.fits`

-p n Also save a FITS file containing the nth percentile valued pixels of the stack.

-d Similar to the previous option, this option will dump a number of stacked images equal to the total number of accepted sources. These stacks all correspond to the total range of percentiles in the stack (in other words, slices from the sorted stack)

-f "a b c" Generate a "fake source" of a simple circular Gaussian centred at the position of the source in the sourcelist, with a peak value of a, using the units of the source images, and a Full-Width Half-Maximum (FWHM) of b pixels. The peak value is either always a if c is set to "d" (for delta distribution), or a uniform distribution with median a if c is set to "u" (for uniform distribution).

4.2 Other Tools

4.2.1 readheader.py

The readheader tool is used to print out the special header values contained in the stacked FITS files. It can either print out a human-readable set of values and explanations, or print out a single row of data columns for generating tables (using the `readheader.py -t` option). All the units on the presented data are in the original FITS header units, which will be printed in the human-readable output along with each value. The columns of this tabular format is described as follows:

1. Minimum Stack Intensity (0^{th} percentile of the stack)
2. Maximum Stack Intensity (100^{th} percentile of the stack)
3. Median Stack Intensity
4. Lowest Quartile Intensity
5. Highest Quartile Intensity
6. Border RMS Noise
7. Border Absolute Median Intensity
8. Maximum Pixel Intensity
9. Maximum Pixel X Location
10. Maximum Pixel Y Location
11. Minimum Pixel Intensity
12. Minimum Pixel X Location
13. Minimum Pixel Y Location

14. Stamp X dimension
15. Stamp Y dimension
16. Number of sources stacked
17. File name

4.3 Polarization Estimator Usage

The polarization estimator tool takes the results of a polarized intensity (PI) stack along with a Stokes Q/U stack and generates an estimate of the true median polarization of the stacked sources. Once a set of these two stacks are created, the noise file from the Q/U stack is used to start the simulator. Using `Q_noise_file.dat`, the simulator is invoked by:

```
statter.sh Q_noise_file.dat
```

Once all of the simulation runs are complete, you can obtain the most likely true polarization by finding the resulting true polarization (as indicated in the output file name) whose file has a median polarization value closest to the stack observed polarization.

5 Internal Operation

5.1 Stack Generation

5.1.1 Stamp Creation

Once a sourcelist and FITS files have been read by genstack.py, the program proceeds to extract "stamps", square sections of the source image with a source centred within them. Each of these stamps has a number of calculations performed on it. If the regridding option has been selected, the image resolution is increased by the specified factor using cubic spline interpolation. The stamp is then shifted to centre the source more accurately, and it is then regridded back to the original resolution. The brightest and dimmest pixel locations are then found, and the pixel coordinates within the stamp are recorded.

A border region is then calculated. This border is simply all the pixels in the image within a threshold value from the edge of the stamp. This thickness threshold is the greater of either 5 pixels or 20 percent of the length of a stamp side. Of the pixels in this border region, and RMS values are calculated and stored, along with the median of the absolute value of each pixel. The median absolute value $\hat{\sigma}$ is used as a robust estimator of the RMS. $\hat{\sigma}$ is somewhat less affected by nearby sources, as well as able to work with Stokes $Q/U/V$ images, in which negative intensities can occur without introducing any ambiguity. This absolute median can be used to calculate the rms. For a Gaussian distribution, the absolute median covers 50% of the distribution. This same amount of area is covered by 0.67443σ, and therefore

$$\sigma \approx \hat{\sigma}/0.674443$$

After these calculations complete, each source is assigned conditional flags. The first check is simply to ensure that the stamp has the same dimensions as specified by the size option (stamps from the borders of images may contain only a portion of a full stamp.) If the absolute median is found to be above the maximum specified noise (default value is infinite), a stamp will be rejected as too noisy. If this flag is not tripped, but the raw RMS of the border is above the maximum specified noise, this is flagged as having a possible interloping source, but still used in the stack. The final check is to ensure that the stamp does not contain any blank or NaN pixels, as these interfere with the statistics of the final stacked results (by giving some pixel columns more sources to stack than others).

If the "save stamps" option has been selected, the stamp will then be written out to a specific fits file. This feature can be useful for creating individual source images of a list of sources, as well as stacking.

5.1.2 Regridding

If regridding is specified in the command-line arguments, before each stamp is finally added to the stack, PASTA will interpolate the image up by the factor specified in the arguments. The interpolation scheme is a simple two-dimensional cubic spline. Once this is complete, the image is shifted to move the source centre (as found in the source list) to the centre of the higher resolution image. This shifting is done using Numpy's `numpy.roll()` method, and the interpolation uses `ndimage.zoom()`. The stamps initially generated with the regridding option enabled have dimensions of $(\mathrm{N}+2)\mathrm{x}(\mathrm{N}+2)$ for a stamp size of N. This allows PASTA to trim the extra border of pixels off the image after it is regridded, in order to prevent edge-located objects and structure from tearing when the image is shifted. Once the image is then interpolated and shifted, the image is interpolated down to the original dimensions, and the extra pixels are trimmed off, resulting in a NxN stamp with a more accurately centred source.

5.1.3 Seeding Fake Sources

If the user requests a fake source to be seeded, and specifies a peak and FWHM value for the simulated sources, the following process will be applied to each stamp. After regridding is completed, a value will be drawn from a uniform distribution between 0 and twice the peak value (thus having a median value of the specified peak) if the uniform distribution is set. If the delta distribution is set, the peak value is simply set to the input peak value. A 2D, circular Gaussian with a FWHM (in pixels) as specified by the user is then generated with `Numpy's ndimage.filters.gaussian\_filter()` method. The peak is then scaled to match the random value previously drawn. This is then added to the stamp, and the stacking process resumes as usual.

