Towards Finding Contrails in Cloud Images

Tuochaolong Zhang

Western University 2016

Towards Finding Contrails in Cloud Images

Tuochaolong Zhang

Thesis for
Computer Science 3380F - Project
Western University
Dr. John Barron

presented by
Tuochaolong Zhang

Octmber 15 2016

First Viewer: Dr. John Barron
Second Viewer: Dr. John Barron
Day of Oral test:

Contents

List of Figures

List of Tables

Summary

This is a project thesis for CS3380 course.

This project is to make a program to detect the contails from the images of sky and clouds.

I uses MatLab functions, like the Canny Edge Detection, the Polynomial curve fitting, and the Hough Transfer to make this project working

Chapter 1

Introduction

This project topic was originally posted by NASA Space Apps Challenges. Below is the introduction given on the project page [4].

> On clear or partly sunny days, people might look up at the sky and see straight lines of what appear to be clouds or white smoke. These lines are not smoke or natural clouds; they are contrails produced by aircraft. Contrails form because water vapor from jet engine exhaust passes through a cold and humid part of the air at high altitudes. Sometimes the jet that created the contrails is not visible overhead because winds aloft have blown the vapor trail into the observed area after the jet has passed. Naturally occurring high thin cirrus clouds do not form straight lines, they are more diffuse and irregular in shape than a contrail. Can an app be developed to help a ground observer determine the probability that an aircraft made the thin lines of white 'clouds' overhead?
>
> Contrails are potentially important sources of global warming. Contrails have been estimated to cause a tropospheric warming of 0.2 to 0.3 degree per decade by a general circulation model simulation of contrails [2]. Contrails reflect solar radiation and absorb and emit thermal infrared radiation. They make a radiative forcing that depends on many factors, especially contrail optical depth and coverage [3]. For scientists, knowing the contrails is an important way to know about climate changes.
>
> Viewing the image to figure out the contrails is very inefficient, as, in an image, there are other clouds data that make detection of contrails difficult. This project uses MatLab to program a solution for this problem. MatLab provides many built-in functions useful for image processing in general and this project in particular.

In this report, I will discuss the background of existing projects, my solution, and some results. Also, I give conclusions and future work for this topic.

Chapter 2

Background

2.1 Past Solution

On the NASA space challenge website, many participants had offered their solution by machine learning or performed prediction/verification using flight timetables and routes. However, these methods usually only can be used to find the existence of contrails, or the possibility of that those contrails exist. Meanwhile, it really relies on the flight data and the photo location, but it is not feasible for scientists to access all the flight database.

Here are some examples of the solutions propoised:

- Contrailers-Exeter
  This team mentioned in their explanation that they determine the probability of contrails by examining recent flights in the area and the air temperature.

- Hot on the Contrail
  This team proposes a machine learning solution, however, they only determine the possibility of contrails existing, but not where the contrails actually are. This solution has been proposed for the NASA space challenge, but it seems not to be useful for detecting the actual contrails locations.

2.2 Aim and Target

In order for scientists to do better research on contrails and clouds, the aim of this project proposes a program to automatically segment the contrails from clouds on different kinds of images, such as satellites images, photos, and so on. After the contrails in the image have been determined, they are highlighted and a comparison image is output.

2.3 Scope and Constraints

In this section, we outline some constraints exhibited by our approach and give the scope of our work.

Our project can not use images with interference information, an example is shown in Figure 2.1, which has latitude and longitude lines and map boards. These lines are much clearer than cloud outlines the images and they will disrupt the detection result.

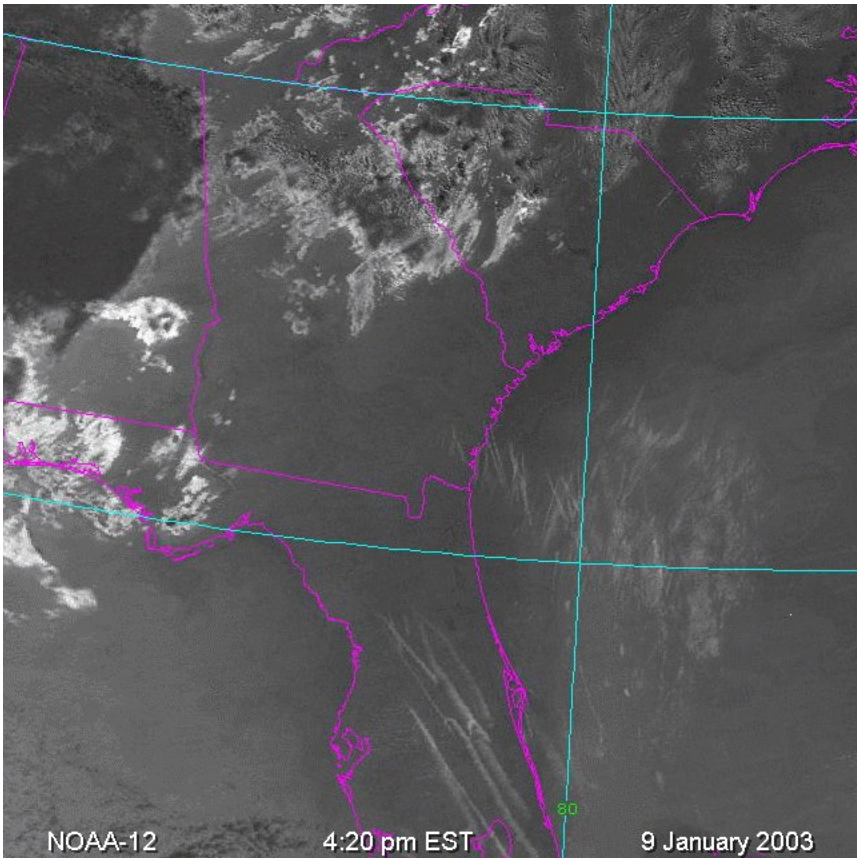

Figure 2.1: This program is inefficient to process this kind of image

Our program is inefficient for processing large images. The strategical removal of non-line edgels is computationally expensive.

Chapter 3

Approach

3.1 Solution Strategy

This projects goal is to distinguish contrails from the clouds, so we start by investigating the main difference between contrails and clouds.

Contrails usually show as a line on an image. Meanwhile, the clouds exhibit a random discrete distribution. Knowing this, we write a program to detect the straight lines in images to recognize the contrails.

However, it should be realized that even when the contrails look like pencil lines on the image, there actually doesnt exist an actual straight line when looking through the image data. Contrails also have some width in the image ([1], page 3). The real shape of contrails seems more like some long and thin rectangles with two fading sides. In this case, the detection problem became much harder.

Since this solution is hard to compute, instead of detecting the contrails widths, it is much easier to just detect the two clear sides of contrails. In this case, we can ignore the image inside the contrails, as well as other pixels inside the cloud by doing the edge detection.

After performing edge detection to detect edgels (pixels that are edge elements), we can detect lines using the Hough Transform on the edgel map. Another problem was that there were too many edgels in the image (see the results in Figure 3.1 below). Those bad edgels can cause to the Hough Transform to give incorrect lines, which do not come from contrails.

To solve this problem, some preprocessing on the edgel maps to get rid of bad edgel is necessary. In order to delete those bad edgels from clouds or other graphics, each pixel in a specific neighbouthood is checked as a small block. Because lines can be divided into an infinity of small lines, if a line goes through a small block, there must exist a small line inside the small block. We can use polynomial curve fitting to get the slope of the best small line in each block. Then we check how well the block edgels fit the computed line. Poor fits, as determined by a line residue, causes an edgel at the center of the neighbourhood to be rejected from the further Hough

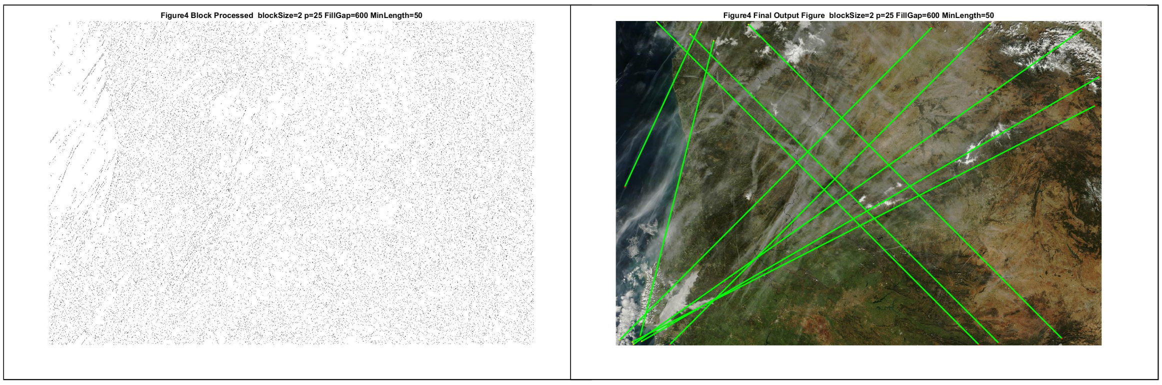

Figure 3.1: Too many bad edgals result

Transform.

After this preprocessing, I use the Hough Transform and obtain a much better line detection result.

The complete algorithm:

1. Convert the color image to a grayvalue image;
2. Use the Canny edge detection to compuyte the edge map of the grayvalue image;
3. Using the neighbouthoods about each edgel (left to right, top to bottom) we perform polynomial curve fitting to get the best small line information (slope and intercept);
4. Put all the edgels surviving the polynomial curve fitting procedure into a new image;
5. Perform the Hough Transform on the new image;
6. Median filter the new image;
7. Plot the results.

