Curvit

Prajwel Joseph

Apr 10, 2024

CONTENTS:

Welcome to Curvit's documentation! Curvit can be used to create light curves from AstroSat UVIT data.

Important: Please include a *Citation* if you use Curvit for work presented in a publication or talk.

Caution: The software is under active development. For new features and bug fixes, please check *Changelog*.

Curvit is an open-source python package to produce light curves from UVIT (Ultraviolet Imaging Telescope) data. The events list from the **official UVIT L2 pipeline (version 6.3 onwards)** is required as input to the package. If you have data from old official pipeline versions or other pipelines, please get in touch with me, and we can try to figure out a solution!

The software paper manuscript is accessible at https://arxiv.org/abs/2101.06377

CHAPTER

ONE

INSTALLATION

Curvit is on the Python Package Index (PyPI), and you can install the package using pip as follows:

```
pip install curvit --user
```

1.1 Requirements

Curvit has the following requirements. If you have an older version of Python and is new to the language, I recommend that you install the Anaconda Distribution. It installs Python in a user directory without requiring root permissions and will work on Linux, macOS, and Windows.

- Python 3.6 or later
- Aafitrans
- Astropy
- Astroquery
- Matplotlib
- Numpy
- Photutils
- Scipy
- Scikit-image

CHAPTER

TWO

GETTING STARTED

Curvit's capabilities can be best demonstrated by examples. First, we need to get events list to be provided as input (an events list is a FITS file containing events). Go to ISSDC's AstroBrowse website and download UVIT data of your interest. If you are new to the Astrobrowse website and UVIT Leve2 data, please visit this page for a quick walkthrough. Here, as an example, the publicly available Level2 data of FO Aqr was chosen (Observation ID: `G06_084T01_9000000710`).

This dataset, `LEVL2AS1UVT20161005G06_084T01_9000000710.zip`, is a compressed file that needs to be extracted. Once extracted, a directory named `20161005_G06_084T01_9000000710_level2` can be found. This directory has the following structure.

```
20161005_G06_084T01_9000000710_level2/
└── uvit
    ├── RAS_NUV
    │   ├── pipeline
    │   ├── uvt_01
    │   ├── uvt_02
    │   ├── uvt_03
    │   ├── ...
    │   ├── ...
    │   └── uvt_ci
    ├── RAS_VIS
    │   ├── pipeline
    │   ├── uvt_01
    │   ├── uvt_02
    │   ├── uvt_03
    │   ├── ...
    │   ├── ...
    │   └── uvt_ci
    ├── DISCLAIMER.txt
    ├── LEVL1AS1UVT20161005G06_084T01_9000000710_05546_V2.2_dqr.xml
    └── README.txt
```

Please read the `README.txt` for details on Level2 data and what it contains. `RAS_VIS` directory contains images that were corrected for satellite drift by using the VIS (visible) channel. For drift correcting the images inside the `RAS_NUV` directory, NUV (near-ultraviolet) channel was used. For most cases, the data from `RAS_VIS` would be suitable. If you download a dataset that is different from the one mentioned above, check the statistics inside `DISCLAIMER.txt` to decide what to use.

Our directory of interest, `RAS_VIS`, has the following contents.

```
RAS_VIS/
├── pipeline
```

(continues on next page)

(continued from previous page)

```
    └── LEVL2AS1UVT20161005G06_084_L2_DM_params.txt
├── uvt_01
    ├── F_01
        ├── AS1G06_084T01_9000000710uvtFIIPC00F1A_l2err.fits
        ├── AS1G06_084T01_9000000710uvtFIIPC00F1A_l2exp.fits
        ├── AS1G06_084T01_9000000710uvtFIIPC00F1A_l2img.fits
        ├── AS1G06_084T01_9000000710uvtFIIPC00F1I_l2img.fits
        └── AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.fits
    ├── N_01
        ├── AS1G06_084T01_9000000710uvtNIIPC00F2A_l2err.fits
        ├── AS1G06_084T01_9000000710uvtNIIPC00F2A_l2exp.fits
        ├── AS1G06_084T01_9000000710uvtNIIPC00F2A_l2img.fits
        ├── AS1G06_084T01_9000000710uvtNIIPC00F2I_l2img.fits
        └── AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.fits
    └── V_01
        └── AS1G06_084T01_9000000710uvtVIIIM00F2_l2dr.fits
├── ...
├── ...
```

Inside the directory `uvt_01`, data are organized in separate folders, each corresponding to overlapping time ranges in UV and VIS channels, as available in Level1 dataset (`F_01`: FUV; `N_01`: NUV; `V_01`: VIS).

The suffixes of the FITS files have the following meaning.

- `...A_l2img.fits`: Image file in astronomical coordinates.
- `...I_l2img.fits`: Image file in instrument coordinates.
- `...A_l2exp.fits`: Exposure map for `A_l2img.fits`.
- `...A_l2err.fits`: Error map for `A_l2img.fits`.
- `...l2ce.fits` : Corrected events list.
- `...l2dr.fits` : the Relative Aspects Series (RAS) file.

This structure of subdirectories shall repeat for all sets - `uvt_01`, `uvt_02`, `uvt_03`, etc.

For the examples given below, we will be using the FUV events list (`...l2ce.fits`) from `uvt_03` as input to curvit.

