COSMILOG: A Cosmic Ray Logger

M. S.NIAEI[1,2], C. YEŞİLYAPRAK,[1,2]

[1]*Atatürk University Astrophysical Research and Application Center, Erzurum, Turkey;* m.shemuni@gmail.com

[2]*Atatürk University, Science Faculty, Astronomy and Astrophysics Department, Erzurum, Turkey*

Abstract. We describe the design and development of COSMILOG (Cosmıc Ray Logger), a pipeline that aims to be completely open-source, licensed under General Public License V3. COSMILOG can detect cosmic rays on a given data file (FITS/PNG-/JPG). For this purpose, COSMILOG uses, Python3, astropy, L.A. Cosmic (a rebuilt version). For a better user experience, COSMILOG also comes with COSMILOG-GUI which offers a mordern and practical interface. With this pipeline, cosmic ray counts can be driven from all kind of raw CCD data including night observation data, All SKy Camera, etc.

1. INTRODUCTION

Cosmic Rays (CR) are a form of high-energy radiation. Particles with high velocities hit the Earth's upper atmosphere and create a shower of other high-energy particles.CRs are Trouble-makers for modern astronomy science since we cannot stop them from hitting the CCD cameras. CRs once was a problem for computers too. They can hit a specific part of a Random Access Memory (RAM) and cause a bit to flip from 0 to 1 state and even may cause a crash. There are various ways to detect CRs. Detectors can drive count of the CRs and even the velocity or energy of each CR.
COSMILOG is trying to be the most cost-efficient way to observe the only number of CRs for a given site.

2. APPROACH

CRs can effect CCD data. This effect especially is more crucial for spectroscopy. For overcoming these problems mathematical methods (Edge detection) for Cosmic Ray removals developed and are used to clean CCD data. As it is obvious to clean data from CRs we mush first detect them. With using these methods on scientific data obtained from an observatory, CR count can easily be found.
One of the best devices to drive CR count could be an All Sky Camera (ASC). Since ASCs uses a fish-eye lens a good part of the CCD won't be exposed to light and must only return Dark level. Any spikes in this section of an ASC data most probably are caused by CRs. COSMILOG can create circular or polygonal masks to avoid CR count-

ing from any part of data. This can be used to drive CR count from only previously explained, not exposed part of the data.

3. COSMILOG & COSMILOG-GUI

COSMILOG[1] uses two different methods to count CRs.

- L.A.Cosmic (van Dokkum et al. 2012)
- Median Filtering(Craig et al. 2017)

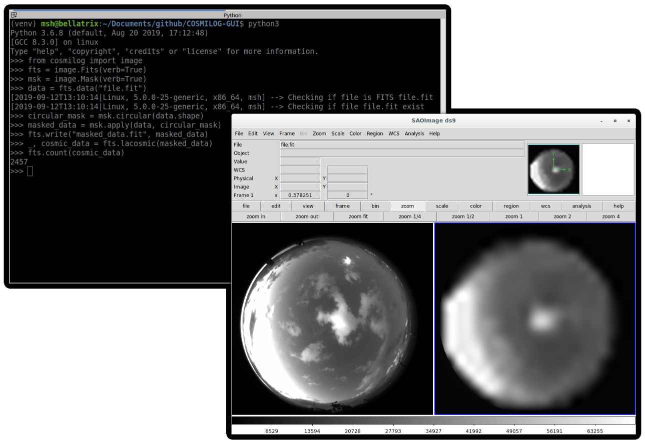
For this purpose, COSMILOG uses Python3 and some other modules. COSMILOG-GUI is a QT graphical user interface created for better user experience.

Module	Origin
ginga	COSMILOG-GUI
ccdproc	COSMILOG
numpy	COSMILOG
PIL	COSMILOG
sep	COSMILOG
matplotlib	COSMILOG-GUI
pyqt5	COSMILOG-GUI
astropy	COSMILOG

Tbl1: Table of dependencies of COSMILOG and COSMILOG-GUI

COSMILOG can be used either as a Python3 module or a piece of software.

