EE475 HOMEWORK5

BERKAY GÜMÜŞ
2015401183

I. CLUSTERING

I used Euclidean distance from seed points selected randomly to other all points compared.

For each pixel I at the RGB images:

$$L(i,j) = argmin_{(n=1,2,...N)} \left\| \begin{bmatrix} I(r)(n) \\ I(g)(n) \\ I(b)(n) \end{bmatrix} - \begin{bmatrix} M(r)(i,j) \\ M(g)(i,j) \\ M(b)(i,j) \end{bmatrix} \right\|$$

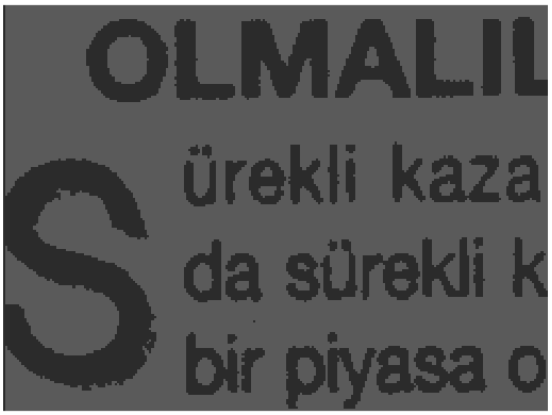
For each pixel I at the gray images:

$$L(i,j) = argmin_{(n=1,2,...N)}(\|I(i,j) - M(n)\|)$$

Firstly, all points are labeled until remaining constant and the first image is labeled image(gray and consists of 1,2,..,N values) and colors are irrelevant to image colors. Secondly, each pixel at the second images has mean color value of its label.

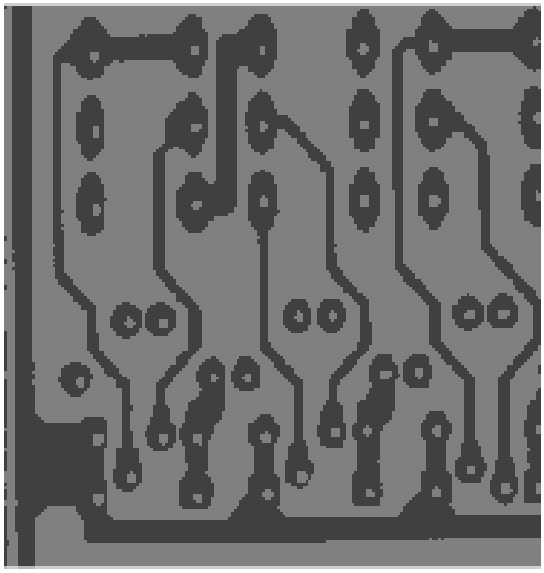
For the newspaper image, I selected N as 2 and mean values are 90.15 43.34

labeled newspaper (N=2)

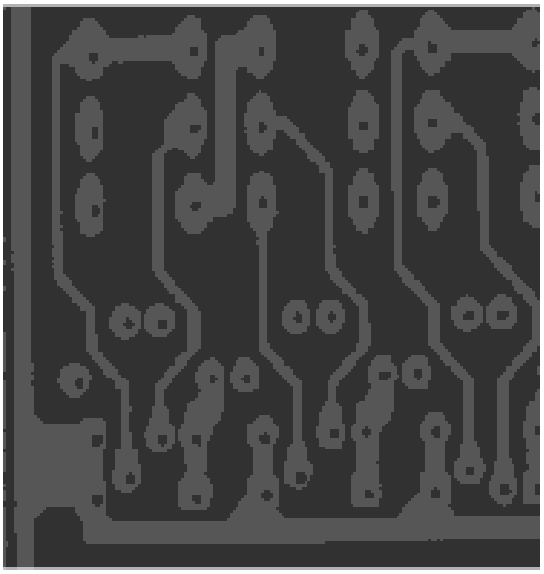


colored newspaper (N=2)