Important: The Level2 directory structure and FITS file naming conventions here explained are for the Level2 data of the 6.3 version obtained from ISSDC. Always refer to the `README.txt` included along with the Level2 data to understand the data structure.

2.1 makecurves

The `makecurves` function of curvit can automatically detect sources from the events list and create light curves. Please note that curvit currently provides source coordinates only in the **instrument coordinate system**.

```
>>> import curvit
>>> curvit.makecurves(events_list = 'AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.fits.gz',
...                   threshold = 5)
```

```
Detected source coordinates saved in file:
* sources_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.coo
Detected sources are plotted in the image:
* sources_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.png

---------------------- light curves ----------------------
* makecurves_3136.64_3651.08_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.png
* makecurves_2530.02_1442.18_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.png
* makecurves_2912.31_3657.17_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.png
...
...

Done!
```

Important: The zero-based indexing scheme is used in curvit. Therefore, if you open the corresponding FITS image file in instrument coordinates (`...I_l2img.fits`) in DS9, there will be a difference of 1 between the source coordinates in DS9 and curvit. For example, the curvit coordinates of (2559, 806) will become (2560, 807) in the FITS convention.

2.2 curve

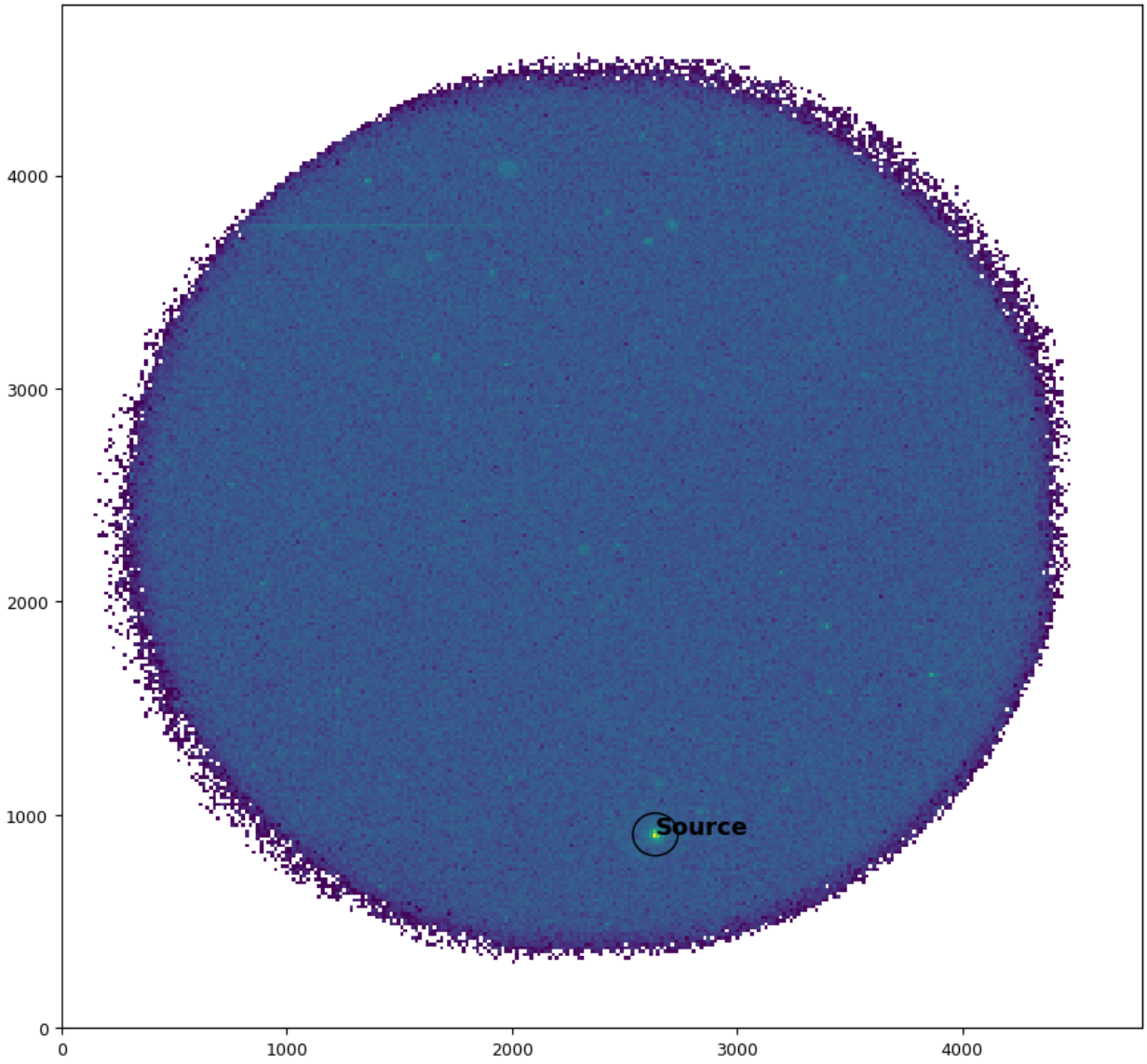
If you know the source coordinates, use the `curve` function of curvit to create light curves.

