JUDE: An Ultraviolet Imaging Telescope Pipeline

Jayant Murthy[a], P. T. Rahna[b], Firoza Sutaria[a], Margarita Safonova[c,*], S. B. Gudennavar[b], S. G. Bubbly[b]

[a]*Indian Institute of Astrophysics, Bengaluru*
[b]*Department of Physics, Christ University, Bengaluru*
[c]*M. P. Birla Institute of Fundamental Research, Bengaluru*

Abstract

The Ultraviolet Imaging Telescope (UVIT) was launched as part of the multi-wavelength Indian *AstroSat* mission on 28 September, 2015 into a low Earth orbit. A 6-month performance verification (PV) phase ended in March 2016, and the instrument is now in the general observing phase. UVIT operates in three channels: visible, near-ultraviolet (NUV) and far-ultraviolet (FUV), each with a choice of broad and narrow band filters, and has NUV and FUV gratings for low-resolution spectroscopy. We have written a software package (*JUDE*) to convert the Level 1 data from UVIT into scientifically useful photon lists and images. The routines are written in the GNU Data Language (GDL) and are compatible with the IDL software package. We use these programs in our own scientific work, and will continue to update the programs as we gain better understanding of the UVIT instrument and its performance. We have released *JUDE* under an Apache License.

Keywords: ultraviolet, UVIT, astronomical software, data analysis

1. Introduction

The Ultraviolet Imaging Telescope (UVIT) was first proposed as part of the multi-wavelength *AstroSat* mission in the late 20th century (Pati and Rao, 1998; Pati, 1999) and was launched on 28 September, 2015 into a low Earth orbit (650 km, 6° inclination) by an ISRO (Indian Space Research Organization) PSLV (Polar Satellite Launch Vehicle) launcher. The UVIT payload doors were opened on 30 November, 2015 and a 6 month performance verification (PV) phase began, ending in March 2016. The ground calibration of the payload has been discussed by Postma et al. (2011) with in-flight tests by Subramaniam et al. (2016).

There are now a complete set of PV phase observations which have been used to characterize the instrument (Tandon et al., 2016) with in-flight calibration and verification done by Tandon et al. (2017) and Rahna et al. (2017), and data are being released to the observers. However, there are still (at the date of writing) issues with the UVIT pipeline and there are data sets which cannot be processed. More importantly from our viewpoint is that the source code is proprietary and difficult to modify.

We have chosen to create an alternative set of routines (*JUDE*: Jayant's UVIT Data Explorer) released under an Apache License[1] which may be freely used and modified. *JUDE* begins with the Level 1 data provided by the Indian Space Science Data Centre (ISSDC) and produces photon lists and images suitable for scientific analysis.

It should be stressed that this package is not intended to be a replacement for the official UVIT pipeline but is rather a tool to examine the data in more detail. However, the authors are using *JUDE* for their own scientific purposes, and are actively maintaining and modifying the software as the instrument characterization improves. *JUDE* is archived at the Astrophysics Source Code Library (Murthy et al., 2016) and the latest version is available on GitHub (https://github.com/jaymurthy/JUDE). We solicit feedback on its operation and further improvements.

2. Instrumentation and Observations

The UVIT instrumentation has been described in Tandon et al. (2016), with a description of the entire *AstroSat* mission by Singh et al. (2014). The

---

*Corresponding author
Email address: margarita.safonova62@iiap.res.in (Margarita Safonova)

[1]*http://www.apache.org/licenses/LICENSE-2.0*

UVIT instrument consists of two 35-cm Ritchey-Chrétien telescopes with three intensified CMOS detectors. One telescope feeds a far ultraviolet (FUV) detector (1300 – 1800 Å), with the other feeding two detectors in the near ultraviolet (NUV: 1800 – 3000 Å) and visible (VIS: 3200 – 5500 Å), respectively, through a dichroic filter. A filter wheel in front of each detector allows different spectral ranges to be selected. The visible channel is not intended for science purposes and is used solely to track the spacecraft motion. Further information about the *AstroSat* mission and the instruments aboard may be obtained from the *AstroSat* Science Support Cell (http://astrosat-ssc.iucaa.in).

The UVIT instrument has observed a number of targets during the PV phase, where an observation may be broken into several exposures of several hundred seconds in length. UVIT has now begun regular observations and is shifting to an observer driven mission with proposals from the Indian and international astronomical community. The official UVIT pipeline is tasked with producing scientifically usable images, including flat-fielding, distortion correction, drift correction, and absolute calibration from the raw data, and is available from the Science Support Cell referenced above.

