VLASS Project Memo: 20

Utilizing a Customized VLASS Single Epoch Continuum Pipeline for End-User Science

Erik V. Carlson (University of Rhode Island, NRAO[1])
Amy E. Kimball (NRAO)
Pallavi Patil (NRAO, Johns Hopkins University)
May 24, 2024

1 Introduction

We have developed a method for utilizing the VLASS imaging pipeline recipe to image VLASS data at a specific location and for a particular image size chosen by an end-user. The necessary workflow is presented here to guide users in producing high-quality images from VLASS data on their own, while conserving disk space and computational resources as much as possible.

Official VLASS images are made at NRAO using the standard VLA imaging pipeline with VLASS-specific recipes. "Quick Look" (QL) images are produced using basic cleaning algorithms and large pixel size ($1''$) that result in some compromises in accuracy, with the advantage that they can be produced and made available to the community in a short time-frame ($\sim$weeks) after observations are completed (see Lacy et al. 2019). The higher-quality Single Epoch (SE) images use smaller pixels ($0''.6$ for a better sampling of the $\sim 2''.5$ synthesized beam) and more advanced imaging algorithms (including iterations of automated masking and self-calibration) that necessitate a longer production time. Creating one SE image using VLASS's default pipeline parameters can take on the order of days or weeks due to the large data sizes, multiple calls to *tclean* for the masking and self-calibration iterations, and deep cleaning.

Given the 5TB disk space quota of observer accounts at NRAO[2], storing/imaging of multiple full VLASS measurement sets is unwieldy. The workflow presented in this memo guides users in identifying and extracting the subset of data that is of interest for their particular science imaging case, saving substantial storage space. Additionally, creating a smaller image (than a standard 1×1 deg^2 VLASS image) saves significant processing time; depending on the desired image size, compute time could be reduced from several days or weeks for standard image production to just a few hours for an image that is 500×500 arcsec^2. Most of the speed-up is due to faster *tclean* runs, but some processing time is saved by allowing the pipeline to import a small subset of data; this is especially important if the user makes multiple imaging attempts with the same data set.

Section 2 of this memo describes the steps necessary to produce a user-defined image with the VLASS SE Continuum Imaging pipeline recipe. Section 3 compares the results from this workflow to SE images made with the official VLASS pipeline processes. Section 4 summarizes with recommendations to the user. The Appendix (§5) presents a specific user-defined imaging example.

---

[1]NRAO Graduate Student Intern; Summer 2021

[2]Remote observers can request a temporary computer account. To request a temporary computer account follow the instructions at https://info.nrao.edu/computing/guide/cluster-processing/user-accounts.

2 Process for Producing a User-Defined Image

The pipeline recipe used by NRAO for VLASS SE Continuum Images was developed specifically to produce standard VLASS data products: images of a fixed size, with certain default values hard-coded into the pipeline recipe, and for a pre-determined set of phase centers across the sky. For finer control of the imaging process, the user can provide a custom input parameter file that will override certain default inputs as needed. This process requires some coordination by the user.

At a high level, the three steps to producing a user-defined image are:

1. Determining angular image size for the pipeline input parameter file

2. Identifying and obtaining the relevant VLASS data

3. Running the VLASS imaging pipeline with user-specified input parameters

Each of these steps will be briefly explained in the subsections below. The Appendix (§5) provides an example guide that demonstrates each of the steps in detail.

2.1 Determining Angular Image Size for the Pipeline

The user should approach this workflow with an idea in mind of how large an image they would like to obtain. Hereafter referred to as the "science image", this can be considered as the rectangular area to be imaged with roughly homogeneous sensitivity, e.g., fully covering the target of interest. The actual image created by *tclean* in the CASA pipeline (hereafter "full image") must be significantly larger than the science image owing to a combination of two reasons: (1) data provided to *tclean* should ideally include all VLASS fields that contribute "significant sensitivity" to the science image, and (2) the full image made by *tclean* must be large enough to include all of the provided fields without aliasing[3] sources into the science image. The pipeline will create the full image, then use CASA's *imsubimage* module to select out the central region of the full image, yielding a final science image. The full image size compared to the science image size is discussed in more detail below, and an illustration is provided in Figure 1, adapted from the VLASS Technical Implementation Plan.[4]

The determination of what constitutes "significant sensitivity" from any particular field is somewhat subjective. The standard VLASS procedure includes fields whose pointing centers are within $R_{\mathrm{g}} = 1000''$ of an edge of the science image. This angular distance corresponds approximately to the radius where the Gaussian primary beam response at the lower end of the VLASS bandwidth—2 GHz—falls to 17% of maximum.[5] A user can opt for a smaller value for R_{g} at the expense of losing some sensitivity at the science image edges.

Regardless of science image size or R_{g} value, an additional buffer must be included for calculating the even larger "full image" size in order to avoid aliasing: the full image must encompass all of the area where the provided fields contribute significant sensitivity, i.e., the area where sensitivity is above $\sim 17\%$ in any of the included fields. In other words, if one edge of the $> 17\%$ sensitivity circle of a field intersects with the science image, then the full image must be large enough to encompass the entire $> 17\%$ sensitivity circle for that field (see Figure 1). In practice, the VLASS team determined that it is sufficient to extend the full image by $840''$ beyond the furthest field positions outside the science image; this value corresponds to where the primary beam sensitivity drops to $f = 0.17$ at a frequency of 2.4 GHz. Therefore, for a science image with rectangular size of L_{sci} that includes field centers a distance R_{g} beyond the science image edges, the full image size should be at least $L_{\mathrm{sci}} + 2 \times (R_{\mathrm{g}} + 840)$ in arcseconds. This value must be converted to pixels (VLASS uses $0''\!.6$-pixels for SE images); the pipeline input parameters "cell" (pixel size) and "imsize" are passed directly from the pipeline input to CASA's *tclean* task.

---

[3] Aliasing is a result of finite sampling in digital signal processing. In interferometric imaging, aliasing can cause sources outside the imaged field of view to falsely appear inside the image.