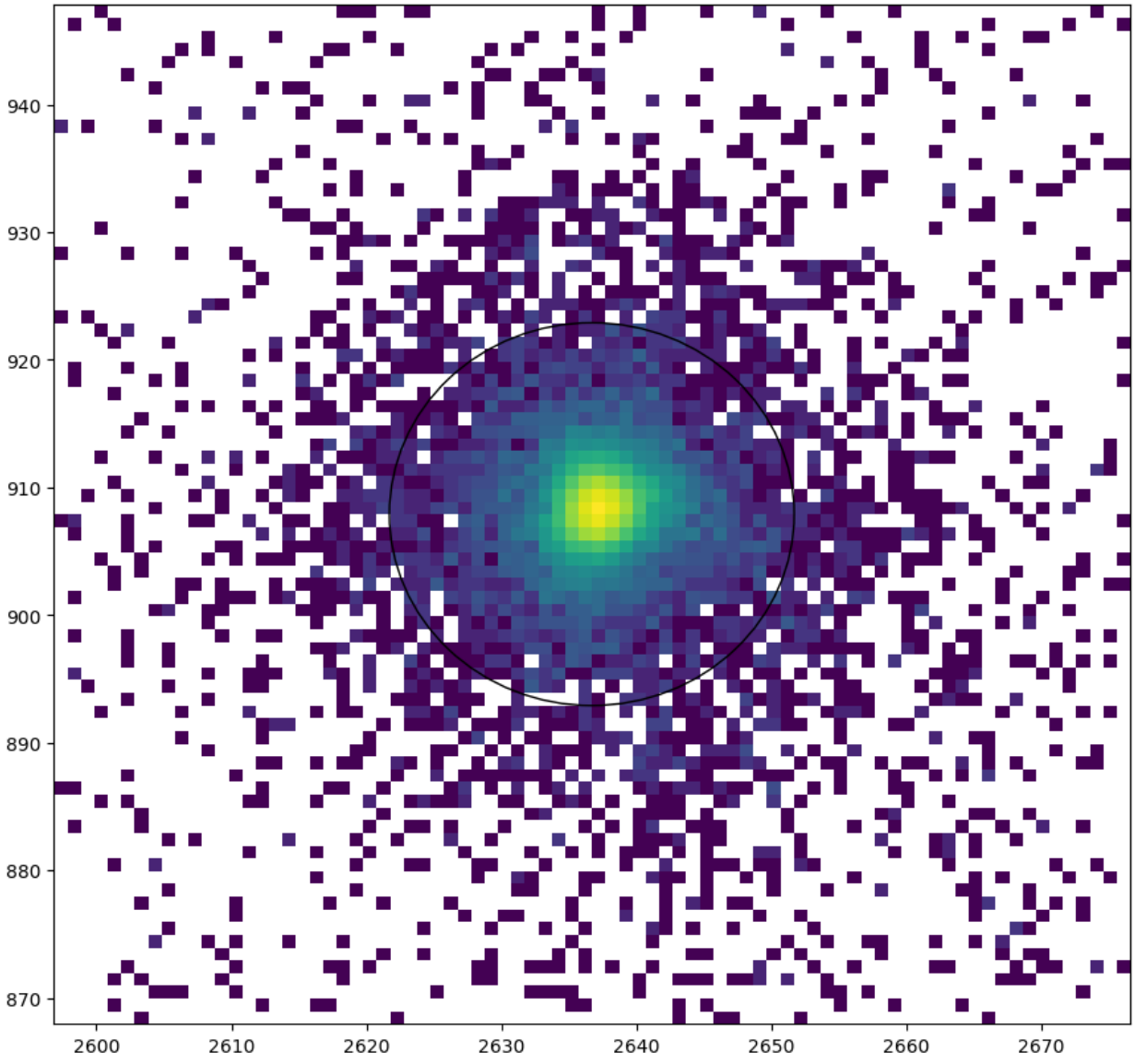
```
>>> curvit.curve(events_list = 'AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.fits.gz',
...              xp = 2636.71, yp = 907.91,
...              radius = 15,
...              bwidth = 50,
...              background = 'auto')
```