3.2 MatLab Methods

The MatLab code for Canny edge detection is given in the subsection below.

3.2.1 Canny Edge Detection

Edge_image = edge (I, ‘canny’);
Input: I, grayscale Image

Output: Edge_image, the black and white edge image

Canny edge detection is a technique for extracting useful structural information from different visual objects and dramatically reducing the amount of data to be processed. These are following steps:

1. Apply Gaussian filtering to remove noise;

2. Compute the intensity gradient of the image;

3. Apply non-maximum suppression to get rid of spurious responses to edge detection;

4. Apply a double threshold to determine potential edges to filter out the edgels with a weak gradient value and preserve edgels with high gradient values;

5. Track edges using hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

3.2.2 Polynomial curve fitting

Again we use MatLab to fit polynomials to the edgel images. *p = polyfit(x,y,n);*

Input:
x and y, vectors containing the x and y data to be fitted;
n, the degree of the polynomial to return;
Output: p, the third-degree polynomial that approximately fits the data.

This algorithm returns the coefficients for a polynomial $p(x)$ of degree n that is the best fit (in a least-squares sense) for the data in y. The coefficients in p are in descending powers and the length of p is $n + 1$.

3.2.3 Hough Transform and Hough Line Transform[5]

We show the MatLab for the Hough transform below. *[H,theta,rho] = hough(BW);*

Input: BW, the black and white image;
Output:
H, the Hough Transform matrix be returned as a numeric array;

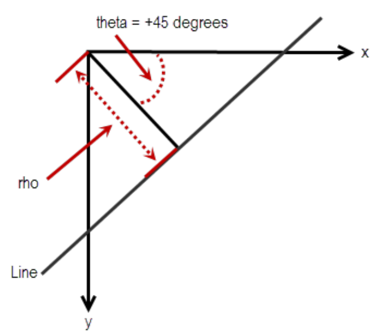

Figure 3.2: theta(θ) and rho(ρ) of Hough Transform

theta(θ), the angle in degree between the x-axis and rho vector;
rho(ρ), the distance from origin to the line along a vector perpendicular to the line;

The Standard Hough Transform (SHT) uses the parametric representation of a line:

$$\rho = xcos(\theta) + ysin(\theta). \tag{3.1}$$

For each edgel in the image, the numbers of lines going through it with all different angles is large. If 2 edgels are on the same line, they have the same ρ and θ valuesr. In a graph of the different angle and distances from the origin, the ρ and θ value copunts will be maximal. As we can see from Figure 3.3, when the Hough Transform

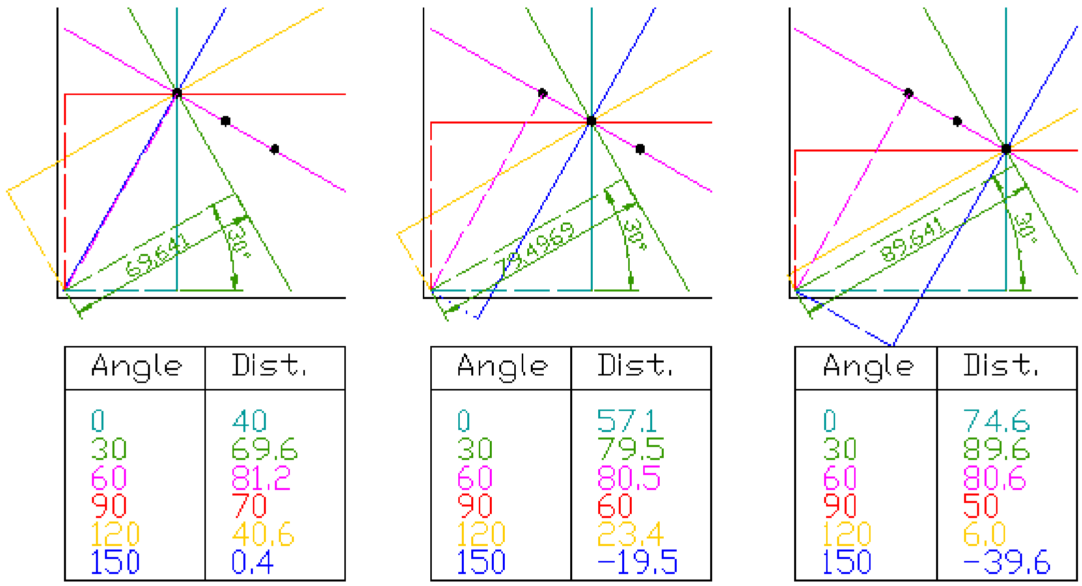

Figure 3.3: $\theta = 60°$ and ρ values are the same for the 3 points, see Wikipedia.com.

algorithm goes through all different angles (θ) for every edgel, there is a certain angle ($\theta = 60°$ in Figure 3.3), that makes the perpendicular distance from the line to origin keeps the same ($\rho = 81.2 \approx 80.5 \approx 80.6$). When we draw all the information below, we have a much clearer view of how the Hough Transform works.

3.2.4 Median Filter

We show how to do median filtering using MatLab. *B = medfilt2(A, [m n]);*

Input:
A the original image;
[m n], the isolate edgels size to be deleted;
Output: B, the image after deleted the isolated edgels.

Median filtering works by puting the $m \times n$ edgels around each target edgel into an array and then sorting all the values in the array. After this, we find the median value of the array and replace the target edgel with that edgel.

3.3 Algorithm to Reduce the Number of Bad Edgels

The input is an edgel image and the output is that edgel image with the small neighbourhoods containing small residual values deleted.

Since Canny edge detection gives many edgels, we need to further process these edgel maps to eliminate edgels that are not parts of straight lines. For each $2 \times s + 1$ by $2 \times s + 1$ square neighborhood about an edgel we fit all the neighborhood edgels to a straight line using polyfit. We fit either $y = mx + b$ for lines, where $|m| \leq 45°$ and $x = \frac{y-b}{m}$, if $|m| \geq 45°$. This takes care of horizontal and vertical lines. Now we have the equation of the best line fit for all the edgels in a neighborhood. But how good is this line fit? For each neighborhood edgel we compute the residual of that edgels neighborhood fit to the line. We compute the overall residual as the square root of the sum of these squared residual values.

Since neighbourhoods with poor line fits will have large overall residual values, we remove bad edgels using a threshold of 15, determined by trial and error.

3.3.1 Pseudo code

Here we present the pseudo code.

```
I = Read image;
grayscale (I);
image = Canny edge detect(I);
set the block size, height and width (s, height, width)

initial number no lines = 0;

for x = (1+s) to (height-s):
    for y = (1+s) to (width-s):
    initial points number = 0:
        if pixel is on an edge:
            for i = (x-s) to (x+s):
                for j = (y-s) to (y+s):
```

```
                    if block pixel is edge:
                    record them;
                    points number ++;
            if at least one line in a block:
                m = polyfit (x and y data recoded);
                if (line s slope >1 or <-1):
                    compute the residual r;
                else:
                    modify the m;
                    compute the residual r;
                save r values
            else:
            number no lines ++;

get non_zero_r;
sort non_zero_r;
set the precentage p;
threshold = cast(p% * non_zero_r);

for x = (1+s) to (height-s):
    for y = (1+s) to (width-s):
        if the r value is in the shreshold and non zero:
            write it on new_image;

new_image = medfilter (new_image);
get Hough Transform matrix(H), theta(T) and rho(R) by hough(new_image);
get peaks(P) = houghpeaks (new_image);
lines = houghlines(new_image, T, R, P);

for k = 1 to number of lines:
    plot lines on origin image(I);
```

Chapter 4

Result

4.1 How to evaluate

In this project we need to manually count the number of contrails and evaluate the correctness of all results with different parameters.

4.1.1 Detect Rules

Here are the rules to count the contrails and detected lines:

1. the contrails and result lines are counted manually
2. in the origin image and final output image, we only count the lines with clear boarder
3. if two lines are overlay each other, and the angle between two lines are less than 5 degree, we count it as one
4. if the lines are more than 10 and hard to count the number, we count it as 10, because the default maximum number of lines to detect is 10 in the Hough Transform in MatLab.

4.1.2 Parameters Definition

Here are the definitions of parameters used in my program:

- Block Radius is the size of blocks in the process of reduce bad pixels. According to our algorithm, for each pixel, we need to do check all the pixels around it within the block radius to check if there is/are line(s) across this block.

- Percentage is the percentage of best blocks to keep in the final step of algorithm to reduce bad edgels.

- Fill Gap is a build-in parameter in the Hough Transform, if there are two line segments are in same slope but no parallel and their distance are less than Fill Gap, the Hough Transform will concatenate two line segments and consider them as one line.

- Minimum Length is a build-in parameter in the Hough Transfer, only if the line segment is larger than a certain length (MinLength), then it will be considered.

4.2 Results and Discussion

According to the results of program's giving, we sort it by largest correct rate firstly and least incorrect rate secondly. Then we can have some discussion here.

4.2.1 Satellite Image

Figure 4.1 is a satellite image with about 7 contrails which are easier to be recognized and others are relatively blurrier.

