Second Update on Polygon Detection Algorithm

Andrew Ho
Aerospace Robotics and Control Laboratory
California Institute of Technology

September 19, 2018

1 Introduction

The goal of the image processing stage is to extract line segments which correspond to true edges or at least to edges available in the spacecraft model. In combination with Polygon tracking, the project aims to find a robust method of mapping polygons.

2 Pipeline

We use the following pipeline to detect line segments within an image, in order to detect larger polygons:

1. Low Pass Filtering
2. Thresholding
3. Edge Detection
4. Corner Detection
5. Convex Hull
6. Further Corner Filtering

7. Polygon Tracking

Low Pass Filtering: This step reduces contrast within image, effectively blurring out less important edges. This allows for higher accuracy when we apply edge detection algorithms.

Thresholding: This step strips the image down to strong features, removing external noise in an image.

Edge Detection: This step implements two different edge detection algorithms. First, Probabilistic Hough Line Transform is applied, extracting an array of important segments. Then the Line Segment Detector algorithm outlined in:

Rafael Grompone von Gioi, Jrmie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: a line segment detector. 2012.

is used to thin out the edges detected from Hough Line Transform.

Corner Detection: This step implements the Shi-Tomasi method of detecting corners. We apply this algorithm to the stripped image from Edge Detection. We set the max number of corners to 8, casting a wide net in case the true corners to detect are not the strongest corners in the image.

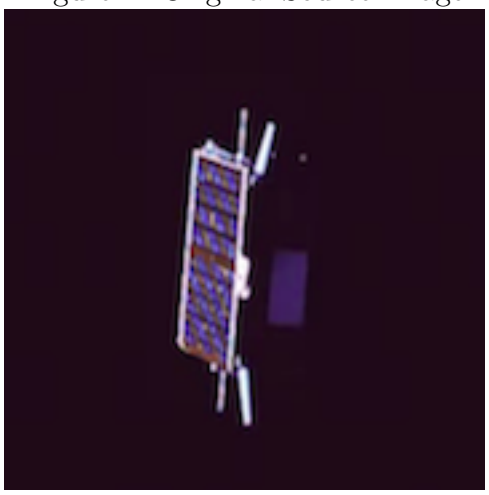
Convex Hull: This step takes the corners detected in the Corner Detection step and finds the convex hull of outer most points, guaranteeing that we preserve true corners.

Further Corner Filtering: This step takes the points of the Convex Hull and removes collinear points, leaving only true corners. From these true corners we draw edges to outline the polygon.

3 Example Process

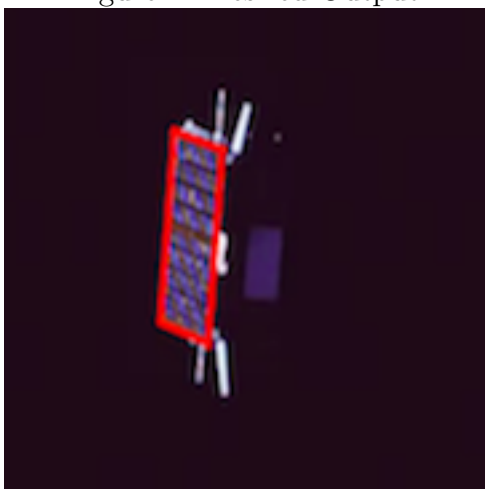
Consider the following image:

Figure 1: Original Source Image



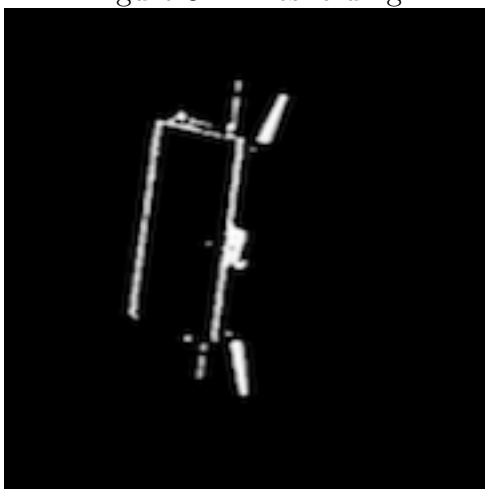
To the human eye, the white rectangle is quite evident. However, there are certain issues for a computer. The bottom length of the rectangle is missing and the antennae of the satellites may prove to turn up as false corners.
We would like to receive a polygon from the image, as shown in the following image:

Figure 2: Desired Output



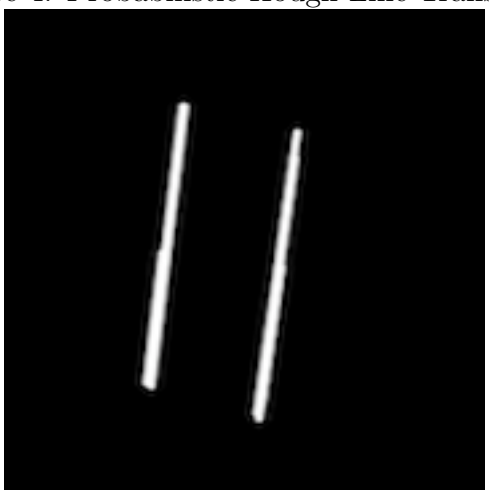
First, we apply thresholding:

Figure 3: Thresholding



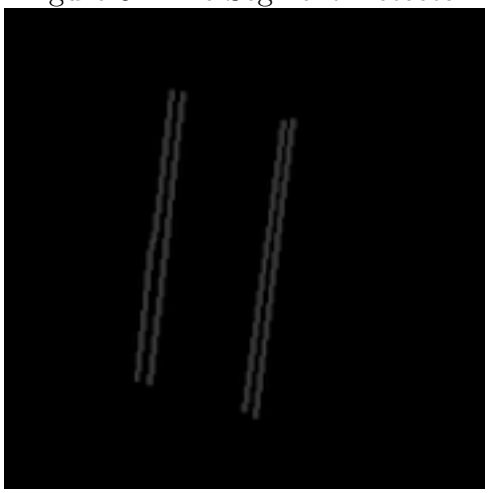
Second, we apply Probabilistic Hough Line Transform:

Figure 4: Probabilistic Hough Line Transform



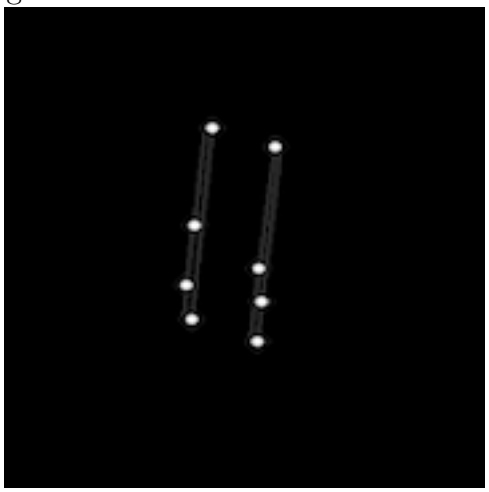
To this image, we apply the Line Segment Detector algorithm from Grompone, Jakubowicz, Morel, and Randall:

Figure 5: Line Segment Detector



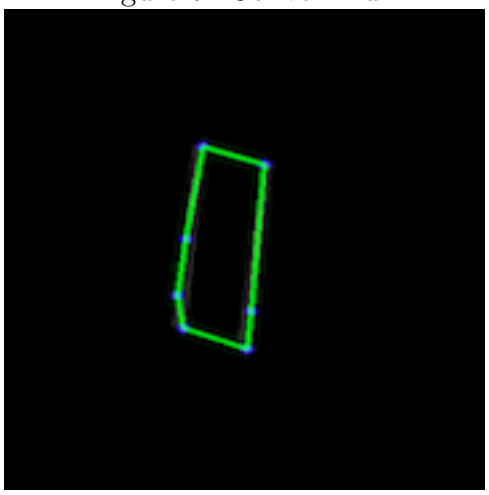
Next, we use the Shi-Tomasi algorithm to detect important corners:

Figure 6: Shi-Tomasi Corner Detection



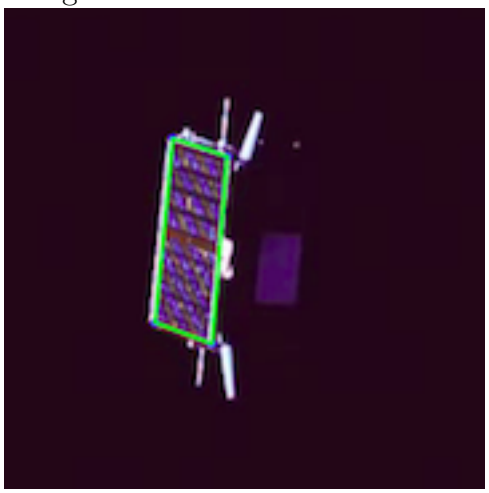
Notice we have over-detected corners. This is to ensure we capture true corners. Next, we find the convex hull:

