Robot Vision [06-25024]
Summative Assignment 1

March 17, 2023

Submission deadline:	**17:00pm (BST), Friday, 17 March 2023**
Instructor:	Dr. Hyung Jin Chang
	Dr. Jianbo Jiao
Total marks:	100
Contribution to overall module mark:	25%
Submission Method:	This assignment must be submitted through Canvas.
Name:	Sagar Vincent
Username:	svx234
Study program:	MSc Robotics

Part 1

- code **sxv234_assignment1_part1.m**

Question 1.1 A picture of mallard was used to perform edge detection. The kernel used for detecting the edges is the Roberts kernel. The kernel was loaded from a predefined MatLab file and the image was converted from RGB to gray scale before performing the convolution. The final image is as shown below :

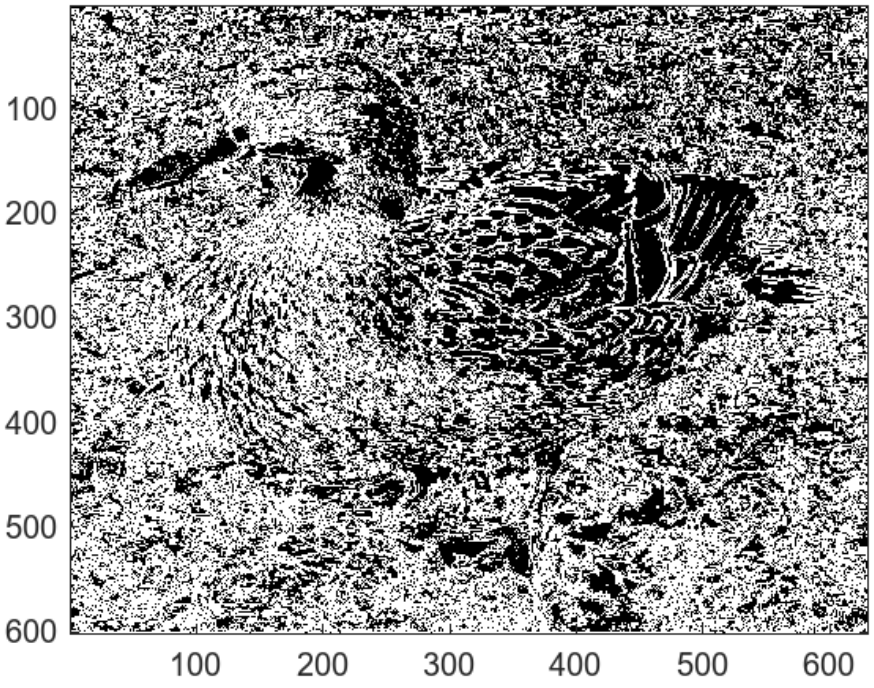


Figure 1: Edge detection using Roberts kernel(with edges greater than 50 threshold)

We can see that in the image, the edges of mallard and background is not visible clearly at bottom left, while the edges are fairly visible in top right. This might due to small size of Roberts kernel, and its inability to gather more information to filter noise.

Question 1.2 In the two matrix given the first matrix is Isotropic while the second one is anisotropic as shown in the figure below:

mat1 = 6×6

0	-1	-1	-1	-1	0
-1	-2	-3	-3	-2	-1
-1	-3	12	12	-2	-1
-1	-3	12	12	-3	-1
-1	2	-3	-3	-2	-1
0	-1	-1	-1	-1	0

(a) Filter 1

mat2 = 6×6

1	1	1	1	1	1
-1	-2	-3	-3	-2	-1
-1	-3	-4	-4	-3	-1
1	3	4	4	3	1
1	2	3	3	2	1
-1	-1	-1	-1	-1	-1