4. VISUALS



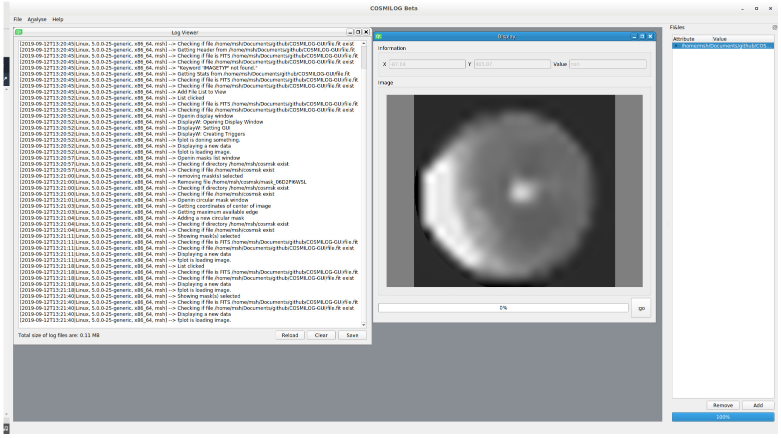
```
(venv) msh@bellatrix:~/Documents/github/COSMILOG-GUI$ python3
Python 3.6.8 (default, Aug 20 2019, 17:12:48)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from cosmilog import image
>>> fts = image.Fits(verb=True)
>>> msk = image.Mask(verb=True)
>>> data = fts.data("file.fit")
[2019-09-12T13:10:14|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file is FITS file.fit
[2019-09-12T13:10:14|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file file.fit exist
>>> circular_mask = msk.circular(data.shape)
>>> masked_data = msk.apply(data, circular_mask)
>>> fts.write("masked_data.fit", masked_data)
>>> _, cosmic_data = fts.lacosmic(masked_data)
>>> fts.count(cosmic_data)
2457
>>> 
```

File file.fit
Frame 1 x 0.378251 0 °

Python
SAOImage ds9
File Edit View Frame Bin Zoom Scale Color Region WCS Analysis Help
Object
Value
WCS
Physical X Y
Image X Y
file edit view frame bin zoom scale color region wcs analysis help
zoom in zoom out zoom fit zoom 1/4 zoom 1/2 zoom 1 zoom 2 zoom 4
Y
X
6529 13594 20728 27793 34927 41992 49057 56191 63255

Fig1: COSMILOG module in use. DS9 shows the data. DS9 left frame: Original ASC data. DS9 right frame: masked ASC data.