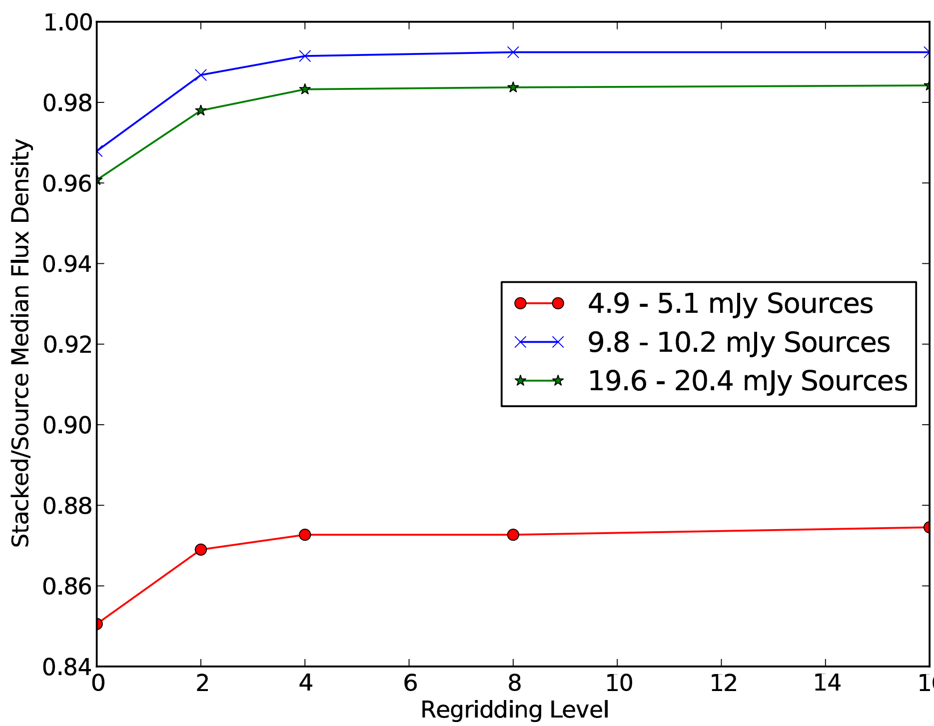


Figure 3: Accuracy of stack results as a function of the regridding level (shown for three different flux levels)

5.1.4 Stack Creation

As the stamps are created, if they are found to have no error flags, they are added to the stack. This stack is simply a 3-dimensional matrix of intensity values, with the first two dimensions (the stamp dimensions) holding spatial information. Once the stack has been fully populated with stamps, two 2-dimensional matrices are generated. These are the median and mean output images, and are generated by taking the mean and median, respectively, along the third dimension of the stack. These values are calculated using Numpy's `numpy.median()` and `numpy.average()` methods. Each of these is written out as its own FITS file, with the additional header values specified previously. The 25^{th} and 75^{th} percentile header values of the central pixels are also calculated along the third dimension. Along with these FITS files, the annotation and noise tables created using data from each stamp are written out.

5.2 Polarization Estimator

The results of a polarized intensity stack cannot be taken at face value to represent the true polarized intensity the result, as it has been long known that polarized observations are subject to bias. Removing this bias has been discussed heavily in the past, and the general distribution of polarized sources has been found to be Ricean [3, Simmons & Stewart 1985]. In order to take the results of a polarized intensity stack and obtain an estimate for the true polarized intensity of that stacked image, a Monte Carlo tool for estimating true polarized intensity has been provided.

5.2.1 Monte Carlo Modeller

To obtain a value for true polarization from an observed polarization an estimate of the percentage for the observed flux due to noise in the image must be determined. The best way to determine this is to find a distribution of the noise, and run Monte Carlo analysis with different intensities to find the most likely true polarized intensity which yields the observed polarization in the stack.

For a given true polarized intensity, the modeller draws a value from a polarized intensity distribution, and scales this value by the true polarization. It then takes a given Q/U absolute median, and draws a value from a noise distribution, scaling this value by the given noise value. It then uses this to generate an observed polarization value. For each stamp in a Q/U stack, the noise is read from the noise file generated by the stacker and run through the modeller. The median of the resulting observed polarization values is taken to create a simulated median stack.

This process is repeated 5000 times, and the true polarization found to be the value that produces a median polarization closest to the result of the polarization stack. Error estimates are produced by calculating the 16.5^{th} and 83.5^{th} percentile values that best fit the observed polarization ($2/3^{\text{rds}}$ of the expected values should lie within this range).

5.2.2 Polarization Distribution

In order to approximate the Rice distribution of polarized sources, an approximated distribution based on the [4, Beck & Gaensler 2004] distribution is used.

5.2.3 Noise Distribution

The included noise distribution used with the modellers was determined empirically from the NVSS survey images. By masking out sources, the background of the image was determined, and then a distribution consisting of the sum of a Gaussian and exponential [1, Stil et al. 2014] was fit to the resulting intensity distribution. As the noise levels, as well as artifacts/systematics, of different survey images will be different, this likely will need to be re-done for each different image survey set used.

6 Stack Performance

The noise level of a stacked image composed of N stamps, with initial RMS noise of σ_0 is roughly

$$\sigma = \frac{\sigma_0}{\sqrt{N}}$$

The actual performance is shown to conform well to this expected rate in polarized intensity, as shown in Figure 3.

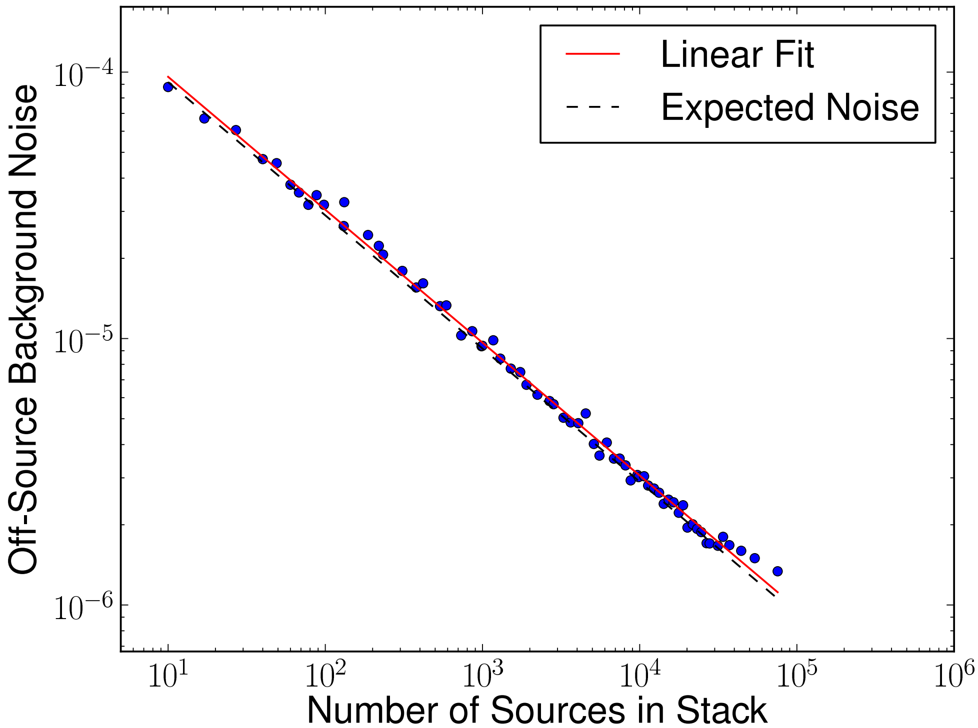


Figure 4: Background rms versus number of sources stacked (in polarized intensity)

The slope of the fitted line was found to be -0.49889, a 0.022% error from the expected -0.5, and the intercept was found to be -3.51871. This corresponds to a σ_0 of 0.303 mJy, while the NVSS images used had a σ_0 of 0.29 mJy [2, Condon et al. 1998], an error of 4.4%. The RMS of this error value was found to be 5.7% of the total RMS.

