Grad Project for ENSC 895

Jordan Lui March 25, 2016

Goal

Recognize buildings and structures in regular satellite images.

In this research I will examine a variety of methods to detect buildings in satellite images. I will focus on detecting edges and corners of 4-sided rectangular buildings in colour images, and will build an algorithm that can recognize buildings in an image automatically and count the numbers of buildings.

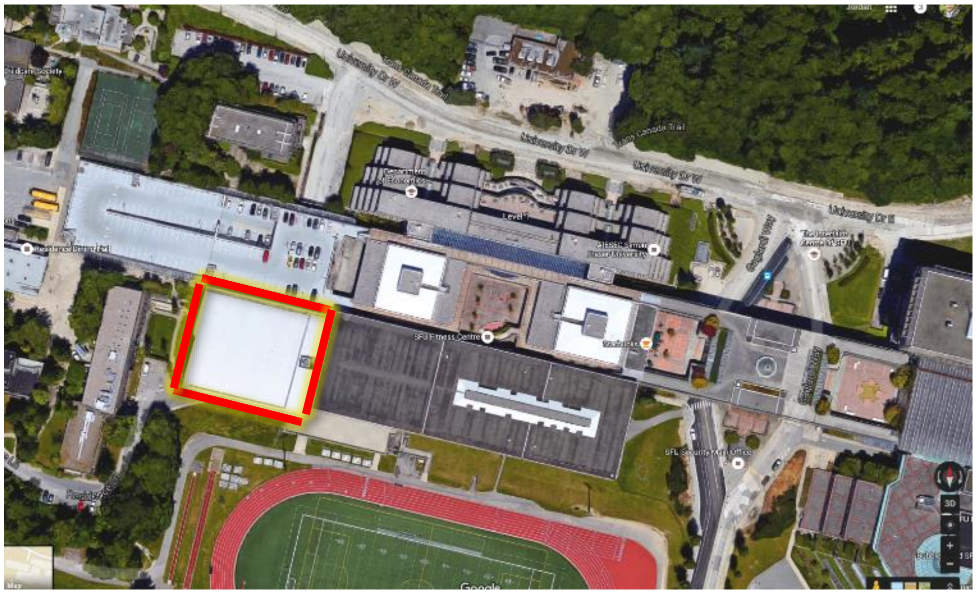


Figure 1 – An example of building detection where buildings are tightly clustered, and trees or shadows could obscure building detection

Source of Images

Images will be obtained from public domain sources, such as Google Maps.

Potential Applications

Work like this would have immediate applications in automated processing of satellite images of Earth. This could applications in intelligence and military industries, as well as applications in urban planning and anthropological studies, where the presence and distribution of human buildings is studied.

The concept of recognizing distinct, smaller entities within a larger image could have broader applications in medical imaging, space exploration, and climate tracking, if the algorithm is tuned to recognize certain types of shapes.

Success Criteria

Several criteria could be used to evaluate the efficiency of the algorithm

- True positives – portion of rectangular buildings successfully detected
- False negative – portion of buildings that algorithm fails to detect
- False positives – algorithm detecting a building when no building is present
- Autonomy – How much or how little human interaction (initial instruction, training, etc.) does the algorithm need to succeed?

Solution Structure

A brief breakdown of the solution structure is described below

Image pre filtering

Image is pre-processed with Gaussian smoothing to remove some image noise.

Burns Edge Detection

Burns edge detection is run to look for gradients in image that indicate edges. This algorithm sorts the lines into "gradient bins" based on the angle of the line. The algorithm then identifies the most dominant lines based on size

Compute Line regions

Line regions are constructed into lines based on a least squares fit of a line of all points in that region

Link lines that are similar or close

Lines are then compared and linked if they are sufficiently close in lateral distance, angle, overlap, and underlap

Line intersection detection

Resulting line segments are then compared to extract all possible line intersections. Note that a tolerance is required here since some line segments do not completely overlap at the building corner point.