3. *JUDE*

3.1. Overview

We have written the *JUDE* software system entirely in the GNU Data Language (Coulais et al., 2010). GDL was chosen for the development environment because it is an interpreted language which lends itself to the interactive analysis of data. This was invaluable in the development of the pipeline where we could check each step and run commands interactively. GDL is an open source version of the Interactive Data Language[2] (IDL) and will run on a wide variety of systems, as will IDL. It will run most IDL programs without modification (and *vice versa*) allowing access to the rich library of utilities developed for IDL over the last four decades, easing the development of *JUDE*. We have tested *JUDE* using both GDL (version 0.9.6) and IDL with identical results on multiple operating systems. In the remainder of the paper, GDL and IDL may be freely interchanged.

The two UV channels typically operate in photon counting mode with the VIS camera operating in integrating mode. The primary science from UVIT will come from the two UV imagers with the

VIS camera intended solely to correct for spacecraft motion. We have therefore focussed on producing scientific products from the two UV channels.

Our starting point is the Level 1 data provided by the ISSDC from which we produce photon lists and images of the sky which may be used for further scientific analysis (Level 2 data). We have tested the *JUDE* system by reducing the entire set of PV observations and are now using it to analyse our GT (Guaranteed Time) observations. In the following sections, we will first describe the Level 1 and 2 data files, the operation of the pipeline and the logic behind the primary components of *JUDE*. The individual modules in *JUDE* are listed in Table 1. Each module is self-documenting with a prologue containing a description of the inputs and outputs. We use a number of routines from the IDL Astronomy Library (Landsman, 1995) without modification (Appendix A: Table A.4) and the MPFIT routines (Markwardt, 2009) for PSF fitting.

3.2. Level 1 Input Files

Data from the *AstroSat* spacecraft are transmitted to the *AstroSat* Data Centre at the ISSDC where they are separated by instrument and written into Level 1 data files. They are then sent to the Payload Operations Centre (POC) at the Indian Institute of Astrophysics (IIA) where they will be processed by the official UVIT pipeline into Level 2 data products, suitable for scientific analysis. After validation, the Level 2 products will be sent back to the ISSDC which has the primary responsibility for data archival and dissemination.

The Level 1 data are distributed as a single zipped archive (earlier tar.gz) for each observation. The files within the archive are in a number of subdirectories organized by orbit number and type of file, all under a single top level directory. All data files are FITS binary tables (Pence et al., 2010) with individual frames tagged by mission time and containing, depending on the file, either housekeeping or science data. Due to the manner in which the data are read and transmitted to the ground, an archive may consist of multiple housekeeping and science files, possibly with duplicated frames and/or broken across multiple files.

We extracted all the files from the archive using unzip (or tar -xf) into a single top-level directory, which we use four sets of files (irrespective of subdirectory):

1. Level 1 data files for VIS ("uvtV"): Used to correct for spacecraft motion. Not required for pipeline operation.

Table 1: JUDE modules

Module	Purpose
Accessory Programs	
jude_get_files	Returns names of data files.
jude_params	Sets operating parameters.
jude_err_process	Error handler.
Level 1 Data	
jude_read_vis	Reads the visible data.
jude_create_uvit_hdr	Creates FITS data header for Level 2 data files.
jude_read_hk_files	Reads housekeeping files.
jude_set_dqi	Checks instrumental parameters.
jude_get_xy	Extracts individual events from Level 1 data.
jude_cnvt_att_xy	Calculates X and Y shifts from boresight.
jude_check_bod	Check and reject bright object detection (BOD).
Level 2 Data	
jude_register_data	Corrects for spacecraft motion.
jude_add_frames	Combines individual frames into image.
jude_vis_shifts	Calculates spacecraft motion from visible data.
jude_add_vis	Adds visible frames together.
jude_obs_log	Creates observation log from Level 2 files.
jude_merge_files	Combines files with overlapping data.
jude_match_vis_offsets	Matches the visible offsets to the UV channels.
jude_apply_cal	Applies photometric calibration to data
jude_centroid	Tracks spacecraft motion through centroids of stars.
Driver Routines	
jude_driver_vis	Chains individual programs to produce Level 2 files in the VIS.
jude_driver_uv	Chains individual programs to produce Level 2 files in the UV.
jude_uv_cleanup	Post processing of Level 2 data.
jude_interactive	Interactive exploration of Level 2 data.

2. Level 1 data files for either FUV ("uvtF") or NUV ("uvtN"): At lease one file must be present.

3. Housekeeping files ("*.lbt"): Frames with no housekeeping information are rejected.

4. Attitude files of spacecraft boresight("*.att"): Required for a preliminary estimate of pointing.

3.3. Running JUDE

3.3.1. Pipeline Operation

We have written *JUDE* as a set of procedures which must be run from within the GDL (or IDL) language. Each procedure is designed to be run independently but will operate as a pipeline if chained together. We have written a shell program (Appendix B) which we have used to process the entire set of PV observations without active monitoring. These files may be further processed by the user to co-add different exposures of a single target, extract point sources, and for other scientific purposes.