[1] https://github.com/mshemuni/COSMILOG



```
[2019-09-12T13:20:45|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/Documents/github/COSMILOG-GUI/file.fit exist
[2019-09-12T13:20:45|Linux, 5.0.0-25-generic, x86_64, msh] --> Getting Header from /home/msh/Documents/github/COSMILOG-GUI/file.fit
[2019-09-12T13:20:45|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file is FITS /home/msh/Documents/github/COSMILOG-GUI/file.fit
[2019-09-12T13:20:45|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/Documents/github/COSMILOG-GUI/file.fit exist
[2019-09-12T13:20:45|Linux, 5.0.0-25-generic, x86_64, msh] --> "Keyword 'IMAGETYP' not found."
[2019-09-12T13:20:45|Linux, 5.0.0-25-generic, x86_64, msh] --> Getting Stats from /home/msh/Documents/github/COSMILOG-GUI/file.fit
[2019-09-12T13:20:45|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file is FITS /home/msh/Documents/github/COSMILOG-GUI/file.fit
[2019-09-12T13:20:45|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/Documents/github/COSMILOG-GUI/file.fit exist
[2019-09-12T13:20:45|Linux, 5.0.0-25-generic, x86_64, msh] --> Add File List to View
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> List clicked
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file is FITS /home/msh/Documents/github/COSMILOG-GUI/file.fit
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/Documents/github/COSMILOG-GUI/file.fit exist
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> Openin display window
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> DisplayW: Opening Display Window
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> DisplayW: Setting GUI
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> DisplayW: Creating Triggers
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> fplot is doning something.
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> Displaying a new data
[2019-09-12T13:20:52|Linux, 5.0.0-25-generic, x86_64, msh] --> fplot is loading image.
[2019-09-12T13:20:57|Linux, 5.0.0-25-generic, x86_64, msh] --> Openin masks list window
[2019-09-12T13:20:57|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if directory /home/msh/cosmsk exist
[2019-09-12T13:20:57|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/cosmsk exist
[2019-09-12T13:21:00|Linux, 5.0.0-25-generic, x86_64, msh] --> removing mask(s) selected
[2019-09-12T13:21:00|Linux, 5.0.0-25-generic, x86_64, msh] --> Removing file /home/msh/cosmsk/mask_06D2PI6WSL
[2019-09-12T13:21:00|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if directory /home/msh/cosmsk exist
[2019-09-12T13:21:00|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/cosmsk exist
[2019-09-12T13:21:01|Linux, 5.0.0-25-generic, x86_64, msh] --> Openin circular mask window
[2019-09-12T13:21:03|Linux, 5.0.0-25-generic, x86_64, msh] --> Getting coordinates of center of image
[2019-09-12T13:21:03|Linux, 5.0.0-25-generic, x86_64, msh] --> Getting maximum available edge
[2019-09-12T13:21:04|Linux, 5.0.0-25-generic, x86_64, msh] --> Adding a new circular mask
[2019-09-12T13:21:04|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if directory /home/msh/cosmsk exist
[2019-09-12T13:21:04|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/cosmsk exist
[2019-09-12T13:21:11|Linux, 5.0.0-25-generic, x86_64, msh] --> Showing mask(s) selected
[2019-09-12T13:21:11|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file is FITS /home/msh/Documents/github/COSMILOG-GUI/file.fit
[2019-09-12T13:21:11|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/Documents/github/COSMILOG-GUI/file.fit exist
[2019-09-12T13:21:11|Linux, 5.0.0-25-generic, x86_64, msh] --> Displaying a new data
[2019-09-12T13:21:11|Linux, 5.0.0-25-generic, x86_64, msh] --> fplot is loading image.
[2019-09-12T13:21:18|Linux, 5.0.0-25-generic, x86_64, msh] --> List clicked
[2019-09-12T13:21:18|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file is FITS /home/msh/Documents/github/COSMILOG-GUI/file.fit
[2019-09-12T13:21:18|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/Documents/github/COSMILOG-GUI/file.fit exist
[2019-09-12T13:21:18|Linux, 5.0.0-25-generic, x86_64, msh] --> Displaying a new data
[2019-09-12T13:21:18|Linux, 5.0.0-25-generic, x86_64, msh] --> fplot is loading image.
[2019-09-12T13:21:40|Linux, 5.0.0-25-generic, x86_64, msh] --> Showing mask(s) selected
[2019-09-12T13:21:40|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file is FITS /home/msh/Documents/github/COSMILOG-GUI/file.fit
[2019-09-12T13:21:40|Linux, 5.0.0-25-generic, x86_64, msh] --> Checking if file /home/msh/Documents/github/COSMILOG-GUI/file.fit exist
[2019-09-12T13:21:40|Linux, 5.0.0-25-generic, x86_64, msh] --> Displaying a new data
[2019-09-12T13:21:40|Linux, 5.0.0-25-generic, x86_64, msh] --> fplot is loading image.
```

Total size of log files are: 0.11 MB

X -87.64 Y 465.07 Value nan

/home/msh/Documents/github/COS...

Fig2: Same process as in Fig1 done by GUI using COSMILOG-GUI

5. CODE

COSMILOG is a modular and Object-oriented piece of code.

```
def data(self, file):
    """Reads FITS files and returns numpy array as data"""
    try:
        if self.check(file):
            data = fts.getdata(file).astype(f64)
            if data.ndim == 2:
                return(data)
            elif data.ndim > 2:
                return(data[self.def_layer])
    except Exception as e:
        self.logger.log(e)
```

One of methods from one of modules created for COSMILOG

COSMILOG-GUI adds COSMILOG module to some graphical elements to achieve its functionality.