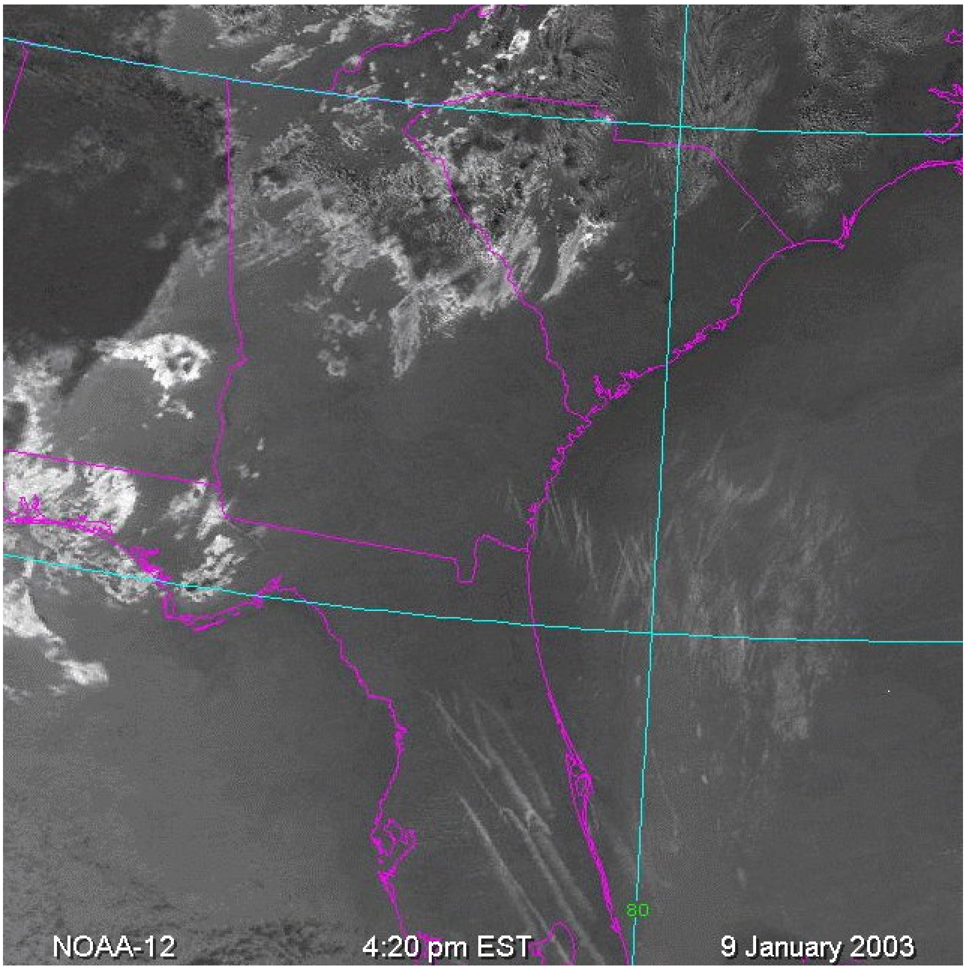

Figure 4.1: A satellite image with longitude and latitude lines (640 x 640)

The result cannot be fully detected with the interference of the lat-lag and broader lines.
The best result has the parameters with block radius as 2, percentage to keep as 15, fill gap as 600, and minimum length can be any in 25, 50 and 100. The best result of Satellite Image is greatly impacted by the lat-lag lines. It covers all the contrails, however, it also covers lots of points are not contrails. We can see the largest correct rate is even less than 50%, which clearly shows this detective is ineffeicent. And the worst result covers a longitude line only, it doesnt cover any contrails because that longitude line has a much more clear edge than contrails. The results image for figure 4.1 can be seen at figure 4.2 for the best result and figure 4.3 for the worst

result.