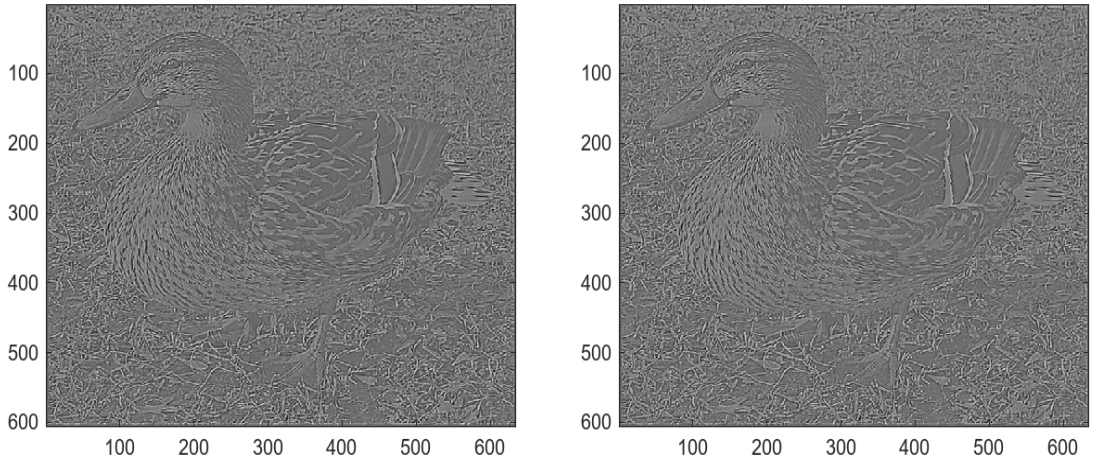
(b) Filter 2

Figure 2: Filters given

The difference between isotropic and anisotropic filter is that, isotropic filter produces the same effect irrespective of the direction of application of the figure. Therefore the nature of each becomes clear by comparing the results produced by these matrix and their transpose. In case of filter one, the images before and after transposing is the same. But for the second filter the transpose produces different image even revealing details not present in the result produced by the original matrix.

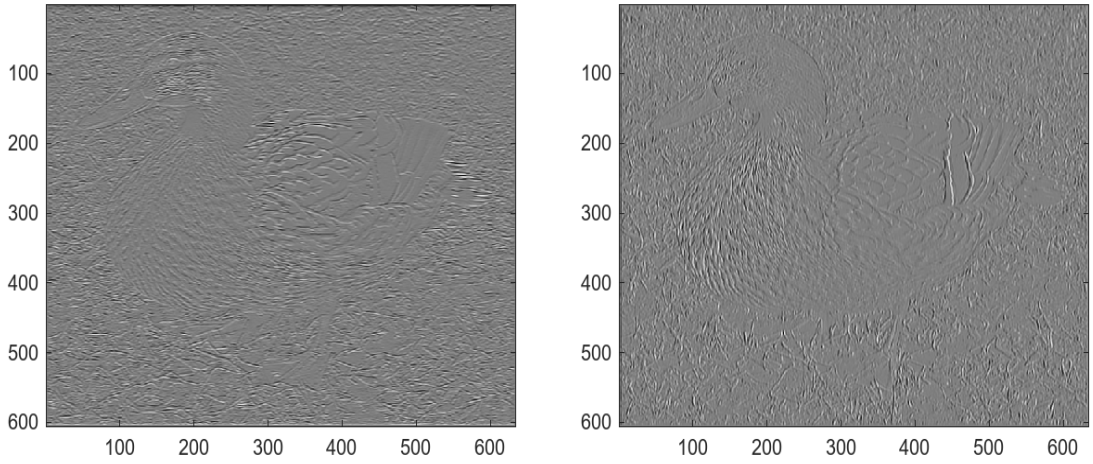
The choice of filters for edge detection is dependent on the data(images used) and cannot be generalised. Usually anisotropic filters such as Sobel's filter and Canny is preferred for edge detection. This is due to fact that isotropic filters tend to smooth out the image.But a particular scenario in which isotropic filters perform better is when is the noise in the image is higher or when the edges are not oriented in any particular direction.This is evident here as based on the results produced the isotropic filter performs better on the *malards.jpg* as the edges are more visible. In case of filter 2, the two images produced are different in the sense, the first image has horizontal edges dominating, while the image produced by its transpose has the vertical edges dominating.

As we discussed, filter one is most useful when the edges in the image are not oriented in any particular direction. There the use of filter one is recommended for uniform texture images. Filter two can be used for edge detection in cases where the required edges are aligned in one direction, or to find edges that are aligned in a certain direction.