The individual *JUDE* procedures are listed in Table 1 and their operation is controlled by a GDL data structure (Appendix C) where the structure is defined in, and defaults are set by, *jude_params.pro*. The user may change the parameters on a global level by editing the parameter file or on the fly through the command line.

3.4. VIS Processing

Observations in the VIS channel are taken in integration mode where the 512×512-pixel CMOS chip is read once per second and the pixels stored in a binary FITS table where the data from each frame are in 261 rows of 1008 pixels each, first with increasing x $(0 - 511)$ and then increasing y $(0 - 511)$. The final 944 $(261 \times 1008 - 512 \times 512)$ elements are left blank and should be skipped. We used *jude_read_vis.pro* to read each VIS frame (Fig. 1) and extracted the x and y positions of the stars in the field with the library routine *find.pro* (based on DAOPHOT: Stetson (1987)). The spacecraft motion is just the shifts between successive frames which we calculated and wrote into a text file using *jude_vis_shifts.pro*.

The NUV and VIS channels share a telescope with the NUV light reflected, and the VIS light transmitted, through a dichroic and the shifts for

Figure 1: A 1 second frame of Holmberg II with the VIS imager.

Table 2: DQI values

DQI	Description	Module
0	Data valid.	
1	BOD	*jude_check_bod*
2	Unused.	
4	Unused.	
8	No HK data.	*jude_set_dqi*
16	Duplicate frames.	*jude_get_xy*
32	Filter closed/invalid.	*jude_set_dqi*
64	Voltage out of bounds.	*jude_set_dqi*
128	> 1000 counts.	*jude_get_xy*
1024	Parity violation.	*jude_get_xy*

the NUV image are mirrored around the x-axis

$$x_{\mathrm{NUV}} = x_{\mathrm{VIS}},$$
$$y_{\mathrm{NUV}} = -y_{\mathrm{VIS}}. \qquad (1)$$

The FUV channel has its own telescope, and we found (empirically) that it is rotated from the VIS by $35°$,

$$x_{\mathrm{FUV}} = x_{\mathrm{VIS}} \cos\left(35°\right) - y_{\mathrm{VIS}} \sin\left(35°\right),$$
$$y_{\mathrm{FUV}} = x_{\mathrm{VIS}} \sin\left(35°\right) + y_{\mathrm{VIS}} \cos\left(35°\right). \qquad (2)$$

We use *jude_match_vis_offsets.pro* to store the offsets into the UV channels, using the spacecraft time as a reference. We have shown the original image with the spacecraft motion and the corrected image in Fig. 2. In practice, we find, as we will show below, that we can obtain better resolution through self-registration, *i.e.* using stars in the NUV (FUV) channel to correct for the motion in the NUV (FUV) channel.

3.5. Housekeeping and Attitude

The housekeeping files (*.lbt) contain a wealth of information about the health and the environmental conditions of the UVIT instrument, most of which are not relevant to our scientific analysis. The attitude files (*.att) contain the spacecraft boresight position as determined by the star sensor approximately every 16 seconds. We use *jude_read_hk.pro* to read the housekeeping and attitude files and *jude_set_dqi.pro* to match the housekeeping and attitude information with the sensor data based on the mission time. We perform basic sanity checks as part of our Level 1

processing and set the DQI (data quality index) as in Table 2. Note that the DQI values are powers of two and additive; it is trivial to diagnose which data were rejected and why.

A bright object detection (BOD) sequence was run at the beginning of every observing sequence in which the gain was ramped up to the operating value. A few frames were sampled at each gain to ensure that there was no potential for damage to the instrument. Not all data files contain the BOD due to the manner in which the data were transmitted to the ground but we checked every observation and flagged those that were part of the BOD sequence.

3.6. Processing UV Data

The UVIT detectors are intensified 512×512 CMOS detectors (Tandon et al., 2016) read 29 times a second with an exposure time of 0.035 seconds per frame. Individual photons fall on the photocathode where they eject electrons that are accelerated through a microchannel plate (MCP) onto a phosphor screen. The resulting flashes of green light, each of which represents a single photon hit, are recorded on the CMOS sensor. Onboard software converts these flashes into individual photon events with a nominal resolution of $\frac{1}{8}$ of a pixel and packs each event into a byte array which is stored on-board and ultimately transmitted to the ground. The ISSDC separates the data by instrument and channel and passes them on to the user as Level 1 data files.

There is another mode in which a small (100×100 pixel) window is read at 600 frames per second as well as a transmission grating which produces low resolution spectra. The processing scheme is the same for those modes and we do produce usable data but have not tested the data products for scientific integrity.

Each event in the Level 1 data is stored as 6 bytes with two bytes for each of the x and y

Figure 2: Holmberg II without correction for spacecraft motion on the left and after correction with the VIS offsets on the right.