For the PCB image, I selected N as 2 and mean values are 86.35 and 166.67

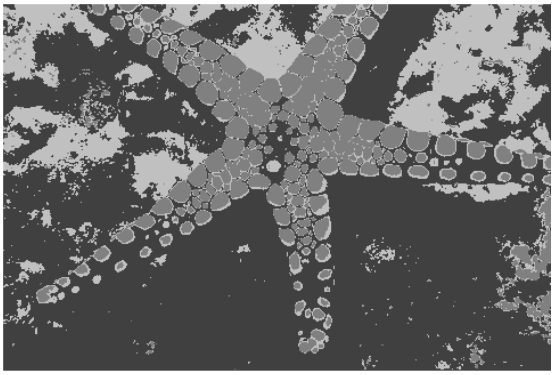


labeled PCB (N=2)

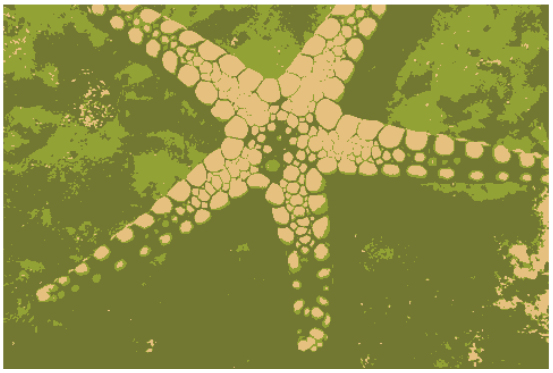


colored PCB (N=2)

For the starfish image, I selected N as 3 and mean values are $\begin{bmatrix}114.11\\118.72\\50.28\end{bmatrix}$, $\begin{bmatrix}230.39\\192.25\\126.42\end{bmatrix}$ and $\begin{bmatrix}41.88\\57.49\\22.79\end{bmatrix}$.



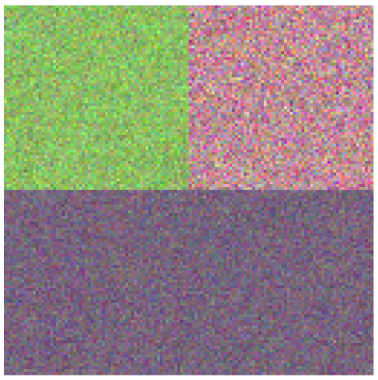
labeled starfish (N=3)



colored starfish (N=3)

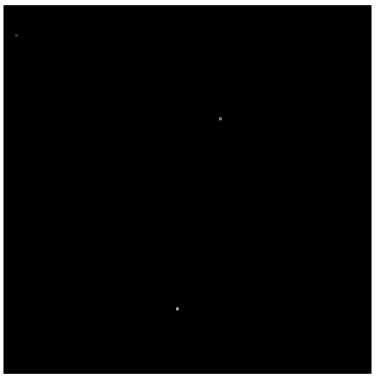
II. REGION GROWING

A. *Square Image*

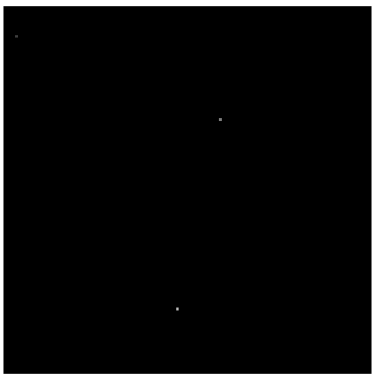


image

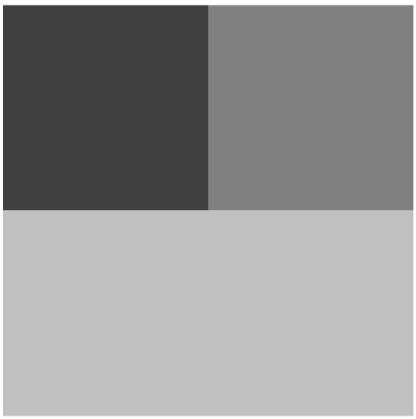
This image is 128X128 pixel^2 and I selected pixels at (5,11), (76,40) and (61,106) as seeds. Distances between seeds and adjacent pixels are more than 5 so no any pixels other than seeds can not be labeled at the first iteration and at the iterations for threshold 5. When the result image is compared to ground truth image, success rate is 0.99805.



