Satellite Detection in ACS/HST Images

David Borncamp, Pey Lian Lim

Space Telescope Science Institute, Baltimore MD

ADASS XXVI

STScI

Introduction

Satellites are a problematic transient event that occurs in astronomical images which will need to be identified and taken care of before any data can be properly used for science. However many people with small programs or doing archival research may not have the resources to manually inspect every images they are using. This necessitates automation of the detection and masking. Though it can be applied to any image, here we show the process by which satellite trails can be identified and properly masked in Hubble Space Telescope (HST) Advanced Camera for Surveys (ACS) images with a very high completeness and low false positive rate. We also provide the Python source code as a stand-alone package to be used by users (Borncamp & Lim, 2016).

Requirements for Software and Detection

Program: Since most ACS users are used to dealing with calibrated, flat-fielded images provided by the MAST OPUS pipe (FLC/FLT images). The code must be able to run on a individual chips of ACS/WFC images. This also means that the code could be useable by other instruments.

Detection: We assume that the satellite trail will cross from one edge of the detector to another. It is possible that resolved sources will span from one side of the image to another, the detection must not view resolved sources or dense star fields as a satellite trail.

Masking: Satellites crossing the field of view may be resolved themselves so the generated mask must extend far enough that the wings of the trail are completely masked. Since we will be working on images containing extreme distortion and observing geometry could make for a curved trail, the mask must be able to curve with the trail.

Analysis of Detections

We tested the algorithm to see if it will flag real astronomical sources where it should not and tested completeness using a truth set. We used datasets from the Orion Nebula, 47 Tuc and the entire first 4 clusters of the Frontier Fields program (HST program ID 14037 PI: Lotz) which shows the algorithm correctly flags distinct lines that cross the images and ignores most real astronomical features (Borncamp & Lim, 2016). Table 1 shows an analysis of the results from testing the truth set.

Dataset	# ACS Chips in Dataset	# Manually Flagged	# Flaggged by Software	# False Positive	# Not Flagged by Software With Satellite Trail
abell2744	138	6	7	1	0
abell2744 – par	177	8	11	3	0
macs0416	180	9	11	5	1
macs0416 – par	140	6	7	1	0
macs0717	161	7	9	3	1
Total	796	36	45	13	1
% of Total	100%	4.52%	5.65%	1.63%	4.44%

Table 1. Table showing completeness and false positive rate of satellite detection algorithm compared to the HFF truth set using only the default values. The algorithm must be run twice on each image, once for each chip, a detection in one chip counts as a detection in the entire image.

Detecting a Satellite

Detection starts by rescaling the image and creating an edge image using a Canny algorithm (Canny,1986) as shown in figure 1. Then a Probabilistic Hough Transform (Galamhos et al, 1999) whose results are also shown in Figure 1. These steps are completed using the implementations in the Python package *skimage* (Van der Walt et al. 2014). The end points of the transform are tested to see if anything crossed the image.