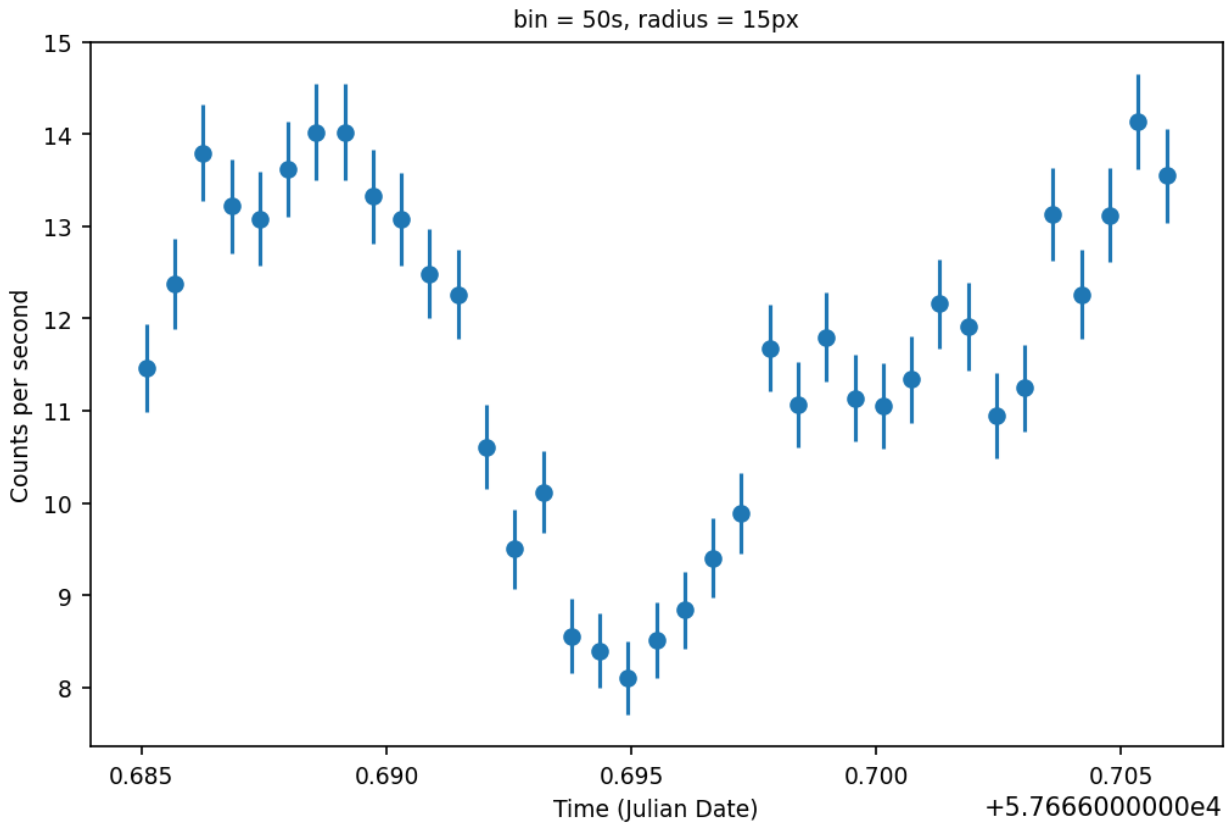
```
Background CPS (scaled to aperture area): 0.02113 ± 0.00421

-------------------------- curve --------------------------
source: source_AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.png
        source_zoomed_AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.png
data: curve_2636.71_907.91_AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.dat
plot: curve_2636.71_907.91_AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.png

Done!
```


Source
0
1000
2000
3000
4000
0
1000
2000
3000
4000


940
930
920
910
900
890
880
870
2600
2610
2620
2630
2640
2650
2660
2670



2.3 Parameters

The curvit package has a set of parameters for which the users can set values. Some of them have default values.

2.3.1 Parameters common to both `makecurves` and `curve`

- **events_list** - The name of the events list (`...l2ce.fits`). The string can also include the path to the file.
- **radius** - The radius of the source aperture in pixels. This parameter has a default value of `6`.
- **sky_radius** - The radius of the background aperture in pixels. The default value is `12`.
- **bwidth** - Time bin width in seconds. The default value is `50`.
- **framecount_per_sec** - Framerate, with a default value of `28.7185` frames per second for 512 x 512 window mode. The most accurate way to get the framerate would be to take the value of (`1 / INT_TIME`). `INT_TIME` value can be found from the corresponding image header. Approximate values of framerate for different window modes of UVIT are given in the table below.

window mode	frames per second
512 x 512	28.7
350 x 350	61
300 x 300	82
250 x 250	115
200 x 200	180
150 x 150	300
100 x 100	640

Note: It is essential to set the correct value of the framerate. Most UVIT observations are carried out in 512 x 512 window mode.

- **background** - Valid inputs are `None`, `'auto'`, or `'manual'`. The parameter affects how the background count-rate estimation is done. The default value is `None`, and no background estimation is carried out. `'auto'` will automatically estimate the background count-rate. If you prefer to specify a background region manually, then give `'manual'` as the value and specify **x_bg** (background X-coordinate) and **y_bg** (background Y-coordinate) parameters.
- **aperture_correction** - Valid inputs are `None`, `'fuv'`, or `'nuv'`. The default value is `None`. The parameter value should be changed to either `'fuv'` or `'nuv'` to apply appropriate aperture corrections to the light curve data.
- **saturation_correction** - Takes either `True` or `False`. The default value is `False`. If the parameter is set to `True`, saturation correction is applied to the light curve data.

2.3.2 Parameters only required for `makecurves`

- **detection_method** - Two source detection methods are available: `'daofind'` and `'kdtree'`. The default method is `'daofind'`.
- **threshold** - The threshold parameter associated with the `'daofind'` method. The default value is `4`.
- **how_many** - The limit for the number of sources to be detected using the `'kdtree'` method. The default value is `4`.

2.3.3 Parameters only required for `curve`

- **xp** - X-coordinate of the source.
- **yp** - Y-coordinate of the source.

CHAPTER

THREE

CITATION

If you use Curvit in your research, please cite the following paper:

Joseph, P., C. S. Stalin, S. N. Tandon, and S. K. Ghosh. "Curvit: An open-source Python package to generate light curves from UVIT data." Journal of Astrophysics and Astronomy 42, no. 2 (2021): 1-10.

To export the citation in any preferred format, use this service of ADS.

CHAPTER

FOUR

API

4.1 makecurves()

curvit.**makecurves**(*events_list=''*, *radius=6*, *detection_method='daofind'*, *threshold=4*, *how_many=4*, *bwidth=50*, *framecount_per_sec=28.7185*, *background=None*, *sky_radius=12*, *x_bg=None*, *y_bg=None*, *aperture_correction=None*, *saturation_correction=False*, *ZEF_correction_factor=1*, *whole_figure_resolution=256*, *sub_fig_size=40*, *fontsize=9*)

Automatically detect sources amd create light curves.