Similar performance can be found in the FIRST, VLSS, and NVSS Stokes I when random positions are stacked. We stacked random positions in order to prevent the increase in noise due to the sidelobes of the beam used to observe the objects. This noise increase can be seen by comparing the relatively low brightness Principal Galaxy Catalogue (PGC) objects stacked in similar numbers: at low enough noise, the flux from the sidelobes decrease the performance of the stack. If the stacked image was CLEANED, or if the beam was weighted out of the noise calculations, similar performance to the theoretical $\sqrt{N}$ noise reduction can likely be obtained.

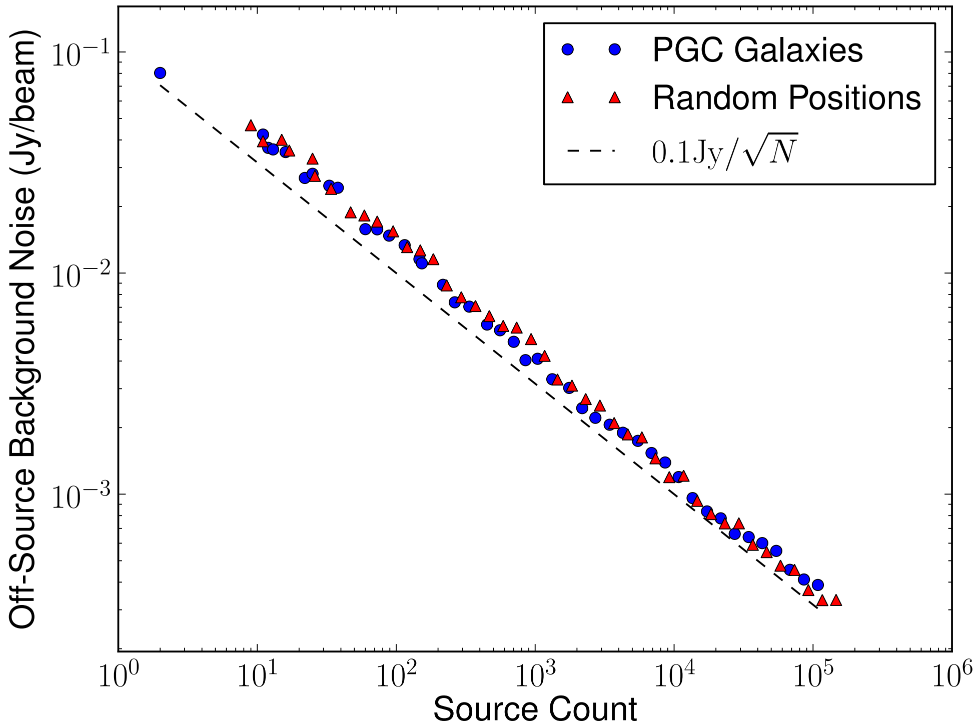


Figure 5: Background rms versus number of sources stacked, random sources and PGC galaxies (VLSS images)

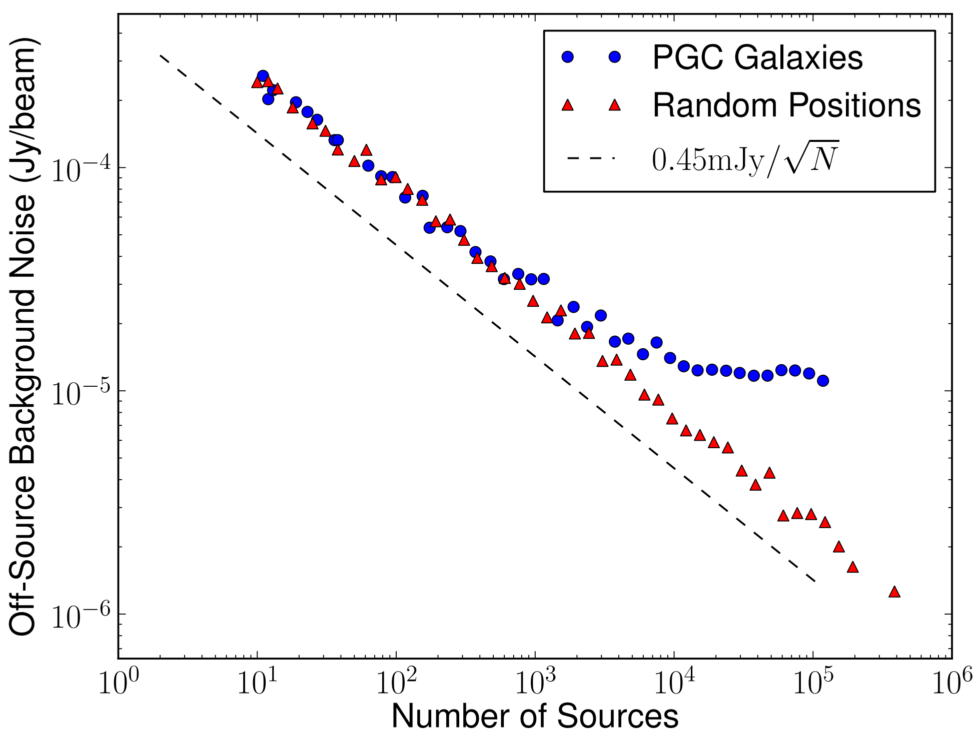


Figure 6: Background rms versus number of sources stacked, random sources and PGC galaxies (NVSS images)