[4] https://www.authorea.com/users/4076/articles/8161/

[5] The VLA primary beam at 2 GHz has FWHM of $21'$. A Gaussian falls to $f = 0.17$ at a radius equal to $0.8 \times$ FWHM, or $1008''$ for a $21'$ primary beam.
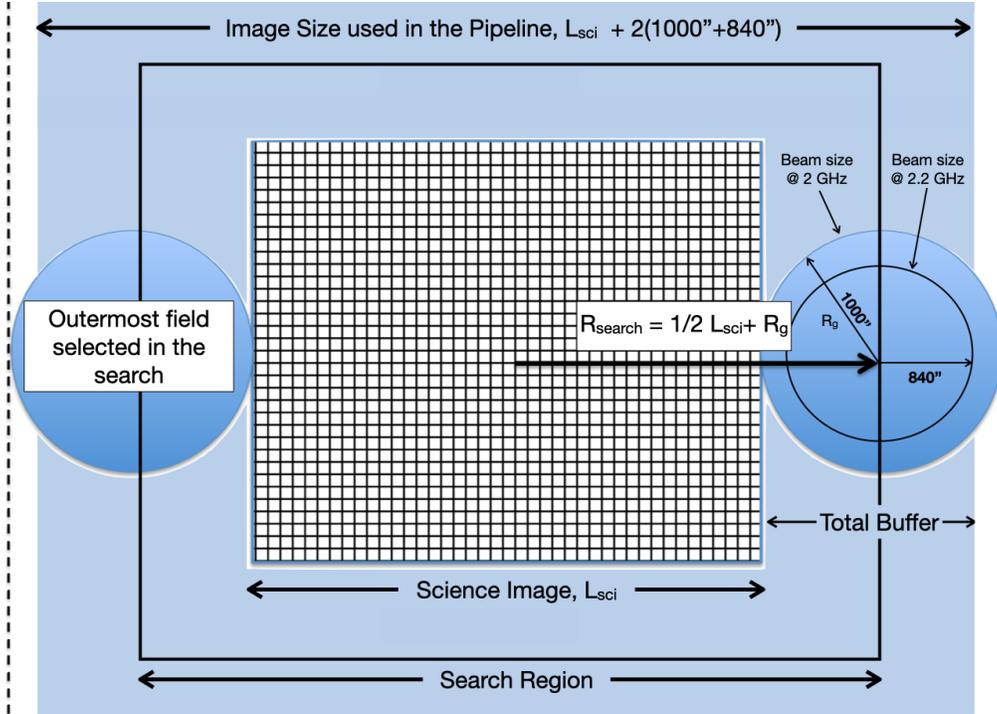
Figure 1: Illustration of the size of the science image compared to the full image size (adapted from Figure 7 of the VLASS Technical Implementation Plan). The innermost rectangle with hatched pattern is the science image with an angular size of L_{sci}. R_{g} represents the $1000''$ radius of a single field's primary beam with sensitivity $> 17\%$ of maximum at 2 GHz. The middle rectangle is the search region used in our workflow to identify all fields whose centers lie within this region. R_{search} is the search radius to identify fields that contribute to the science image. The largest blue rectangle is the full image made in the pipeline with a size of $L_{\text{sci}} + 2 \times (R_{\text{g}} + 840)$; in the example shown here, $R_{\text{g}} = 1000''$.

2.2 Identifying and Obtaining VLASS Data

Having determined the desired image size, the user must identify the specific VLASS observation(s) containing the science image. Imaging is most straightforward when the entirety of the science image has been observed in a single VLASS Scheduling Block. Unfortunately, the imaging pipeline at present is not able to combine data from multiple VLASS measurement sets, and thus cannot make an image with combined data from multiple scheduling blocks or VLASS observing epochs. Allowing the imaging pipeline to combine multiple measurement sets is an active area of development at NRAO (J. Tobin and A. Kimball, *private communication*). For science images that cross scheduling-block boundaries, we recommended for now that the user split off the required fields from each measurement set; they can either be imaged independently using the pipeline or joint-imaged manually.

There are a couple of different approaches a user could take to identify the relevant VLASS scheduling block(s). First, we note that the NRAO data archive[6] allows a search by radial position with user-input search radius. However, although this method is reliable for standard (non-VLASS-style) observations, the search mode currently does not work well for the on-the-fly (OTF) observing mode that is used by VLASS.[7] Therefore, we recommend that the user start by identifying the relevant VLASS "tile(s)" that cover their science image: VLASS observing blocks cover pre-defined survey tiles whose boundaries and

---

[6]`data.nrao.edu`

[7]Positional metadata for OTF scans in the archive currently reports only the *starting* position for the scan, rather than all positions covered by the scan. However, available search metadata will be updated in the future such that a radial search in VLASS data should work as expected (*J. Tobin, private communication)*.

names are presented in Section 2 of VLASS Memo #7[8]; we briefly summarize relevant information here. The VLASS project has divided the sky into 32 "Tiers" numbered with increasing declination; Tier 01 covers $-40° < \delta < -36°$ and Tier 32 covers $\delta > 85°$. Tier 32 and Tier 24 ($52°$–$57°$) are each $5°$ tall in declination; all other Tiers are $4°$ tall in declination. Each Tier is divided evenly into N tiles numbered from 01 to N in increasing RA, with tile 01 in each Tier having its western boundary at RA = 0 = $360°$ = 24h. The tiles are rectangular in RA/dec coordinates, and thus appear trapezoidal on the celestial sphere. The naming scheme is T16t12 for the 12th tile in Tier 16, and so on. Tiles have been further sub-divided into an integral number of images, each covering approximately one square degree: all VLASS science images are $1°$ tall in declination, and approximately $1°$ wide in Right Ascension. Images are named according to their central RA/decl. position. Users can search in the NRAO data archive for VLASS images that cover a specific target; due to the 1-deg^2 image size, we recommend performing an image search using a search radius $\gtrsim (\sqrt{2})° \sim 0.7°$. Note that not all published images are currently available in the archive: quality assurance (QA)-accepted images are uploaded to the archive sometime after the completion of each VLASS half-epoch. However, all images (including those that were QA-rejected due to, e.g., poor deconvolution) are almost immediately available on the NRAO servers.[9]

To simplify the tile and measurement set identification, author EC developed Python code (Carlson & Kimball 2022) that accepts a user-provided phase center and science image size, then reports the associated VLASS measurement sets to produce that image; the measurement sets can be downloaded from the NRAO data archive. Under the hood, this code works by calculating the extreme RA and declination positions for the input image size[10], uses those coordinates for a positional search based on the known tile boundaries, and parses the weblog of a published image from that tile to determine the contributing VLASS measurement set(s). A guide for using this script and downloading VLASS measurement sets is provided in the Appendix (§5).