Identify building corners

From these line intersections, building corners are determined as those which intersect at an angle close to 90 degrees. Note this 90 degrees tolerance can change drastically based on the acquisition angle of the imaging satellite. The result of this stage is a series of x,y points where a building corner has been identified, as well as information on the two walls that intersect it.

Building Hypothesis

Algorithm looks at all possible building corners and identifies sets which close a loop.

Results

Results with simplistic images

The function was able to detect most building edges and corners in an artificial image containing rectangles placed arbitrarily on a monochromatic background. The small green dots indicate any intersections of line segments detected, and the blue x denotes intersections that are potential buildings corners.

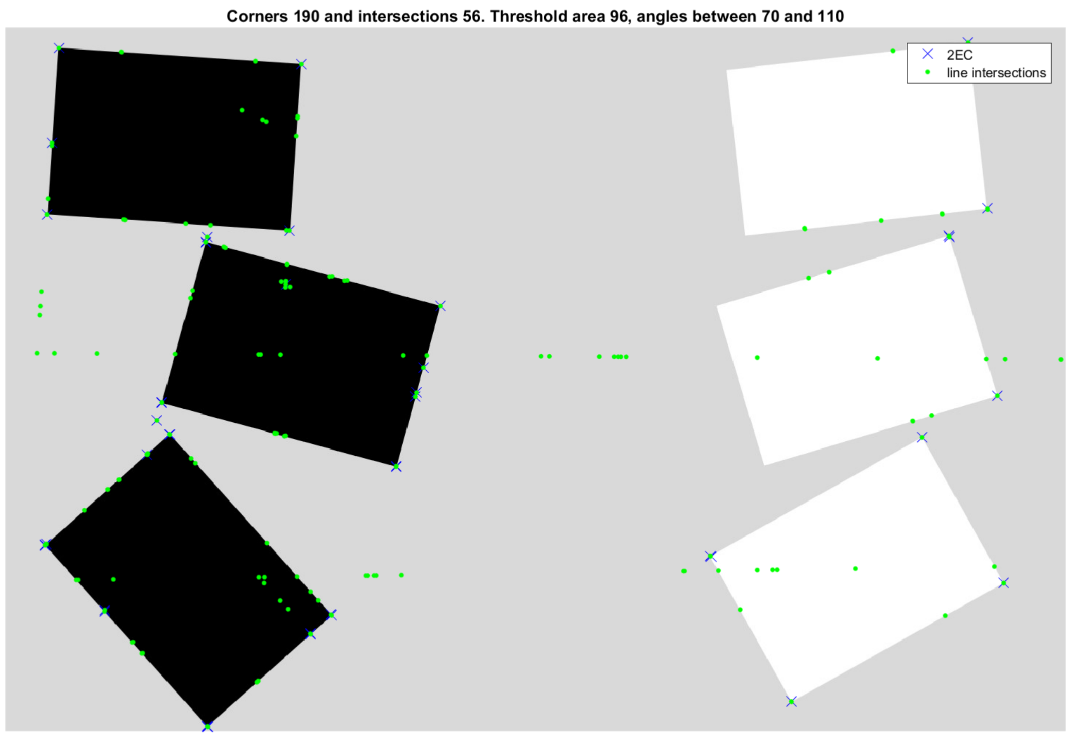


Figure 2 Edge intersection and building corners

Shown below are some lines detected in the image, as well as the potential building corners. Note that the algorithm currently detects most building corners but continues to fail to detect some of these corners. Tuning of various algorithm parameters such as minimum line length could increase the sensitivity of the solution.

