

The Orville Wideband Imager

Ver. 1

J. Dowell* S. Varghese† G. B. Taylor‡

September 9, 2020

Contents

1	Introduction	2
2	Hardware	2
3	Software	2
4	Data	4
5	Document History	8

*University of New Mexico. E-mail: jdowell@unm.edu

†University of New Mexico. E-mail: savin@unm.edu

‡University of New Mexico. E-mail: gbtaylor@unm.edu

1 Introduction

The purpose of this document is to describe that Orville Wideband Imager system that is deployed at the Long Wavelength Array station located on the Sevilleta National Wildlife Refuge (LWA-SV). This includes not only the data products and how they are made but also the hardware that the software runs on.

Briefly, Orville is a realtime, wideband all-sky imager for LWA-SV. It is similar to the LWA All-Sky Imager system (Obenberger et al. [4]) that uses TBN data but has several notable differences:

1. Unlike LASI, which is both a FX correlator and an imager, Orville only images. Its input data are visibilities from the ADP COR mode.
2. Orville creates all-sky images over 19.8 MHz of bandwidth in 198, 100 kHz wide channels with full linear polarization products. This almost a factor of 200 more than what the LASI produces.
3. Orville uses a custom w -stacking imager built in the Bifrost framework¹ instead of CASA [3] which is used by LASI.

2 Hardware

The Orville Wideband Imager runs on a 4U Silicon Mechanics Rackform R2504.v7 platform. The machine has:

- Two 8-core Intel Xeon Silver 4208 processors with a base clock of 2.10GHz,
- 96 GB of 2666 MHz DDR4 memory,
- Two Nvidia Titan RTX GPUs with 24 GB of memory each,
- A 46 TB RAID6 array,
- A 480 GB Intel Optane 900P NVMe drive, and
- A Mellanox ConnectX-3 10GbE fiber card.

The Orville software is bound to one of the CPUs and one of the GPUs, and writes the output images to the RAID6 array. The other resources on the machine are used by the Wilbur Single Dispersed Pulse system which is described in LWA Memo 216 [2].

3 Software

The Orville imager is written in Python using the Bifrost framework (Cranmer et al. [1]) and the source code is available on GitHub at https://github.com/lwa-project/orville_wideband_imager. Figure 1 shows the Bifrost blocks used for Orville. At the top of the diagram are the 197,376 data sources. These represent the 32,896 baselines and autocorrelations that are generated by the six DRX pipelines running on the ADP backend at LWA-SV. These data are complex visibilities with Stokes I, Q, U, and V polarization products stored as 32+32-bit complex floating point integers. Each baseline/autocorrelation is transmitted by ADP in its own packet and the input data rate for Orville is about 45 MB/s for 5 s integrations.

¹<https://github.com/jaycedowell/bifrost/tree/wideband-imager/src>

Pipeline: orville_imager.py

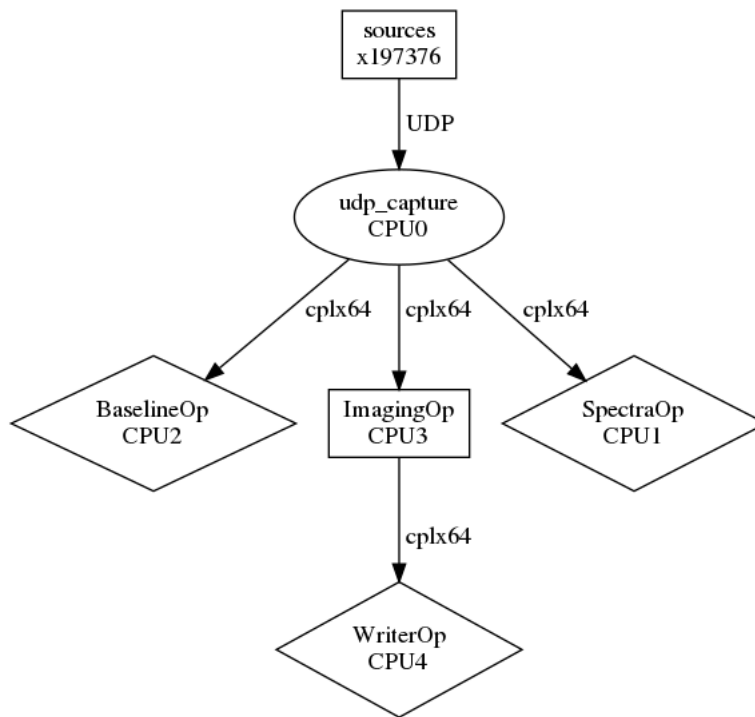


Figure 1: Bifrost block diagram for the Orville Wideband Imager.