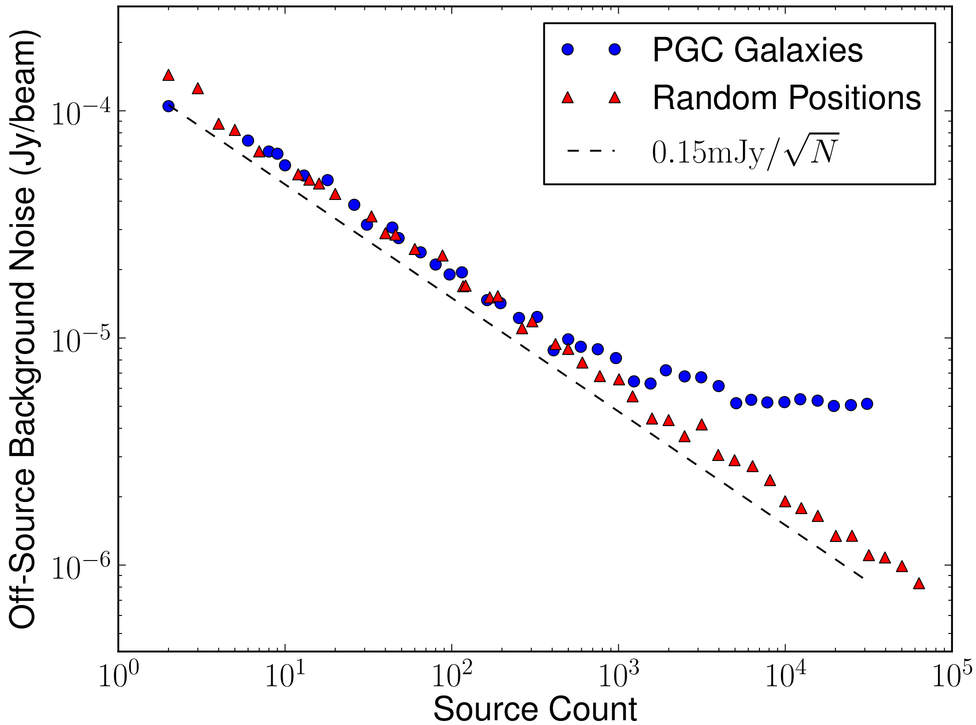


Figure 7: Background rms versus number of sources stacked, random sources and PGC galaxies (FIRST images)

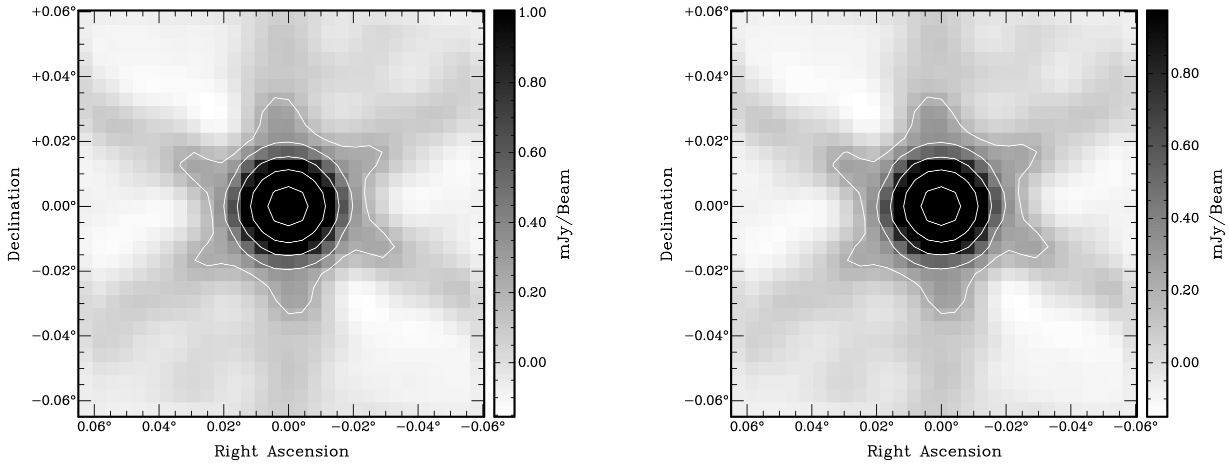
6.1 Limitations

6.1.1 Positional Accuracy

Because PASTA relies on the ability to place sources in an overlapping stack centred on each source's peak, the accuracy of positions given in the sourcelist is extremely important. Inaugurate positions cause flux to "leak" into adjacent pixels to the central pixels, causing the peak flux to be lower than the true median of the sources, and creating a larger source than should be found. This issue is why we regrid sources before stacking them, to ensure that sub-pixel resolution positions are used (images stacked without regridding exhibit the same limitations as images stacked with less accurate positions)

The effects of positional accuracy can be shown in the comparison of a sample of sources stacked using positions from the NVSS and the Faint Images of the Radio Sky at Twenty Centimetres (FIRST) survey. Both surveys were made with the VLA, but the FIRST used the VLA B configuration, with a synthesized beam of roughly 5", where the NVSS was observed using the VLA D configuration, with a beam of 45". Thus, the FIRST source positions are expected to be more accurate than the positions in the NVSS catalogue, with an expected positional error $\approx 1/9^{\mathrm{th}}$ the magnitude of the NVSS. A subset of sources found in both the NVSS and FIRST was selected with a NVSS peak intensity of between 3 and 5 mJy, and then stacked using both the NVSS positions and the FIRST positions for both the NVSS and FIRST images. The differences in flux and source size are not as obvious in the stacks derived from NVSS images, but in the stacks generated from the FIRST images, the source is clearly "blurred". The blurred stack shows a larger apparent source, with a lower peak intensity.

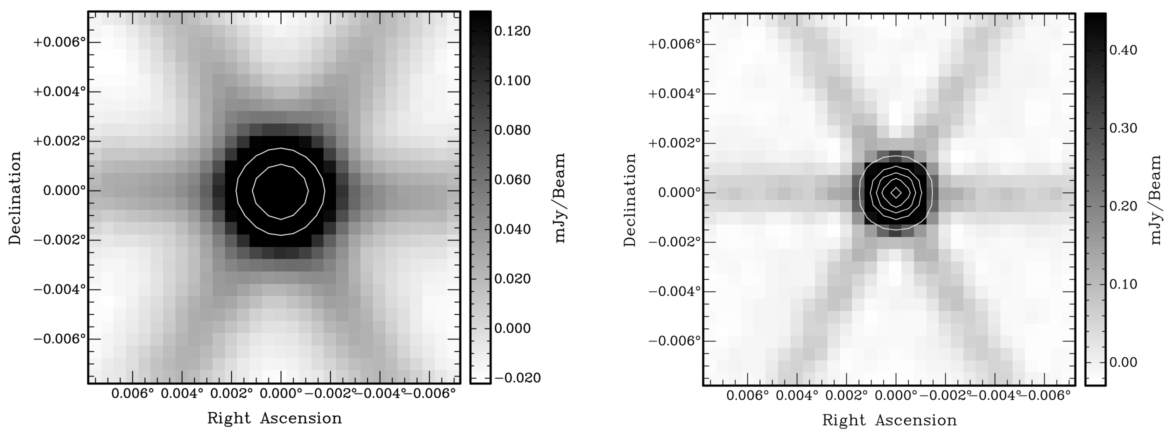
Fitting of source sizes is a simple test one can run on the results of a stack. This can be used as a test of both the positional accuracy, as was shown in comparing the NVSS and FIRST positions, and in testing that the sources stacked are indeed unresolved. If stacked image produces a source the same size of the beam, this will confirm that the positional accuracy is sufficient, and that the stack is dominated by unresolved sources.