the first iteration



iterations for threshold 5



ground truth

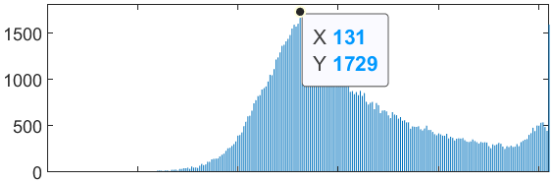
result image (success=0.99805)

B. Horses Image

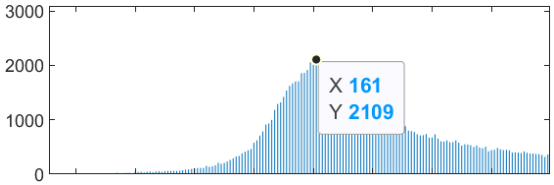
image

This image is a bit complicated and the result success depends on the positions of seeds and values. So, I investigated their(background, horse1 and horse2) histograms.

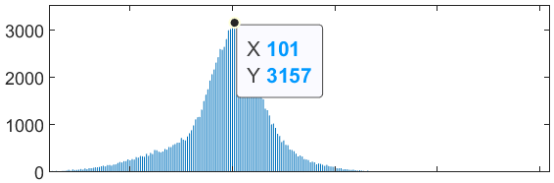
background



histogram of red component

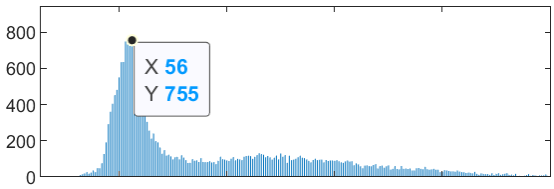


histogram of green component

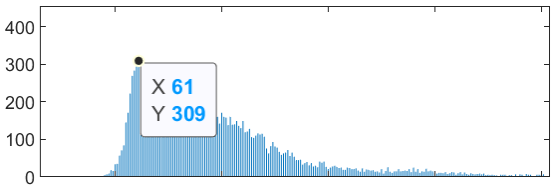


histogram of blue component

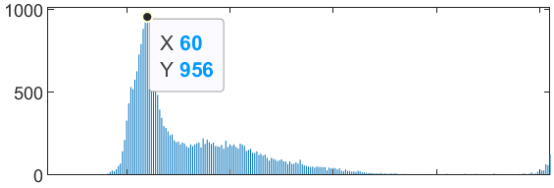
horse 1



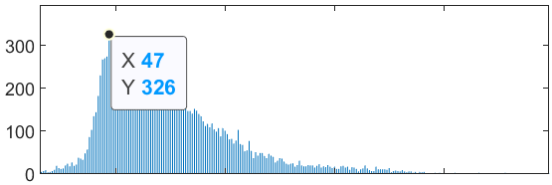
histogram of red component



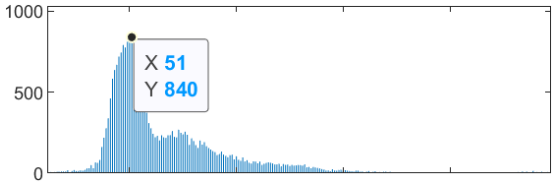
histogram of green component



histogram of green component



histogram of blue component



histogram of blue component

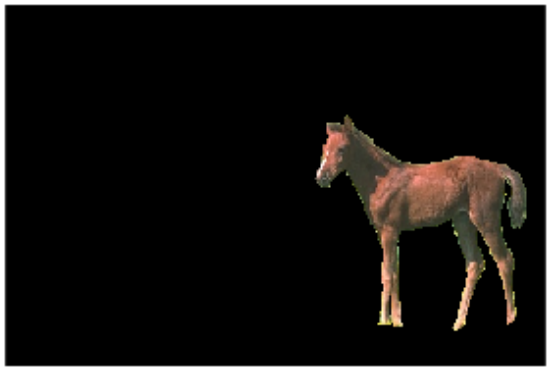
I try to select seed points whose color values are similar to these peak values of histograms. Peak values for the background, horse1 and horse2 are

$$\begin{bmatrix}131\\161\\101\end{bmatrix}, \begin{bmatrix}56\\60\\51\end{bmatrix} \text{ and } \begin{bmatrix}62\\61\\47\end{bmatrix} \text{,orderly.}$$

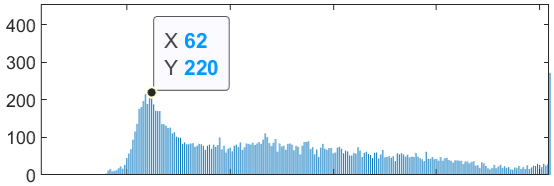
I select the pixels at (280, 472) for background, (134, 254) for horse1 and (128, 308) for horse2 as seeds and their values are

$$\begin{bmatrix}170\\195\\112\end{bmatrix}, \begin{bmatrix}54\\56\\51\end{bmatrix} \text{ and } \begin{bmatrix}62\\58\\47\end{bmatrix} \text{,orderly.}$$

I select a seed with more values than peak values for background because background has many pixels at the right side of the peak values because of yellow pixels. The success rate is 0.88851.