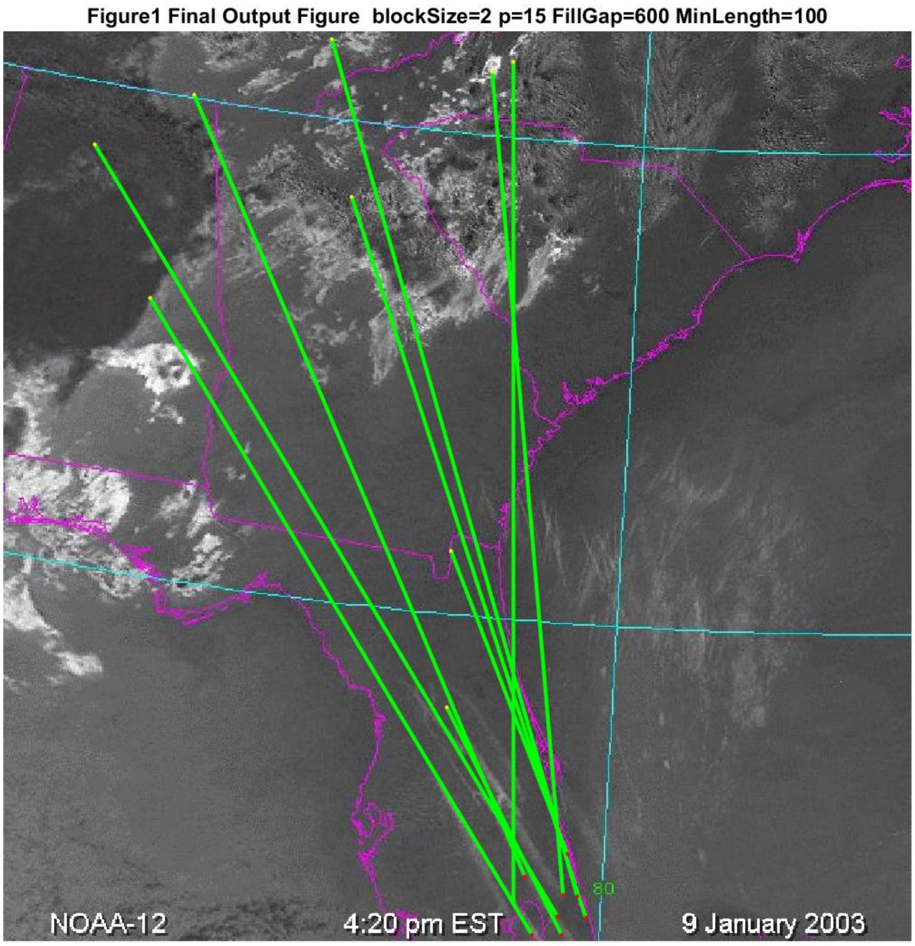

Figure 4.2: Best Result for Figure 4.1

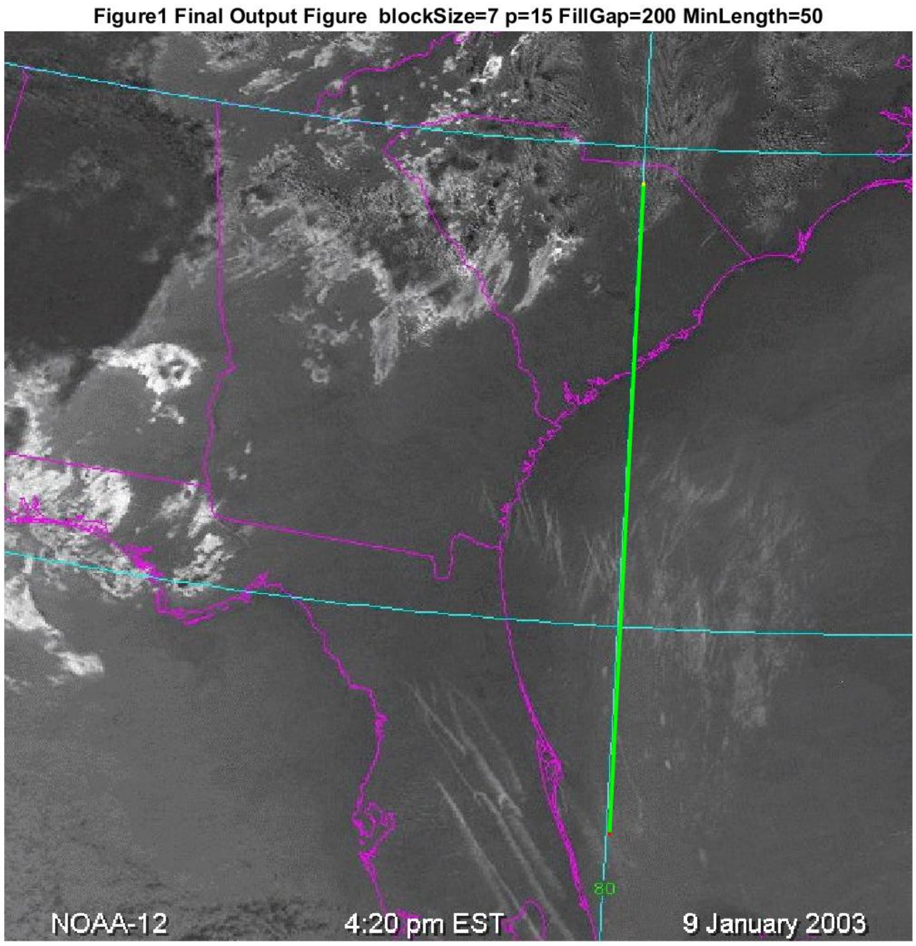

Figure 4.3: Worst Result for Figure 4.1, only covers lagtitude line

Block Radius	Percentage	FillGap	Minimum Length	Correct Number	Incorrect Number	Missingr Number	Correct Rate	Incorrect Rate
2	15	600	50	3	5	4	42.86%	62.50%
2	15	600	100	3	5	4	42.86%	62.50%
2	15	600	25	3	5	4	42.86%	62.50%
2	15	200	50	2	5	5	28.57%	71.43%
2	15	200	100	2	5	5	28.57%	71.43%
2	25	400	25	2	5	5	28.57%	71.43%
2	25	400	50	2	5	5	28.57%	71.43%
2	25	400	100	2	5	5	28.57%	71.43%
2	15	400	25	2	6	5	28.57%	75.00%
2	15	400	50	2	6	5	28.57%	75.00%
2	15	400	100	2	6	5	28.57%	75.00%
2	25	600	25	2	6	5	28.57%	75.00%
2	25	600	50	2	6	5	28.57%	75.00%
2	25	600	100	2	6	5	28.57%	75.00%
2	15	200	25	2	7	5	28.57%	77.78%
2	5	200	25	2	8	5	28.57%	80.00%
2	5	200	50	2	8	5	28.57%	80.00%
2	5	200	100	2	8	5	28.57%	80.00%
2	5	400	25	2	9	5	28.57%	81.82%
2	5	400	50	2	9	5	28.57%	81.82%
2	5	400	100	2	9	5	28.57%	81.82%
2	5	600	25	2	9	5	28.57%	81.82%
2	5	600	50	2	9	5	28.57%	81.82%
2	5	600	100	2	9	5	28.57%	81.82%
2	25	200	50	2	9	5	28.57%	81.82%
2	25	200	100	2	9	5	28.57%	81.82%
2	25	200	25	2	10	5	28.57%	83.33%
7	5	200	100	0	1	7	0.00%	100.00%
7	5	400	25	0	1	7	0.00%	100.00%
7	5	400	50	0	1	7	0.00%	100.00%
7	5	400	100	0	1	7	0.00%	100.00%
7	5	600	25	0	1	7	0.00%	100.00%
7	5	600	50	0	1	7	0.00%	100.00%
7	5	600	100	0	1	7	0.00%	100.00%
7	15	200	25	0	1	7	0.00%	100.00%
7	15	200	50	0	1	7	0.00%	100.00%
7	15	200	100	0	1	7	0.00%	100.00%
7	15	400	25	0	1	7	0.00%	100.00%
7	15	400	50	0	1	7	0.00%	100.00%
7	15	400	100	0	1	7	0.00%	100.00%
7	15	600	25	0	1	7	0.00%	100.00%
7	15	600	50	0	1	7	0.00%	100.00%

7	15	600	100	0	1	7	0.00%	100.00%
7	25	200	25	0	1	7	0.00%	100.00%
7	25	200	50	0	1	7	0.00%	100.00%
7	25	200	100	0	1	7	0.00%	100.00%
7	25	400	25	0	1	7	0.00%	100.00%
7	25	400	50	0	1	7	0.00%	100.00%
7	25	400	100	0	1	7	0.00%	100.00%
7	25	600	25	0	1	7	0.00%	100.00%
7	25	600	50	0	1	7	0.00%	100.00%
7	25	600	100	0	1	7	0.00%	100.00%
12	5	200	25	0	1	7	0.00%	100.00%
12	5	200	50	0	1	7	0.00%	100.00%
12	5	200	100	0	1	7	0.00%	100.00%
12	5	400	25	0	1	7	0.00%	100.00%
12	5	400	50	0	1	7	0.00%	100.00%
12	5	400	100	0	1	7	0.00%	100.00%
12	5	600	25	0	1	7	0.00%	100.00%
12	5	600	50	0	1	7	0.00%	100.00%
12	5	600	100	0	1	7	0.00%	100.00%
12	15	200	25	0	1	7	0.00%	100.00%
12	15	200	50	0	1	7	0.00%	100.00%
12	15	200	100	0	1	7	0.00%	100.00%
12	15	400	25	0	1	7	0.00%	100.00%
12	15	400	50	0	1	7	0.00%	100.00%
12	15	400	100	0	1	7	0.00%	100.00%
12	15	600	25	0	1	7	0.00%	100.00%
12	15	600	50	0	1	7	0.00%	100.00%
12	15	600	100	0	1	7	0.00%	100.00%
12	25	200	25	0	1	7	0.00%	100.00%
12	25	200	50	0	1	7	0.00%	100.00%
12	25	200	100	0	1	7	0.00%	100.00%
12	25	400	25	0	1	7	0.00%	100.00%
12	25	400	50	0	1	7	0.00%	100.00%
12	25	400	100	0	1	7	0.00%	100.00%
12	25	600	25	0	1	7	0.00%	100.00%
12	25	600	50	0	1	7	0.00%	100.00%
12	25	600	100	0	1	7	0.00%	100.00%
7	5	200	25	0	1	7	0.00%	100.00%
7	5	200	50	0	1	7	0.00%	100.00%

Table 4.1: Experimental results for Figure 4.1.

4.2.2 Contrails in Trees and Sky Image

Figure 4.4 is a color image with 1 thin contrail which is easy to realize. However, there also exists some clear boarder of trees and sky. Even thought the boarder is clear, their edge curves are not as straight as contrails.
The best result happens when parameters without the block radius as 2, others can all be counted as the best, their correctness rate are 100%, and incorrect rate are 0%. It covers only contrail very well, only a small part of this contrail not be covered because it sightly curved. As for the worst, it happens when blockSize = 2, p = 15, FillGap = 600, MinLength = 25, it has more than 10 incorrect lines, none of them covers that contrail. The results image for figure 4.4 can be seen at figure 4.5 for the best result and figure 4.6 for the worst result.

Figure 4.4: Contrails in Trees and Sky Image (640 x 480)

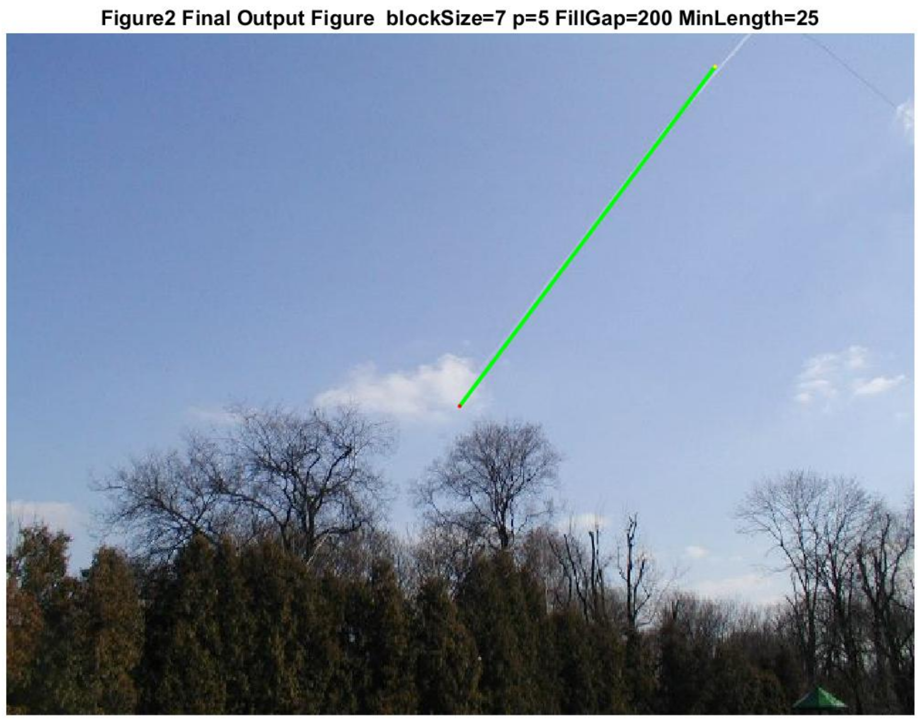

Figure 4.5: Best Result for Figure 4.4

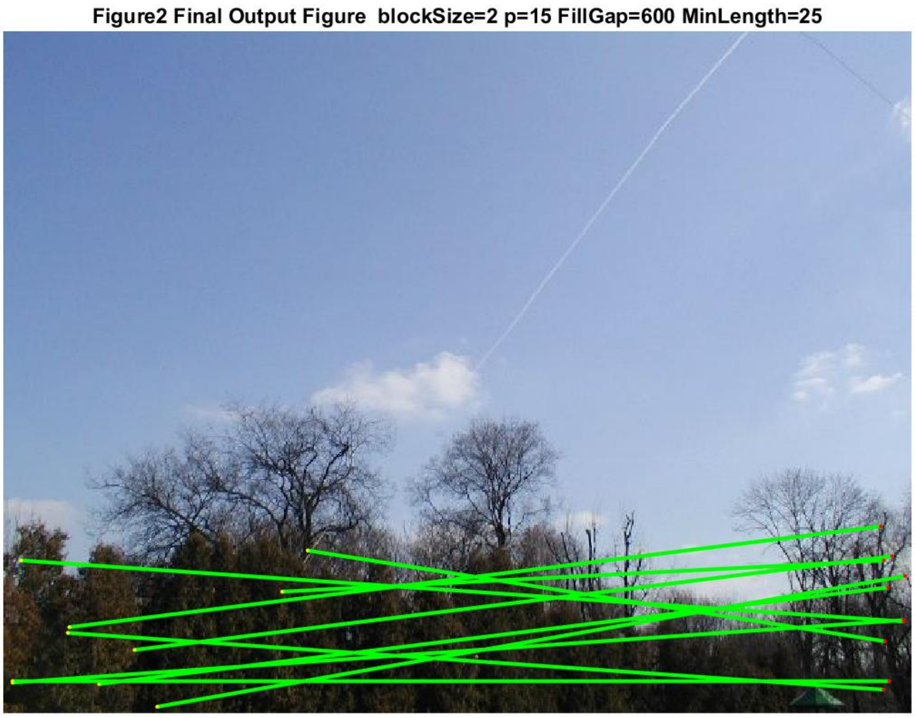

Figure 4.6: Worst Result for Figure 4.4, no contrail covered, and too many useless lines