Parameters

- **events_list** (*file path*) – The name of the events list FITS file.
- **radius** (*float, optional*) – The source aperture radius in pixels. This parameter has a default value of 6.
- **detection_method** (*{'daofind', 'kdtree'}, optional*) – The parameter to choose between available detection methods.
  - `'daofind'`: To use the DAOFIND algorithm. This is the default method.
  - `'kdtree'`: Source detection method based on a k-d tree implementation.
- **threshold** (*float, optional*) – The threshold parameter associated with the `'daofind'` method. The default value is 4.
- **how_many** (*int, optional*) – The limit for the number of sources to be detected using the `'kdtree'` method. The default value is 4.
- **bwidth** (*float, optional*) – Time bin width in seconds. the default value is 50.
- **framecount_per_sec** (*float, optional*) – The framerate of the observation, with a default value of 28.7185 frames per second for 512 x 512 window mode. The most accurate way to get the framerate would be to take the value of (`1 / INT_TIME`). `INT_TIME` value can be found from the corresponding image header. Approximate values of framerate for different window modes of UVIT are given in the table below.

window mode	frames per second
512 x 512	28.7
350 x 350	61
300 x 300	82
250 x 250	115
200 x 200	180
150 x 150	300
100 x 100	640

- **background** (*{'auto', 'manual', None}, optional*) – The parameter affects how the background count-rate estimation is done.
  - `'auto'`: Automatic estimation of the background count-rate.
  - `'manual'`: To manually specify a background region using **x_bg** and **y_bg** parameters.
  - `None`: No background estimation is carried out. This is the default method.
- **sky_radius** (*float, optional*) – The background aperture radius in pixels. The default value is 12.
- **x_bg** (*float, optional*) – The X-coordinate of the background region.
- **y_bg** (*float, optional*) – The Y-coordinate of the background region.
- **aperture_correction** (*{'fuv', 'nuv', None}, optional*) – The parameter affects how the aperture correction is done.
  - `'fuv'`: Aperture correction for the FUV channel is applied.
  - `'nuv'`: Aperture correction for the NUV channel is applied.
  - `None`: No aperture correction is applied. This is the default method.
- **saturation_correction** (*bool, optional*) – If *True*, saturation correction is applied. The default value is *False*. Note that aperture correction should be applied if you apply saturation correction.
- **ZEF_correction_factor** (*float, optional*) – To apply the correction for zero event frames. The default value is 1 (no correction).

Note: It is essential to set the correct value of the framerate. Most UVIT observations are carried out in 512 x 512 window mode.

Example

```
>>> import curvit
>>> curvit.makecurves(events_list = 'AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.fits.
↪gz',
...                   threshold = 5)
```

```
Detected source coordinates saved in file:
* sources_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.coo
Detected sources plotted in the image:
* sources_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.png

---------------------- light curves ----------------------
* makecurves_3136.64_3651.08_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.png
* makecurves_2530.02_1442.18_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.png
* makecurves_2912.31_3657.17_AS1G06_084T01_9000000710uvtNIIPC00F2_l2ce.png
...
...

Done!
```

4.2 curve()

curvit.**curve**(*events_list=''*, *xp=None*, *yp=None*, *radius=6*, *bwidth=50*, *framecount_per_sec=28.7185*, *background=None*, *sky_radius=12*, *x_bg=None*, *y_bg=None*, *aperture_correction=None*, *saturation_correction=False*, *ZEF_correction_factor=1*, *whole_figure_resolution=256*, *sub_fig_size=40*, *fontsize=9*)

Create light curve for a source.

Parameters

- **events_list** (*file path*) – The name of the events list FITS file.
- **xp** (*float*) – The X-coordinate of the source.
- **yp** (*float*) – The Y-coordinate of the source.
- **radius** (*float, optional*) – The source aperture radius in pixels. This parameter has a default value of 6.
- **bwidth** (*float, optional*) – Time bin width in seconds. the default value is 50.
- **framecount_per_sec** (*float, optional*) – The framerate of the observation, with a default value of 28.7185 frames per second for 512 x 512 window mode. The most accurate way to get the framerate would be to take the value of (`1 / INT_TIME`). `INT_TIME` value can be found from the corresponding image header. Approximate values of framerate for different window modes of UVIT are given in the table below.

window mode	frames per second
512 x 512	28.7
350 x 350	61
300 x 300	82
250 x 250	115
200 x 200	180
150 x 150	300
100 x 100	640

- **background** (*{'auto', 'manual', None}, optional*) – The parameter affects how the background count-rate estimation is done.
  - `'auto'`: Automatic estimation of the background count-rate.
  - `'manual'`: To manually specify a background region using **x_bg** and **y_bg** parameters.
  - `None`: No background estimation is carried out. This is the default method.
- **sky_radius** (*float, optional*) – The background aperture radius in pixels. The default value is 12.
- **x_bg** (*float, optional*) – The X-coordinate of the background region.
- **y_bg** (*float, optional*) – The Y-coordinate of the background region.
- **aperture_correction** (*{'fuv', 'nuv', None}, optional*) – The parameter affects how the aperture correction is done.
  - `'fuv'`: Aperture correction for the FUV channel is applied.
  - `'nuv'`: Aperture correction for the NUV channel is applied.
  - `None`: No aperture correction is applied. This is the default method.