Once the data are stored on disk, the user should split out the subset of fields necessary for imaging. The official VLASS imaging pipeline performs an analogous split based on metadata that are reported in the pipeline task *hif_editimlist*. What is not trivial in this step is the identification of fields to split out. To streamline this step for users, author EC modified a version of *hif_editimlist* into a Python function (see §5) that reads the measurement set metadata, takes in a science image size and phase center, and provides the user with a list of fields that lie within the Search Region[11] (see Figure 1). Those fields can then be extracted from the full measurement set using CASA's *mstransform* or *split* task. The downloaded measurement set can then be deleted from disk, saving considerable disk storage space.[12]

2.3 Generating an Image with User-Specified Input Parameters to the VLASS Pipeline

The recommended way to set parameters for the VLASS imaging pipeline is by creating *SEIP_parameter.list*— a text file in the working directory that contains one input parameter per line. There are three required parameters, and additional optional parameters may be included. The three required parameters are *image-name*, *phasecenter*, and *imaging_mode*. The most common optional parameter will be *imsize*. Other optional parameters can be included to assist with imaging if the science area of interest contains bright sources that complicate deconvolution; the most common in this case will be *cycleniter*. To see all available input

---

[8]https://library.nrao.edu/public/memos/vla/vlass/VLASS_007.pdf

[9]VLASS images are published immediately upon QA-acceptance or QA-rejection at an NRAO repository online located at https://data.nrao.edu/vlass/. The Canadian Initiatives for Radio Astronomy Data Analysis (CIRADA) project also provides published VLASS images, at https://cirada.ca/vlasscatalogueql0, having acquired them from the NRAO repository. VLASS images are also mirrored to the Canadian Astronomy Data Centre within a day or so and are available from https://www.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/en/vlass/. Finally, CIRADA provides an image cutout server at http://cutouts.cirada.ca/.

[10]The code adds an additional buffer in order to determine all data sets that contribute to the "Search Region" as shown in Figure 1; a default value of 0.17 is used for the primary beam sensitivity in determining the Search Region size but the code allows a user to select a different value if answering "Y" when asked for Advanced Options.

[11]The default Search Region assumes contribution to the science image from fields with > 0.17 primary beam sensitivity.

[12]NRAO provides a 5 TB disk space quota for each user on the Observatory's lustre computing system; VLASS calibrated measurement sets can be over 1 TB in size.

parameters, type *help(hif_editimlist)* at the CASA prompt.

The parameter file should be named 'SEIP_parameter.list', as this is the file that the pipeline will look for automatically. The *imagename* and *phasecenter* parameters will be passed directly to *tclean*, so they must match CASA's expected formats. The parameter *imagename* provides the pre-pended name for output images, and *phasecenter* provides the location on the sky of the image center. The *imaging_mode* parameter specifies the pipeline recipe; set it to "VLASS-SE-CONT-MOSAIC" for SE continuum imaging.

If optional parameters are not included, the default values used by the pipeline for SE imaging are imsize=[12500,12500] and pixel size = $0\rlap{.}''6$. Thus, the default full image size is $12500 \times 0\rlap{.}''6 \sim 2$ deg on each side. The default final image size, i.e., the science image cutout from the central section of the full image, is 1 deg^2 for VLASS. To determine the user's value of *imsize*, divide the full image size from Section 2.1 by the pixel size ($0\rlap{.}''6$/pixel).

Two parameters govern the value of the *tclean* input parameter *cycleniter* (which sets the maximum number of clean iterations or "minor cycles" per major cycle in tclean) in different stages in the SE pipeline. The reason for having two separate parameters is that the SE imaging pipeline has several calls to *tclean*, per the workflow described in VLASS Memo 15.[13] The pipeline has input parameter *cycleniter* with a default value of 500, and input parameter *cycleniter_final_image_nomask* with a default value of 100. The *cycleniter* parameter is used as the *tclean* input parameter of the same name in the first two *hif_makeimages* stages of the SE pipeline, and in the two calls to *tclean* in the third *hif_makeimages* stage to produce the "iter1" and "iter2" intermediate images. In the first *hif_makeimages* stage, *tclean* is used to create an image that sets the initial model. In the second *hif_makeimages* stage, *tclean* is used to create an intermediate image that determines the improved mask which will be used in the third and final *hif_makeimages* stage. In the latter, two intermediate cleaning stages (with the two versions of the mask) again use the pipeline input parameter "cycleniter" as the *tclean* parameter of the same name. Then, a final *tclean* call in the last *hif_makeimages* stage performs an additional step of cleaning, without any mask, using the value of pipeline input parameter *cycleniter_final_image_nomask* (default value 100) for *tclean* input parameter *cycleniter*. Therefore, the choice of whether to change the input values of *cycleniter* or *cycleniter_final_image_nomask* should depend on whether and where *tclean* divergence is seen in one or more of the stages of the pipeline. For initial imaging attempts by end users, we recommend trying the default parameters; i.e., there may be no need to include "cycleniter" or "cycleniter_final_image_nomask" in the SEIP_parameter.list file. An example SEIP_parameter.list file is shown in Figure 2.
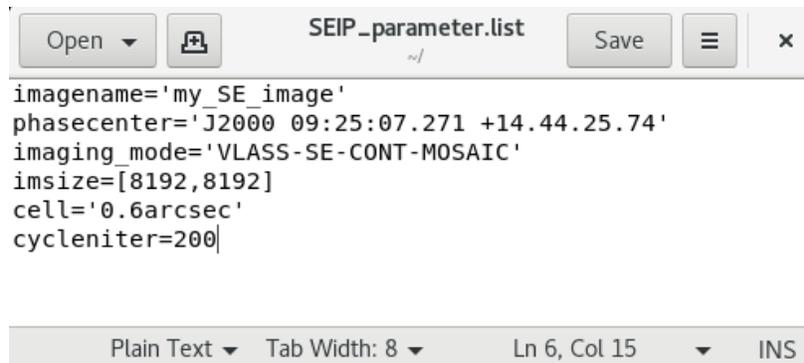

```
imagename='my_SE_image'
phasecenter='J2000 09:25:07.271 +14.44.25.74'
imaging_mode='VLASS-SE-CONT-MOSAIC'
imsize=[8192,8192]
cell='0.6arcsec'
cycleniter=200
```

Figure 2: An example input parameter file for the VLASS SE imaging pipeline. Parameter values are described in Section 2.3.

The versions of the pipeline that have been used for VLASS data processing are listed on a VLASS pipeline versions webpage.[14] We recommend using the most recent version; at the time of this memo's publication the latest version is casa-6.5.4.9-pipeline-2023.1.1.8. We recommend that users run the pipeline on the NRAO

---

[13]https://library.nrao.edu/public/memos/vla/vlass/VLASS_015.pdf
[14]https://science.nrao.edu/vlass/vlass-pipeline-versions

servers by submitting a batch job for processing, as described in the NRAO computing guide[15] and as outlined in the Appendix of this memo.