Block Radius	Percentage	FillGap	Minimum Length	Correct Number	Incorrect Number	Missingr Number	Correct Rate	Incorrect Rate
7	5	200	25	1	0	0	100.00%	0.00%
7	5	200	50	1	0	0	100.00%	0.00%
7	5	200	100	1	0	0	100.00%	0.00%
7	5	400	25	1	0	0	100.00%	0.00%
7	5	400	50	1	0	0	100.00%	0.00%
7	5	400	100	1	0	0	100.00%	0.00%
7	5	600	25	1	0	0	100.00%	0.00%
7	5	600	50	1	0	0	100.00%	0.00%
7	5	600	100	1	0	0	100.00%	0.00%
7	15	200	25	1	0	0	100.00%	0.00%
7	15	200	50	1	0	0	100.00%	0.00%
7	15	200	100	1	0	0	100.00%	0.00%
7	15	400	25	1	0	0	100.00%	0.00%
7	15	400	50	1	0	0	100.00%	0.00%
7	15	400	100	1	0	0	100.00%	0.00%
7	15	600	25	1	0	0	100.00%	0.00%
7	15	600	50	1	0	0	100.00%	0.00%
7	15	600	100	1	0	0	100.00%	0.00%
7	25	200	25	1	0	0	100.00%	0.00%
7	25	200	50	1	0	0	100.00%	0.00%
7	25	200	100	1	0	0	100.00%	0.00%
7	25	400	25	1	0	0	100.00%	0.00%
7	25	400	50	1	0	0	100.00%	0.00%
7	25	400	100	1	0	0	100.00%	0.00%
7	25	600	25	1	0	0	100.00%	0.00%
7	25	600	50	1	0	0	100.00%	0.00%
7	25	600	100	1	0	0	100.00%	0.00%
12	5	200	25	1	0	0	100.00%	0.00%
12	5	200	50	1	0	0	100.00%	0.00%
12	5	200	100	1	0	0	100.00%	0.00%
12	5	400	25	1	0	0	100.00%	0.00%
12	5	400	50	1	0	0	100.00%	0.00%
12	5	400	100	1	0	0	100.00%	0.00%
12	5	600	25	1	0	0	100.00%	0.00%
12	5	600	50	1	0	0	100.00%	0.00%
12	5	600	100	1	0	0	100.00%	0.00%
12	15	200	25	1	0	0	100.00%	0.00%
12	15	200	50	1	0	0	100.00%	0.00%
12	15	200	100	1	0	0	100.00%	0.00%
12	15	400	25	1	0	0	100.00%	0.00%
12	15	400	50	1	0	0	100.00%	0.00%
12	15	400	100	1	0	0	100.00%	0.00%

12	15	600	25	1	0	0	100.00%	0.00%
12	15	600	50	1	0	0	100.00%	0.00%
12	15	600	100	1	0	0	100.00%	0.00%
12	25	200	25	1	0	0	100.00%	0.00%
12	25	200	50	1	0	0	100.00%	0.00%
12	25	200	100	1	0	0	100.00%	0.00%
12	25	400	25	1	0	0	100.00%	0.00%
12	25	400	50	1	0	0	100.00%	0.00%
12	25	400	100	1	0	0	100.00%	0.00%
12	25	600	25	1	0	0	100.00%	0.00%
12	25	600	50	1	0	0	100.00%	0.00%
12	25	600	100	1	0	0	100.00%	0.00%
2	25	400	50	1	10	0	100.00%	90.91%
2	25	400	100	1	10	0	100.00%	90.91%
2	25	600	25	1	10	0	100.00%	90.91%
2	25	600	50	1	10	0	100.00%	90.91%
2	25	600	100	1	10	0	100.00%	90.91%
2	25	200	25	1	10	0	100.00%	90.91%
2	25	200	50	1	10	0	100.00%	90.91%
2	25	200	100	1	10	0	100.00%	90.91%
2	25	400	25	1	10	0	100.00%	90.91%
2	5	200	25	0	9	1	0.00%	100.00%
2	5	200	50	0	9	1	0.00%	100.00%
2	5	200	100	0	9	1	0.00%	100.00%
2	5	400	25	0	9	1	0.00%	100.00%
2	5	400	50	0	9	1	0.00%	100.00%
2	5	400	100	0	9	1	0.00%	100.00%
2	5	600	25	0	9	1	0.00%	100.00%
2	5	600	50	0	9	1	0.00%	100.00%
2	5	600	100	0	9	1	0.00%	100.00%
2	15	200	25	0	10	1	0.00%	100.00%
2	15	200	50	0	10	1	0.00%	100.00%
2	15	200	100	0	10	1	0.00%	100.00%
2	15	400	25	0	10	1	0.00%	100.00%
2	15	400	50	0	10	1	0.00%	100.00%
2	15	400	100	0	10	1	0.00%	100.00%
2	15	600	25	0	10	1	0.00%	100.00%
2	15	600	50	0	10	1	0.00%	100.00%
2	15	600	100	0	10	1	0.00%	100.00%

Table 4.2: Experimental results for Figure 4.4.

4.2.3 Clear Sky with Contrails

Figure 4.7 is a color image with 1 thick contrail which is easy to realize, this contrail is wide enough to have 2 edges, one side is clear, another side is relatively blurrier. Also, there is a tree in this image which has clear edge with sky.
The best result happens when parameters without the FillGap as 200 and when BlockRadius as 2, Percentage as 5 and FillGap as 400 and 600. Note here the largest correct rate is 200%, there are actually lines on same contrail, then it is clear that 100% correct rate is better than 200% correct rate, therefor we donnot consider FillGap with 200 as the best parameter. The best result of Clear Sky with Contrails covers more than half length of the contrail on the image. And the worst one also covers the right contrail on the image, however, the incorrect rate is too high because it also covers so many incorrect lines.The results image for figure 4.7 can be seen at figure 4.8 for the best result and figure 4.9 for the worst result.

Figure 4.7: Clear Sky with Contrails (1488 x 1984)