Figure 3: Normalised histogram of the count rate per frame in the FUV (solid line) and NUV (dashed line).

positions and the third for diagnostic information about that event (not used in *JUDE*). One frame contains a maximum of 336 events with further events resulting in a new frame. We read each frame and extracted the photon events using *jude_get_xy.pro*. The CMOS sensors are subject to cosmic ray hits giving rise to Cherenkov radiation within the MCP. This blue light results in a huge splash on the detector which is interpreted as a large number of individual photon hits by the on-board centroiding algorithm, all spurious. The actual count rate in one of our observations is plotted in Fig. 3 where the cosmic ray hits are represented by the long tail at higher count rates. The effect of including the spurious counts is to increase the overall background level in the image and we rec-

ommend that they be discarded. The default in *JUDE* is to define the threshold as $m + 3\sigma$ where m is the median of the counts per frame over the entire observation and $\sigma = \sqrt{m}$ for the photon counting UVIT detectors. If desired, the threshold may be changed using the max_counts parameter (Table C.5). Note that we record no more than 1000 events per frame, a most unlikely level for any observation to reach.

Our final step in the production of Level 2 data was to remove duplicate frames and merge files which were broken in the middle of an exposure. This creates a number of FITS binary tables (Table 3 which contain the time-tagged photon lists with selected housekeeping data. These can be used for further scientific processing without returning to the Level 1 data and the housekeeping files, saving considerable time. One of the most time-consuming aspects of *JUDE* is matching the event lists with the housekeeping data.

3.7. Level 2 Output Files

Our primary Level 2 data product is a UV photon list for every exposure (a contiguous set of UV data). This list is time-tagged and stored in a FITS binary table (Table 3). Each row of the FITS binary table represents a single 0.035 second frame identified by the time in seconds from Jan. 1, 2010. The x and y position of every photon in a frame are stored in two arrays ("X" and "Y") and the spacecraft motion is stored in "XOFF" and "YOFF". The output files from *JUDE* are described in Appendix D, including the VIS output and the UV photon lists and images.

5

Table 3: Level 2 Data

	Variable	Data Type	Sample Value
1	FRAMENO	LONG	17007
2	ORIG_INDEX	LONG	17451
3	NEVENTS	INT	153
4	X	FLOAT	Array[1000]
5	Y	FLOAT	Array[1000]
6	MC	INT	Array[1000]
7	DM	INT	Array[1000]
8	TIME	DOUBLE	1.8824165e+08
9	DQI	INT	0
10	FILTER	FLOAT	271.232
11	ROLL_RA	DOUBLE	11.761461
12	ROLL_DEC	DOUBLE	85.226608
13	ROLL_ROT	DOUBLE	166.56405
14	ANG_STEP	DOUBLE	0.0030735851
15	XOFF	FLOAT	-0.998805
16	YOFF	FLOAT	-1.00348

Figure 4: Spacecraft motion derived from the VIS channel (solid line). The jerk at about frame 3000 is likely due to the operation of the Scanning Sky Monitor (SSM). The difference between the motion derived from the UV channel and the VIS channel is plotted as a dashed line. The offset of about 5 pixels is due to differing starting points.

3.8. Image Production

In principle, it is easy to add the individual photon events from the Level 2 UV files to form an image of the sky, as we have done in Fig. 2 (using *jude_add_frames.pro*). However, the quality of the images will be determined largely by how well we are able to correct for the spacecraft motion. The VIS images were intended to track stars in the field and, as discussed in Section 3.4, we have matched the calculated VIS offsets with the UV frames to populate "XOFF" and "YOFF" in Table 3.

If the VIS data are not available, we obtained a first estimate of the motion from the on-board star sensor which reports the spacecraft boresight (RA, Dec, Roll Angle) every 16 seconds (464 frames). We used these to shift and add frames and, in the same manner as the VIS data, found the shifts between successive frames. Most of the UV sources observed to date are relatively faint, largely to protect the detectors from damage, and we had to integrate over $100 - 200$ frames ($3.5 - 7$ seconds) to get enough counts for *find.pro* to work and interpolate to find the shifts for each frame.

We found that we could dramatically improve the resolution if we adopted a more robust centroiding algorithm (*jude_centroid.pro*) in which we followed a single star in the image throughout the observation. The registration worked best when we added together $10 - 20$ frames with a star that had $3 - 10$ counts in that time. We calculated the centroid of the star in x and y in each set of frames and used the shifts between frames to derive the spacecraft motion. The difference between the spacecraft motion derived here and that inferred from the VIS channel is plotted in Fig. 4 and is on the order of 0.5 pixels (other than a constant offset of 5 pixels due to a different choice for the reference frame). The better resolution obtained with *jude_centroid.pro* is apparent in Fig. 5 with more detail seen in the central portion of Holmberg II. The effective FWHM of the stars is $1.7''$ when we use *jude_centroid*, and $4.7''$ when we use the VIS offsets.