3 Data Products and Analysis

Table 1: Selected High-Powered Quasars for Imaging

J2000 Name	z	RA	Dec	Source	SDSS Class	$\log_{10}(P_{1.4\mathrm{GHz}})$	D
		(hh:mm:ss.sss)	(dd:mm:ss.ss)			W/Hz	$''$
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
J0807+0432	2.876	08:07:57.538	4:32:34.53	OJ+008	QSO	28.24	15.4
J0925+1444	0.896	09:25:07.271	14:44:25.74	OK+136	QSO	27.31	32.1

 NOTE - Column 1: J2000 Name; Column 2: Spectroscopic Redshift; Column 3: Right Ascension; Column 4: Declination; Column 5: Alternative Name; Column 6: SDSS Source Classification; Column 7: 1.4 GHz Radio Luminosity; Column 8: Distance from Standard Phase Center in arcsec

To explore the time-saving benefits of the supplied functions, we selected as imaging targets two quasars from Gobeille (2011) and Gobeille et al. (2014) due to the wide availability of pointed VLA observations and their well-known radio morphology. These sources are listed in Table 1.

Our entire processing was performed using a partition of 4 cores (Intel Xeon Gold 6136) on a single computing node on the NRAO cluster; we requested 256GB of RAM per imaging job. We used the VLASS SE imaging pipeline with varying values of the *imsize* parameter, producing three images of each source with different image sizes to compare the pipeline run times. The three science image sizes we ran were $500''$, $1000''$, and $1428''$; the additional $1840''$ buffer (see Section 2.1 and Figure 1) yielded full-image sizes approximately $4180''$, $4680''$, and $5108''$ on a side. The cell (pixel) size used by default[16] in the SE imaging pipeline is $0.6''$/pixel, meaning the science-image sizes in number of pixels on a side were 833, 1667, and 2380, with full image sizes of 6968, 7800, and 8514 pixels respectively. In Figure 3, we show the pipeline run time needed to produce the images for each target as a function of science image size. Run times were similar for the two sources. The pipeline ran for about 4.75 hours for the smallest image size; this represents a reduction in run time by a factor of 6 compared to making a $2° \times 2°$ full image with the default[17] VLASS SE imaging parameters.

We also looked at the result of shifting the phase-center for imaging each source: for a standard VLASS image (with survey pre-defined phase-center), a user's source of interest could be located anywhere within the 1 deg^2 image. For this investigation, we looked at each source when imaged at the phase-center (image center) compared to the same source in the official published VLASS image. To investigate the results, we subtracted the off-source image from the on-source image using CASA's *immath* command (after re-gridding one of the images to ensure matching pixel coordinates). We then used the image visualization software package CARTA (Comrie et al., 2021) to apply a linear color scheme. These images are shown in Figure 4.

We compared the total integrated flux density for the core and extended structure for each target in its user-defined image and official VLASS image. Flux density values were derived using PyBDSF's *process_image()*

---

[15]https://info.nrao.edu/computing/guide/cluster-processing/access-and-running-jobs

[16]If a user elects to use a different (non-default) cell size, it is important to maintain the (minimum) required $1840''$ buffer for any requested image size; the buffer is crucial to avoid aliasing as discussed in Section 2.1 and is a function of the primary beam (field of view) rather than the image or pixel size.

[17]The default SE imaging pipeline actually makes a slightly larger full image size: 12500×12500 pixels corresponding to an angular size of 2 degrees plus 5 arcminutes on a side.
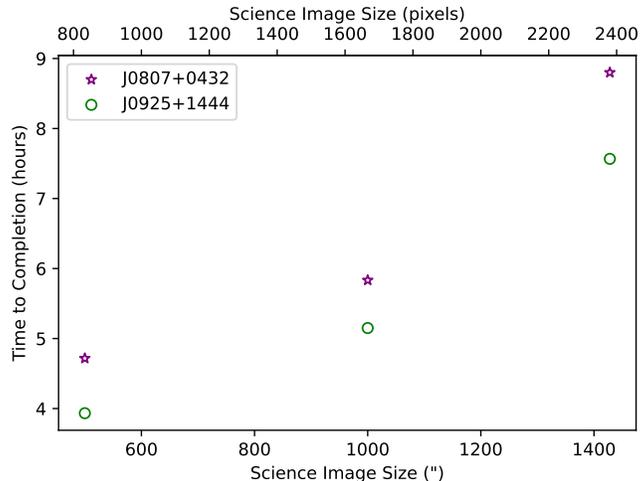
Figure 3: Production Time versus Science Image size (in pixels) for two quasars, J0807+0432 (purple points) and J0925+1444 (green points). The numbers on the x-axes represent the *one*-dimensional length of a square image; the total number of pixels in the image is equal to the square of the provided values. Production time includes the entirety of the standard VLASS SE imaging pipeline with standard parameters. Run times are heavily dependent on user-defined imaging parameters (e.g., the value of the *cycleniter* parameter used by *tclean*, which helps determine the number of major cycles).

with default inputs (Mohan & Rafferty, 2015); additionally, a manually drawn polygon region surrounding the target (identical in both its images) was used to estimate total flux density. The results are presented in Table 2, while pixel-by-pixel percent error images are presented in Figure 5. We note that the values in the table in columns 3–6 are from default runs of PyBDSF, while column 8 shows the total flux density from the manually drawn polygon regions. Tweaking of PyBDSF parameters would likely yield different results, but such testing is beyond the scope of this memo.

Table 2 demonstrates that PyBDSF catalogs made from images where a source is or is not at the phase center can vary significantly in recovered flux density (columns 3–6); the differences can be attributed to different fitting results from PyBDSF in the face of extended emission. A solid demonstration that the differently-produced images actually do provide similar results is shown in column 8, where an identical region surrounding each source yields total flux densities that are in agreement within 1%. We therefore highly recommend that flux density measurements for extended objects should be based on the use of apertures rather than component fitting with PyBDSF. In-depth characterization of and validation of SE images (and source parameters measured from them) based on various standard calibrators—which are point sources— have been presented in memo 17 of the VLASS Memo Series[18].

The results demonstrated here should not be used to make conclusions as to whether on- vs. off-source imaging yields *truer* results, for particular individual sources. Much of image fidelity hinges on the OTF observing mode used for VLASS, regardless of the phasecenter used for imaging. Imaging a target of interest from OTF observations necessarily consists of mosaicking together many individual fields containing the target; each individual OTF "pointing" will be centered at a different sky position near the source, and thus the target will lie at a different location in each pointing's primary beam (sensitivity) pattern. For fields where the target is far from the pointing center (and thus far out in the sensitivity pattern of the primary beam), there is an increase in the w-term error due to sky curvature. For that reason, users may consider whether they wish to make even smaller images (i.e., with a smaller buffer than recommended in Section 2), thus excluding fields where the target is far from the primary beam center, at the expense of increased noise.