(a) NVSS positions

(b) FIRST positions

Figure 8: NVSS images stacked using FIRST and NVSS positions



(a) NVSS positions

(b) FIRST positions

Figure 9: FIRST images stacked using FIRST and NVSS positions

6.1.2 Unresolved Structure & Directionality

For objects such as spiral galaxies, two similar unresolved sources can have significantly different flux due to unresolved properties. One such example is the axial ratio of spiral galaxies (B_{maj}/B_{min}), ie. face-on versus edge-on orientation. A face-on galaxy will obviously have a larger solid angle than an edge-on galaxy at the same distance. This will affect the flux observed from that galaxy. This effect can be observed by stacking lists of similar galaxies split into bins by their axial ratio. The following stacks and plots were generated by doing this to a list of spiral galaxies obtained from the PGC type Sb and Sc spiral galaxies, using NVSS Stokes I images. The peak flux density obtained from stacking these galaxies altogether, with axial ratio below 1.3 (face on), and above 4.0 (edge on), were 727.03 μJy, 1.04 mJy, and 430.06 μJy respectively.

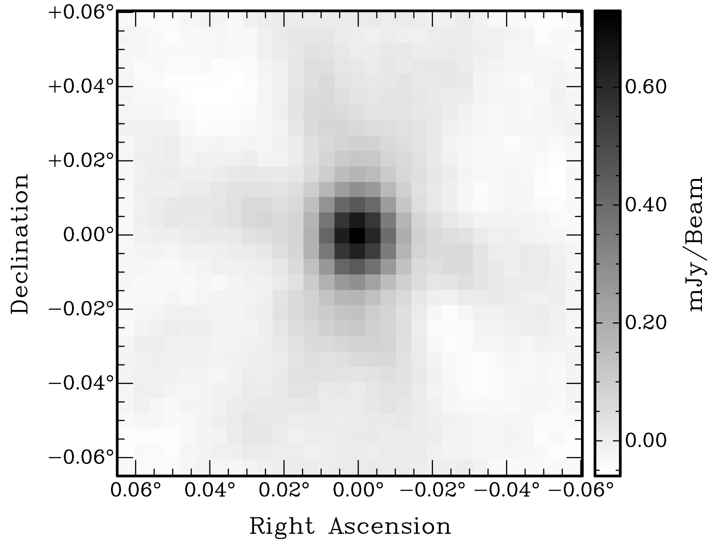
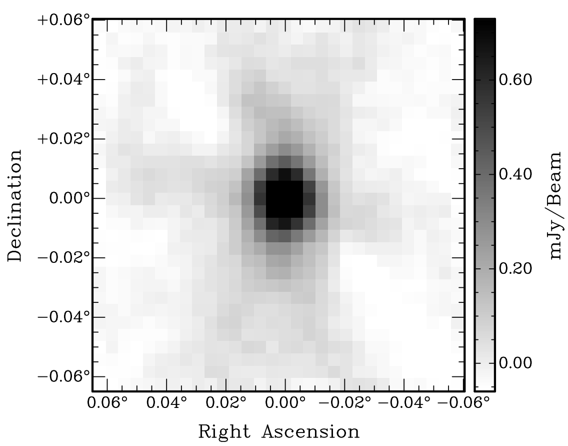
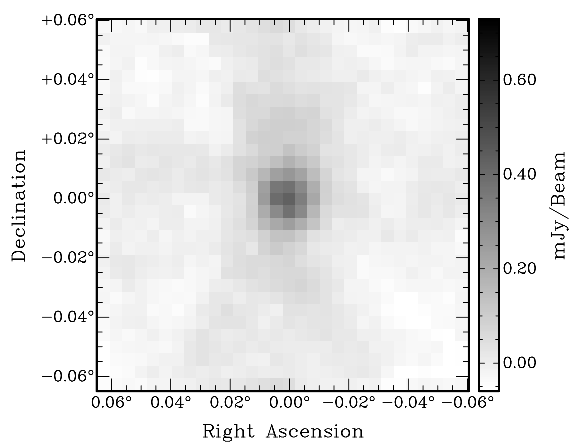


Figure 10: Stack of spiral galaxies without selection by axial ratio



(a) Axial Ratio below 1.3 (face on)



(b) Axial Ratio above 4.0 (edge on)

Figure 11: Stacks of spiral galaxies with axial ratio below 1.3 and above 4.0

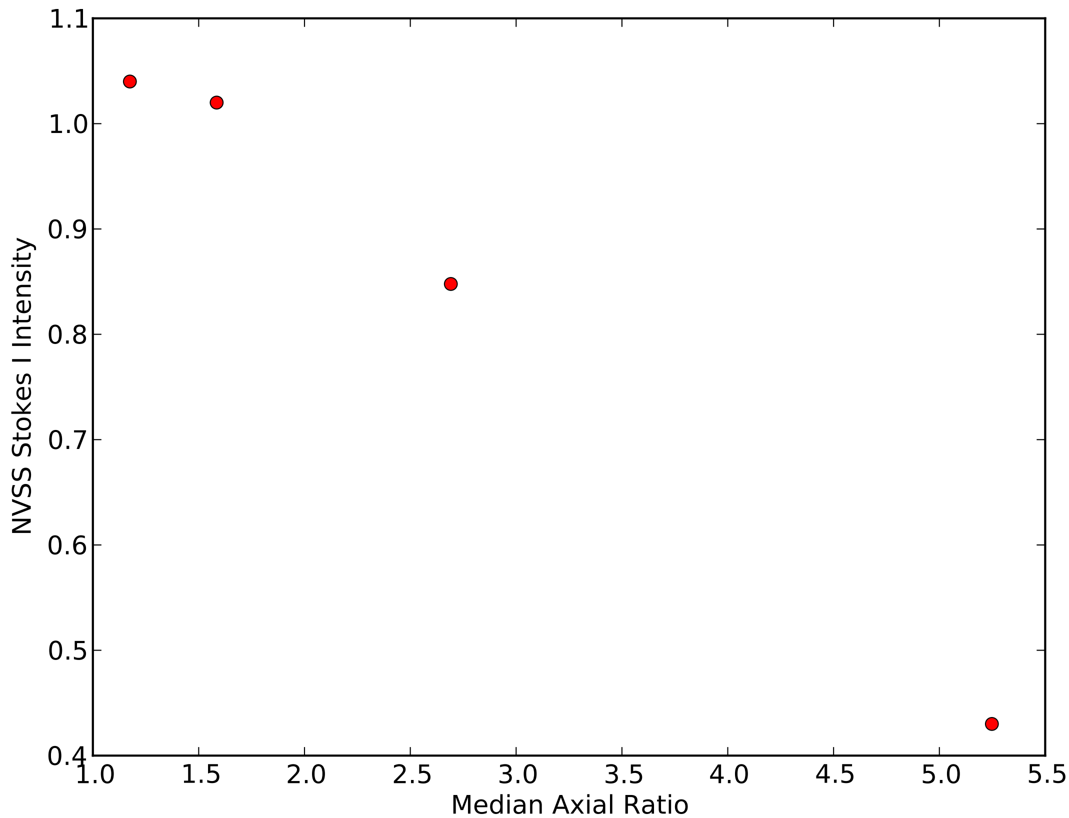


Figure 12: Median Stokes I flux as a function of axial ratio B_{maj}/B_{min} in spiral galaxies

7 Usage Example: Simulation Stacking

7.1 Introduction

In order to test the behaviour of the modeller and the stacking process in comparison to known results. In order to do this, a set of simulated images were seeded with sources of a known percentage polarization, and stacked. This allowed us to verify that the true polarization produced by the stacking and modelling matched the polarization of the simulated images.

