AllSkEye Network Research

Joseph Cohen jsc6121@terpmail.umd.edu
Aerospace Engineering Science, Discovery and the Universe

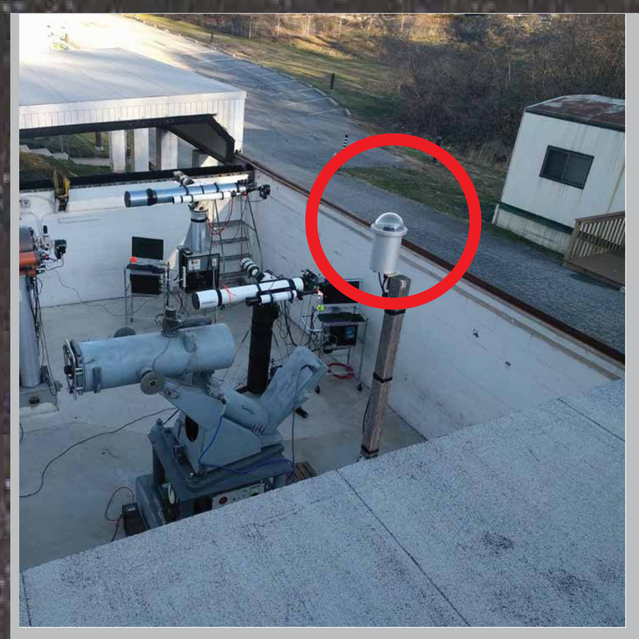
Introduction

Goal

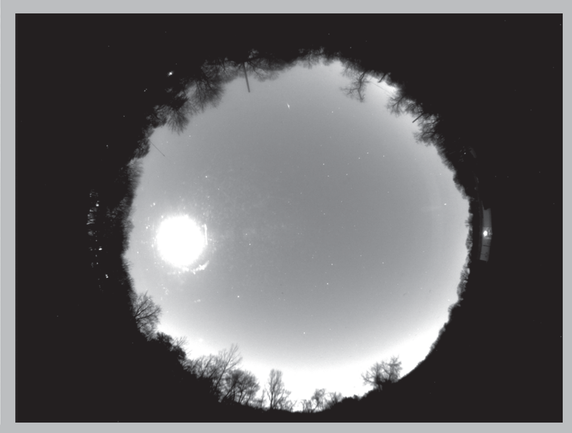
Throughout my research, my goal has been to create a program that takes hundreds of pictures from an All-Sky Camera (pictured to the right) and analyzes these pictures to detect if there were any meteors, which leave streaks on the images, during the night. In the future we hope to be able to autonomously go through all of these steps and upload successful detections to the UMD Observatory [1] website.

Technical

- GitHub [2] - used for group coding stored on the cloud
- Python - coding language for both analysis and pipeline software
- AllSkEye [3] - software that controls pictures being taken each night
  - Exposure - 30-60 seconds
  - Meteors are usually only visible for around 2 seconds
  - Around 1000 pictures taken each night



Photos courtesy of Elizabeth Warner/ UMD Observatory



Pictured above and to the left is the UMD All-Sky Camera. The difference between a normal camera and this camera is that an All-Sky Camera can take 180 degree pictures, like the picture above and to the right.

Process

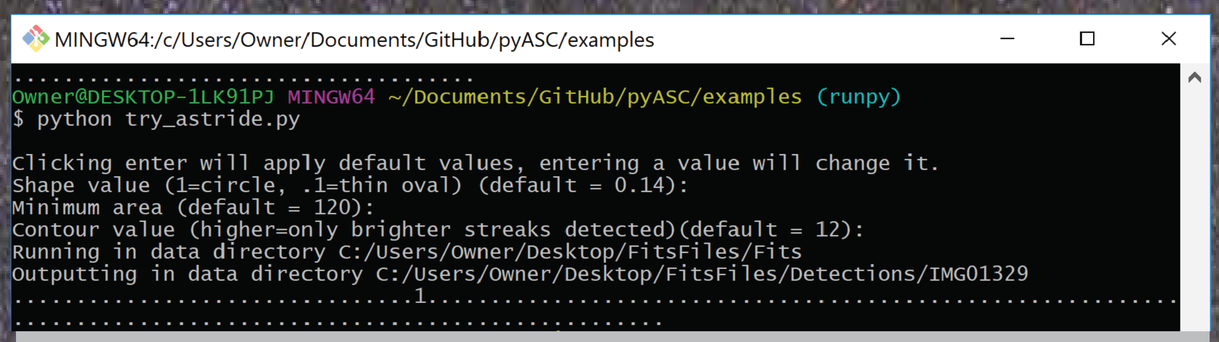
Program

The astride [4] program detects “streaks” which are the result of long exposure pictures constantly being taken throughout the night. To detect a meteor that could potentially be in one of the thousands of different pictures we had to create a program which would go through each picture and on each one run the astride algorithm (created by Dae-Won Kim).