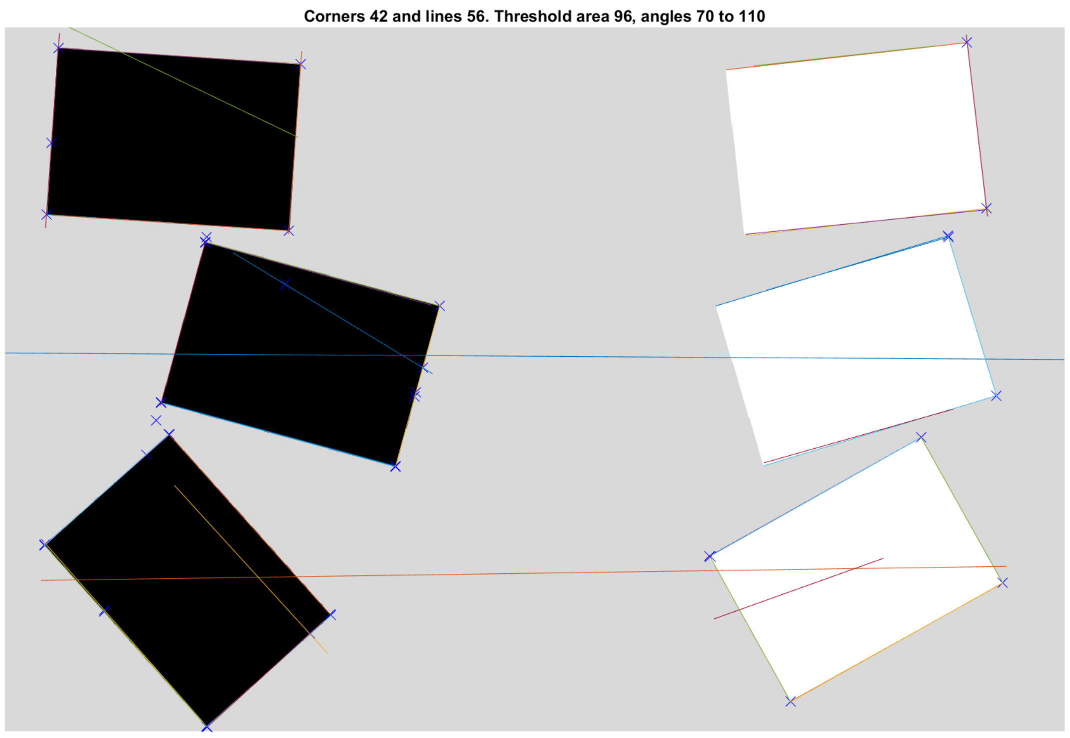

Corners 42 and lines 56. Threshold area 96, angles 70 to 110

Figure 3 – Building corners and lines

The script then compares all detected building corners and does a graph search to determine if any sets of points meet the criteria for a building. The algorithm will start at one building corner and search along one of its edges, looking for other building corners with one wall that shares an angle similar to our first point. If such a point is found, the algorithm continues to search for third point, applying the same success criteria. If the algorithm finds a 4th point, it will compare this to the slope and position of our first corner. If the points alight in slope and position, we determine this as a match and record the match. The resulting match can then be displayed as an overlay, as seen below.