---

[18]https://library.nrao.edu/vlass.shtml

There is, however, a different factor—dependent on imaging rather than on OTF observing—that will likely lead to improved imaging deconvolution when using an on-source phasecenter: variation of the point spread function (PSF). By default, CASA's *tclean* task determines the PSF via Fourier transform of the *uv*-coverage for the image phasecenter, and that same PSF is used for deconvolution across the entire image. For mosaic data, cleaning a source that's far from the phasecenter—i.e., observed in pointings with slightly different *uv*-coverage—leaves deconvolution errors; the errors manifest at a level that depends on cleaning depth, but are a particular fraction of the source brightness. Preliminary testing suggests that accounting for this effect can improve imaging, but that it is not the main factor in yielding good image fidelity (S. Myers, *private communication*). We recommend that users first check the official VLASS image(s) for their target-of-interest; if deconvolution errors are not evident around the source (especially if it is not close to an edge), then re-imaging with the source at phasecenter is not likely to yield substantial difference in the final image.

Even when the PSF is accounted for, small w-term errors remain present in the VLASS imaging due to the mosaicking of many pointings that cover a large (square-degree) region of sky; the mosaic gridder algorithm in *tclean* currently does not account for the w-term.[19] Thus, it is not sufficient to compare on-source to off-source phasecenter images to determine overall image fidelity. In general, pointed observations (rather than OTF) should be used as a best estimate for "true" sky emission; indeed pointed observations were used for characterizing and validating VLASS Single Epoch continuum image products in VLASS Memo # 17[20]. As demonstrated therein, flux density calibration for VLASS is expected to be accurate within $\pm 3\%$ for Single Epoch images, improving upon the survey goal of $\pm 5\%$ (and significantly better than the official requirement of $\pm 10\%$).

4   Conclusions

In summary, we have presented several use cases along with tools to produce user-defined to achieve VLASS program requirements. First, for basic use cases such as non-standard phase centers, non-standard image sizes, and the combination of non-standard phase centers and image sizes, the VLASS SE pipeline is able to successfully produce an image that can be used for end-user science. The process requires minimal user effort, with a complete process being compiled in Section 5. We explored images from two sources to ensure that changing the phase center did not drastically affect any imaging algorithms.

Official VLASS SE images should be sufficient for most science use cases, and thus the method presented here will not be needed by most users. We anticipate a few situations where our method will be particularly helpful: (1) an official SE image containing the desired target has not yet been published; (2) a user has looked at the official VLASS SE image and determined that the area around their target of interest shows deconvolution errors that affect their science analysis; (3) a target of interest lies on the edge of multiple official images or tiles, and a user wishes to joint-deconvolve the data in one *tclean*; (4) a user wishes to deconvolve the image of their target with *tclean* algorithms or parameters that differ from the default versions used by the VLASS project.

---

[19]The main issue with w-term for VLASS is that the magnitude of the offset depends on frequency, affecting the measurement of the spectral index. VLASS Memo # 17 (see next footnote) shows that this error is small for sources at low zenith distance.
[20]https://library.nrao.edu/public/memos/vla/vlass/VLASS_017.pdf

Table 2: Flux Density Measurements from User-defined ("On-Source") vs. official VLASS ("Off-Source") SE Images

Source	Image Type	S_p	S_{tot}	S_{ext}	S_{core}	σ_{rms}	S_{tot} [manual]
		mJy/bm	mJy	mJy	mJy	mJy/bm	mJy
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
J0807+0432	SE On-Source	341.25 ± 1.33	455.81 ± 5.12	$-^{\dagger}$	455.81 ± 5.12	1.12	443.5
	SE Off-Source	343.92 ± 1.16	465.61 ± 4.68	$-^{\dagger}$	465.61 ± 4.68	1.16	447.9
J0925+1444	SE On-Source	99.70 ± 0.17	339.02 ± 1.79	305.47 ± 1.79	33.55 ± 0.55	0.15	331.0
	SE Off-Source	99.82 ± 0.15	338.92 ± 1.61	305.58 ± 1.61	34.43 ± 0.58	0.17	333.0

