FIBRIL ORIENTATION TRY

Table of Contents

Program explores the use of a radon transform and hough transform to identify the orientation of firils in an image

Before the images are undergone a radon transform and hough transform the images are undergone through a processing algorithm built in Java, this algorithm essentially consists of these steps:

- Adaptive Histogram normalization of the image
- The adaptive histogram equalaizatoin is done so by looping through all the possible values of normalization (i.e. 0-255) and identifying the highest constrast producing normalization by comparing the normalization of an image histogram with an ideally distributed image histogram
- And finally median filtering for noise reduction

(This algoritm can be added as MATLAB code)

Then the image that was processsed through Java is ran through a series of processes in MATLAB, which consists of:

- A second histogram normalization of the image
- A Diffusion Filter to reduce more noise
- A third and final histogram normalization that expands the image intensity range over the whole 8-bit range (i.e 0-255)
- Sharpening of the image
- Thresholding of the image

After the image has undergone those processing steps the resulting image is then ran through the Hough and Radon transforms

ORIGINAL IMAGE

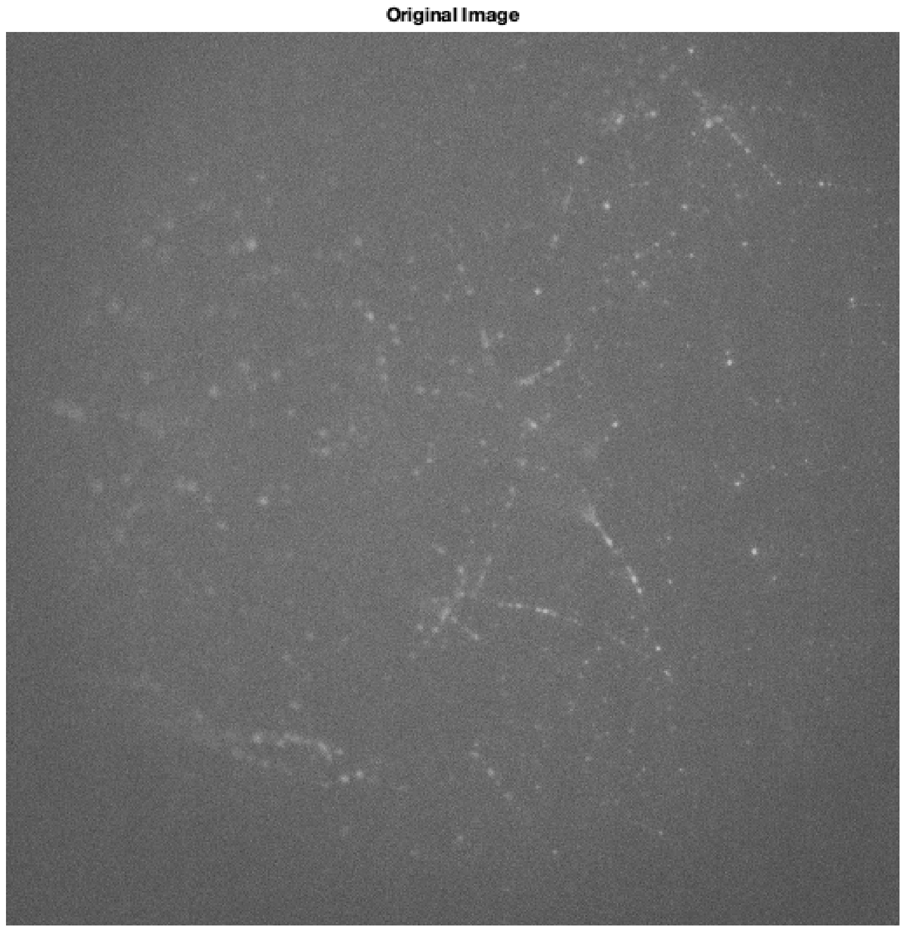


IMAGE PROCESSED IN JAVA

See above for processing steps

Image Processed in Java

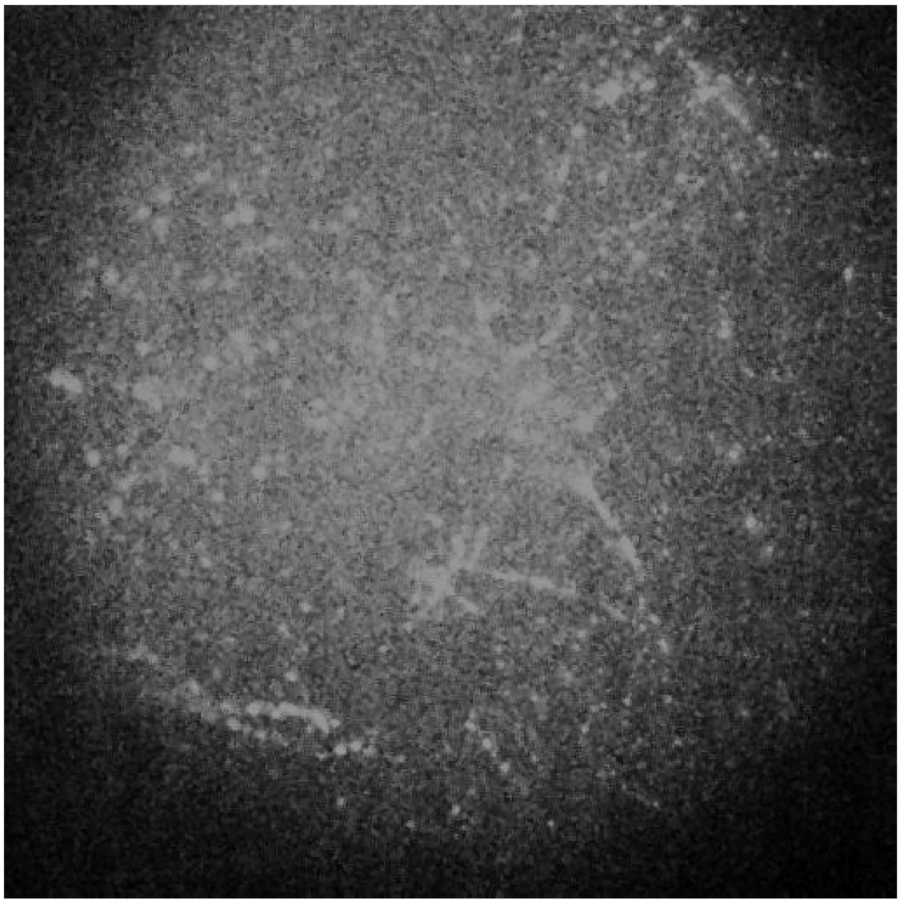
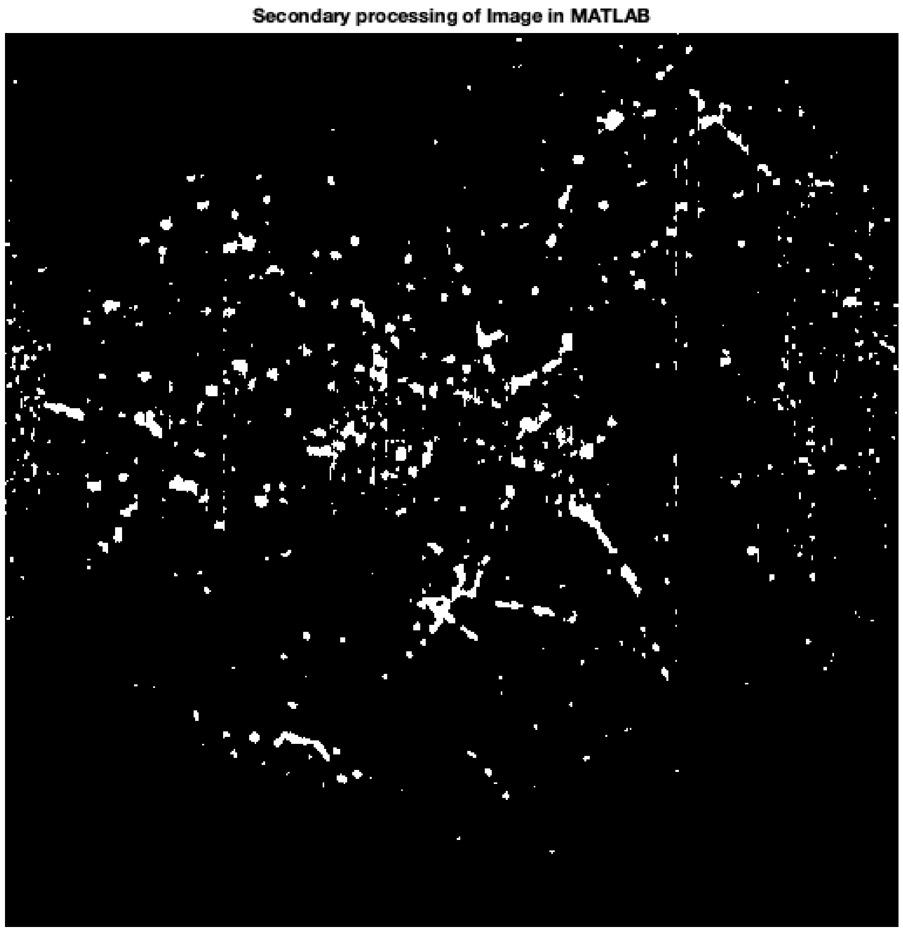