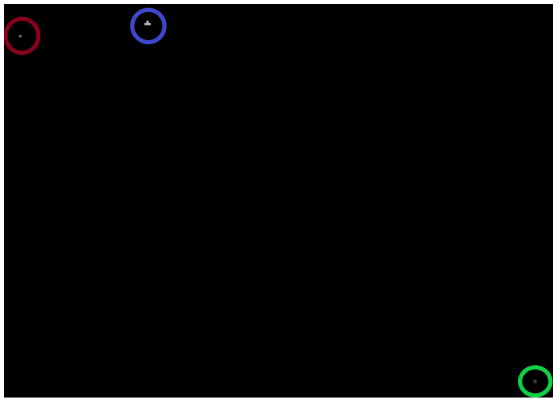


horse 2



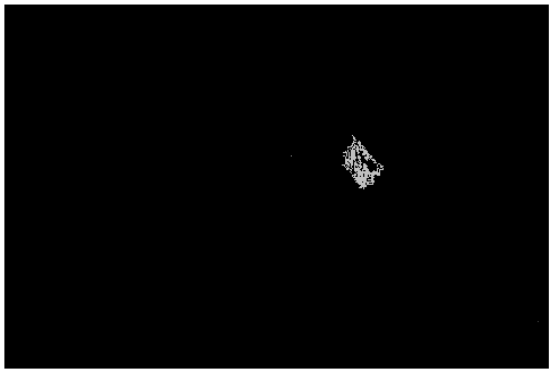
histogram of red component

one iteration for each seed

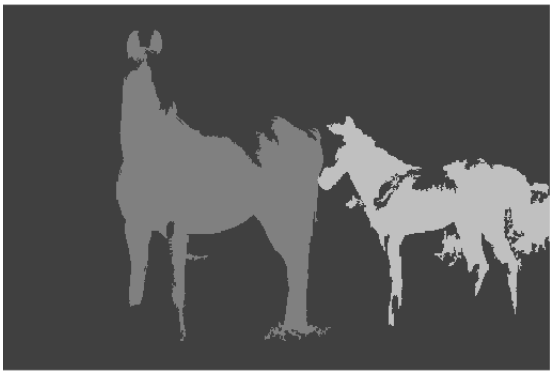


zoom in

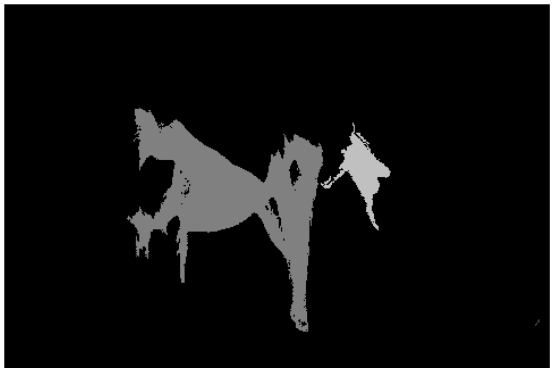
iterations for threshold 65



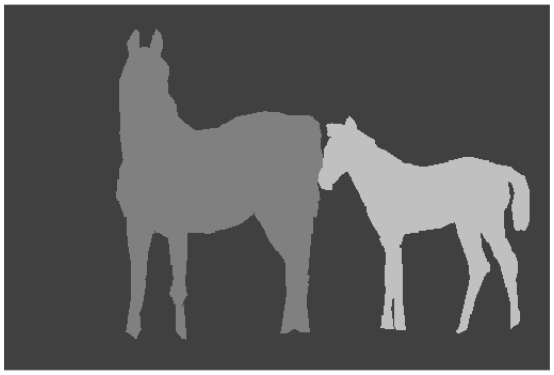
iterations for threshold 5



result (success=0.88851)