4. Calibration

We have used *JUDE* to perform an in-flight calibration of UVIT (Rahna et al., 2017) and have incorporated those results back in the programs. We have left the Level 2 data in units of counts per second but have added a field to the header with the scale factor for the appropriate conversion into energy units (ergs cm^2 s^{-1} Å^{-1} (cps)$^{-1}$). We found no evidence for flat field variations, especially in view of the long star tracks due to the spacecraft motion. We have derived a distortion correction for the data but found that applying it did not improve the resolution. We therefore recommend that the distortion correction be done as part of the astrometric correction after the final image production.

The astrometric information in the header is based on the relatively poor attitude information from the star sensor. We made the decision that Astrometry.net (Lang et al., 2010) has solved this problem and, in keeping with our philosophy of reusing existing components, recommend it to the user for precise astrometry.

Figure 5: We achieve better resolution when we use self-registration through *jude_centroid* (right) than with the VIS-derived spacecraft motion (left).

5. Conclusions

We have designed a set of software routines that begin with Level 1 UVIT data from the ISSDC and produces images suitable for scientific purposes. We have tested the software on all our GT observations, and have produced Level 2 event lists and images for all observations through an automated process. A few observations required manual intervention, primarily due to registration problems. We have used *JUDE* to characterize the in-flight performance of UVIT (Rahna et al., 2017) and now expect to move on to the scientific exploitation of the data.

We have released *JUDE* under the Apache License 2.0,[3] and it is available on the Astrophysics Source Code Library (Murthy et al., 2016) and on GitHub (*https://github.com/jaymurthy/JUDE*). We are using data from the UVIT for our science and, therefore, will continue to maintain and improve *JUDE* as needed.

As a final note, the scientific exploitation of the UVIT data has been limited till now owing to the delays in the UVIT pipeline. We hope that with the release of *JUDE* to the astronomical community at large, more and better scientific results from the UVIT will be published. The only valid test of scientific software is if it is used widely, and we will continue to update the routines as we find errors or as they are reported to us. We welcome feature requests to improve the scientific utility of the programs.

Acknowledgements

References

References

Coulais, A., Schellens, M., Gales, J., Arabas, S., Boquien, M., Chanial, P., Messmer, P., Fillmore, D., Poplawski, O., Maret, S., Marchal, G., Galmiche, N., Mermet, T., Dec. 2010. Status of GDL - GNU Data Language. In: Mizumoto, Y., Morita, K.-I., Ohishi, M. (Eds.), Astronomical Data Analysis Software and Systems XIX. Vol. 434 of Astronomical Society of the Pacific Conference Series. p. 187.

Landsman, W. B., 1995. The IDL Astronomy User's Library. In: Shaw, R. A., Payne, H. E., Hayes, J. J. E.

---

[3]https://www.apache.org/licenses/LICENSE-2.0.txt

(Eds.), Astronomical Data Analysis Software and Systems IV. Vol. 77 of Astronomical Society of the Pacific Conference Series. p. 437.

Lang, D., Hogg, D. W., Mierle, K., Blanton, M., Roweis, S., May 2010. Astrometry.net: Blind Astrometric Calibration of Arbitrary Astronomical Images. *Astronom. J.* 139, 1782–1800.

Markwardt, C. B., Sep. 2009. Non-linear Least-squares Fitting in IDL with MPFIT. In: Bohlender, D. A., Durand, D., Dowler, P. (Eds.), Astronomical Data Analysis Software and Systems XVIII. Vol. 411 of Astronomical Society of the Pacific Conference Series. p. 251.

Murthy, J., Rahna, P. T., Safonova, M., Sutaria, F., Gudennavar, S. B., Bubbly, S. G., Jul. 2016. JUDE: An Utraviolet Imaging Telescope pipeline. Astrophysics Source Code Library.

Pati, A. K., 1999. Proposed UV - optical payload for the Indian astronomy satellite. *Bulletin of the Astronomical Society of India.* 27, 295.

Pati, A. K., Rao, N. K., Aug. 1998. UV imaging telescope for the Indian Multiwavelength Astronomy Satellite. In: Bely, P. Y., Breckinridge, J. B. (Eds.), Space Telescopes and Instruments V. Vol. 3356 of *Proc. SPIE*. pp. 635–640.

Pence, W. D., Chiappetti, L., Page, C. G., Shaw, R. A., Stobie, E., Dec. 2010. Definition of the Flexible Image Transport System (FITS), version 3.0. *Astron. Astrophys.* 524, A42.