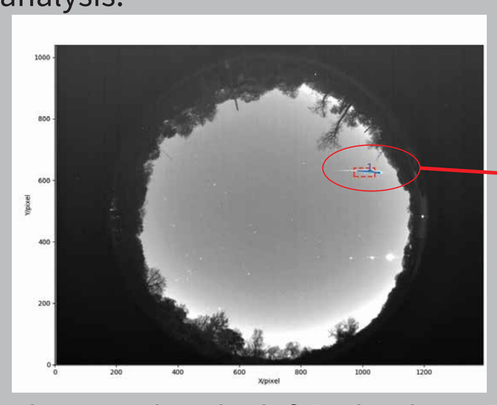
Detection

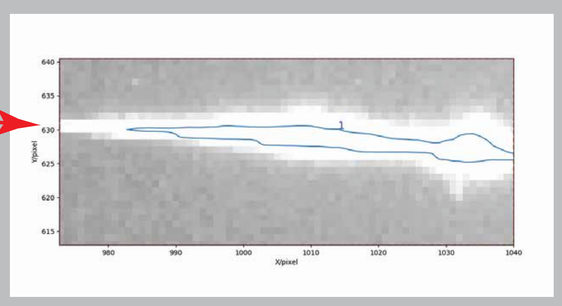
Currently, there are three different customizable parameters which affect how the program searches for meteors during its detection process.

- *Shape value* (Range: .01 -> 1): this value allows the user to pinpoint how oval/circular a meteor streak should be.
- *Minimum area* (Range: 7->1000): this value lets users define the minimum pixel count a streak could be to be classified as a streak.
- *Contour value* (Range: 1-> 100): this value allows the user to define how bright the meteor should be, this is useful because of varying exposure

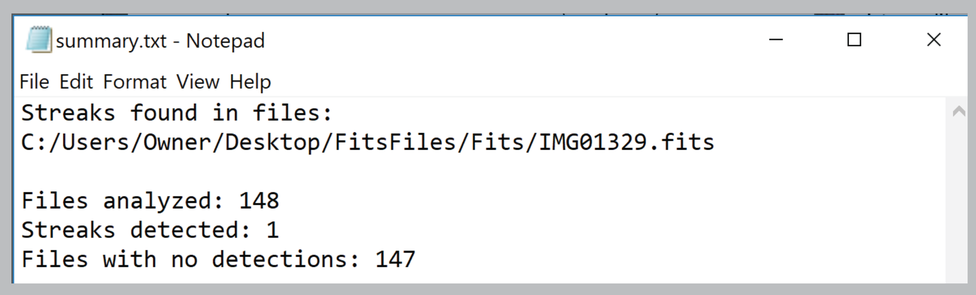


This is an example of the program being run [5]. As shown above, the program has numerous different parameters that can be changed to customize the analysis.





Above and to the left is the detection view where the program points out where in the sky the meteor was detected and to it’s right is a close up of the detected meteor.



Once the program has successfully analyzed all of the fits (picture) files, a summary is automatically created providing the user with useful information about the detections.

Conclusion

While the astride algorithm works very well, it does not run without its flaws. Currently, the algorithm picks up on most meteors in images but has a tendency to detect false positives, for example, detecting trees and planes as meteors. In the future, after detecting all the meteors from the previous night the program will automatically upload all detections to the UMD Observatory’s website.

References

[1] https://www.facebook.com/UMObservatory
[2] https://medium.com/@abhishekj/an-intro-to-git-and-github-1a0e2c7e3a2f
[3] www.sxccd.com
[4] github.com/dwkim78/ASTRiDE
[5] GitHub/pyASC

Thank you so much to Elizabeth Warner and Peter Teuben for giving me this amazing opportunity to learn and work alongside you both for the past year