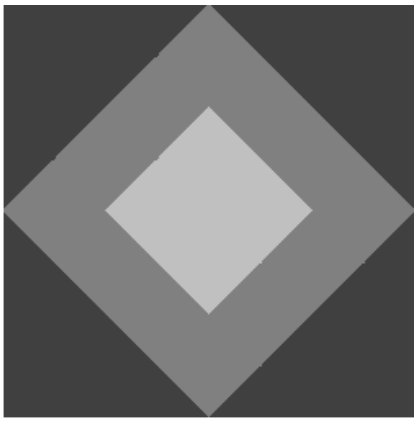


iterations for threshold 25

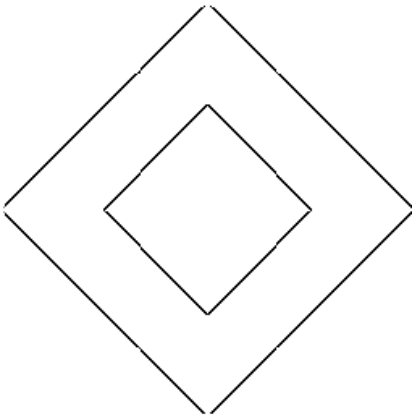


ground truth

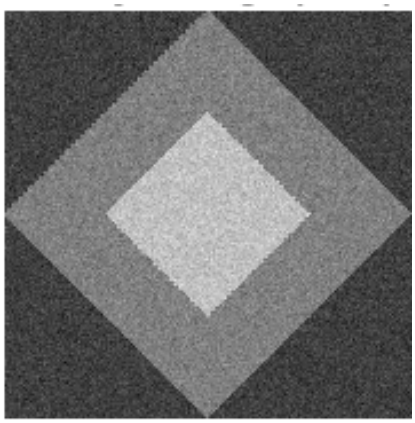
III. EDGE DETECTION



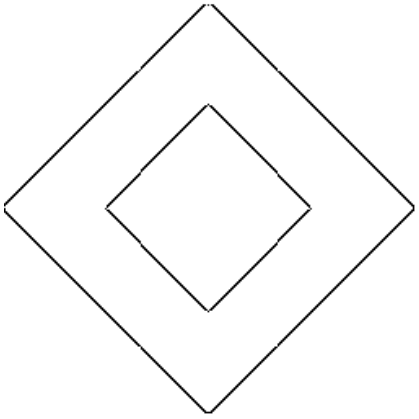
diamond image



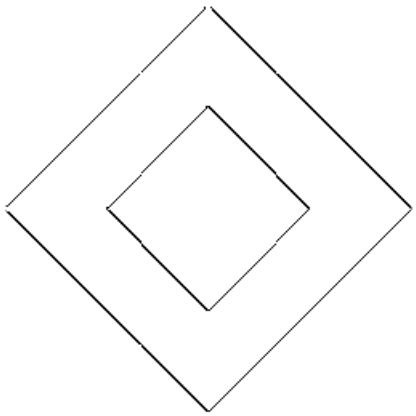
LoG edge detection for image



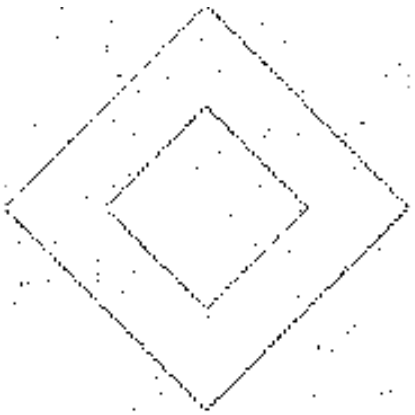