Postma, J., Hutchings, J. B., Leahy, D., Jul. 2011. Calibration and Performance of the Photon-counting Detectors for the Ultraviolet Imaging Telescope (UVIT) of the Astrosat Observatory. *Publ. Astr. Soc. Pac.* 123, 833–843.

Rahna, P. T., Murthy, J., Safonova, M., Sutaria, F., Gudennavar, S. B., Bubbly, S. G., 2017. Investigating the In-Flight Performance of the UVIT Payload on AstroSat. E-print arXiv:1706.02523.

Singh, K. P., Tandon, S. N., Agrawal, P. C., Antia, H. M., Manchanda, R. K., Yadav, J. S., Seetha, S., Ramadevi, M. C., Rao, A. R., Bhattacharya, D., Paul, B., Sreekumar, P., Bhattacharyya, S., Stewart, G. C., Hutchings, J., Annapurni, S. A., Ghosh, S. K., Murthy, J., Pati, A., Rao, N. K., Stalin, C. S., Girish, V., Sankarasubramanian, K., Vadawale, S., Bhalerao, V. B., Dewangan, G. C., Dedhia, D. K., Hingar, M. K., Katoch, T. B., Kothare, A. T., Mirza, I., Mukerjee, K., Shah, H., Shah, P., Mohan, R., Sangal, A. K., Nagabhusana, S., Sriram, S., Malkar, J. P., Sreekumar, S., Abbey, A. F., Hansford, G. M., Beardmore, A. P., Sharma, M. R., Murthy, S., Kulkarni, R., Meena, G., Babu, V. C., Postma, J., Jul. 2014. ASTROSAT mission. In: Space Telescopes and Instrumentation 2014: Ultraviolet to Gamma Ray. Vol. 9144 of *Proc. SPIE*. p. 91441S.

Stetson, P. B., Mar. 1987. DAOPHOT - A computer program for crowded-field stellar photometry. *Publ. Astr. Soc. Pac.* 99, 191–222.

Subramaniam, A., Tandon, S. N., Hutchings, J., Ghosh, S. K., George, K., Girish, V., Kamath, P. U., Kathiravan, S., Kumar, A., Lancelot, J. P., Mahesh, P. K., Mohan, R., Murthy, J., Nagabhushana, S., Pati, A. K., Postma, J., Rao, N. K., Sankarasubramanian, K., Sreekumar, P., Sriram, S., Stalin, C. S., Sutaria, F., Sreedhar, Y. H., Barve, I. V., Mondal, C., Sahu, S., Jul. 2016. In-orbit performance of UVIT on ASTROSAT. In: Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray. Vol. 9905 of *Proc. SPIE*. p. 99051F.

Tandon, S. N., Hutchings, J. B., Ghosh, S. K., Subramaniam, A., Koshy, G., Girish V, V., Kamath, P. U., Kathiravan, s., Kumar, A., Lancelot, J. P., Mahesh, P. K., Mohan, R., Murthy, J., Nagabhushana, S., Pati A, A. K., Postma, J., Kameswara Rao, N., Sankarasubramanian, K., Sreekumar, P., Sriram, S., Stalin, C. S., Sutaria, F., Sreedhar, Y. H., Barve, I. V., Mondal, C., Sahu, S., Dec. 2016. In-orbit Performance of UVIT and First Results. E-print arXiv:1612.00612.

Tandon, S. N., Subramaniam, A., Girish, V., Postma, J., Sankarasubramanian, K., Sriram, S., Stalin, C. S., Mondal, C., Joseph, S. S. P., Hutchings, J., Ghosh, S. K., Barve, I. V., George, K., Kamath, P. U., Kathiravan, S., Kumar, A., Lancelot, J. P., Mahesh, P. K., Mohan, R., Nagabhushana, S., Pati, A. K., Kameswara Rao, N., Sreedhar, Y. H., Sreekumar, P., Sutaria, F., May 2017. In-orbit Calibrations of the Ultra-Violet Imaging Telescope. E-print arXiv:1705.03715.

Appendix A. IDL External Modules

JUDE uses routines from the IDL Astronomy Library (Landsman, 1995) and from the MPFIT library (Markwardt, 2009). These are listed in Table A.4 and may be downloaded individually or as part of the entire library.

Appendix B. Running the pipeline

We have written a GDL command file (*process_uvit.com*) to call each of the programs in order. The user will have to change the first two lines to conform to their environment.

1. **!path='jude_dir:'+!path** — *!path* is a system variable specifying the directories where the programs are located, if not in the working directory. It should include *JUDE* files and any additional programs required.

2. **dname='root_dir/'** — Sets the variable *dname* to the root directory for the UVIT Level 1 data files.

3. **jude_driver_vis,dname** — Processes the VIS files and derives the spacecraft motion. The VIS processing should occur before the two UV channels because the image correction comes from the VIS channel.

4. **jude_driver_uv,dname,/nuv** — Processes the NUV files to produce event lists.