Figure 4.8: Best Result for Figure 4.7

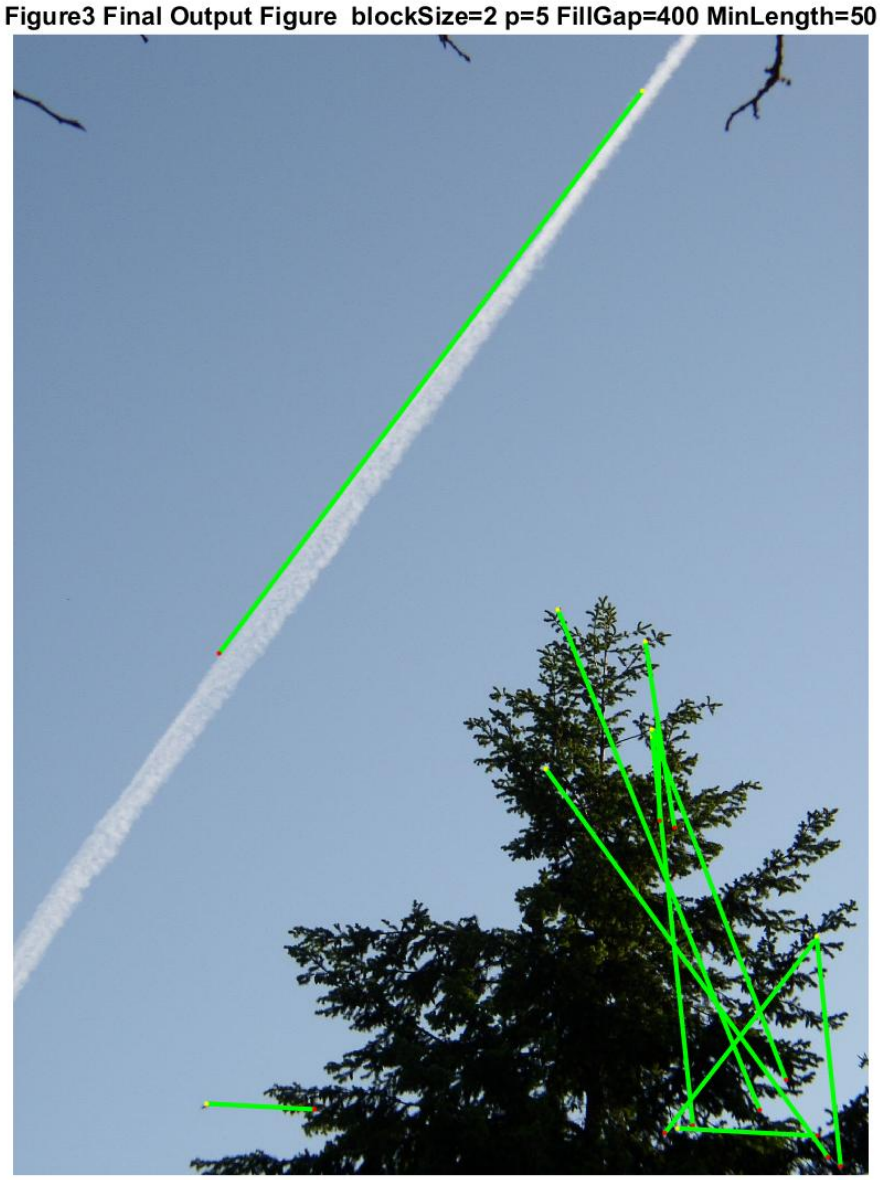

Figure 4.9: Worst Result for Figure 4.7, too many bad lines

Block Radius	Percentage	FillGap	Minimum Length	Correct Number	Incorrect Number	Missingr Number	Correct Rate	Incorrect Rate
12	5	200	50	2	0	-1	200.00%	0.00%
2	15	200	50	2	0	-1	200.00%	0.00%
2	15	200	100	2	0	-1	200.00%	0.00%
2	25	200	50	2	0	-1	200.00%	0.00%
2	25	200	100	2	0	-1	200.00%	0.00%
2	15	200	25	2	0	-1	200.00%	0.00%
2	25	200	25	2	0	-1	200.00%	0.00%
7	5	200	100	2	0	-1	200.00%	0.00%
7	15	200	50	2	0	-1	200.00%	0.00%
7	15	200	100	2	0	-1	200.00%	0.00%
7	25	200	50	2	0	-1	200.00%	0.00%
7	25	200	100	2	0	-1	200.00%	0.00%
7	5	200	25	2	0	-1	200.00%	0.00%
7	15	200	25	2	0	-1	200.00%	0.00%
7	25	200	25	2	0	-1	200.00%	0.00%
12	5	200	100	2	0	-1	200.00%	0.00%
12	15	200	50	2	0	-1	200.00%	0.00%
12	15	200	100	2	0	-1	200.00%	0.00%
12	25	200	50	2	0	-1	200.00%	0.00%
12	25	200	100	2	0	-1	200.00%	0.00%
12	5	200	25	2	0	-1	200.00%	0.00%
12	15	200	25	2	0	-1	200.00%	0.00%
12	25	200	25	2	0	-1	200.00%	0.00%
7	5	200	50	2	0	-1	200.00%	0.00%
2	5	200	100	2	6	-1	200.00%	75.00%
2	5	200	25	2	7	-1	200.00%	77.78%
2	5	200	50	2	7	-1	200.00%	77.78%
2	15	400	25	1	0	0	100.00%	0.00%
2	25	400	25	1	0	0	100.00%	0.00%
2	15	400	50	1	0	0	100.00%	0.00%
2	15	400	100	1	0	0	100.00%	0.00%
2	15	600	25	1	0	0	100.00%	0.00%
2	15	600	50	1	0	0	100.00%	0.00%
2	15	600	100	1	0	0	100.00%	0.00%
2	25	400	50	1	0	0	100.00%	0.00%
2	25	400	100	1	0	0	100.00%	0.00%
2	25	600	25	1	0	0	100.00%	0.00%
2	25	600	50	1	0	0	100.00%	0.00%
2	25	600	100	1	0	0	100.00%	0.00%
7	15	400	25	1	0	0	100.00%	0.00%
7	25	400	25	1	0	0	100.00%	0.00%
7	5	400	50	1	0	0	100.00%	0.00%

7	5	400	100	1	0	0	100.00%	0.00%
7	5	600	25	1	0	0	100.00%	0.00%
7	5	600	50	1	0	0	100.00%	0.00%
7	5	600	100	1	0	0	100.00%	0.00%
7	15	400	50	1	0	0	100.00%	0.00%
7	15	400	100	1	0	0	100.00%	0.00%
7	15	600	25	1	0	0	100.00%	0.00%
7	15	600	50	1	0	0	100.00%	0.00%
7	15	600	100	1	0	0	100.00%	0.00%
7	25	400	50	1	0	0	100.00%	0.00%
7	25	400	100	1	0	0	100.00%	0.00%
7	25	600	25	1	0	0	100.00%	0.00%
7	25	600	50	1	0	0	100.00%	0.00%
7	25	600	100	1	0	0	100.00%	0.00%
12	15	400	25	1	0	0	100.00%	0.00%
12	25	400	25	1	0	0	100.00%	0.00%
12	5	400	50	1	0	0	100.00%	0.00%
12	5	400	100	1	0	0	100.00%	0.00%
12	5	600	25	1	0	0	100.00%	0.00%
12	5	600	50	1	0	0	100.00%	0.00%
12	5	600	100	1	0	0	100.00%	0.00%
12	15	400	50	1	0	0	100.00%	0.00%
12	15	400	100	1	0	0	100.00%	0.00%
12	15	600	25	1	0	0	100.00%	0.00%
12	15	600	50	1	0	0	100.00%	0.00%
12	15	600	100	1	0	0	100.00%	0.00%
12	25	400	50	1	0	0	100.00%	0.00%
12	25	400	100	1	0	0	100.00%	0.00%
12	25	600	25	1	0	0	100.00%	0.00%
12	25	600	50	1	0	0	100.00%	0.00%
12	25	600	100	1	0	0	100.00%	0.00%
7	5	400	25	1	0	0	100.00%	0.00%
12	5	400	25	1	0	0	100.00%	0.00%
2	5	400	25	1	6	0	100.00%	85.71%
2	5	400	50	1	6	0	100.00%	85.71%
2	5	400	100	1	6	0	100.00%	85.71%
2	5	600	25	1	6	0	100.00%	85.71%
2	5	600	50	1	6	0	100.00%	85.71%
2	5	600	100	1	6	0	100.00%	85.71%

Table 4.3: Experimental results for Figure 4.7.

4.2.4 Aerial Image

Figure 4.10 is a colored aerial image with about too many contrails which are hard to be clear recognized and counted. These contrails cross each other, and most of them are all thin and blurry. According to the rule 4 above, we count it as 10, for it is so difficult to find out the actual number.
The best result has the parameters with block radius as 2, percentage to keep as 15, fill gap as 600, and minimum length can be any in 25, 50 and 100. The best result covers more than half of the contrails on the image. Anyway, because too many unclear contrails exist on the image, and the default contrails to find is only 10, we cannot find all the contrails. And the worst one covers very few contrails on the image with most confused thing is the results lines are parallel or perpendicular with each other. The results image for figure 4.10 can be seen at figure 4.11 for the best result and figure 4.12 for the worst result.

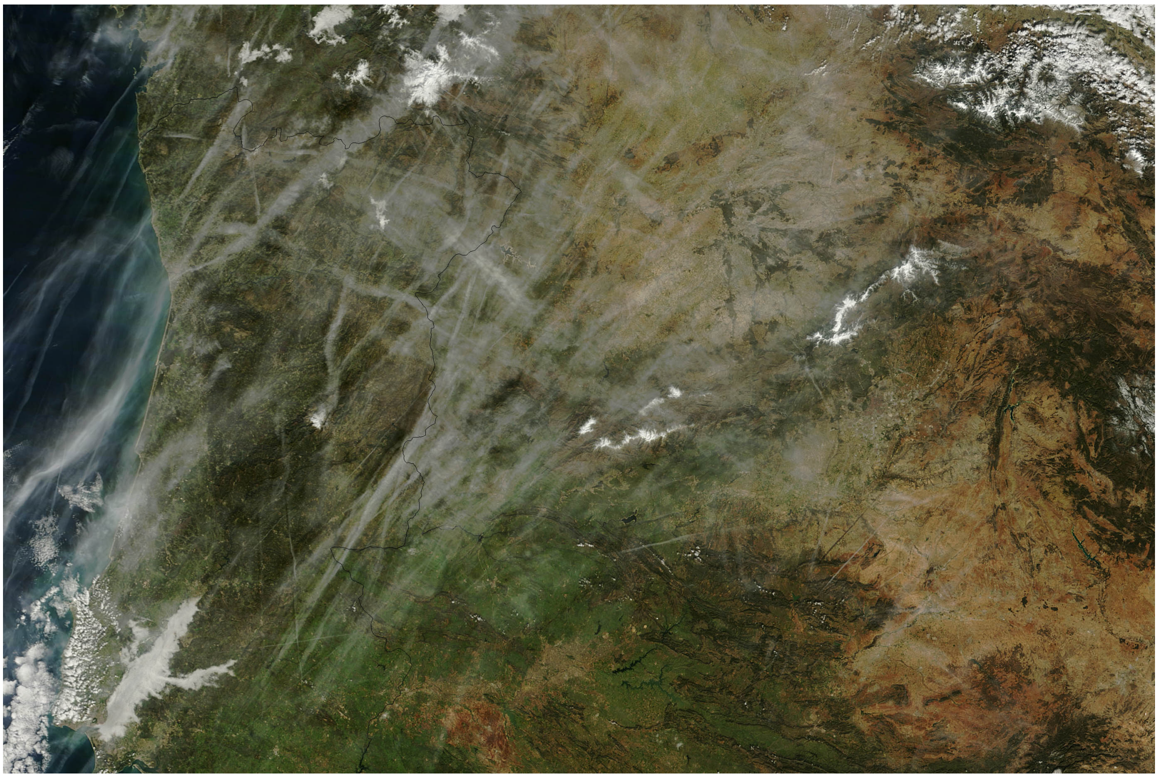

Figure 4.10: Aerial Image (2768 x 1845)