noisy image with variance 144



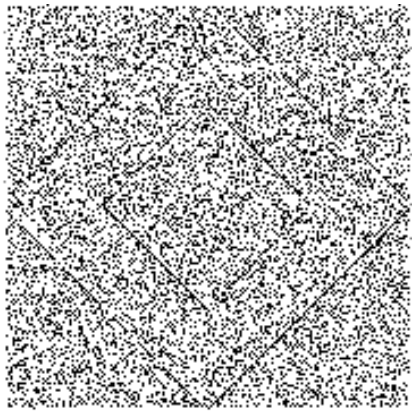
canny edge detection for image



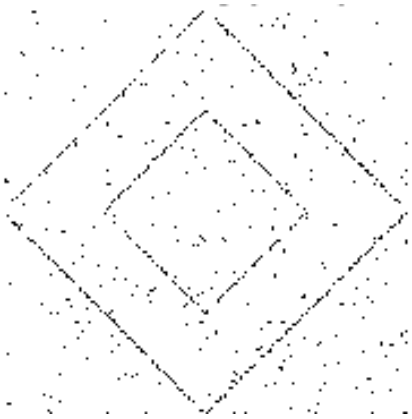
sobel edge detection for image



sobel edge detection for noisy image(144)

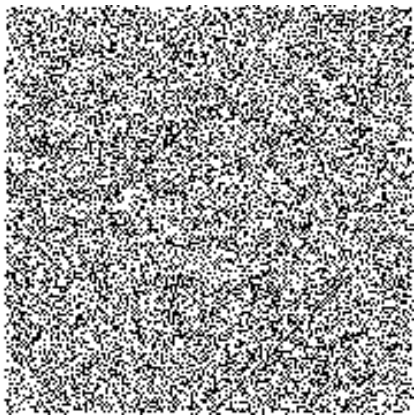


LoG edge detection for noisy image(144)

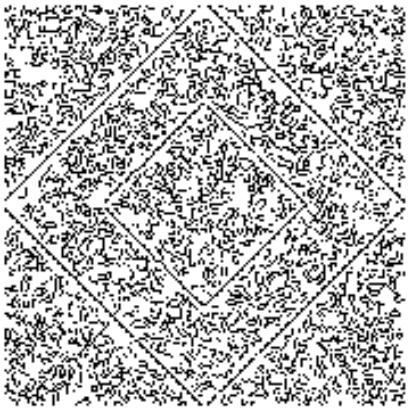


sobel edge detection for noisy image(484)