- **saturation_correction** (*bool, optional*) – If *True*, saturation correction is applied. The default value is *False*. Note that aperture correction should be applied if you apply saturation correction.
- **ZEF_correction_factor** (*float, optional*) – To apply the correction for zero event frames. The default value is 1 (no correction).

Note: It is essential to set the correct value of the framerate. Most UVIT observations are carried out in 512 x 512 window mode.

Example

```
>>> import curvit
>>> curvit.curve(events_list = 'AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.fits.gz',
...              xp = 2636.71, yp = 907.91,
...              radius = 15,
...              bwidth = 50,
...              background = 'auto')
```

```
Background CPS (scaled to aperture area): 0.02155 ± 0.00425

------------------------- curve -------------------------
source: source_2636.71_907.91_AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.png
        source_zoomed_2636.71_907.91_AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.png
data: curve_2636.71_907.91_AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.dat
plot: curve_2636.71_907.91_AS1G06_084T01_9000000710uvtFIIPC00F1_l2ce.png

Done!
```

4.3 curve_orbitwise()

curvit.**curve_orbitwise**(*events_list=''*, *xp=None*, *yp=None*, *radius=6*, *time_separation=1800*, *framecount_per_sec=28.7185*, *background=None*, *sky_radius=12*, *x_bg=None*, *y_bg=None*, *aperture_correction=None*, *saturation_correction=False*, *ZEF_correction_factor=1*, *whole_figure_resolution=256*, *sub_fig_size=40*, *fontsize=9*)

Create a light curve for a source, generating one data point per orbit. This function works best with the combined events list produced using *combine_events_lists()*.

Parameters

- **events_list** (*file path*) – The name of the events list FITS file.
- **xp** (*float*) – The X-coordinate of the source.
- **yp** (*float*) – The Y-coordinate of the source.
- **radius** (*float, optional*) – The source aperture radius in pixels. This parameter has a default value of 6.
- **time_separation** (*float, optional*) – The time separation in seconds. This parameter has a default value of 30 * 60 seconds.

- **framecount_per_sec** (*float, optional*) – The framerate of the observation, with a default value of 28.7185 frames per second for 512 x 512 window mode. The most accurate way to get the framerate would be to take the value of (`1 / INT_TIME`). `INT_TIME` value can be found from the corresponding image header. Approximate values of framerate for different window modes of UVIT are given in the table below.

window mode	frames per second
512 x 512	28.7
350 x 350	61
300 x 300	82
250 x 250	115
200 x 200	180
150 x 150	300
100 x 100	640

- **background** (*{'auto', 'manual', None}, optional*) – The parameter affects how the background count-rate estimation is done.
  - `'auto'`: Automatic estimation of the background count-rate.
  - `'manual'`: To manually specify a background region using **x_bg** and **y_bg** parameters.
  - `None`: No background estimation is carried out. This is the default method.
- **sky_radius** (*float, optional*) – The background aperture radius in pixels. The default value is 12.
- **x_bg** (*float, optional*) – The X-coordinate of the background region.
- **y_bg** (*float, optional*) – The Y-coordinate of the background region.
- **aperture_correction** (*{'fuv', 'nuv', None}, optional*) – The parameter affects how the aperture correction is done.
  - `'fuv'`: Aperture correction for the FUV channel is applied.
  - `'nuv'`: Aperture correction for the NUV channel is applied.
  - `None`: No aperture correction is applied. This is the default method.
- **saturation_correction** (*bool, optional*) – If *True*, saturation correction is applied. The default value is *False*. Note that aperture correction should be applied if you apply saturation correction.
- **ZEF_correction_factor** (*float, optional*) – To apply the correction for zero event frames. The default value is 1 (no correction).

Note: It is essential to set the correct value of the framerate. Most UVIT observations are carried out in 512 x 512 window mode.

Example

```
>>> import curvit
>>> curvit.curve_orbitwise('AS1G05_240T01_9000000674uvtFIIPC00F2_l2ce_all_orbits.
↪fits',
...                         xp = 1425.26, yp = 1861.78,
...                         radius = 15,
...                         background = 'auto')
```

```
Background CPS (scaled to aperture area): 0.01473 ± 0.00073

------------------------- curve -------------------------
source: source_1425.26_1861.78_AS1G05_240T01_9000000674uvtFIIPC00F2_l2ce_all_orbits.
↪png
        source_1425.26_1861.78_zoomed_AS1G05_240T01_9000000674uvtFIIPC00F2_l2ce_all_
↪orbits.png
data: curve_orbitwise_1425.26_1861.78_AS1G05_240T01_9000000674uvtFIIPC00F2_l2ce_all_
↪orbits.dat
plot: curve_orbitwise_1425.26_1861.78_AS1G05_240T01_9000000674uvtFIIPC00F2_l2ce_all_
↪orbits.png

Done!
```

4.4 make_image()

curvit.**make_image**(*events_list=''*, *framecount_per_sec=28.7185*)

Create a FITS image from the input events list. The image unit is in counts.