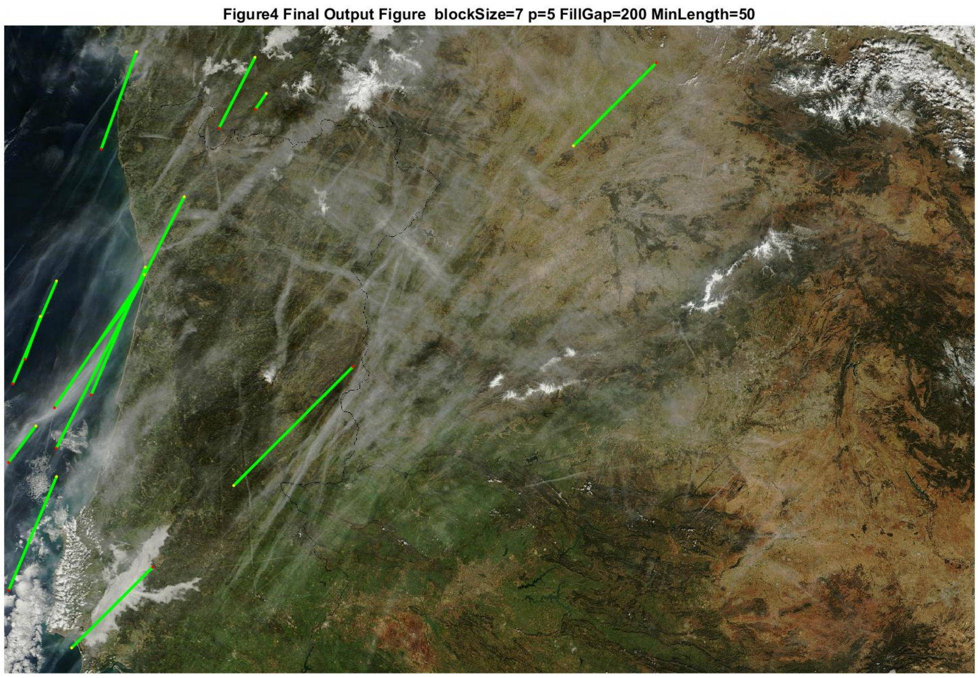

Figure 4.11: Best Result for Figure 4.10

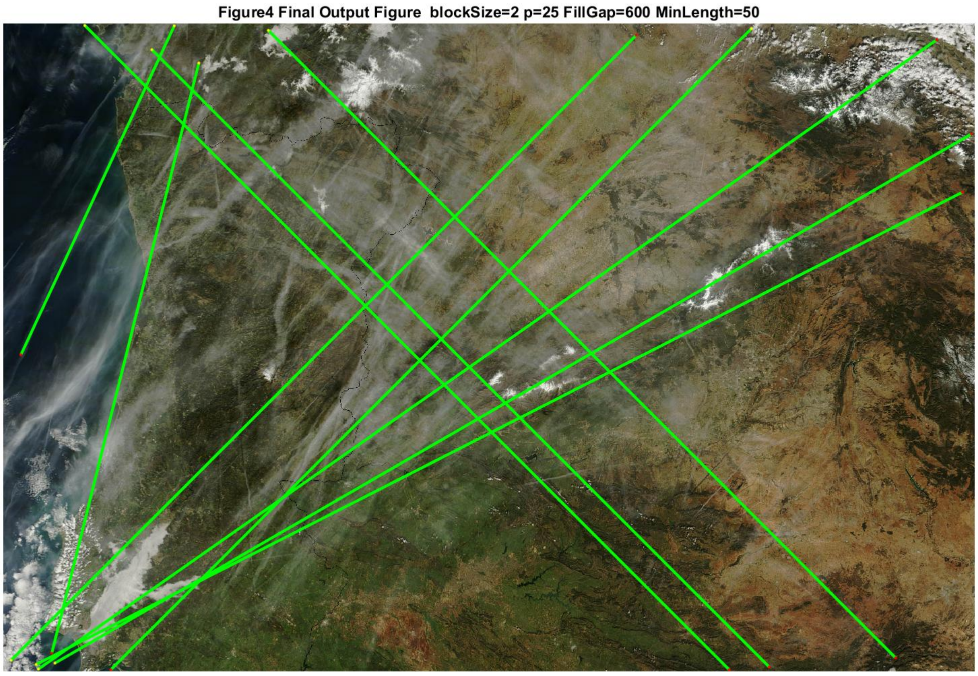

Figure 4.12: Worst Result for Figure 4.10, lines parallel or perpendicular

Block Radius	Percentage	FillGap	Minimum Length	Correct Number	Incorrect Number	Missingr Number	Correct Rate	Incorrect Rate
7	5	200	50	6	3	4	60.00%	33.33%
7	5	200	25	6	4	4	60.00%	40.00%
2	15	200	25	4	1	6	40.00%	20.00%
2	15	200	50	4	1	6	40.00%	20.00%
2	15	200	100	4	1	6	40.00%	20.00%
2	15	400	25	3	0	7	30.00%	0.00%
2	15	400	50	3	0	7	30.00%	0.00%
2	15	400	100	3	0	7	30.00%	0.00%
2	15	600	25	3	0	7	30.00%	0.00%
2	15	600	50	3	0	7	30.00%	0.00%
2	15	600	100	3	0	7	30.00%	0.00%
7	5	400	25	3	1	7	30.00%	25.00%
7	5	400	50	3	1	7	30.00%	25.00%
7	5	400	100	3	1	7	30.00%	25.00%
7	5	600	25	3	1	7	30.00%	25.00%
7	5	600	50	3	1	7	30.00%	25.00%
7	5	600	100	3	1	7	30.00%	25.00%
2	25	200	25	3	8	7	30.00%	72.73%
2	25	200	50	3	8	7	30.00%	72.73%
2	25	200	100	3	8	7	30.00%	72.73%
12	5	200	50	2	2	8	20.00%	50.00%
12	5	200	100	2	2	8	20.00%	50.00%
12	5	400	25	2	2	8	20.00%	50.00%
12	5	400	50	2	2	8	20.00%	50.00%
12	5	400	100	2	2	8	20.00%	50.00%
12	5	600	25	2	2	8	20.00%	50.00%
12	5	600	50	2	2	8	20.00%	50.00%
12	5	600	100	2	2	8	20.00%	50.00%
12	5	200	25	2	3	8	20.00%	60.00%
2	25	400	25	2	8	8	20.00%	80.00%
2	25	400	50	2	8	8	20.00%	80.00%
2	25	400	100	2	8	8	20.00%	80.00%
2	25	600	25	2	8	8	20.00%	80.00%
2	25	600	50	2	8	8	20.00%	80.00%
2	25	600	100	2	8	8	20.00%	80.00%
7	15	200	25	2	10	8	20.00%	83.33%
7	15	200	50	2	10	8	20.00%	83.33%
7	15	200	100	2	10	8	20.00%	83.33%
2	5	200	50	1	0	9	10.00%	0.00%
2	5	200	100	1	0	9	10.00%	0.00%
2	5	400	25	1	0	9	10.00%	0.00%
2	5	400	50	1	0	9	10.00%	0.00%

2	5	400	100	1	0	9	10.00%	0.00%
2	5	600	25	1	0	9	10.00%	0.00%
2	5	600	50	1	0	9	10.00%	0.00%
2	5	600	100	1	0	9	10.00%	0.00%
7	15	400	25	1	10	9	10.00%	90.91%
7	15	400	50	1	10	9	10.00%	90.91%
7	15	400	100	1	10	9	10.00%	90.91%
7	15	600	25	1	10	9	10.00%	90.91%
7	15	600	50	1	10	9	10.00%	90.91%
7	15	600	100	1	10	9	10.00%	90.91%
7	25	200	25	0	10	10	0.00%	100.00%
7	25	200	50	0	10	10	0.00%	100.00%
7	25	200	100	0	10	10	0.00%	100.00%
7	25	400	25	0	10	10	0.00%	100.00%
7	25	400	50	0	10	10	0.00%	100.00%
7	25	400	100	0	10	10	0.00%	100.00%
7	25	600	25	0	10	10	0.00%	100.00%
7	25	600	50	0	10	10	0.00%	100.00%
7	25	600	100	0	10	10	0.00%	100.00%
12	15	200	25	0	10	10	0.00%	100.00%
12	15	200	50	0	10	10	0.00%	100.00%
12	15	200	100	0	10	10	0.00%	100.00%
12	15	400	25	0	10	10	0.00%	100.00%
12	15	400	50	0	10	10	0.00%	100.00%
12	15	400	100	0	10	10	0.00%	100.00%
12	15	600	25	0	10	10	0.00%	100.00%
12	15	600	50	0	10	10	0.00%	100.00%
12	15	600	100	0	10	10	0.00%	100.00%
12	25	200	25	0	10	10	0.00%	100.00%
12	25	200	50	0	10	10	0.00%	100.00%
12	25	200	100	0	10	10	0.00%	100.00%
12	25	400	25	0	10	10	0.00%	100.00%
12	25	400	50	0	10	10	0.00%	100.00%
12	25	400	100	0	10	10	0.00%	100.00%
12	25	600	25	0	10	10	0.00%	100.00%
12	25	600	50	0	10	10	0.00%	100.00%
12	25	600	100	0	10	10	0.00%	100.00%
2	5	200	25	0	10	10	0.00%	100.00%

Table 4.4: Experimental results for Figure 4.10.

4.2.5 Line of Trees and Open Sky

Figure 4.13 is a color image with about 3 contrails which are easier to be recognized, and there exists an apparent road and trees whose edges are close to the straight line.
The best result has the parameters with block radius as 12, percentage to keep as 15, fill gap as 200, and minimum length as 100. The best result cover the contrail, however, it doesnt avoid the highway line in the bottom because its edge is much clear than contrails. And the worst one only covers the highway line in the bottom but no contrails. The results image for figure 4.13 can be seen at figure 4.14 for the best result and figure 4.15 for the worst result.