6. RESULTS

Fig3 shows CR count for 10 days and as seen JD2458445 has a rises on CR count. This might belong to a supernova happened at the day in question. The sunset and sunrise effects can be seen at the beginning and end of each day's graph. Please notice data was taken from DAG site in Turkey with the timezone of GMT+3.

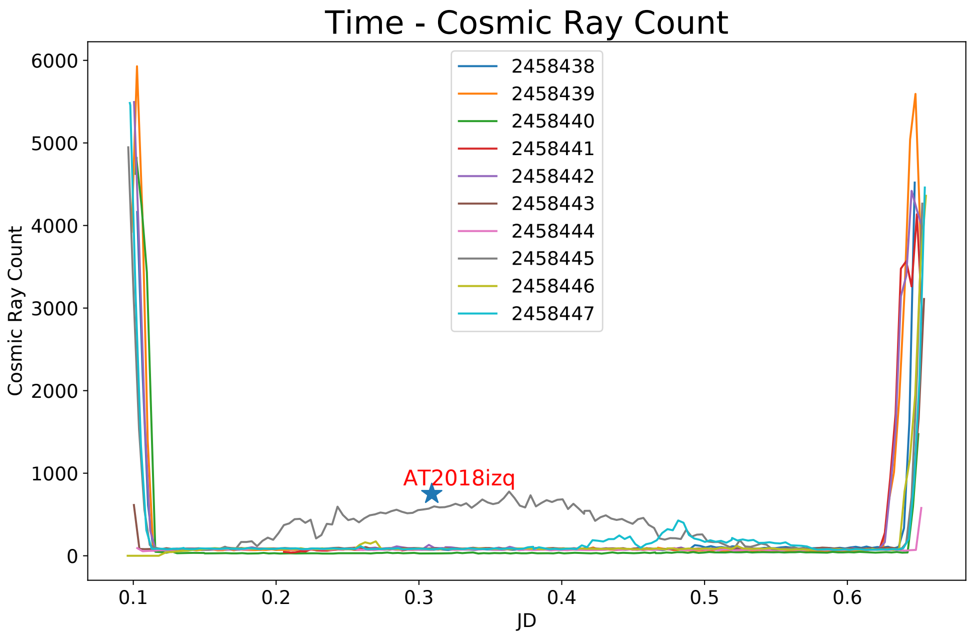


Fig3: CR count of 10 continues days obtained from an Alcor-System All Sky Camera. Blue star shows the time of the Supernova "AT2018izq"

7. CONCLUSION

This method can be used to count CR belongs to a certain site and as seen in Fig3 is responsive to CRs from the sun or other CR sources.
The downside of this method is it cannot determine the direction of the incident. If we could process data from different sites over the globe, we might be able to roughly determine the direction of incidence.

Acknowledgments. This study is supported by DAG Project (Project ID: 2011K120230), Atatürk Universtiy and ATASAM. Authors are grateful to DAG Project, Atatürk Universtiy and ATASAM for continuous and valuable supports.

References

Craig, M., Crawford, S., Seifert, M., Robitaille, T., Sipőcz, B., Walawender, J., Vinícius, Z., Ninan, J. P., Droettboom, M., Youn, J., Tollerud, E., Bray, E., Walker, N., Janga, V. R., Stotts, C., Günther, H. M., Rol, E., Bach, Y. P., Bradley, L., Deil, C., Price-Whelan, A., Barbary, K., Horton, A., Schoenell, W., Heidt, N., Gasdia, F., Nelson, S., & Streicher, O. 2017, astropy/ccdproc: v1.3.0.post1. URL `https://doi.org/10.5281/zenodo.1069648`

van Dokkum, P. G., Bloom, J., & Tewes, M. 2012, L.A.Cosmic: Laplacian Cosmic Ray Identification, Astrophysics Source Code Library. `1207.005`