(a) Convolution with the Filter 1 (b) Convolution with transpose of filter 1

Figure 3: Images produced by Filter 1



(a) Convolution with filter 2 (b) Convolution with transpose of filter 2

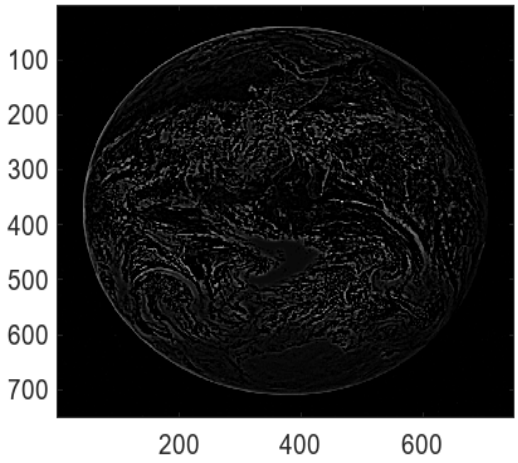
Figure 4: Images produced by Filter 2

Question 1.3 The Gaussian filter has the highest value in the centre element and value decreases from centre to edges. Each value in the filter matrix is obtained by the formula :

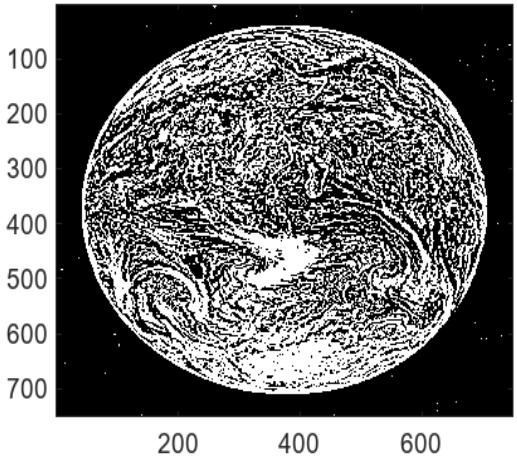
$$M(i,j) = \frac{1}{(2\pi\sigma^2)}\, e^{-\frac{(i-(n+1)/2)^2 + (j-(n+1)/2)^2)}{(2\sigma^2)}} \tag{1}$$

The *(i,j)* corresponds to indexes of the filters, n=*15* is the filter size and σ=*5* is the standard deviation of the underlying Gasussian distribution of the filter. The above equation is used to create two different Gaussian filters to calculate the Difference of Gaussians(DoG). Then the original image is transformed using the Laplacian of Gaussian operator to get a laplacian tranformation of the image. The DoG images are converted to binary for better visualisation. The threshold used for the binary conversion is *0.1*. Then the DoG image is compared with original image and then it is compared with Laplacian of Gaussian image to assess how efficient the approximation is. We can see from the image that approximation is fairly good.