NOTE - Column 1: J2000 source name. Column 2: Image type: "On-Source" indicates a user-defined image centered on the target of interest, whereas "Off-Source" corresponds to a standard VLASS SE image where the target happens to be far from the phasecenter. Columns 3–7 were measured with default input parameters by PyBDSF (see text): Column 3: Peak flux density; Column 4: Total integrated flux density; Column 5: Total flux density in the extended component; Column 6: Total flux density in the core component; Column 7: 1σ RMS noise level. Column 8: total flux density in a manually drawn polygon region surrounding the source (for each of the two sources, an identical region was used for both images).
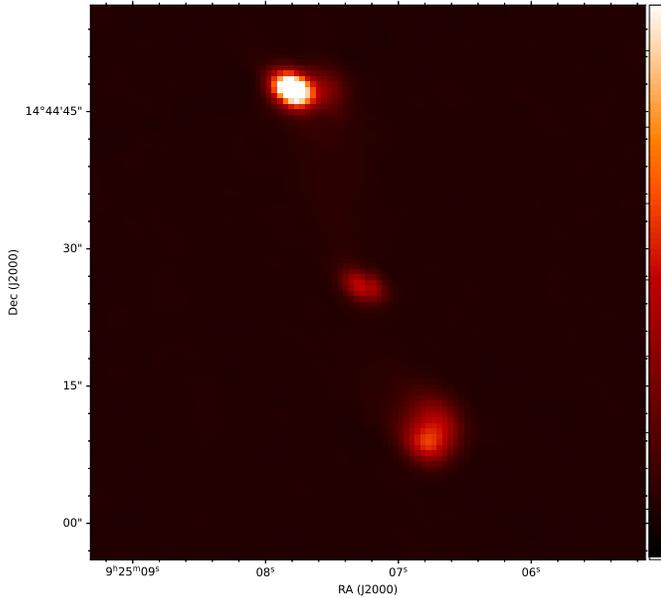
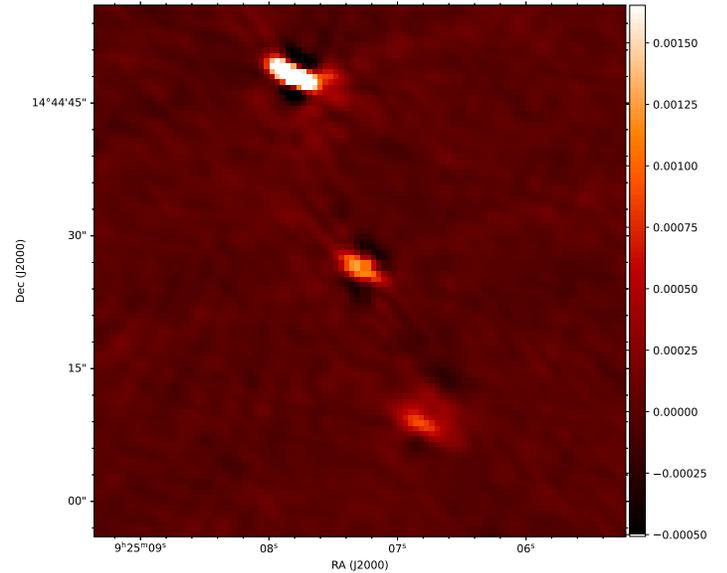$\dagger$: PyBDSF did not identify any extended structure.
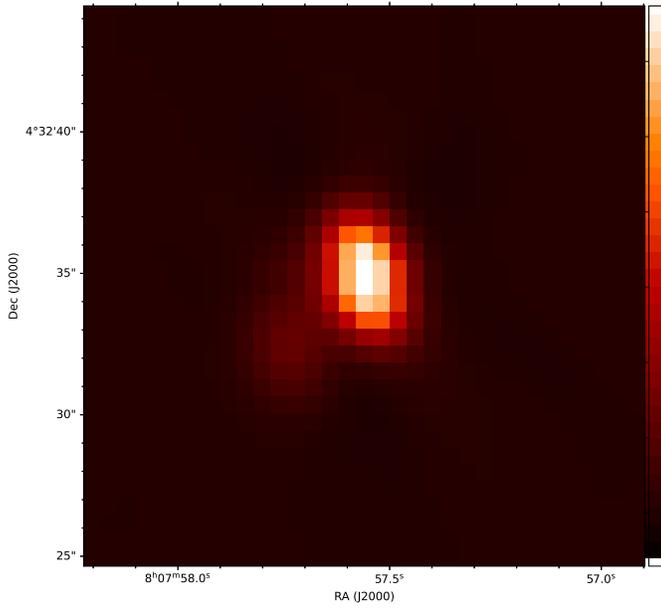
(a) J0807+0432 On-Source
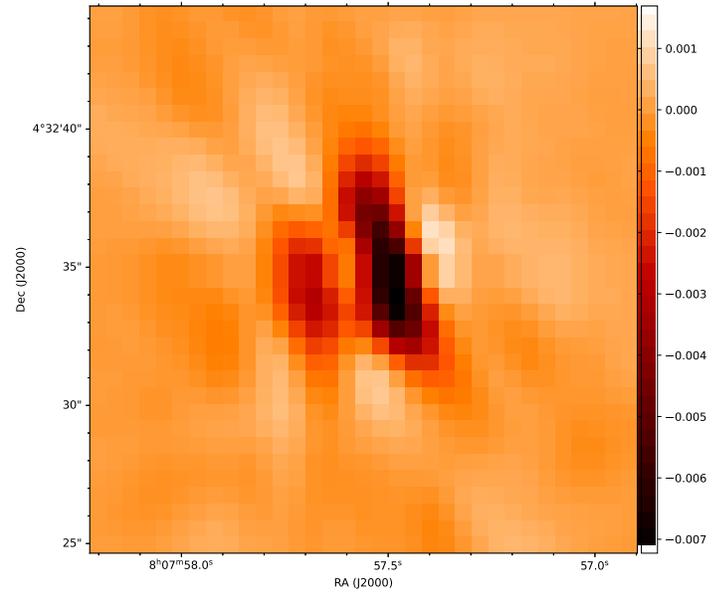
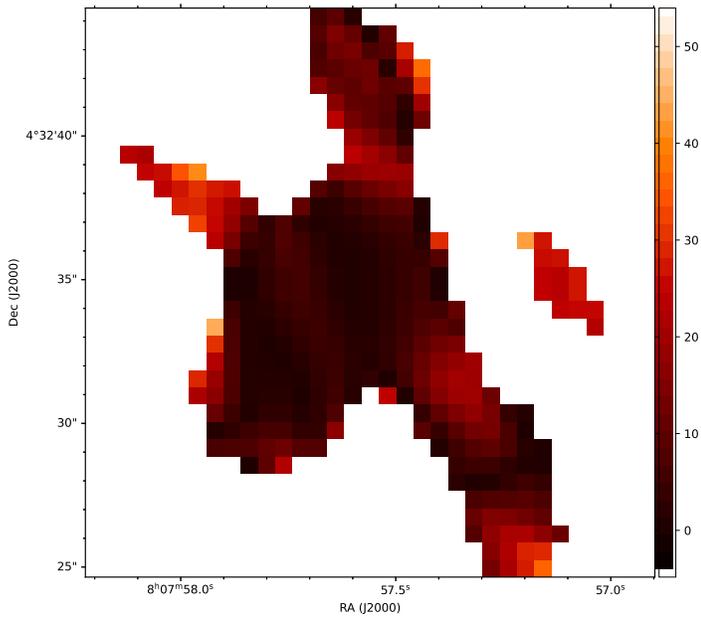(b) J0807+0432: Difference between SE On-Source and Off-Source
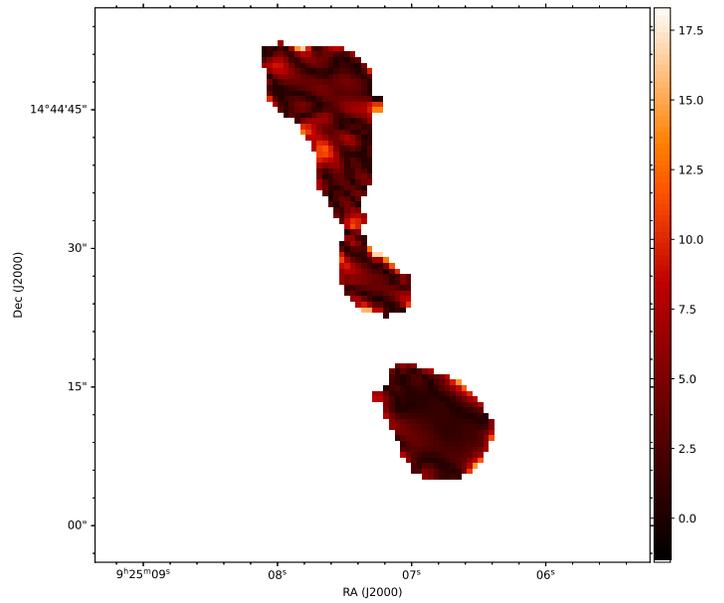
(c) J0925+1444 On-Source

(d) J0925+1444: Difference between SE On-Source and Off-Source

Figure 4: Comparison of images made with the targets of interest at the phase center ("On-Source") or away from the phase center as in the standard VLASS images ("Off-Source") as explained in Section 3. Panels a and c show the On-Source images for quasars J0807+0432 and J0925+1444 respectively; panels b and d show the difference between on-source and off-source imaging. Units for each color bar is in Jy/bm. Note that the color scaling differs across panels.