Parameters

- **events_list** (*file path*) – The name of the events list FITS file.
- **framecount_per_sec** (*float, optional*) – The framerate of the observation, with a default value of 28.7185 frames per second for 512 x 512 window mode. The most accurate way to get the framerate would be to take the value of (`1 / INT_TIME`). `INT_TIME` value can be found from the corresponding image header. Approximate values of framerate for different window modes of UVIT are given in the table below.

window mode	frames per second
512 x 512	28.7
350 x 350	61
300 x 300	82
250 x 250	115
200 x 200	180
150 x 150	300
100 x 100	640

Warning: If you plan to use the generated FITS image for science, make sure to give the proper framerate value.

Example

```
>>> import curvit
>>> curvit.make_image('test_events_list.fits', 28.7185)
```

The above script will generate a FITS image called `test_events_list_image.fits`. You may open it in software such as DS9 to view the image.

4.5 process_ccdlab()

curvit.**process_ccdlab**(*output=None*, *time_list=None*, *XY_integers=None*, *XY_fractions=None*, *flat_list=None*, *framecount_per_sec=28.7185*)

Generate a Curvit compatible events list from CCDLAB files.

Parameters

- **output** (*file path*) – The name of the output events list FITS file.
- **time_list** (*file path*) – The name of the CCDLAB time list FITS file
- **XY_integers** (*file path*) – The name of the CCDLAB XY integers FITS file
- **XY_fractions** (*file path*) – The name of the CCDLAB XY fractions FITS file
- **flat_list** (*file path*) – The name of the CCDLAB flat list FITS file
- **framecount_per_sec** (*float, optional*) – The framerate of the observation, with a default value of 28.7185 frames per second for 512 x 512 window mode. The most accurate way to get the framerate would be to take the value of (`1 / INT_TIME`). `INT_TIME` value can be found from the corresponding image header. Approximate values of framerate for different window modes of UVIT are given in the table below.

window mode	frames per second
512 x 512	28.7
350 x 350	61
300 x 300	82
250 x 250	115
200 x 200	180
150 x 150	300
100 x 100	640

Note: It is essential to set the correct value of the framerate. Most UVIT observations are carried out in 512 x 512 window mode.

Warning: This function is new; please report if you find any bugs.

Example

```
>>> import curvit
>>> process_ccdlab(output = 'output_events_list.fits',
...                time_list = 'sample_TimeList.fits',
...                XY_integers = 'sample_XYInts_List.fits',
...                XY_fractions = 'sample_XYFrac_List.fits',
...                flat_list = 'sample_FlatList.fits',
...                framecount_per_sec = 28.7185)
```

The above script will generate a FITS table called `output_events_list.fits`. You may then use it as input to `curve` or `makecurves`.

4.6 combine_events_lists()

curvit.**combine_events_lists**(*events_lists_paths=None*, *threshold=8*, *num_nearest_neighbors=8*, *min_matches=4*, *pixel_tolerance=2*, *minimum_detections=3*, *maximum_detections=100*, *shift_algorithm='multiple_star'*, *min_exptime=30*, *framecount_per_sec=28.7185*)

Align and combine the events lists from different orbits.

Parameters

- **events_lists_paths** (*list*) – The list of events list FITS file paths.
- **threshold** (*float, optional*) – The threshold parameter associated with the source detection method. The default value is 8.
- **num_nearest_neighbors** (*int, optional*) – The number of nearest neighbors of a given star (including itself) to construct the triangle invariants for matching. The default value is 8. Only relevant for `'multiple_star'` method.
- **min_matches** (*int, optional*) – The minimum number of triangle matches to be found. A value of 1 refers to 1 triangle, corresponding to 3 pairs of coordinates. The default value is 4. Only relevant for `'multiple_star'` method.
- **pixel_tolerance** (*int, optional*) – The maximum residual error for matches. The default value is 2. Only relevant for `'multiple_star'` method.
- **minimum_detections** (*int, optional*) – The minimum number of sources to be detected. The threshold will be modified to meet the criteria.
- **maximum_detections** (*int, optional*) – The maximum number of sources to be detected. The threshold will be modified to meet the criteria.
- **shift_algorithm** (*{'single_star', 'multiple_star'}, optional*) – The parameter to choose between available aligning methods.
  - **`'single_star'`: To use a single star for aligning orbits.**
    Single_star option is useful only when there is only one star in the field and no rotation between frames. The minimum number of source detection is 2.
  - **`'multiple_star'`: To use multiple stars for aligning orbits.**
    This is the **recommended** and default method.
- **min_exptime** (*float, optional*) – Orbits having exposure time below this limit will be ignored. the default value is 30 seconds.

- **framecount_per_sec** (*float, optional*) – The framerate of the observation.

Warning: While combining events lists, do not mix data from RAS_VIS and RAS_NUV. In most cases, RAS_VIS data should be preferred.

Note: Please cite the Astroalign article if you are using this function.

Example

```
>>> import curvit
>>> from glob import glob
>>> file_path_list = glob('*/*/*/RAS_VIS/*/F*/*F1*ce*')
>>> curvit.combine_events_lists(file_path_list)
```

4.7 image_astrometry()

curvit.**image_astrometry**(*UV_image=None*, *threshold=3*, *API_key='ujmrvwqqyelxmzcj'*)

Carry out astrometry on a UVIT image using Astrometry.net.

Parameters

- **UV_image** (*file path*) – The name of the UVIT FITS image.
- **threshold** (*float, optional*) – The threshold parameter associated with the source detection method. The default value is 3.
- **API_key** (*string, optional*) – The Astrometry.net API key. Ideally, you should get your API key from Astrometry.net and use it.