5. **jude_driver_uv,dname,/fuv** — Processes the FUV files to produce event lists. The two UV routines may be run in either order.

6. **jude_uv_cleanup,/nuv** — Merges the NUV event lists, matches with offsets from the VIS channel, and produces images.

7. **jude_uv_cleanup,/fuv** — Merges the FUV event lists, matches with offsets from the VIS channel, and produces images.

Appendix C. Parameters

The parameters used in *JUDE* are listed in Table C.5 and are passed to the pipeline through a structure defined in *jude_params.pro*. The advantage of using a structure is that it allows for further parameters to be added if required, without changing the program invocation.

Table A.4: External Library Routines

Module	Purpose
add_distort.pro	Add the distortion parameters in an astrometry structure to a FITS header.
ad2xy.pro	Coordinate to pixel conversion.
cgErrorMsg.pro	a device-independent error messaging function.
correl_images.pro	Compute the 2-D cross-correlation function of two images.
corrmat_analyze.pro	Find the optimal (x, y) offset to maximize correlation of 2 images.
detabify.pro	Replace tabs with spaces.
data_chk.pro	Checks input data.
datatype.pro	Returns data type of a variable.
daycnv.pro	Converts Julian dates to Gregorian calendar dates.
gcirc.pro	Computes rigorous great circle arc distances.
getcoords.pro	Converts a string with angular coordinates to floating point values.
get_date.pro	Return the UTC date in CCYY-MM-DD format.
get_equinox.pro	Returns equinox.
getopt.pro	Convert a string into a valid scalar or vector.
gettok.pro	String manipulation.
get_wrd.pro	Return the nth word from a text string.
file_exist.pro	Tests for the existence of a file.
find.pro	Find point sources.
file_stat.pro	Returns information on a file.
fxaddpar.pro	Add parameter to FITS header.
fxbhmake.pro	Create basic FITS header.
fxhclean.pro	Removes keywords from FITS header.
fxpar.pro	Obtains value of a parameter in the FITS header.
fxparpos.pro	Finds position to insert par.
fxposit.pro	FITS file manipulation.
fxmove.pro	Skip extensions in a FITS file.
make_astr.pro	Builds astrometry structure.
match.pro	Find subscripts where vectors match.
mean.pro	Find mean.
mkhdr.pro	Makes FITS header.
mpfit.pro	Perform Levenberg-Marquardt least-squares minimization.
mpfit2dfun.pro	Perform Levenberg-Marquardt least-squares fit to a 2-D IDL function.
mpfit2dpeak.pro	Fit a gaussian, lorentzian or Moffat model to data.
mrd_hread.pro	Reads FITS header.
mrdfits.pro	Reads FITS files.
mrd_skip.pro	Skip bytes in a file.
mrd_struct.pro	Create structure.
mwrfits.pro	Writes FITS files.
putast.pro	Adds astrometry to FITS header.
quadterp.pro	Interpolation.
remove.pro	Remove elements from vector.
reverse.pro	Reverse the order of rows or columns in an array.
srcor.pro	Match point sources in image.
strsplit.pro	STRTOK wrapper.
sxaddhist.pro	Adds HISTORY to FITS header.
sxaddpar.pro	Adds parameter to FITS header.
sxpar.pro	Obtain the value of a parameter in a FITS header
tag_exist.pro	Check for tags in a structure.
valid_num.pro	Checks that a string is a valid number representation.
wcs_check_ctype.pro	Checks CTYPE parameters and return the projection type and coordinate type.
wcs_rotate.pro	Rotate between standard and native coordinates.
wcssph2xy.pro	Coordinate to pixel conversion.
write_png.pro	Writes PNG (Portable Network Graphics) file.
zparcheck.pro	Check type and size of a parameter.

Table C.5: Program parameters

Parameter	Description	Default Value
resolution	Number of sub-pixels in one physical pixel (1, 2, 4 or 8).	8
min_counts	Frames with less counts are rejected (Fig. 3).	0
max_counts	Frames with more counts are rejected. If 0, we use the $median + 3 \times \sqrt{median}$.	0
min_frame	Starting frame. In practice, the first frame where the DQI value is 0.	0
max_frame	Ending frame. If 0, last frame where DQI = 0.	0
coarse_bin	Number of frames binned to find stars in *jude_register.pro*.	200
fine_bin	Number of frames binned to find centroids in *jude_centroid.pro*.	20
ps_threshold_fuv	Threshold for FUV point sources in counts s^{-1} pixel^{-1}. Used in *jude_register*.	3×10^{-4}
ps_threshold_nuv	Threshold for NUV point sources in counts s^{-1} pixel^{-1}. Used in *jude_register*.	1.5×10^{-3}
flat_field	Place holder if we want to use a flat field. Not implemented as yet.	"No flat field"
events_dir	Output directory for Level 2 event lists. Separated by channel (VIS, NUV, FUV).	"events/"
image_dir	Output directory for FITS images.	"images/"
mask_dir	Holds mask files. Each mask is a FITS file containing an 512×512 array where the region of interest is marked with 1 and the other elements are 0. Masks are not used in the pipeline but the functionality is in *jude_interactive.pro* and the masks are written by *jude_mask.pro*.	"masks/"
def_nuv_dir	Default directory for NUV channel.	"nuv/"
def_fuv_dir	Default directory for FUV channel.	"fuv/"
def_vis_dir	Default directory for VIS channel.	"vis/"
png_dir	Output directory for PNG diagnostic images.	"png/
vis_L2_dir	Output directory for VIS files.	"vis_files/"
vis_off_dir	Output directory for files with spacecraft motion.	"vis_off/"
vis_add_dir	Output directory for VIS images.	"vis_add/"
temp_dir	Directory for temporary files.	"jude_temp/"