(a) J0807+0432 Percent Difference

(b) J0925+1444 Percent Difference

Figure 5: Single Epoch Percent Difference between On and Off-Source Phase Centers

5   Appendix: Guide for Generating Images

The following documentation outlines how to generate a custom image utilizing a VLASS observation of a source. For the first half of this documentation, we will look at the high-powered quasar [HB89] 0805+046, known in this guide as J0807+0432.

The first step in generating a custom image is to identify the required measurement sets for which data will be pulled from the NRAO archive. The code to identify measurement sets is located in EC's Github repository.[21] It is recommended that the user use an online iPython interface like a Jupyter Notebook or Google Collab. Although utilizing Python in a command-line setting or inside of CASA will also work, it may require changes that are beyond the scope of this guide.

Running the code results in prompts to enter Right Ascension and Declination in decimal degrees, as well as an integer value for the science image size rounded to the nearest arcsecond. Figure 6 shows an example of running the script *VLASS_Tile_Puller.py* to identify measurement sets that contain observations of a particular area of sky, including the "advanced options" to change the image buffer (§2.1).[22]

```
Please Enter your Right Ascension in Decimal Format: 121.98974
Please Enter your Declination in Decimal Format: 4.54293
Please Enter the Proposed Image Size to the nearest Arcsecond: 250
Advanced Options (Y/n): Y
What fraction of the full sensitivity of the primary beam would you like to sample: (i.e 0.17 is the standard 1000 a
rcseconds): 0.17
Your image phasecenter resides in tile: T12t13
Your image also draws data from tile(s):
VLASS3.1.sb43247986.eb43449519.59968.0862703125 - MS contains data or tile: T12t13
VLASS1.1.sb34647560.eb34700758.58075.26425702547 - MS contains data or tile: T12t13
VLASS2.1.sb38561374.eb38565040.59070.62333981482 - MS contains data or tile: T12t13
```

Figure 6: Prompts and Results Presented by the Jupyter Notebook

For the purposes of this guide, we will use the VLASS2.1.sb38561374.eb38565040.59070.62333981482 measurement set. To obtain our calibration products, we will go to `data.nrao.edu`. Simply copying and pasting the string of our measurement set, we are presented with the screen as shown in Figure 7.
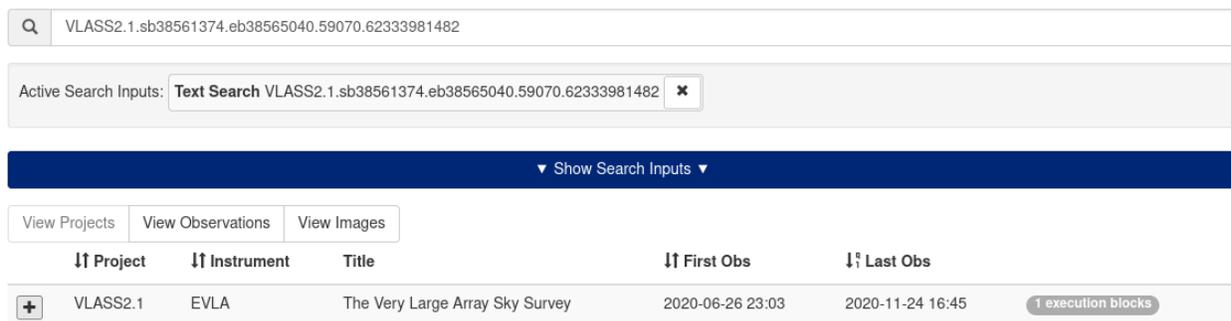

Figure 7: A screenshot taken from the `data.nrao.edu` for searching the relevant VLASS measurement set. Simple copying and pasting of the measurement set name into the archive is sufficient to list the available data.

After selecting the cross button just before the project name, click the clipboard and then the "Download" button. We are then presented with the user interface as shown in Figure 8.

---

[21]https://github.com/erikvcarlson/VLASS_Scripts/blob/main/Measurement_Set_Identification/VLASS_Tile_Puller.py

[22]In most cases, utilizing the advanced options flag and changing the sensitivity for which the "buffer" is calculated is unnecessary. However, the "Advanced Options" flag is provided for images where additional sensitivity beyond the default 17% is required (e.g., images with bright sources near the edge of the beam). See §2.1 for more details.

Figure 8: An example download GUI at the NRAO data archive.

We must then enter our email and submit the request.

The delivery request turnaround time is roughly 1–2 days for the standard size of VLASS data sets. We do *not* recommend selecting "Return results as a tar file" as this will increase the delivery request turnaround time to 3–4 days! When the data are available for download, the user will receive an email similar to the one in Figure 9.
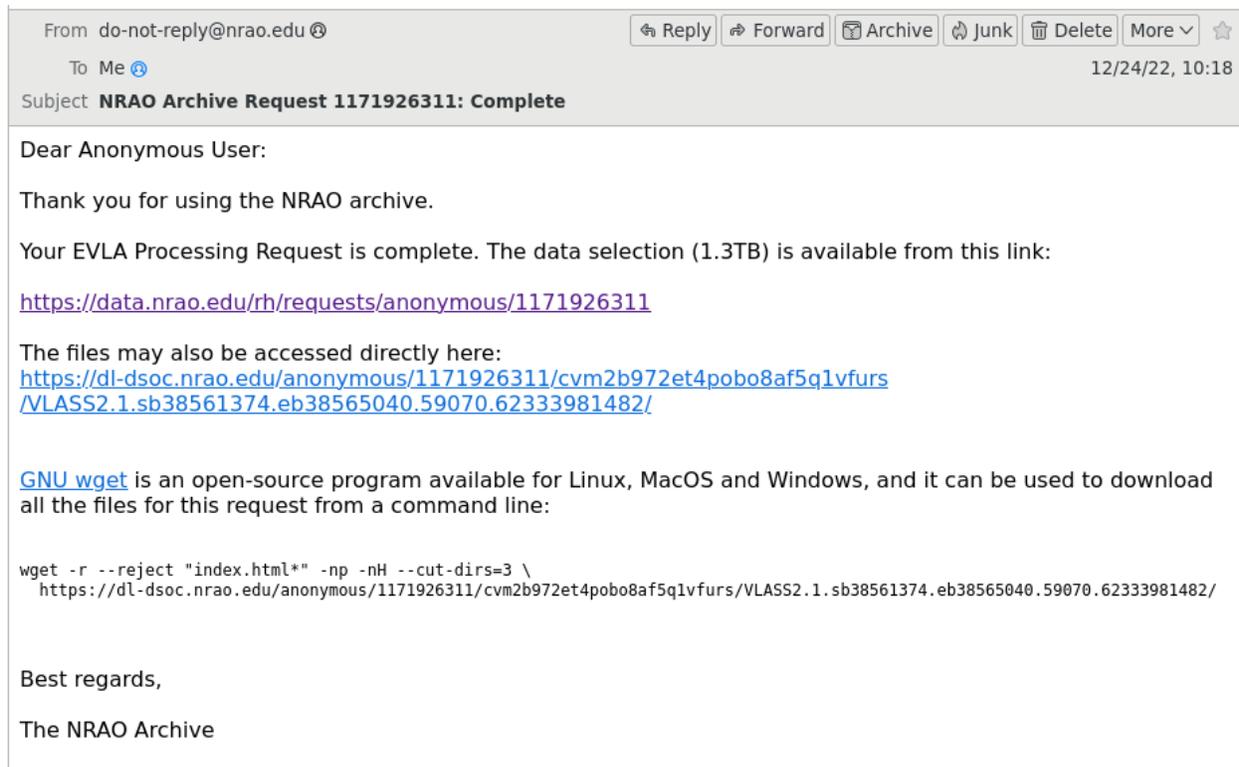
Figure 9: A sample email sent by the NRAO archive alerting the user that their data are ready for download. The email provides a *wget* command that can be used from the command line to download the data.