7.2 Simulated Image Generation

The simulated images are intended to match as closely as possible the images of the real NVSS. The steps used to generate these images are as follows:

1. Each 4° by 4° simulated image is first seeded with 1639 sources, the average number of polarized sources found in a real NVSS tile. These sources have random intensity with a median polarized intensity constant across brightness at 2% of the total source brightness. These sources are seeded using the same beam produced by the Very Large Array (VLA) antenna configuration in the NVSS

2. Once all 1639 sources have been seeded, Gaussian noise is added to each image, along with a random weighting to simulate areas of poor coverage in our real NVSS images.

3. The images are used to create FITS files of PI, as well as Stokes I, Q, U, and the source information is written out into a sourcelist for this set of images.

4. This process is repeated 479 times to produce an area covering a band about the celestial equator from -20°to 20°declination.

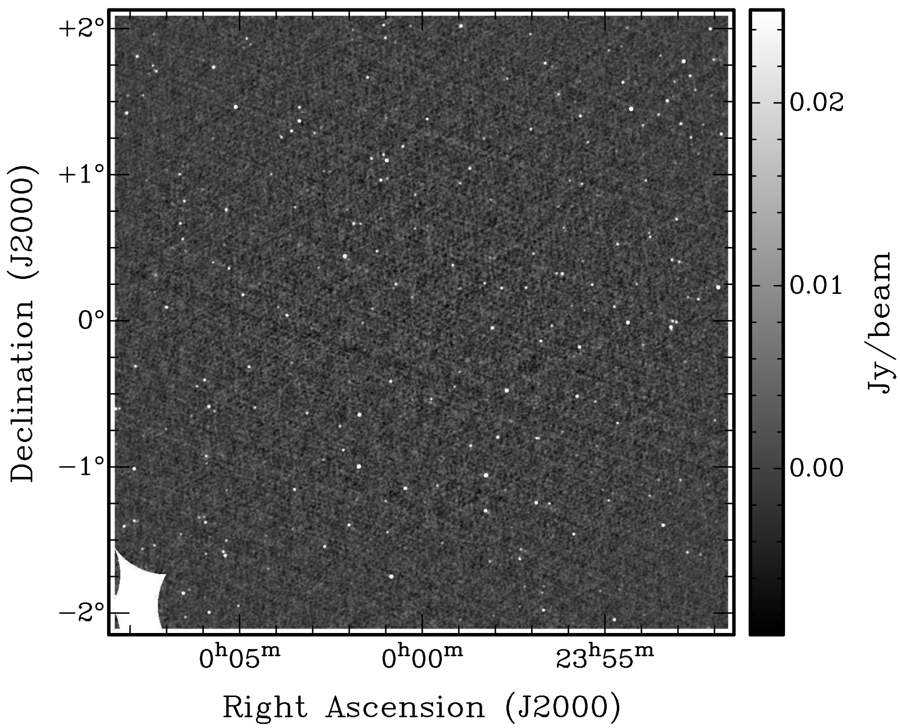


Figure 13: Simulated NVSS Stokes I 4° by 4° image tile

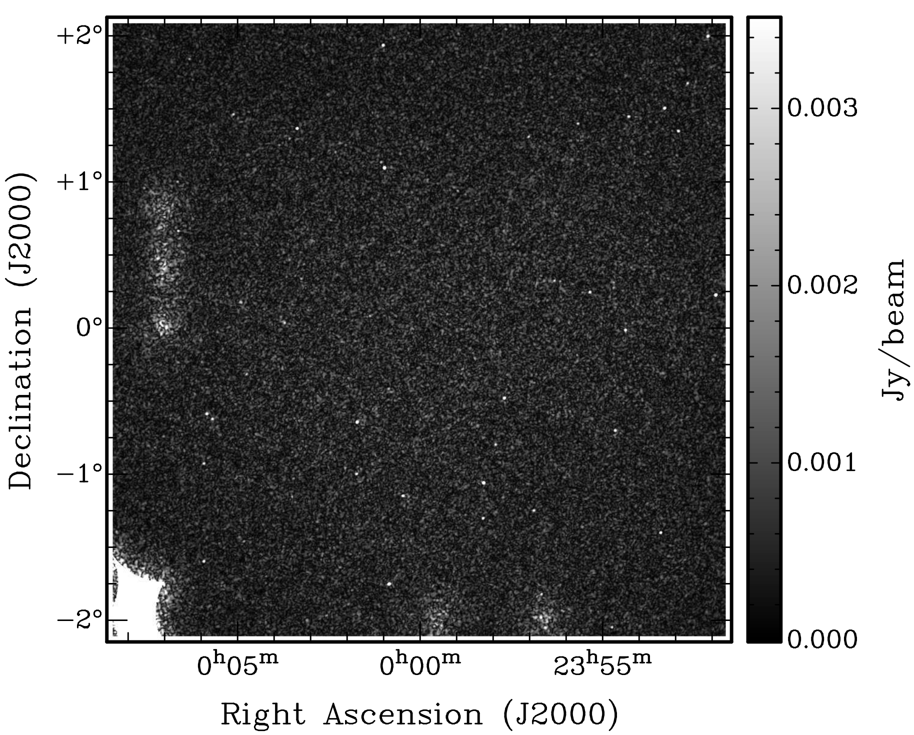


Figure 14: Simulated NVSS Polarized Intensity 4° by 4° image tile

7.3 Stacking the Simulation

The process of stacking the simulated images is essentially the same as the process for stacking the real NVSS images. Obviously, the only difference is that the sourcelists and images are the simulated images and the sourcelist for the simulated sources.

The modeller was running using the same PI distribution as the NVSS, with a similar Gaussian plus exponential distribution for the noise. The noise distribution used slightly different parameters that were obtained empirically from the images using the same technique as the one used to generate the noise distribution parameters for the NVSS images.