Appendix D. Level 2 Data Format

Appendix D.1. VIS Files

1. A GDL save set containing three variables:
   (a) GRID — Floating point array of dimensions $512 \times 512 \times NFRAMES$. The VIS data consist of individual CMOS frames with 512×512 pixels with a total number of frames in the Level 1 data denoted by $NFRAMES$.
   (b) TIMES — Double precision array with $NFRAMES$ elements containing the UVIT time (the number of seconds since Jan. 1, 2010).
   (c) DATA_HDR — String array with the header information from the Level 1 VIS file. The format is the standard GDL format for FITS headers with one line for each keyword in the original FITS header.

2. A text file containing the offsets between each frame in the VIS data and the first frame. The first line has two elements: the starting and ending time for that file and each successive line has 4 elements, with the first column being the reference time and the second being the time for that particular frame. The last two columns are the X and Y shifts of that frame with respect to the reference frame. The times are specified to 6 decimal places with the first 4 being significant, and the X and Y offsets are specified to 2 decimal places.

3. A FITS file with the co-added VIS data, including the correction for spacecraft motion.

Appendix D.2. UV Data

1. A FITS binary table for each of the two UV bands containing the photon event list as a function of frame number with relevant housekeeping and attitude information (Table 3). An optional second extension contains the offsets from the VIS data. The components of the binary table are as follows:
   (a) FRAMENO is the frame number from the Level 1 data. A single frame may run across multiple lines if there are more than 336 events recorded, usually due to a cosmic ray hit. There may also be duplicated frames.
   (b) ORIG_INDEX is the row number (beginning with 0 as in the GDL convention) in the original data file (recorded in the FITS header). ORIG_INDEX is $\geq$ than FRAMENO due to the repetition of frames in the Level 1 data.

   (c) NEVENTS is the total number of photon events in the frame with a maximum of value of 999.
   (d) X is an array contains the column number of each event up to NEVENTS ranging from 0 to 511.875.
   (e) Y is an array contains the column number of each event up to NEVENTS ranging from 0 to 511.875.
   (f) MC is the minimum corner value in the 5×5 on-board centroiding.
   (g) DM is the (maximum - minimum) 5×5 corner value. MC and DM are reported as they may be used for diagnostics but are not used in *JUDE*.
   (h) TIME is the elapsed mission time in seconds from Jan. 1, 2010.
   (i) DQI is a diagnostic flag set by the pipeline. The values are defined in Table 2.
   (j) FILTER is the angle of the filter wheel where the conversion from angle to filter is in *jude_set_dqi.pro.*
   (k) ROLL_RA is the right ascension (RA) of the spacecraft boresight as reported by the star sensor.
   (l) ROLL_DEC is the declination (DEC) of the spacecraft boresight as reported by the star sensor.
   (m) ROLL_ROT is the roll angle of the spacecraft boresight as reported by the star sensor.
   (n) ANG_STEP is the shift in the spacecraft pointing between one frame and the next in arcseconds.
   (o) XOFF is the shift in sensor pixels (where the CMOS sensor is a 512×512 element array) between the frame and the reference frame in the X direction, where the reference frame is defined by the FITS header of the image file.
   (p) YOFF is the same as XOFF but in the Y direction.

2. A FITS image file for each UV band with two extensions. The first extension is the co-added data from a single Level 1 file in units of counts s^{-1} pixel^{-1} with the second extension containing the exposure time per pixel. Although the file contains astrometric information, it is only as accurate as the spacecraft star sensor. The scale factor to convert from counts s^{-1} pixel^{-1} into ergs cm^{-2} s^{-1} Å^{-1} is in the FITS header.

3. A PNG file for a quick look containing the scaled image, a plot of the DQI values, a histogram of the number of counts, and a plot of the spacecraft motion.

4. A text file with a comma separated observation log.