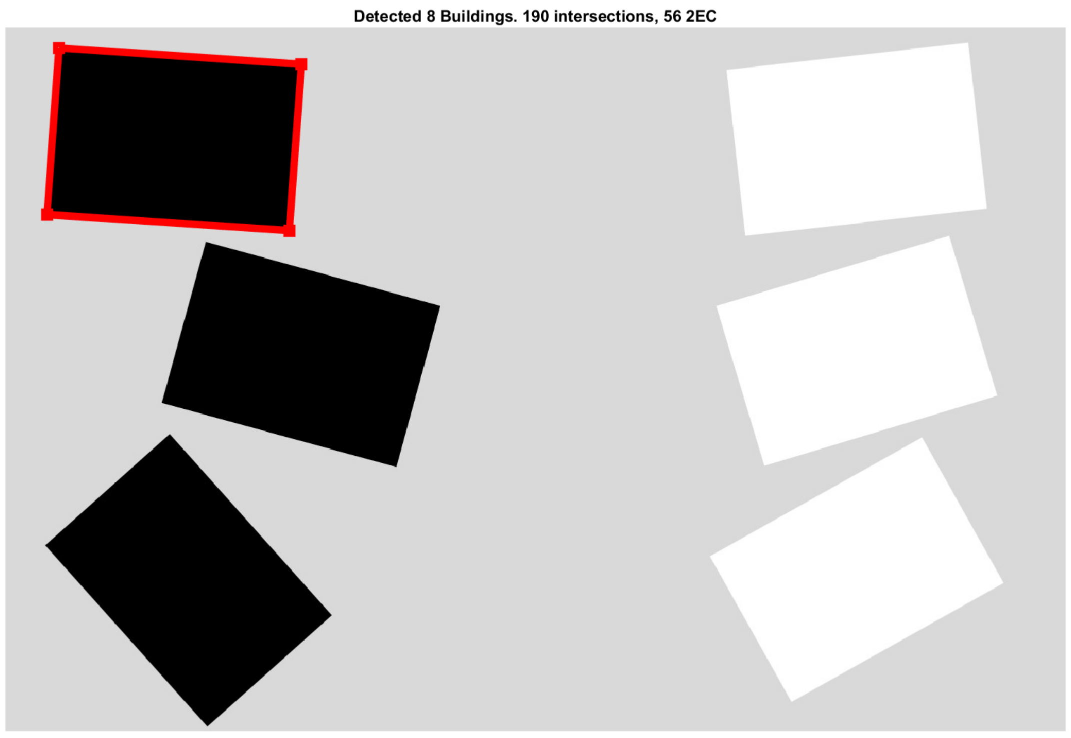


Figure 4 – Detected building shapes

As can be seen above, a building shape has successfully been detected. The corners and walls of the building are completely covered by the shape detected by this algorithm.

False negative error

Note that several buildings were not successfully detected. This may be due in part to an issue in the building detection algorithm, since we can see above that we have detected building corners on each corner of other rectangles, but the algorithm may have failed to examine them. Improvements to the building detection algorithm will mitigate this issue.

Repeat results

Note as listed in the title of the figure, that 8 buildings have been detected, but they all overlap the same building. This could easily be eliminated in the future through use of filtering to recognize and eliminate detected buildings that are similar.

Runtime

Note the run time is quite short for artificial images such as this, which will have less lines and corners detected. The script above runs in about 4 seconds.

Results with real satellite images

The algorithm is able to detect several edges and corners in satellite images, as can be seen below. One of the larger issues with real satellite images is filtering out the several detected lines and intersections.

Tuning threshold values such as minimum line length can help in reducing the number of lines and intersections detected.

Shown below is the algorithm's initially detected corners and intersections.