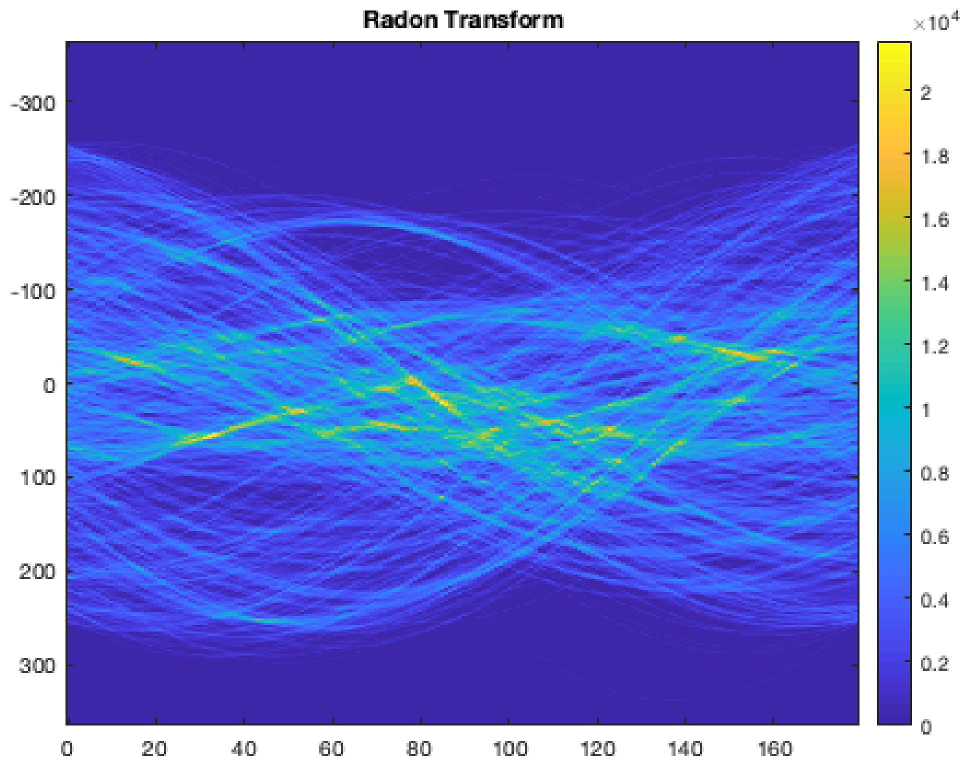
IMAGE PROCESSED IN MATLAB

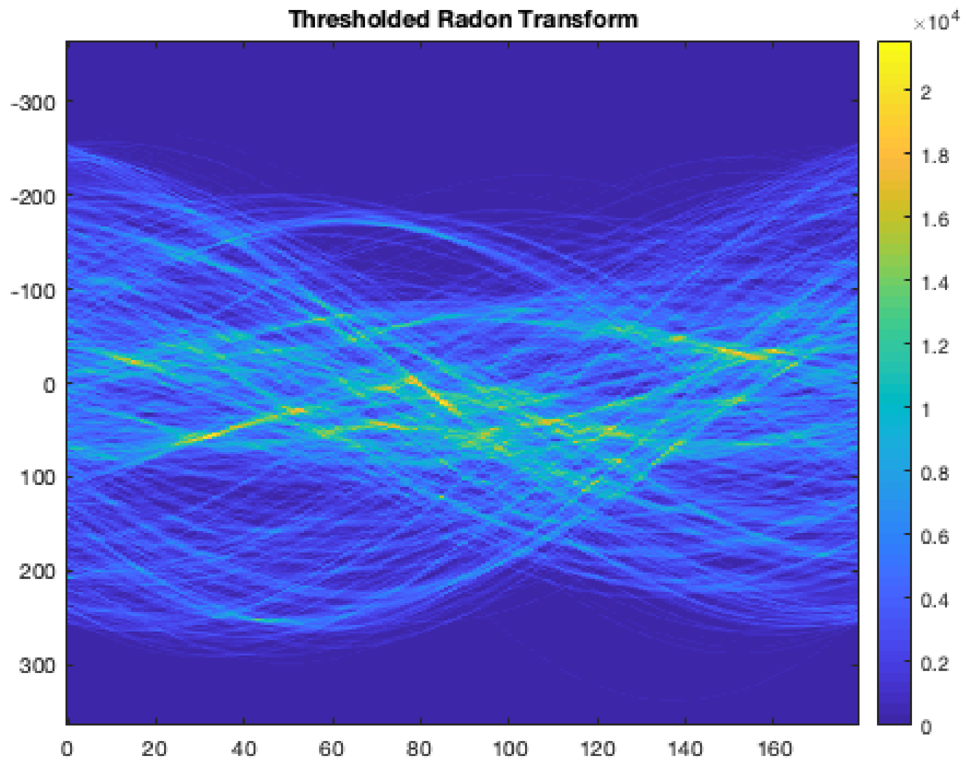
See above for processing steps

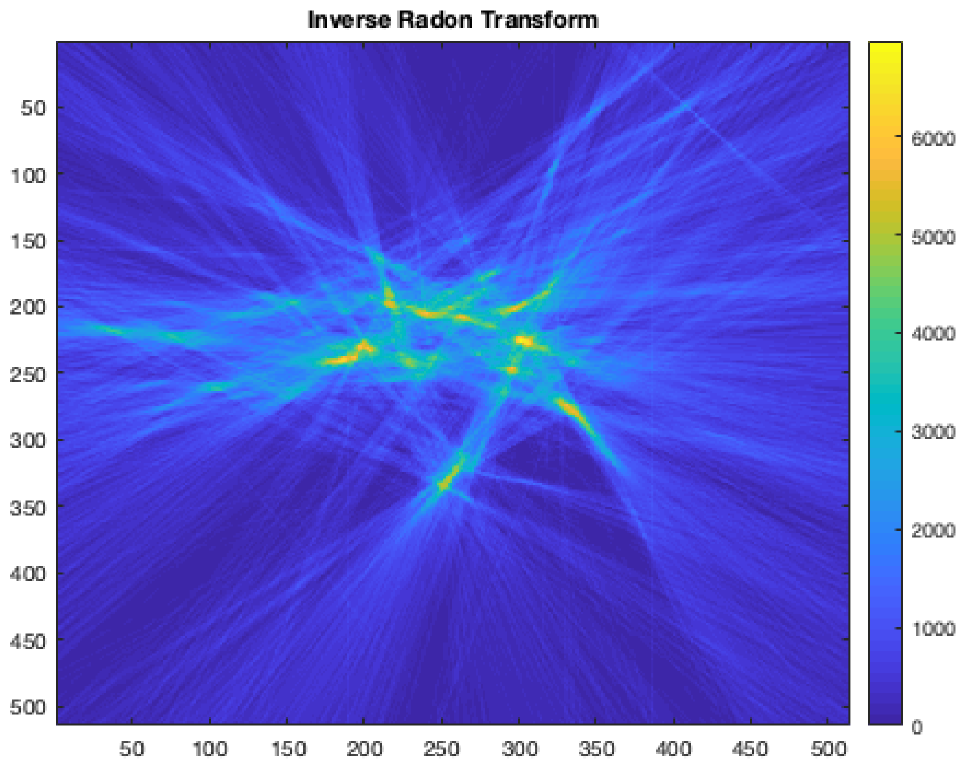


RADON TRANSFORM

Here the image is essentially first run through a Radon transform using the radon() function in MATLAB. Then the image is 'filtered' by thresholding out all the values that are less than the maximum value of the Radon transform. The inverse radon transform is then taken of that thresholded Radon transform using the iradon() function in MATLAB.