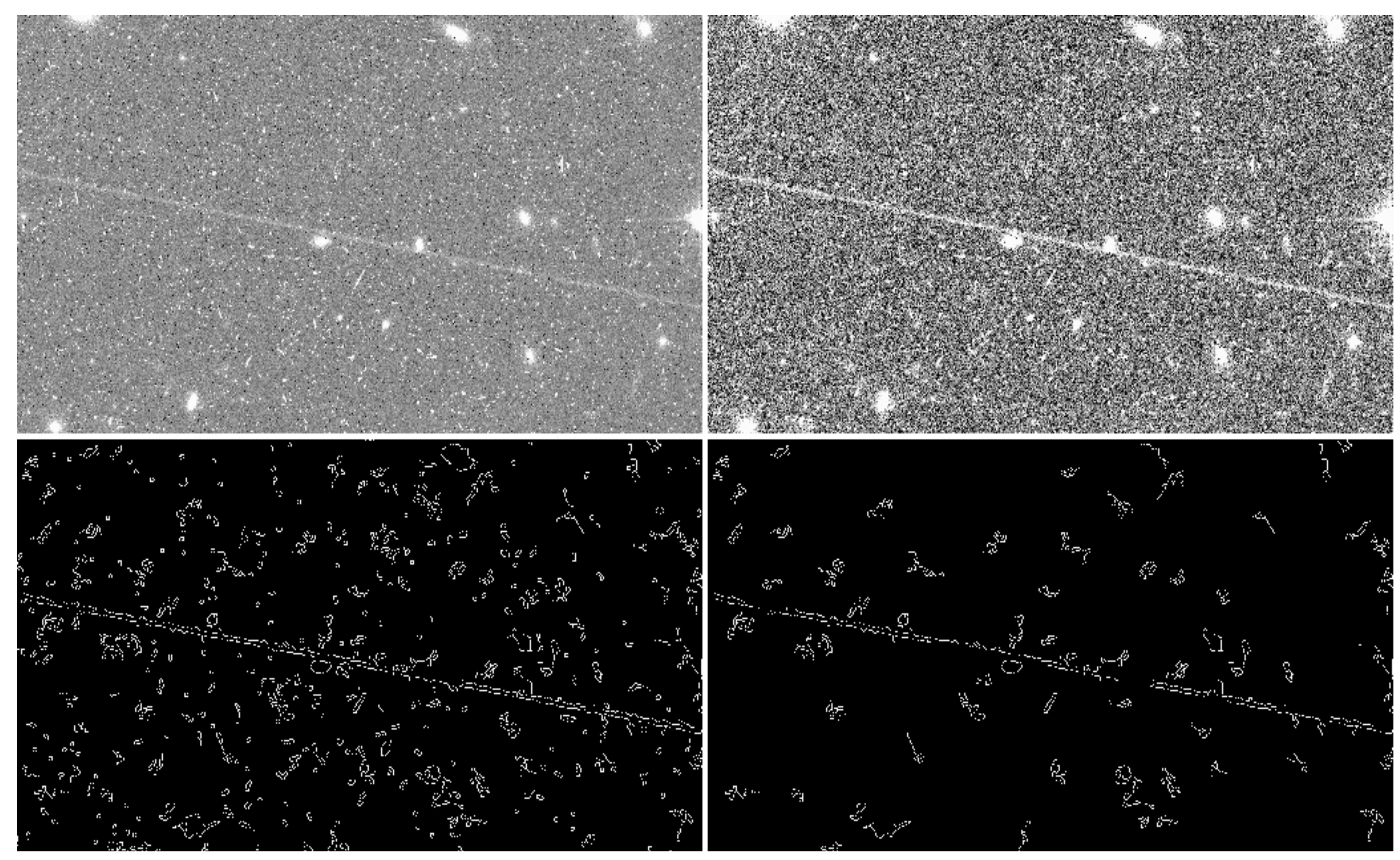
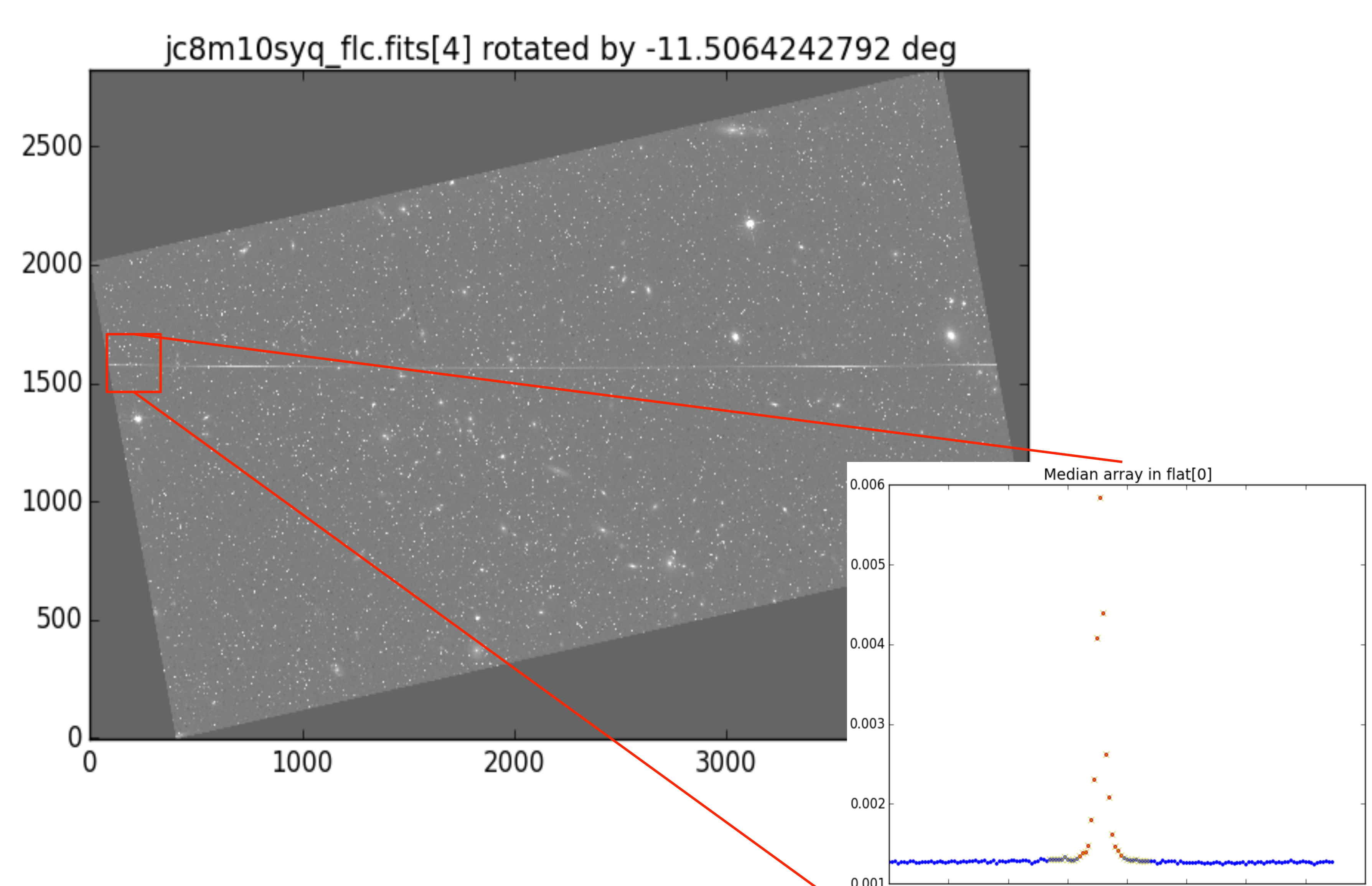
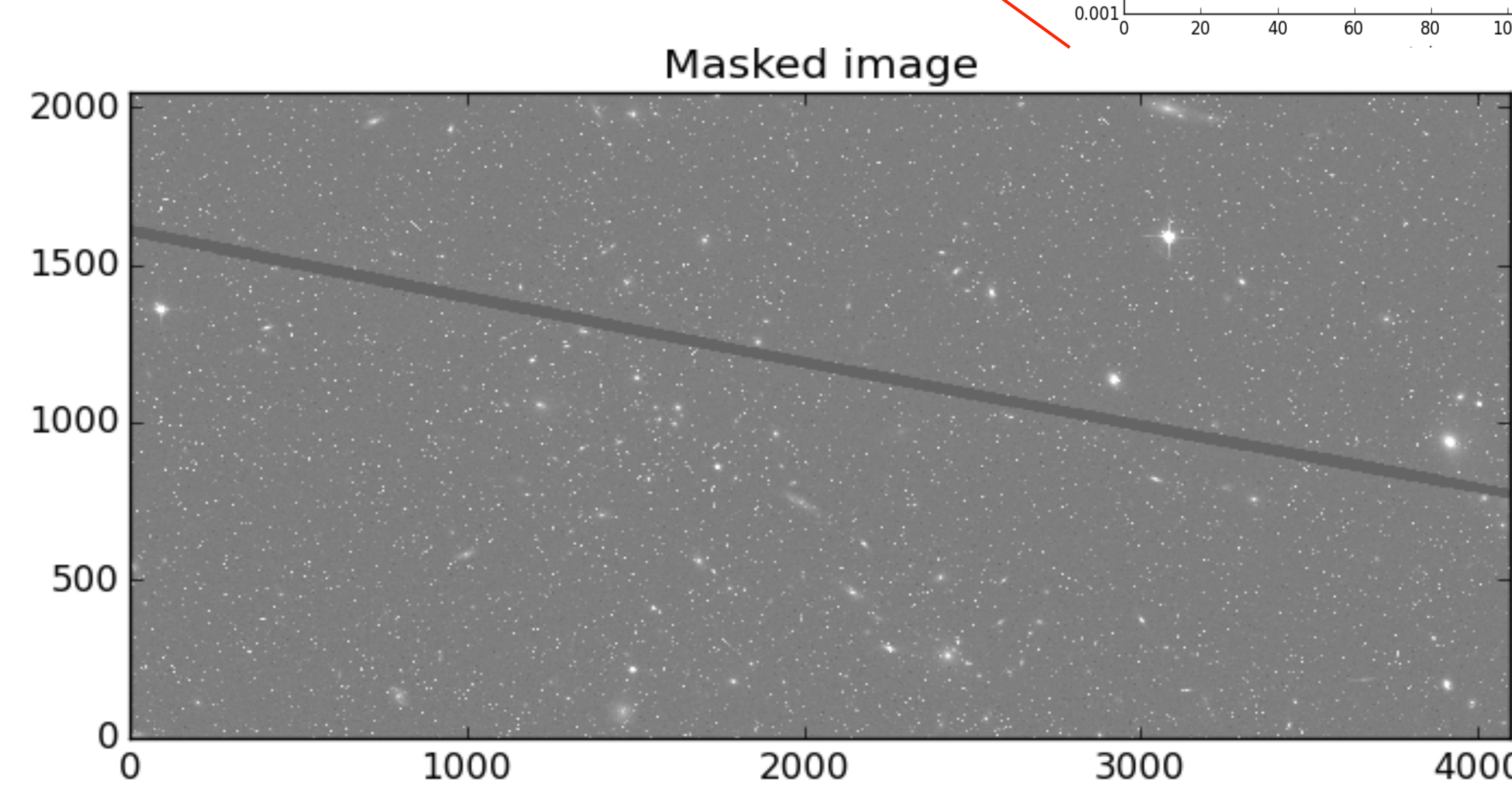


Figure 1. Satellite trail in jc8n81r7q_flc.fits. Top Left: Original FLC image. Top Right: Rescaled image of the same location. The satellite trail in the rescaled image is much clearer which makes edge detection much easier. Bottom Left: Edge detection on rescaled image. Bottom Right: Cleaned up edge image with small objects removed. The trail of the satellite is preserved and complete in the cleaned up edge image. Taken from Borncamp & lim 2016.

Masking

The image is first rotated and a sub-region around the trail is used to fit the trail. The trail is then iteratively fit across its length which allows it to vary as it fits the trail.





References

- D Borncamp & P-L Lim, ACS ISR 16-01(Baltimore:STScI)
- Canny, J., A Computational Approach to Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986
- C. Galamhos, J. Matas and J. Kittler, "Progressive probabilistic Hough transform for line detection", in IEEE Comp
- Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu and the scikit-image contributors. scikit-image: Image processing in Python. PeerJ 2:e453 (2014) http://dx.doi.org/10.7717/ peerj.453