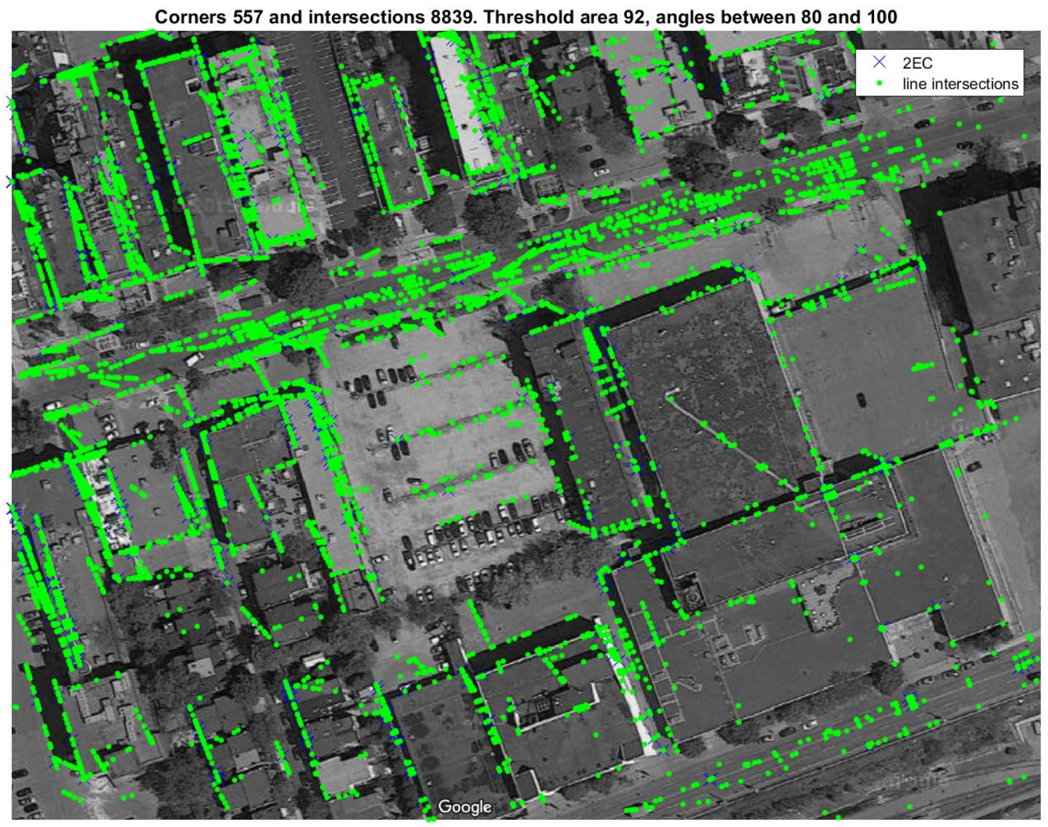

Corners 557 and intersections 8839. Threshold area 92, angles between 80 and 100
2EC
line intersections
Google

Figure 5 - Detected line intersections in satellite image

As can be seen below, several lines and potential building corners are noticed in areas where will not expect to see a building, such directly on the roadway of the image below. However the algorithm for detecting buildings looks for building corners that close a loop, so several of the corners seen below are eliminated.