Warning: Astrometry should be successful on most fields. However, failures found during tests on some crowded fields. Please try changing the source detection threshold in such cases.

Note: Please also cite Astrometry.net and Astroquery if you are using this function.

Example

```
>>> import curvit
>>> curvit.image_astrometry('test.fits')
```

CHAPTER

FIVE

REPORT BUGS AND CONTRIBUTE

Curvit development happens on https://github.com/prajwel/curvit/. If you find bugs or other mistakes, the best way to get them addressed is to report them on the GitHub issue tracker. If you also know how to fix the problem, please consider contributing!

CHAPTER

SIX

CHANGELOG

Important: If you have an older version, please upgrade:

```
pip install curvit --upgrade
```

6.1 1.7.2

The aperture radius is mentioned in the zoomed source images. Colour bars are added to the image plots. Matplotlib is allowed to use the default backend.

6.2 1.7.1

The input source coordinates are included in the *curve()* and *curve_orbitwise()* functions generated `source_*png` and `source_zoomed_*png` filenames.

6.3 1.7.0

A new parameter, `ZEF_correction_factor`, has been added to *curve()*, *makecurves()*, and *curve_orbitwise()* functions.

6.4 1.6.8

The default value of the *min_matches* parameter was changed to 4 from 1.

6.5 1.6.7

Replaced the Astroalign package with the Aafitrans package; this change only affects the *combine_events_lists()* function. *combine_events_lists()* function parameters `NUM_NEAREST_NEIGHBORS` and `MIN_MATCHES_FRACTION` have been removed. Three new parameters have been added to the *combine_events_lists()* function: `num_nearest_neighbors`, `min_matches`, and `pixel_tolerance`.

6.6 1.6.6

The events with having a `FrameCount` value of 1 are masked. This frame contains bad events.

6.7 1.6.5

Bug fix: Background subtraction was not taking place if `saturation_correction` was `True`.

6.8 1.6.4

A new parameter, `time_separation`, has been added to the *curve_orbitwise()* function.

6.9 1.6.3

The saturation correction technique has been made more accurate.

6.10 1.6.2

An internal function that estimates the weighted centroid of all events have been improved.

6.11 1.6.1

Bug fix: Corrected inaccurate MJD values for orbitwise data points in *curve_orbitwise()* function.

6.12 1.6.0

Added *curve_orbitwise()* function to create a light curve for a source, generating one data point per orbit. This function works best with the combined events list produced using *combine_events_lists()*.

6.13 1.5.9

Replaced the previous fixed image centre value of (2400, 2400) with a function call that estimates the weighted centroid of all events.

6.14 1.5.8

combine_events_lists() function can now be used to combine already combined events lists from multiple observation IDs.

6.15 1.5.7

Bug fix for the *combine_events_lists()* function. In the previous versions, the threshold value was not getting reset after automatic threshold modification.

6.16 1.5.6

Parameters `minimum_detections` and `maximum_detections` have been added to the *combine_events_lists()* function. The function will automatically modify the threshold to meet the minimum and maximum number of detections criteria.

6.17 1.5.5

The *combine_events_lists()* function has been updated with new default parameters. The function will automatically modify the threshold to limit the maximum number of detections to below 200.

6.18 1.5.4

Parameters `NUM_NEAREST_NEIGHBORS` and `MIN_MATCHES_FRACTION` have been added to the *combine_events_lists()* function.

6.19 1.5.3

The FITS file writing bug in the *image_astrometry()* function was fixed. This bug was reported by Vikrant Jadhav.

6.20 1.5.2

The `threshold` parameter has been added to the *combine_events_lists()* function.

6.21 1.5.1

The makefits() function has been renamed to *make_image()*.

6.22 1.5.0

Added *image_astrometry()* function to carry out astrometry on a UVIT image using Astrometry.net.

6.23 1.4.0

Added *combine_events_lists()* function to combine events lists from multiple orbits and create a single combined events list.

6.24 1.3.3

Curvit now reads and applies the "BAD FLAG" column from the official UVIT L2 pipeline (UL2P) generated events lists before processing the data. Note that the same "BAD FLAG" column generated by the *process_ccdlab()* function is only a placeholder.

6.25 1.3.2

Bug fix for the makefits() function.

6.26 1.3.1

Bugfix for background estimation. The scaling of the background counts to the source aperture was affected by a bug; this has been fixed.

6.27 1.2.5

The last release before the publication of Curvit software on JAA AstroSat special issue.

CHAPTER

SEVEN

MISCELLANEOUS

7.1 The UVIT filters in VIS, NUV and FUV channels

VIS Filter ID	Old filter name	New filter name	NUV Filter ID	Old filter name	New filter name	FUV Filter ID	Old filter name	New filter name
F1	VIS3	V461W	F1	Silica - 1	N242W	F1	CaF2 - 1	F148W
F2	VIS2	V391M	F2	NUVB15	N219M	F2	BaF2	F154W
F3	VIS1	V347M	F3	NUVB13	N245M	F3	Sapphire	F169M
F4	ND1	V435ND	F4	Grating		F4	Grating - 1	
F5	BK7	V420W	F5	NUVB4	N263M	F5	Silica	F172M
			F6	NUVN2	N279N	F6	Grating - 2	
			F7	Silica - 2	N242Wa	F7	CaF2 - 2	F148Wa

INDEX