7.4 Analyzing the results

As the figure on the next page shows, the simulation images have similar properties to the real NVSS images when stacked. In the bottom pane showing the percentage polarization, it is clear that from 1 mJy to 500 mJy Stokes I intensity, the percentage polarization remains fairly constant, consistent with the seeded polarization.

In the high intensity end of the curve, we find the percentage polarization diverging above the constant value, as does the low intensity end. Both of these regions are expected to have less reliable polarization estimates. The higher intensity region suffers from lower source counts, due to the fact that sources bright enough for those bins are outliers on the intensity distribution. Low source counts suffer from both higher RMS noise, as well as simple small number statistics. The low intensity region is also found to have both high errors and divergent percentage polarization. This is simply due to the fact that the signal to noise level in this region is extremely low. A 2% polarized source with 1 mJy total flux has only 20 μJy flux in PI. In a single image, this corresponds to a S/N ratio of 0.07, as the background RMS was seeded to match the NVSS noise of 290 μJy.

A greater number of stacked images would reduce the uncertainty in the percentage polarization, so if one finds that the values obtained from a set of stacked bins are too uncertain, this can be somewhat reduced by decreasing the total number of bins, and thus increasing the number of sources in each bin. Obviously, this can only be extended so far. This issue must be considered when stacking populations of very dim or very sparse sources.

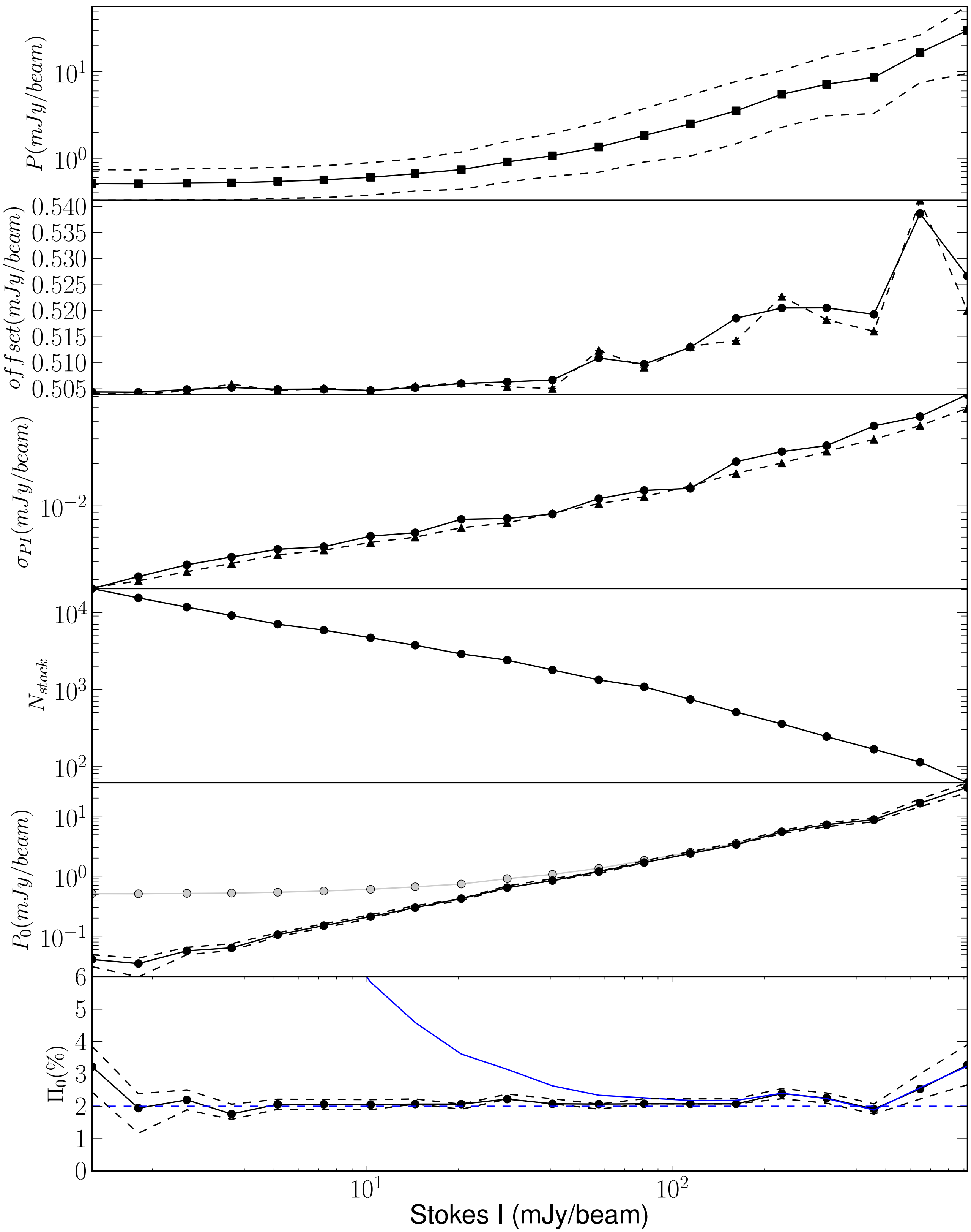


Figure 15: Statistics of Simulated NVSS Stack

8 Usage Example: NVSS Stacking

8.1 Setup

The PASTA suite of tools can be used for determining a percentage polarization curve for the NVSS extragalactic sources. In order to do this, 4°x 4°images in Stokes I, Stokes Q, and PI are obtained. In addition to the two sets of image tiles, a sourcelist containing the RA, Dec, and Stokes I value of each source is used. This list is split into logarithmic Stokes I flux bins, defined as:

$$\mathrm{F_{upper}} = 10^{0.05}\mathrm{F_{lower}}$$

with a minimum flux of 2.000 mJy and a maximum flux of $\approx$ 1 Jy. The RMS background of the NVSS is 290 μJy in Stokes Q and U, and 450 μJy in Stokes I. Running at a maximum noise of 2σ yields a maximum noise value of 580 μJy for the Q stack.

8.2 Stacking Stokes Q

The first stack to be generated is a Stokes Q stack. Our noise rejection is done based on the noise in Q, as this noise value is what is used by the modeller to generate an estimate on the true polarization. In order to generate the stacked results, the following command is used:

```
genstack.py -g 8 -m 0.00058 fluxbinlist  list_of_Q_images 30
```

This will generate a stack for the given list consisting on stamps 30 pixels square pulled from the list of Q images. Each stamp will have an absolute median border value of less than 580 μJy, and will have been regridded by a factor of 8.

8.3 Stacking Stokes I and PI images

Once a set of Q stacks has been completed, the noise files from each stack are used to generate I and PI stacks. The command used to generate these stacks is:

```
genstack.py -n -g 8 noise_fluxbinlist.dat list_of_I_images 30
genstack.py -n -g 8 noise_fluxbinlist.dat list_of_PI_images 30
```

This will generate stacks for I and PI containing only sources accepted in the Q stack run earlier.