Once the data have been received and reordered by the `udp_capture` block, there are three blocks that make use of the data. The first is the `BaselineOp` block. This block takes the visibility data and generates (u, v) radial plots for each of the six sub-bands sent by the ADP servers. These plots server a diagnostic for RFI and ADP server health. An example plot is shown in Figure 2. Similarly, the `SpectraOp` block produces plots of the autocorrelation spectra for all 256 stands for the XX and YY polarizations (Figure 3).

The final block that reads data from the `udp_capture` block is the `ImagingOp` block. This block is responsible for take the visibility data, applying the station delay calibration information, and generating the images. The processing flow inside this block is:

1. At the start of each unique frequency setup:
 - (a) Check the disk cache for (u, v, w) coordinates for the baselines and cable delay information. If these values are not in the cache, compute them. To minimize the w term when imaging, Orville uses a phase center for the images that is slightly off zenith. The phase center is located at an hour angle of $-0^h07^m59.82^s$ and a declination of $+33^\circ21'27.5''$ (90.97° azimuth, 88.07° elevation).
 - (b) Create the output image grid where the grid size, s , and resolution, r , are set by the high frequency. For the shortest wavelength, λ_{min} , the grid size is:

$$s = \max \left(128 \text{ px}, \left\lceil \frac{743.9 \text{ px/m}}{\lambda_{min}} \right\rceil \right), \quad (1)$$

and the resolution is:

$$r = \frac{130^\circ}{s}. \quad (2)$$

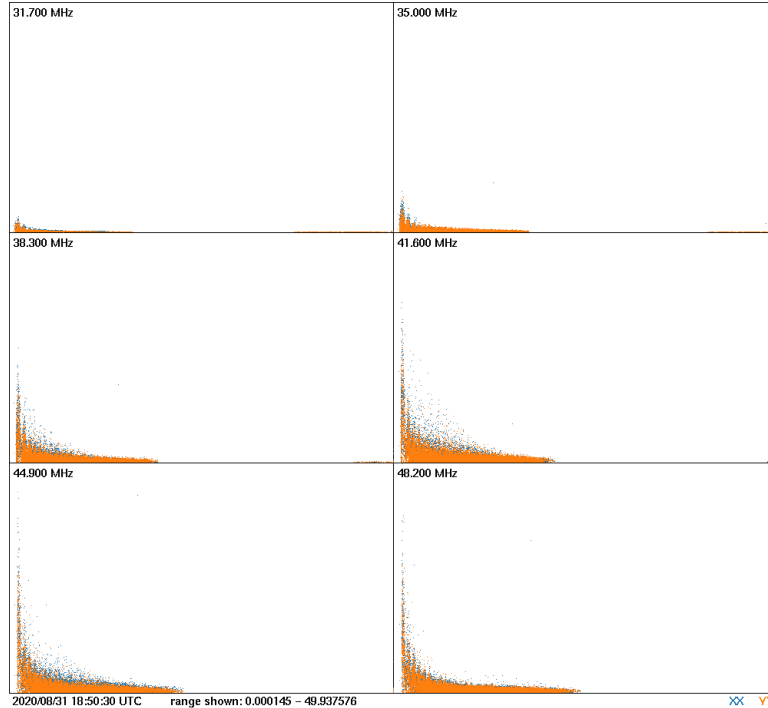


Figure 2: Example (u, v) radial plots from the Orville Wideband Imager showing the visibility amplitude as a function of (u, v) distance for all baselines. The frequency for each of the six sub-bands is label in the upper left of each box and the points are color coded by polarization.

The grid is four dimensional with dimensions of channels by polarizations by s by s .

2. Re-order the visibility data, apply the cable delay compensation via phase rotators.
3. Convert from XX, XY, YX, and YY to Stokes I, Q, U, and V visibilities and fill in the conjugate baselines.
4. Grid the data in the (u, v) plane using a w -stacking algorithm [5] with a w step of 0.1λ and then invert the image plane. A Kaiser-Bessel function is used as the gridding kernel similar to what is done with `wsclean` [5].

After the data are imaged they are sent to the `WriterOp` to be saved to disk in two different forms: a full frequency and polarization image cube and an archival image cube with six 3.3 MHz channels and full polarization information². The `WriterOp` also creates an “LWA TV” style image from a single 100 kHz channel from the center of the band (Figure 4).