Figure 4.13: Line of Trees and Open Sky (3264x2448)

Figure 4.14: Best Result for Figure 4.13

Figure 4.15: Worst Result for Figure 4.13, no contrail covered

Block Radius	Percentage	FillGap	Minimum Length	Correct Number	Incorrect Number	Missingr Number	Correct Rate	Incorrect Rate
12	15	200	100	3	4	1	75.00%	57.14%
12	15	200	50	3	5	1	75.00%	62.50%
12	15	200	25	3	6	1	75.00%	66.67%
12	5	200	50	2	1	2	50.00%	33.33%
12	5	200	100	2	1	2	50.00%	33.33%
12	5	200	25	2	2	2	50.00%	50.00%
12	15	600	25	2	4	2	50.00%	66.67%
12	15	600	50	2	4	2	50.00%	66.67%
12	15	600	100	2	4	2	50.00%	66.67%
12	15	400	50	2	5	2	50.00%	71.43%
12	15	400	100	2	5	2	50.00%	71.43%
12	15	400	25	2	5	2	50.00%	71.43%
7	5	200	25	1	0	3	25.00%	0.00%
7	5	200	50	1	0	3	25.00%	0.00%
7	5	200	100	1	0	3	25.00%	0.00%
7	5	400	25	1	0	3	25.00%	0.00%
7	5	400	50	1	0	3	25.00%	0.00%
7	5	400	100	1	0	3	25.00%	0.00%
7	5	600	25	1	0	3	25.00%	0.00%
7	5	600	50	1	0	3	25.00%	0.00%
7	5	600	100	1	0	3	25.00%	0.00%
12	5	400	25	1	2	3	25.00%	66.67%
12	5	400	50	1	2	3	25.00%	66.67%
12	5	400	100	1	2	3	25.00%	66.67%
12	5	600	25	1	2	3	25.00%	66.67%
12	5	600	50	1	2	3	25.00%	66.67%
12	5	600	100	1	2	3	25.00%	66.67%
2	25	600	25	1	3	3	25.00%	75.00%
2	25	600	50	1	3	3	25.00%	75.00%
2	25	600	100	1	3	3	25.00%	75.00%
7	15	600	25	1	3	3	25.00%	75.00%
7	15	600	50	1	3	3	25.00%	75.00%
7	15	600	100	1	3	3	25.00%	75.00%
2	25	200	25	1	5	3	25.00%	83.33%
2	25	200	50	1	5	3	25.00%	83.33%
2	25	200	100	1	5	3	25.00%	83.33%
2	25	400	25	1	5	3	25.00%	83.33%
2	25	400	50	1	5	3	25.00%	83.33%
2	25	400	100	1	5	3	25.00%	83.33%
7	15	200	100	1	6	3	25.00%	85.71%
7	15	400	100	1	6	3	25.00%	85.71%
7	15	400	25	1	7	3	25.00%	87.50%

7	15	400	50	1	7	3	25.00%	87.50%
2	15	200	100	1	9	3	25.00%	90.00%
2	15	400	25	1	9	3	25.00%	90.00%
2	15	400	50	1	9	3	25.00%	90.00%
2	15	400	100	1	9	3	25.00%	90.00%
2	15	600	25	1	9	3	25.00%	90.00%
2	15	600	50	1	9	3	25.00%	90.00%
2	15	600	100	1	9	3	25.00%	90.00%
2	15	200	25	1	9	3	25.00%	90.00%
7	15	200	25	1	9	3	25.00%	90.00%
7	15	200	50	1	9	3	25.00%	90.00%
2	15	200	50	1	9	3	25.00%	90.00%
7	25	200	25	1	10	3	25.00%	90.91%
7	25	200	50	1	10	3	25.00%	90.91%
7	25	200	100	1	10	3	25.00%	90.91%
7	25	400	25	1	10	3	25.00%	90.91%
7	25	400	50	1	10	3	25.00%	90.91%
7	25	400	100	1	10	3	25.00%	90.91%
7	25	600	25	1	10	3	25.00%	90.91%
7	25	600	50	1	10	3	25.00%	90.91%
7	25	600	100	1	10	3	25.00%	90.91%
2	5	200	25	0	4	4	0.00%	100.00%
2	5	200	50	0	2	4	0.00%	100.00%
2	5	200	100	0	2	4	0.00%	100.00%
2	5	400	25	0	4	4	0.00%	100.00%
2	5	400	50	0	4	4	0.00%	100.00%
2	5	400	100	0	4	4	0.00%	100.00%
2	5	600	25	0	3	4	0.00%	100.00%
2	5	600	50	0	3	4	0.00%	100.00%
2	5	600	100	0	3	4	0.00%	100.00%
12	25	200	50	0	2	4	0.00%	100.00%
12	25	200	100	0	2	4	0.00%	100.00%
12	25	400	25	0	2	4	0.00%	100.00%
12	25	400	50	0	2	4	0.00%	100.00%
12	25	400	100	0	2	4	0.00%	100.00%
12	25	600	25	0	1	4	0.00%	100.00%
12	25	600	50	0	1	4	0.00%	100.00%
12	25	600	100	0	1	4	0.00%	100.00%
12	25	200	25	0	2	4	0.00%	100.00%

Table 4.5: Experimental results for Figure 4.13.

4.3 Effect of Parameters

4.3.1 Block Radius

As long as in computer graphics, all the curves are made by the joint of lots of small pieces of line segments. When the block radius is smaller, there must be more blocks can pass the block process to be considered that they are in a straight line. This can make us get more the possible lines. From graph most of the figure results, we can see when the block radius is larger, indeed we can get higher correctness rate. However, if we have more pixels each block pass the algorithm, it means we filter less bad pixels per block, it also increase the incorrectness rate.
Also when the block size is larger, the average time to process will be longer.

4.3.2 Percentage to Retain

As I mentioned in 3.3.1, the less percentage will keep less block in the this step. The block with higher residual value will reminds on the top, thus what we deleted is the blocks less likely to be in the contrails or lines. However, if we have many enough blocks are in lines, a smaller percentage will erase small details of the image which can mislead final result. According to the table, I think 10% is a good value for this process.

4.3.3 Fill Gap

This parameter will help we find better contrails. Because contrails are not strictly straight lines, there are pixels on contrails may not always lay on the same line. this parameter can help us avoid this kind of problem. Like we discussed for 4.7, the too small Fill Gap parameter makes the two lines on same contrail were detected as two.
However, as long as the Hough Transfer cannot tell if the lines are break because the lines are slightly curved or there is actually lines in between. Thus it will also give us some bad concatenation results.
FillGap should be set as property value to make sure it is not either too large or too small when compare with the image size.

4.3.4 Minimum Length

As mentioned in the FillGap, the curves are all made up by small line segments. This parameter is a so important one to make us do not need to see infinitely many line segments. But also, this should keep in a relative threshold of the image to make sure it is in property.

4.4 Problems

This project solution still has some problems:

1. Parameters need to be manually set if the best result is to be obtained, such as the canny edge threshold; the block size; the r value threshold; and Hough Transform minimum length, minimum gap, line numbers. Otherwise, the results would surprise some details when processing the multiple contrails images.

2. The images with multiple contrails that cross each other will be not be processed well because some blocks which have multiple contrails image cross each other might be deleted because of too large r value.

3. Super large image will be processed very slow because of the pixel by pixel processing algorithm. For example, figure4s size is 2768 x 1845, and it took 1902 seconds (31.7 minutes) to process it (details see this project package /rst/figure4.txt).

Chapter 5

Future Work

Further work is needed to solve the problems that exist in current solution. Here are two directions to improve the program to make it more efficient: easier to use and faster to process the image.:

1. Automatically set some parameters, such as the parameters for Canny edge detection, block size, or r value threshold, etc.

2. Find out a way to make this program to be more accurate, for example, in figure1 we should make it auto-remove those lat-lag lines, then we should have a better detecting result.

3. Also consider the color for contrails always in simlar color, then we can aviod some bad situations also.

Chapter 6

Conclusion

This project is designed to recognize the contrails from clouds image. Using canny edge detection, polynomial curve fitting, and Hough Transform to solve this problem.

The result is optimistic. Even we do not have a perfect result, I believe if we try have a much better gradient parameters (such as the block size from 5..10..15 to 5,6,7.., 14, 15), we can get a better results.

Following the algorithms below:

1. Grayscale the original image
2. Use canny edge detection to get the edge
3. By getting the small blocks pixel by pixel, and do the polynomial curve fitting to get the small line information
4. Put pixels survive the block processes into a new image
5. Do the Hough Transform on new image
6. Plot the results

From the results, it can be seen that the problem has been fairly solved. However, still, some problems exist, such as parameters setting, data miss deleted, and time issues.
We can still do some further research on solving this problem, for example, making the program more efficient or easier to use.

Bibliography

[1] CONTRAILS FACTS." US Environmental Protection Agency, https://www3.epa.gov/otaq/regs/nonroad/aviation/afd-051013-001.pdf

[2] Minnis, P., Ayers, J. K., Palikonda, R. and Phan, D. (2004). Contrails, cirrus trends, and climate. Journal of Climate, 17(8), 1671-1685.

[3] Sassen, Kenneth. "Contrail-cirrus and their potential for regional climate change." Bulletin of the American Meteorological Society 78.9 (1997): 1885-1903.

[4] Space Apps 2016, Sept. 2016, https://2016.spaceappschallenge.org/challenges/aero/clouds-or-contrails

[5] "Hough Transform", Wikipedia. Wikimedia Foundation, Sept. 2016, https://en.wikipedia.org/wiki/Hough_transform

thanks

Thanks to my professor John Barron, it is you always helping me to make me finish my first research.