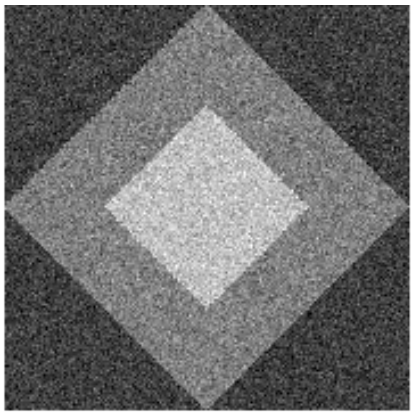
canny edge detection for noisy image(144)



LoG edge detection for noisy image(484)



canny edge detection for noisy image(484)



noisy image with variance 484

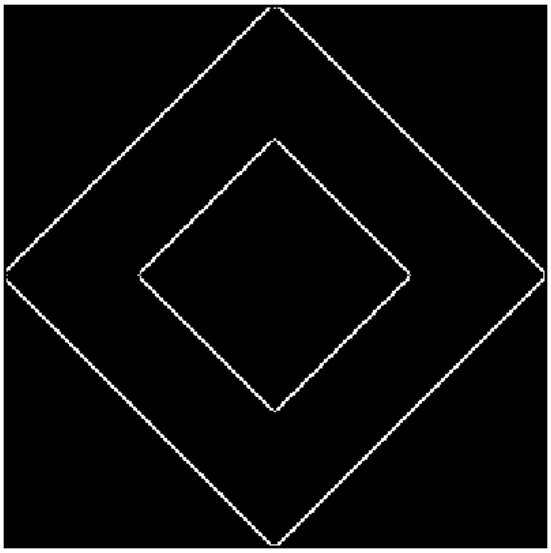
images vs methods	Sobel	LoG	Canny
image with variance 144	0.7883	0.8796	0.9960
image with variance 484	0.6986	0.8234	0.9653

Edge Performances

When only edge pixels at ground truth image are considered, the best edge detector is canny and the second best is LoG

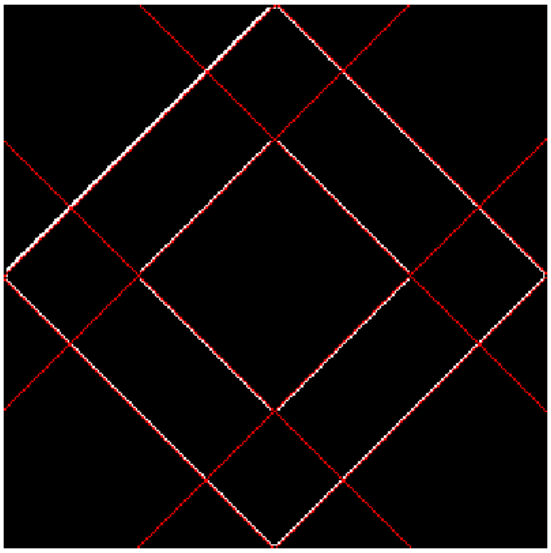
and the worst is Sobel. However, when the result images are considered and checked, Sobel edge detection gives better results for each noisy image, LoG and canny give unsuccessful results for noisy images.

IV. HOUGH TRANSFORMATION



canny edge detection

I set the angular resolution at 1 degree and distance at 1 pixel. The image is 256X256 pixel^2 and N is 256 because of 1 pixel distance. Angle range is (-90,90) and distance range is $(-N\sqrt{2}, N\sqrt{2})$. N is 256 and the distance range is (-363, 363). The peak values(ro,degree) at hough transformation are (91,-45), (45,-45), (-45,-45), (-91,-45), (91,45), (136,45), (226,45) and (272,45) for 8 edges. These 8 hough lines(red lines) can be seen at the image below.



edge detection and hough lines

Hough Transformation