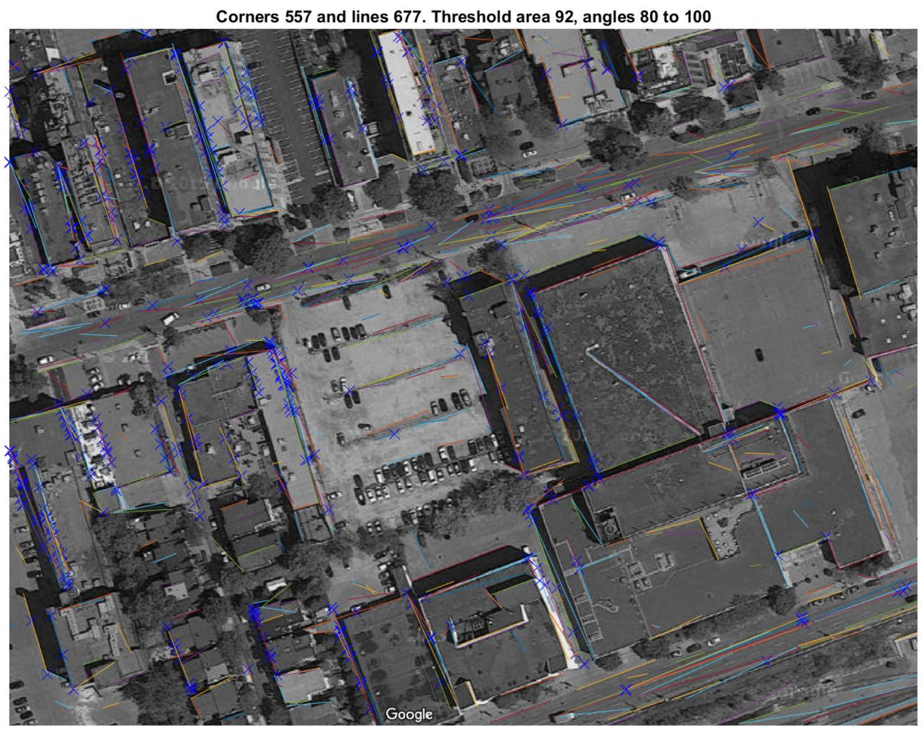
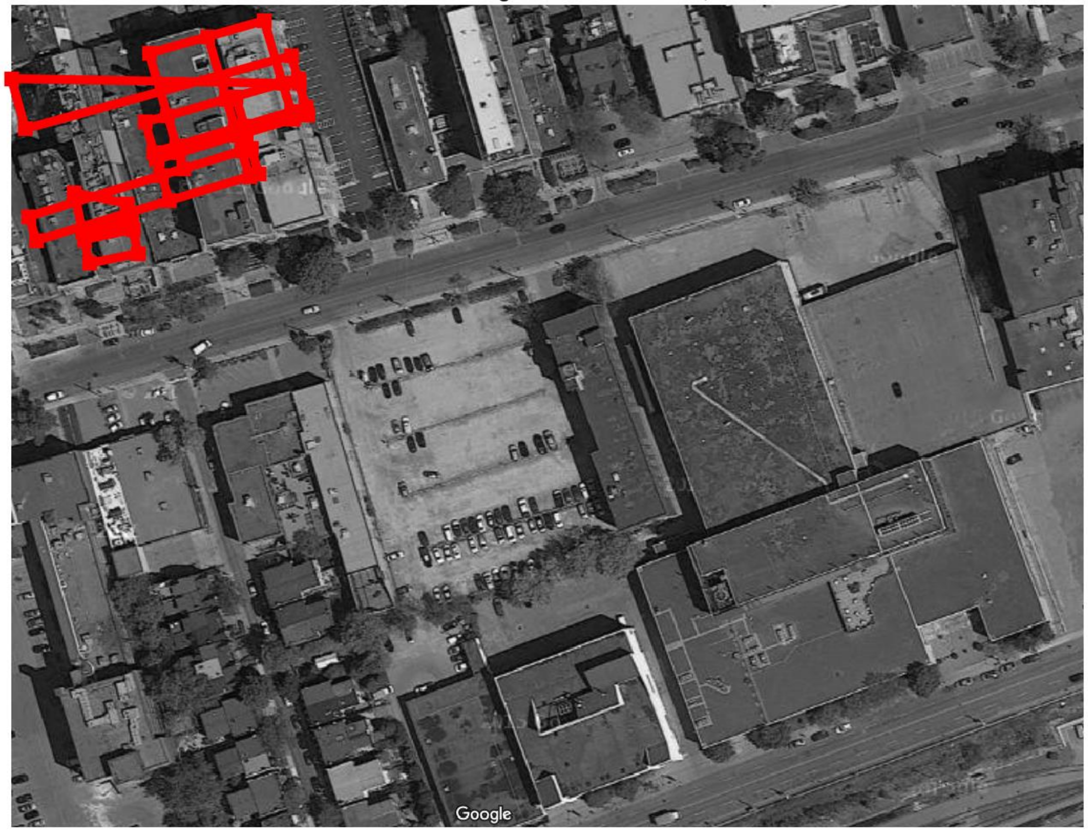