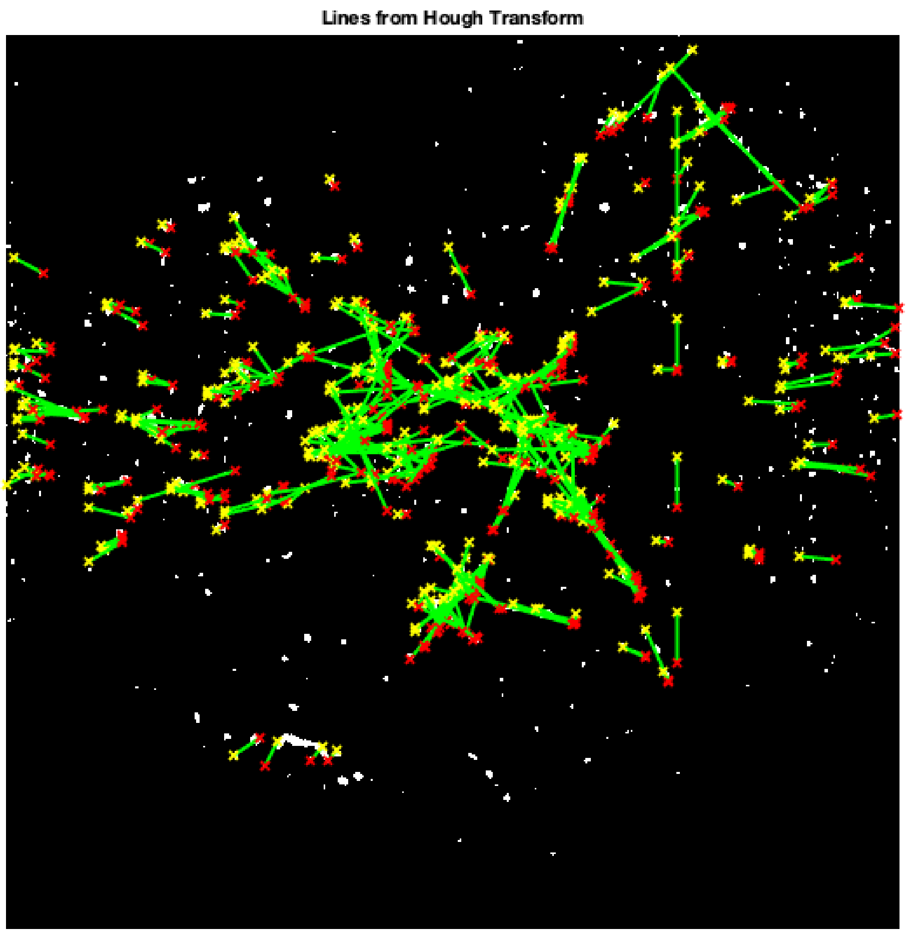

Radon Transform
×10^4
-300
-200
-100
0
100
200
300
0
20
40
60
80
100
120
140
160
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0


Thresholded Radon Transform
×10^4
-300
-200
-100
0
100
200
300
0
20
40
60
80
100
120
140
160
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0



HOUGH TRANSFORM

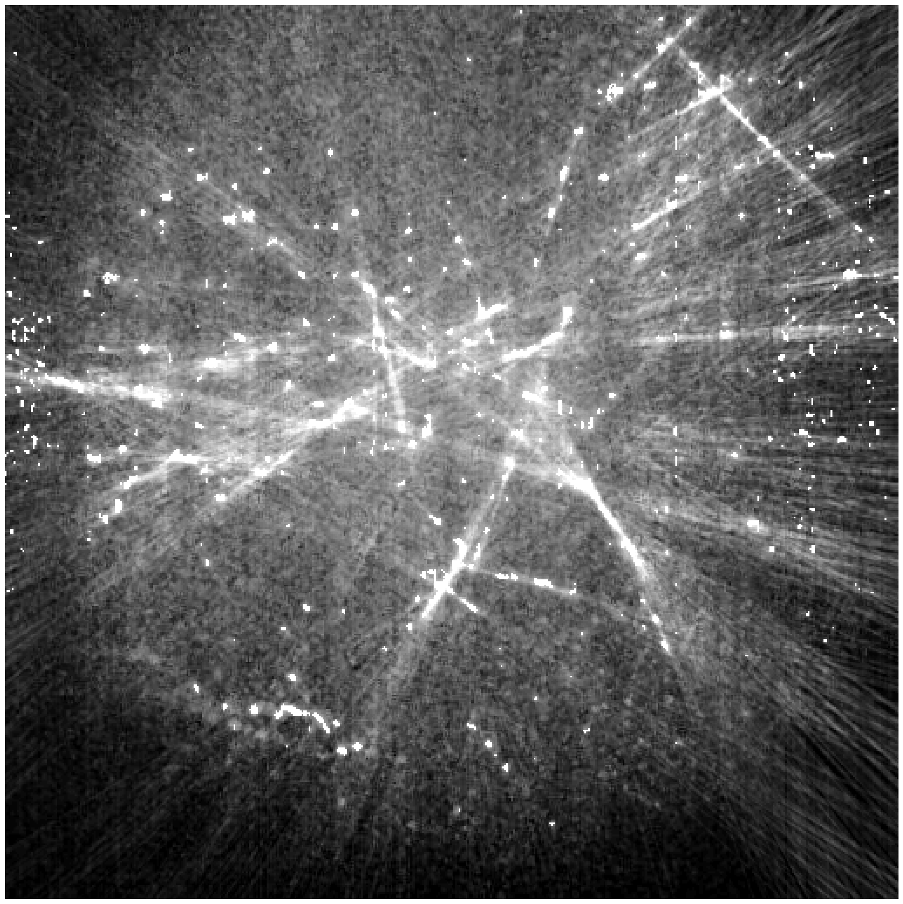
Here the image is essentially ran through the funtions provided by MATLAB to perform a hough transform to identify lines in the image. These functions are hough(), houghpeaks(), and houghlines().