4 Data

4.1 File Format

Both of these data products generated by the `WriterOp` are stored in a little endian binary format called the Orville Image Format (OIMS). This is a frame-based format that has a file header, described in Table 1 and one or more data frames. Each data frame has its own frame-specific header (Table 2) and the images are packed within the frame as 32-bit floating point numbers.

²<https://lda10g.alliance.unm.edu/Orville/>

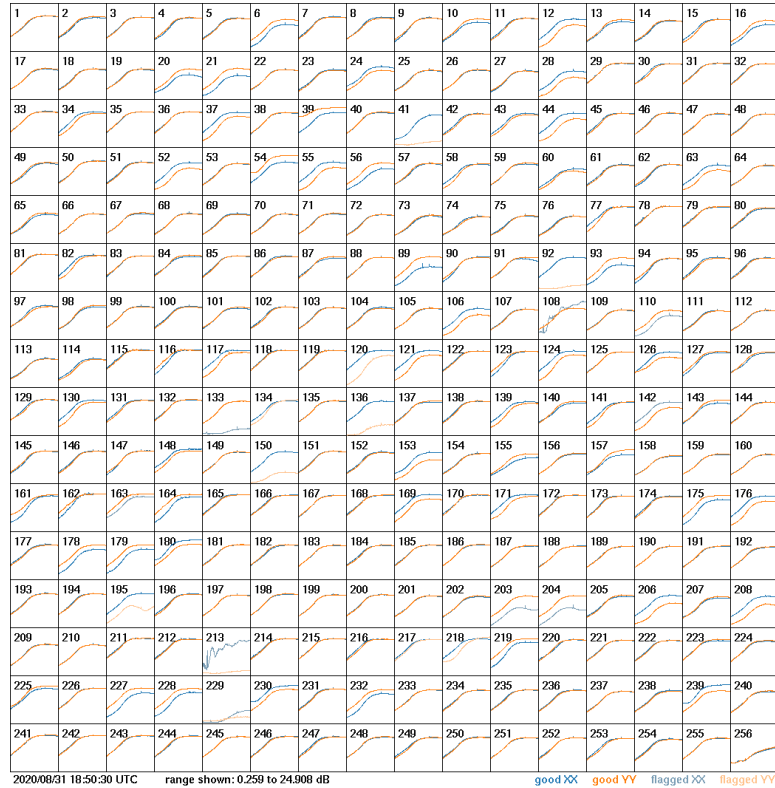


Figure 3: Example autocorrelation spectra plots from the Orville Wideband Imager for all 256 stands. The stand number is labeled in the upper left corner of each box and the spectra are color coded by polarization.

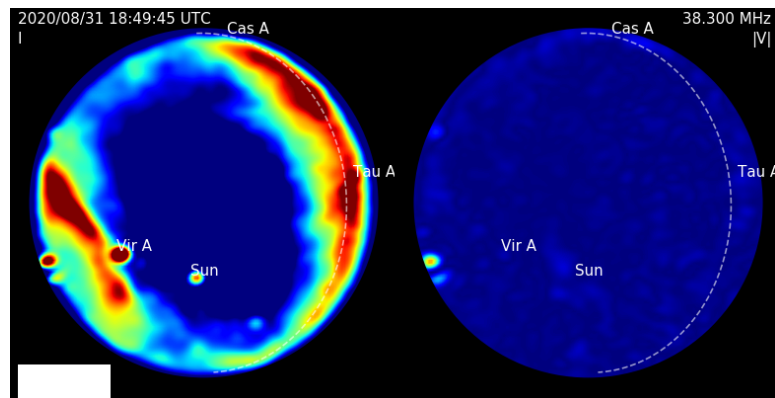


Figure 4: Example “LWA TV”-style image generated by the Orville Wideband Imager. The left panel shows the Stokes I sky while the right panel shows the absolute value of Stokes V. The time and frequency for the image are labelled in the upper left and right-hand corners, respectively.

Keyword	Description	Units
imager_version	Version of the imager	
station	Station name	
stokes_params	Comma-separated list of Stokes parameters	
ngrid	Image Size	pixels
pixel_size	Pixel size at the center of the image	degrees
nchan	Number of channels per image	
flags	File and/or image flags	
start_time	UTC time for first frame in the file	MJD
stop_time	UTC time for the last frame in the file	MJD

Table 1: OIMS file header

Keyword	Description	Units
sync_word	Frame header marker	0xC0DECAFE
start_time	UTC time at the start of the frame	MJD
int_len	Integration time for the frame	days
fill	Fractional packet fill	
lst	Mean local sidereal time of the frame	days
start_freq	Frequency of the first channel	Hz
stop_freq	Frequency of the last channel	Hz
bandwidth	Total bandwidth of the images	Hz
center_ra	RA of the phase center	degrees
center_dec	Declination of the phase center	degrees
center_az	Azimuth of the phase center	degrees
center_alt	Altitude of the phase center	degrees

Table 2: OIMS frame header

4.2 Image Format

The images are in orthographic sine projection with an image resolution given in the *pixel_size* keyword of the file header. The phase center is at the center of each image and its coordinates are given by the *center_ra* and *center_dec* values stored in the associated frame header. Each image is a four dimensional image cube with dimensions *nchan* by the number of polarization products indicated in *stokes_params* by *ngrid* by *ngrid*.