Corners 557 and lines 677. Threshold area 92, angles 80 to 100
Google

Figure 6 – Detected corners and lines in satellite image

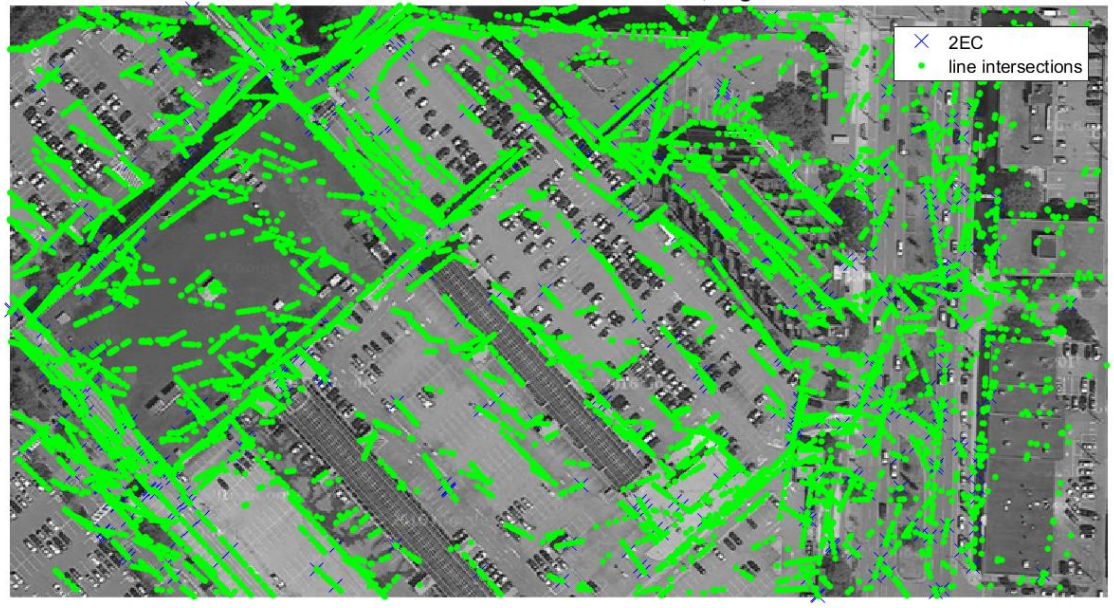
As can be seen below, the algorithm proposes several potential buildings directly above an area with several buildings, but has generally failed to properly detect building corners and edges of the larger building features. There are several tuning parameters