Logging into our NRAO compute cluster account, we used[23] the *wget* or *wget2* command (provided in the notification email from the archive) to transfer the data to a working directory. Once the data have been transferred (generally >100 minutes), we can begin to run some of our custom scripts before we begin the VLASS pipeline. The first script is used to identify what fields are required. To utilize the "carlson_editimlist_prep.py" script, located in the Github repository, the user must first open a version of CASA containing the pipeline[24]. This must be done in the working directory containing the VLASS measurement set.

Additionally, the user must have also generated a text file (named SEIP_parameter.list) that contains their list of image parameters. The text file should contain the desired image name, image center, image mode, and image size. To calculate the image size, this[25] Python script can be used. If a custom cell size is to be selected, it must be put into the image parameters file. A sample of a complete image parameters file is located here[26]. This text file must also be in the working directory.

Once these preliminary steps are complete, the user can copy and paste the carlson_editimlist_prep.py file into CASA to initialize the function. The user should then run the following command:

---

[23]It is possible to request data delivery directly to a specified local directory in one's observer account, which is faster than *wget*.

[24]The following command will run the pipeline version of CASA on the NRAO computing cluster: `casa-vla --pipeline`

[25]https://github.com/erikvcarlson/VLASS_Scripts/blob/main/Misc/image_size_value_calculator.py

[26]https://github.com/erikvcarlson/VLASS_Scripts/blob/main/Misc/SEIP_parameter.list

```
field_list = carlson_editimlist_prep('VLASS2.1.sb38561374.eb38565040.59070.
    ↪62333981482.ms', 500, 'J2000 08:07:57.5 +04.32.34.6', matchregex=['^0',␣
    ↪'^1', '^2'])
```

Figure 10

Where the first argument is the string for the measurement set, the second is an integer with the desired science image size in arcseconds and the desired phase center. The matchregex argument should be left as the default value.

Once the script is done running, the user can now run the following split command (Figure 11) on the original dataset and remove it from the disk.

```
split(vis = 'VLASS2.1.sb38561374.eb38565040.59070.62333981482.ms', outputvis =␣
    ↪'name_for_ms.ms',field = field_list)
```

Figure 11

Once the data have been properly split off, the user can then run a normal VLASS SE imaging pipeline script with the exception of custom image parameters. To complete this, the user must generate a working directory with the following structure:

```
SourceName
    working
        command_script.py
        run_SE_SourceName.sh
        image_parameter.list
    products
```

Examples of the command script, run script, and image parameter script are located in the GitHub repository.[27] To calculate the required image size, the user must first choose a cell size. The default cell size for quick-look images is 1 arcsecond, whereas the default for Single Epoch images is 0.6 arcseconds. Once the cell size has been determined, the user must then calculate an image size using the following formula:

$$P = (I + 2 \times 1840)/C \tag{1}$$

I is the science image size in arcseconds, C is the cell size in arcseconds and P is the full image size in pixels. It is this value P that is appended to the image_parameter.list file. It is important to note that *tclean* is optimized for *imsize* values that are even and factorizable by 2,3,5,7 only. A helpful tool to assist with determining the proper image size is available online.[28]

Once the directory structure, image parameter list, command script, and shell script are properly formatted, the user can now submit a non-interactive job using the methodology outlined in the link given below.[29] Upon completing the run, the user should inspect the weblog for warnings and errors. Interpreting and troubleshooting errors during the VLASS imaging pipeline should be directed to the NRAO help desk, submitted to the VLA Pipeline department.[30]

---

[27]https://github.com/erikvcarlson/VLASS_Scripts/tree/main/Run_Files

[28]https://github.com/erikvcarlson/VLASS_Scripts/blob/main/Misc/image_size_value_calculator.py

[29]https://info.nrao.edu/computing/guide/cluster-processing/access-and-running-jobs

[30]https://help.nrao.edu

References

Carlson, E., & Kimball, A. 2022, User-Defined Imaging of the Very Large Array Sky Survey, vv1.0, Zenodo, doi:10.5281/zenodo.7076243. https://doi.org/10.5281/zenodo.7076243

Comrie, A., Wang, K.-S., Hsu, S.-C., et al. 2021, CARTA: The Cube Analysis and Rendering Tool for Astronomy, v2.0.0, Zenodo, doi:10.5281/zenodo.3377984

Gobeille, D. B., Wardle, J. F. C., & Cheung, C. C. 2014, arXiv e-prints, arXiv:1406.4797

Gobeille, D. B. P. 2011, PhD thesis, Brandeis University

Kimball, A. E. 2017, VLASS Project Memo 7: VLASS Tiling and Sky Coverage,. https://library.nrao.edu/public/memos/vla/vlass/VLASS_007.pdf

Lacy, M., Patil, P., & Nyland, K. 2022, VLASS Project Memo 17: Characterization of VLASS Single Epoch Continuum Validation Products. https://library.nrao.edu/public/memos/vla/vlass/VLASS_017.pdf

Lacy, M., Myers, S. T., Chandler, C. J., et al. 2019, VLASS Project Memo 13: Pilot and Epoch 1 Quick Look Data Release,. https://library.nrao.edu/public/memos/vla/vlass/VLASS_013.pdf

Mohan, N., & Rafferty, D. 2015, PyBDSF: Python Blob Detection and Source Finder, Astrophysics Source Code Library, record ascl:1502.007, , , ascl:1502.007

Tobin, J. J., Marvil, J., Ott, J., & Bhatnagar, S. 2020, VLASS Project Memo 15: VLASS Imaging Project: Proposed Workflow. https://library.nrao.edu/public/memos/vla/vlass/VLASS_015.pdf