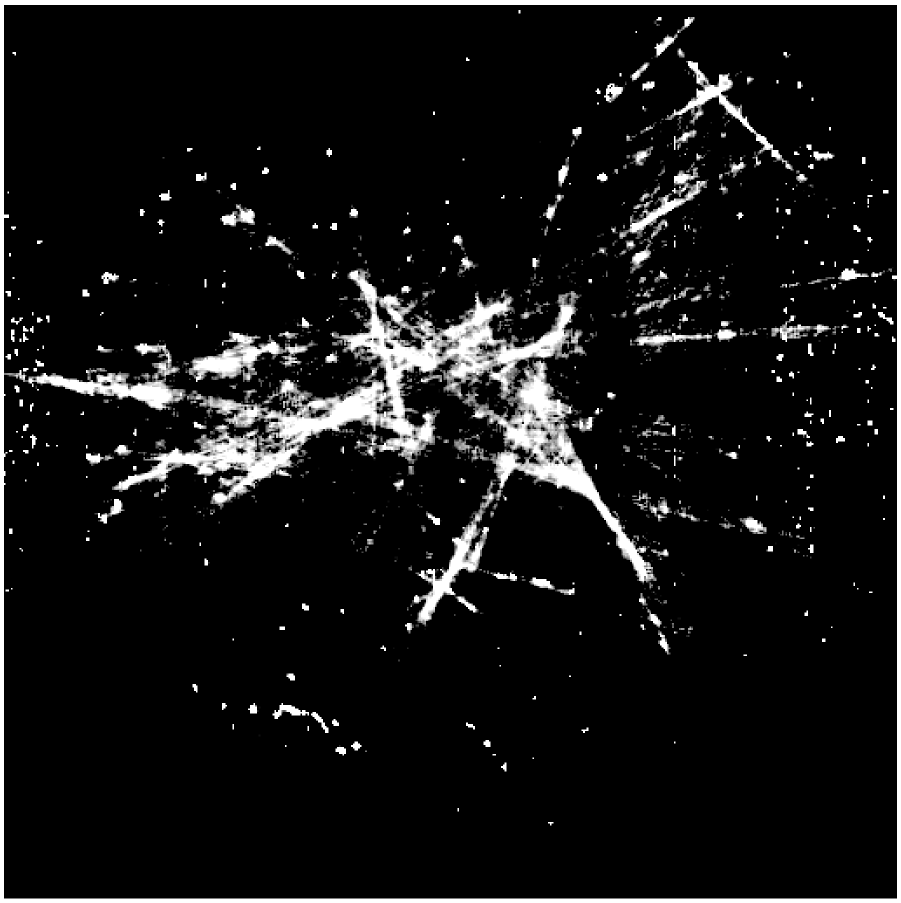
RADON AND HOUGH TRANSFORM COMBINATION ATTEMPT

Here is an attempt at combining the two transforms together by first summing all values of the processed image, the image of the inverse radon transform, and the image of the detected lines from the hough transform. Then the image values are normalized to be in uint8 data.

Combined Image of Processed Image, Radon Image, and Hough Image



Thresholded Image of Combined Image



Published with MATLAB® R2019a