Detected 448 Buildings. 8839 intersections, 557 2EC

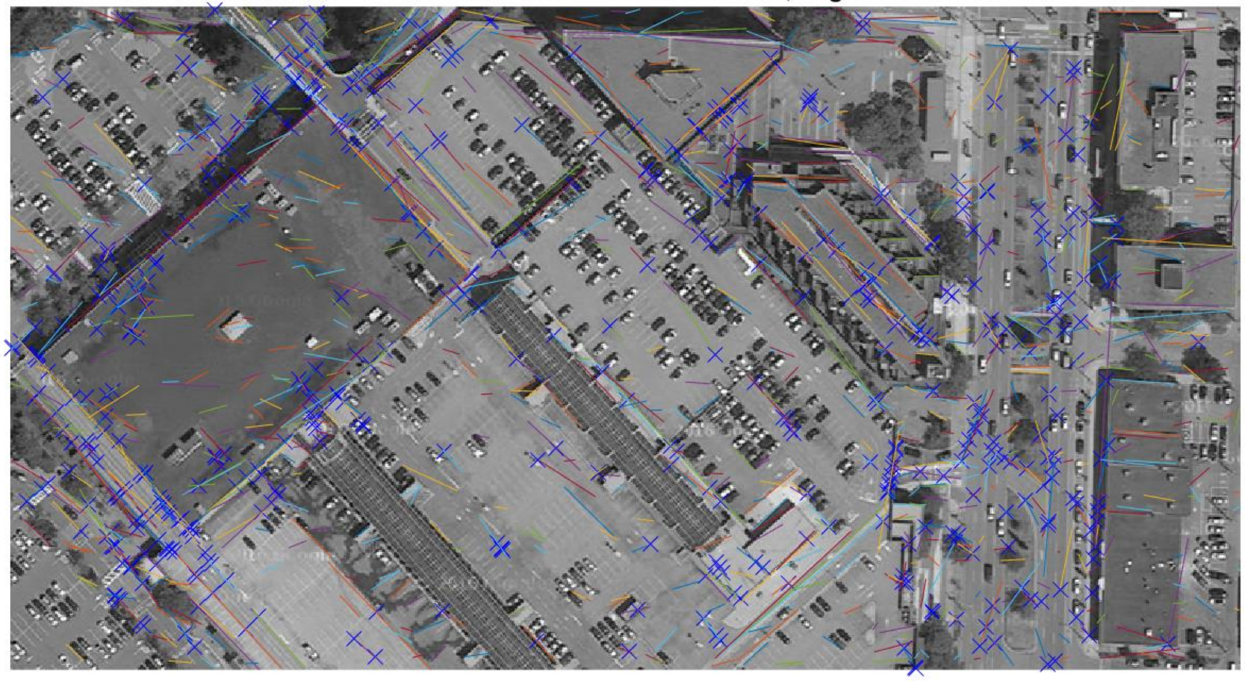


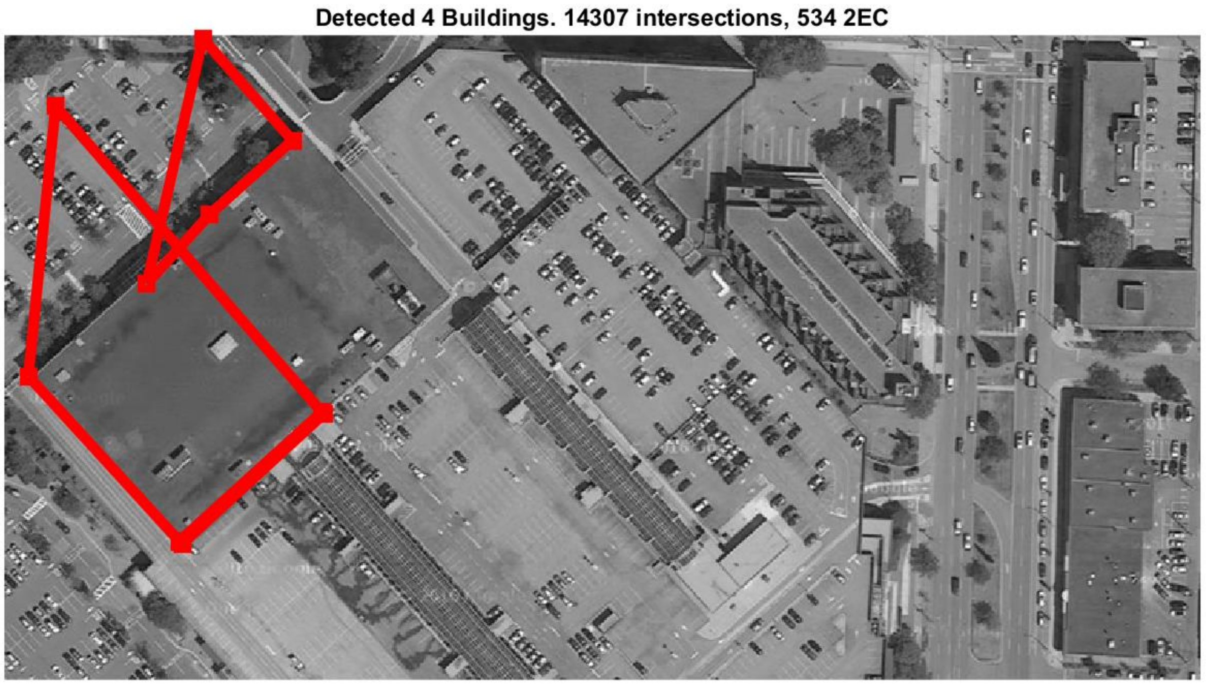
Another set of satellite image results are shown below. Note that in similar fashion, the algorithm was able to detect several edges and corners, and has specifically identified key building corners (blue crosses), the majority of which are well aligned to building faces and corners. While the algorithm is unable to currently produces a comprehensive and accurate list of buildings in the image, a combination of threshold value tuning and improvement of the finding building loop search algorithm will likely produce ideal results.

Corners 534 and intersections 14307. Threshold area 40, angles between 80 and 105



Corners 534 and lines 930. Threshold area 40, angles 80 to 105





Conclusion

In summary, this algorithm is able to correctly identify the majority of building faces and corners, but needs some amount of optimization to improve the final building detection.

Improvements for the future

Filtering the post-run results could consolidate buildings that are repeatedly detected.