Figure 7: Convex Hull

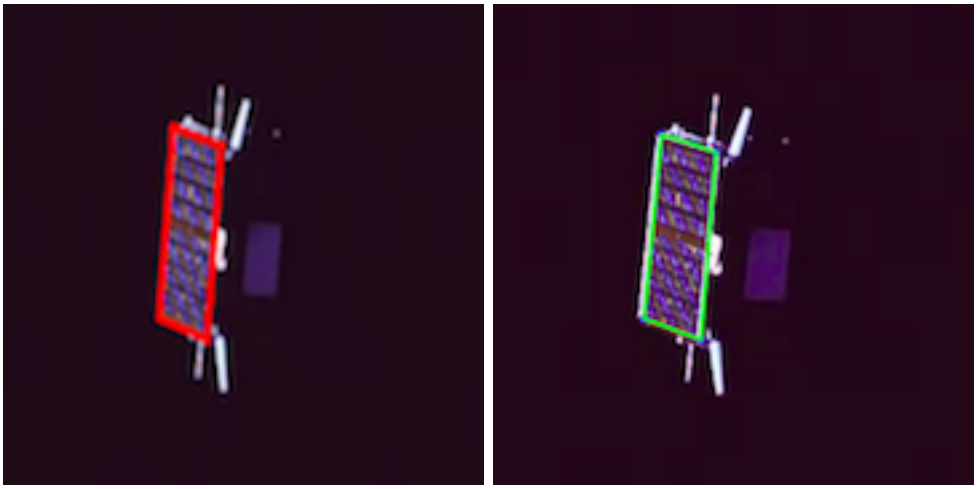


From the convex hull, we remove collinear lines, leaving us with a simple polygon:

Figure 8: Filtered Convex Hull

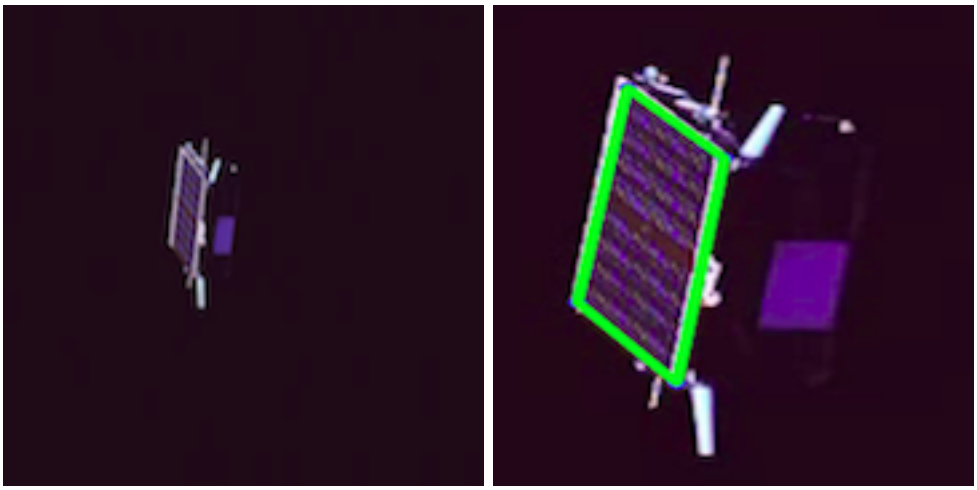


Compared to our target polygon, the algorithm does a tremendous job of finding the correct polygon:

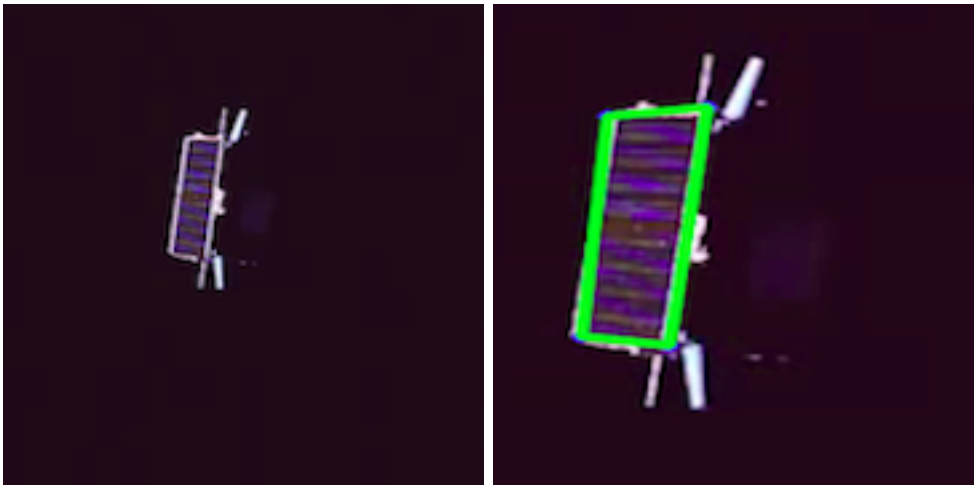


Below are additional examples compared side by side:

Example 1



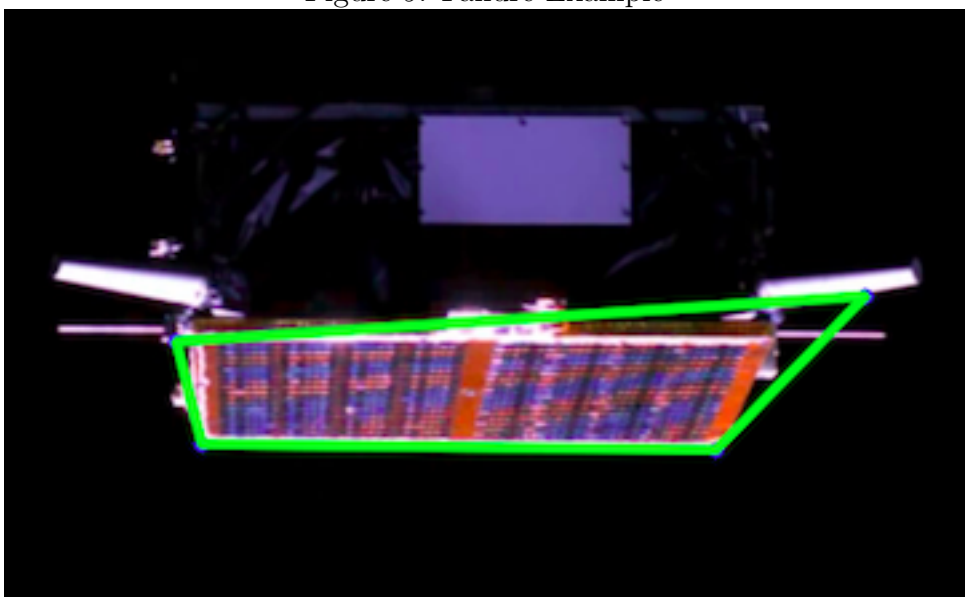
Example 2



4 Next Steps and Challenges

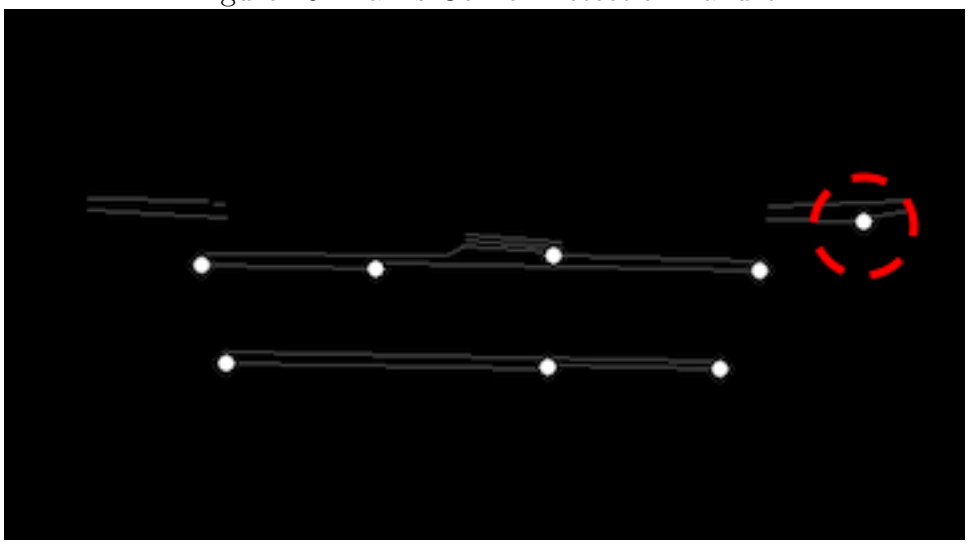
There are certain cases in which the antennae provides too strong of a corner. Consider the following image:

Figure 9: Failure Example



From the Harris Corner Detection step output, it's clear to see that there is an extra corner detected that skews the results

Figure 10: Harris Corner Detection Failure



To amend this issue, we can instead implement a Traveling Salesman algorithm to find include every point in the outer layer, and detect whether or not corners detected are legitimate by comparing whether pixels between the points are black or white. If there are not enough pixels between two corners that are white, then that corner must not exist.