8.4 Running the Noise Modeller

Once the stacks have completed running, the modeller is run on the output. In order to generate a set of modelled stacks, the following command was issued for each Q noise file:

```
mk_stat_varp0 noise_fluxbinlist.dat min_flux max_flux step
```

The `min_flux, max_flux,` and `step` are the values of the minimum and maximum true polarized intensity to test, and the step size between them. These units were in Jy, as the NVSS images store in units of Jy. This generates a series of files, with file names containing the true polarization values used to generate them. To obtain the most likely polarization for a given stack, the median value of the observed polarizations stored in these files was compared to the observed stack polarization. The closest result is the most likely true polarization.

8.5 Analyzing Results

The resulting stack results can be passed to the modelling software in order to obtain an estimated P_0. The result can be read by `mktable.sh` to generate a plot of the statistics of the NVSS stack. To do this, simply pass the name of the Q noise file as an argument to `mktable.sh`:

```
mktable.sh noise_fluxbinlist.dat
```

The table shown on the following page is a plot of the resulting statistics generated by the stacking process. The 6 panels, from top to bottom, are described as follows:

1. Shows the observed PI of the stacked image, with the solid line corresponding to the median polarization, with the two dashed lines corresponding to the upper and lower quartiles of the stack.

2. Shows the absolute median background levels for the stack (solid line), and the modeller (dashed line).

3. Shows the RMS value of the stack (solid line), and the modeller (dashed line)

4. Shows the number of stamps in the stacked image.

5. Shows the true polarization value (black line), along with the 16.5^{th} and 83.5^{th} percentile values (dashed lines), and the observed polarization (grey line)

6. Shows the percentage polarization of a number of values. The dashed blue line shows a constant 2% polarization. The solid blue line shows the observed polarization. The solid black and dashed black lines are the most likely true polarization and 16.5^{th} and 83.5^{th} percentile values obtained from the modeller.

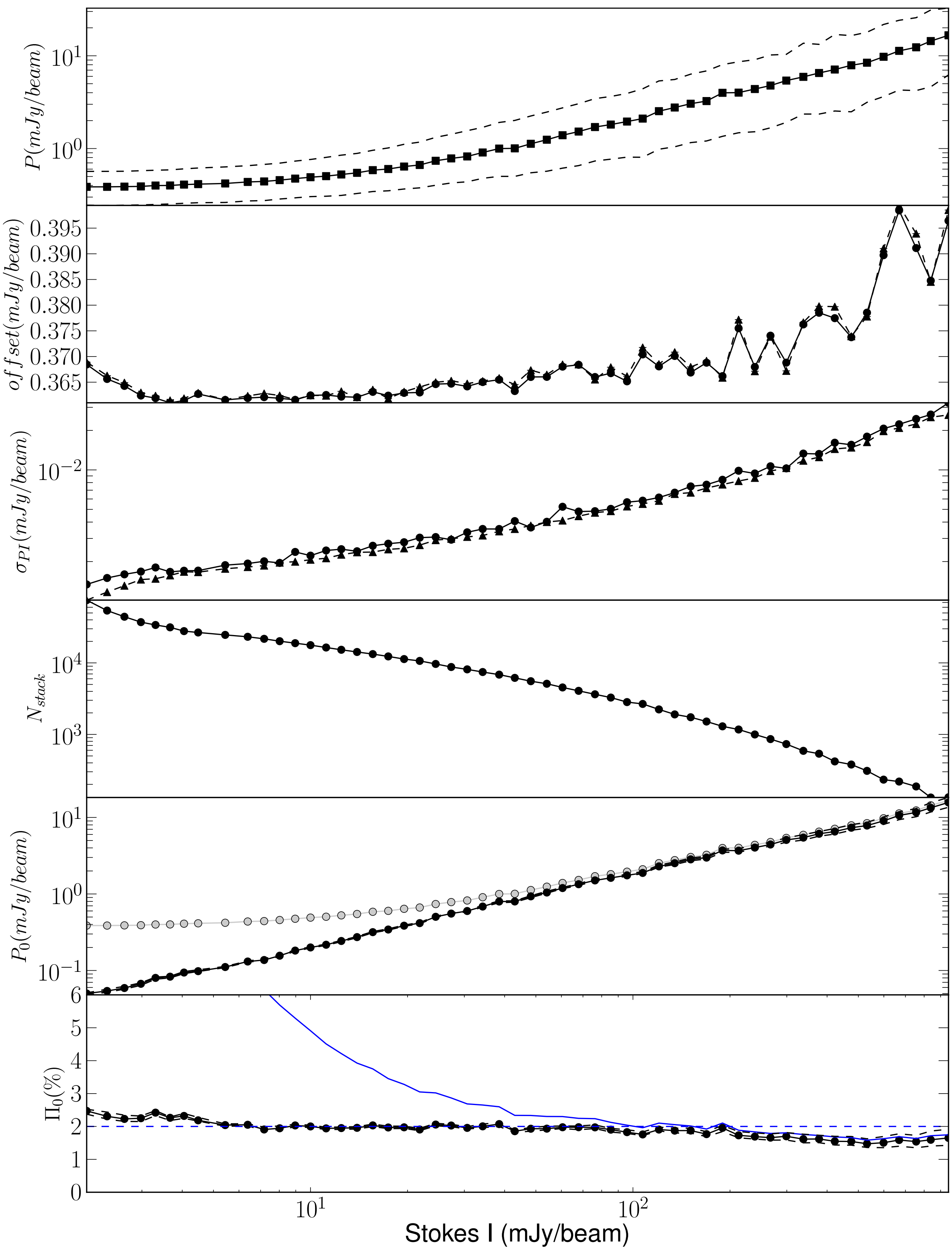


Figure 16: Statistics of NVSS Stacks as a function of Stokes I Intensity

9 Acknowledgements

PASTA was written by BW Keller, with scientific guidance from Jeroen Stil. This work was supported by an Discovery Grant from the Natural Sciences and Engineering Research Council to Jeroen Stil.

10 References

[1] Stil, J. M., Keller, B. W., George, S. J., Taylor, A. R.: 2014 *The Astrophysical Journal* **787** 99

[2] Condon, J. J., Cotton, W. D., Greisen, E. W., Yin, Q. F., Perley, R. A., Taylor, G. B., Broderick, J. J.: 1998 *The Astronomical Journal* **115** 1693

[3] J.F.L Simmons, B.G. Stewart: 1985, *Astronomy and Astrophysics* **142** 100

[4] R. Beck, B.M. Gaensler: 2004 *New Astronomy Reviews* **48** 1289