4.3 Calibration

In principle the Orville images can be calibrated in the same way as LASI image . However, scintillation complicates the procedure since it introduces frequency-dependent time variability for the observations of calibration sources. Due to this variability care must be taking to select data for calibration during quiet times when the ionosphere is more calm.

In order to determine scintillation quiet times, we define a quantity known as scintillation index. The bandpass response of a single dipole antenna changes as a function of elevation. The shape of the bandpass response remains almost stationary as the source transit 1 degree in elevation (during ideal conditions). But during times of high scintillation, the bandpass response changes very rapidly across the bandwidth with lot of frequency structures.

For calculating the scintillation index, we need to collect the bandpass response of the calibrator as the source transits one degree in elevation. Then calculate the median of the bandpass responses and divide each of the responses with the median value. Ideally this should give a flat spectrum for

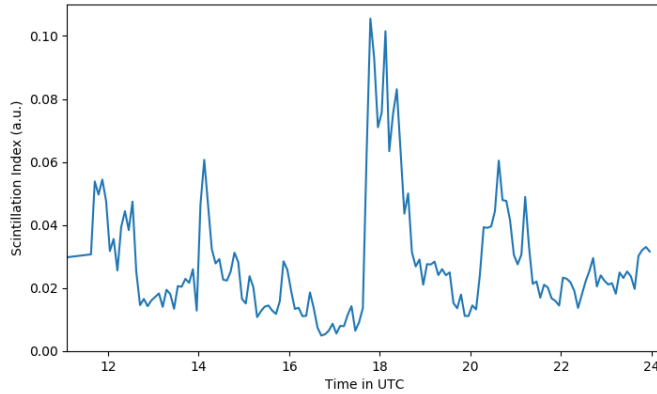


Figure 5: Scintillation index for MJD 58888 (2020 Feb 9) as measured in the direction of Cyg A. Taken from Varghese et al. [6].

all responses and the standard deviation of each divided response should be zero. But in reality even during scintillation low times, bandpass response changes slowly and the standard deviation of the divided response will not be zero. We define the mean of standard deviation of the median divided responses to be the scintillation index. The spectrum of Cas A was calibrated at different times in a day using Cyg A. A good calibration of Cas A is obtained only when the scintillation index is less than 0.05 (or the shape of the bandpass should not change more than 5% from the median response). Ideally if there is no variability in the data due to ionosphere, the scintillation index should be zero. We define scintillation quiet times when the index is less than 0.05 and scintillation high when the index is higher than 0.05. The scintillation index measured on MJD 58888 with Cyg A is shown in Figure 5 The figure shows UTC times when the scintillation is high and low. Therefore, in order to carry out accurate calibrations, finding scintillation quiet times is necessary.

5 Document History

- Version 1 (Sep 9, 2020): First version.

References

- [1] M. Cranmer et al., “Bifrost: a Python/C++ Framework for High-Throughput Stream Processing in Astronomy”, 2017, JAI, 650007.
- [2] J. Dowell & G. B. Taylor, “The Wilbur Single Dispersed Pulse Search System,” Ver. 1, Long Wavelength Array Memo 216, Aug 31, 2020. [online] <http://www.phys.unm.edu/~lwa/memos/index.html>.
- [3] J. McMullin et al., “CASA Architecture and Applications”, 2007, Astronomical Data Analysis Software and Systems XVI (ASP Conf. Ser. 376), ed. R. A. Shaw, F. Hill, & D. J. Bell (San Francisco, CA: ASP), 127.
- [4] K. Obenberger, et al., “Monitoring the Sky with the Prototype All-Sky Imager on the LWA1”, 2015, JAI, 4, 450004.
- [5] A. Offringa, et al., “WSCLEAN: an Implementation of a Fast, Generic Wide-Field Imager for Radio Astronomy” 2014, MNRAS, 444, 606.
- [6] S. Varghese et al., “Broadband Imaging of Meteor Radio Afterglows” 2020, *in prep.*