

City Lights Cleaning and Visualization Code

Python software/Installation

Tested with Python 3.12.4

Scripts and libraries needed to run the process

city_lights.py (main script/driver code)

despeckle.py

destreak.py

model_bkgnd.py

Python libraries

matplotlib

numpy

netCDF4

datetime

os

glob

time

trapz (from scipy.integrate)

ma (from numpy.ma)

pandas

How to run the process :

Have all scripts in the same location and type "city_lights.py" on the command line.

If you use an IDE such as Spyder you can just run the main script from the IDE window.

Steps completed in code:

Set configuration flags to save the city data to files (True) or not (False).

Declare the variable names 'wvla' and 'wvlb' used to access the appropriate variables for VIS or UV (or another use-case of user's choice).

for VIS use wvla = 540 and wvlb = 740

for UV use wvla = 290 and wvlb = 490

Declare the values of frame clean up configuration parameters.

Define the directory path of where to retrieve all netcdf files for the desired

granules.

Note that this version of the code works with a directory where there is a single granule. The iteration over more granules is left up to the user to implement.

Find the NetCDF file(s) in the directory and its subdirectories.

Process one granule from the directory

Read the data from the radiance group

Find the hot pixels and bad quadrants and find the anomalously
oversubtracted quadrants

Start processing a frame; each mirror step is a frame

Compute mean of top of focal plane

Compute mean of bottom of focal plane

Identify bad quadrants where a quadrant is a spectral band in the top or
bottom of the focal plane

Note that when using the Median Absolute Difference (MAD) filter,
1.4826 is the ratio of sigma to the median absolute difference for a
normal distribution

Find persistently hot pixels using despeckle

Clean up the radiance by despeckle, destreak, and background
subtraction via model_bkgnd.

At this point we start the generation of a city lights map

Define the wavelength range for numerical integration

For the VIS case use wvl1 = 540 and wvl2 = 640

For the UV case use wvl1 = 390 to wvl2 = 490

Use the cleaned up radiance from previous processing and

read the nominal wavelengths from the input netCDF file.

Average the wavelength over the spatial samples.

The wavelengths outside of the range are masked and the start and end indices of
the wavelength array to be used in the numerical integration are identified.

Convert radiance to watts using J_per_photon which is Joules per photon

computed using Planck's constant and speed of light.

```
# Process one mirror step at a time
```

```
    # Apply a trapezoidal numerical integration over  
to each frame.
```

wavelengths

```
# Convert from watts to nanowatts
```

```
#Plot out a colormap of the city data variable
```

```
# Save the city data in a '.csv' file or a new NetCDF file if desired.
```

Currently using the same file name as the granule filename with '_output' appended to it.

Testing

This current version of the code has been tested with this granule:

TEMPO_RADT_L1_V03_20240328T103053Z_S001G01.nc.

The city image obtained with this granule is below.