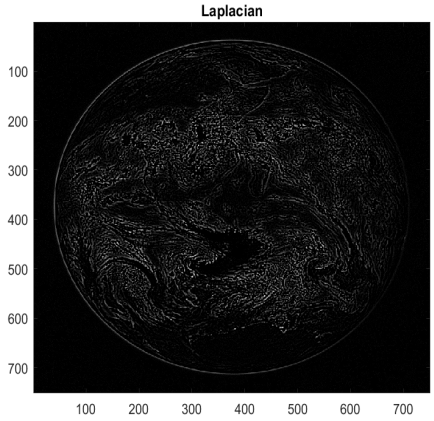
(a) Original Image



(b) Image after application of DoG



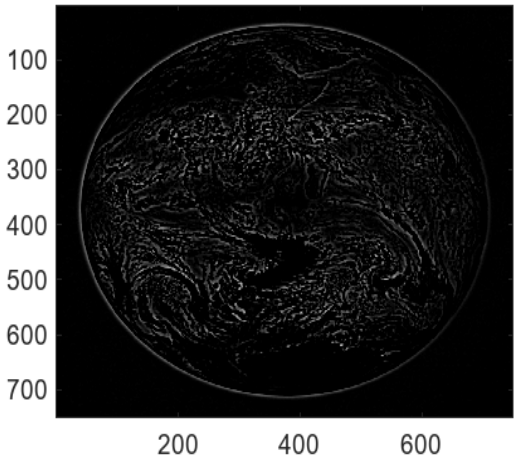
(c) DoG and converting to binary



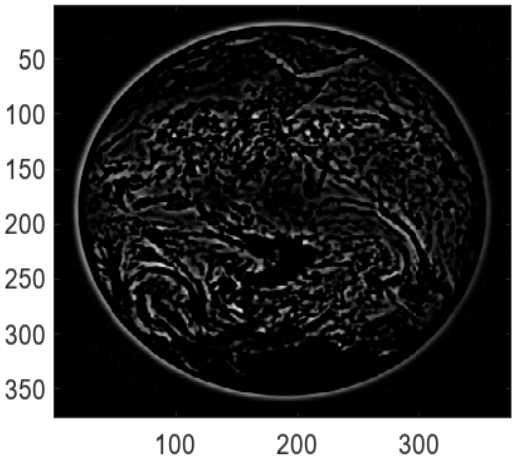
(d) Image after application LoG

Figure 5

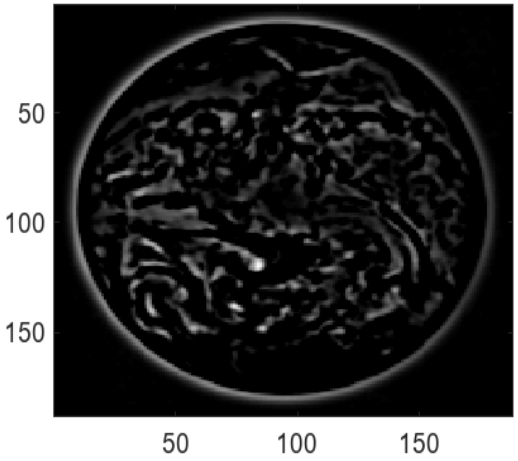
Question 1.4 The previous function DiffGauss is used to produce a scale space pyramid with 4 different octaves and 2 scales $(\sigma=1,\sigma=2\sqrt{2})$ per octave. The Scale space images are converted to binary with a threshold of 0.01 before constructing the pyramid for ease of visualisation.



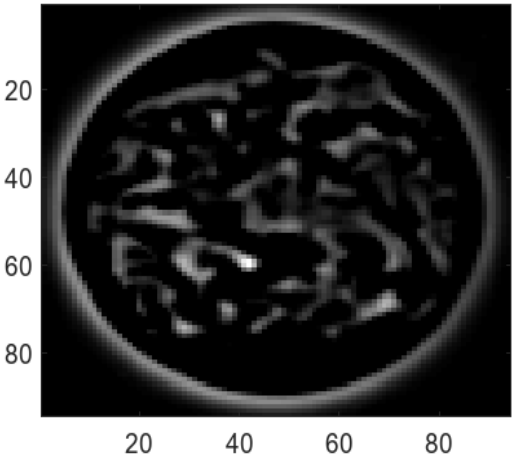
(a) First Octave



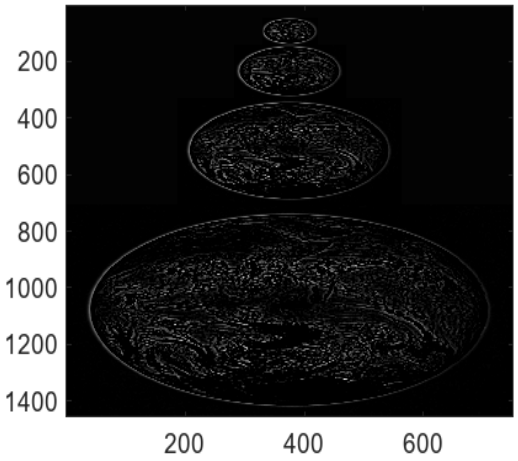
(b) Second Octave



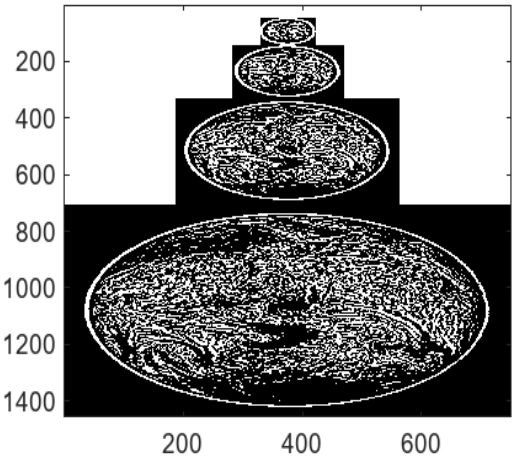
(c) Third octave



(d) Fourth Octave



(e) Scale space period in grayscale



(f) Scale space Pyramid in binary

Part 2

For this task, you will have to submit the following file:

- code **sxv234_assignment1_part2.m**

Question 2.1 The following figure shows a convolution of Gaussian filter over a image.

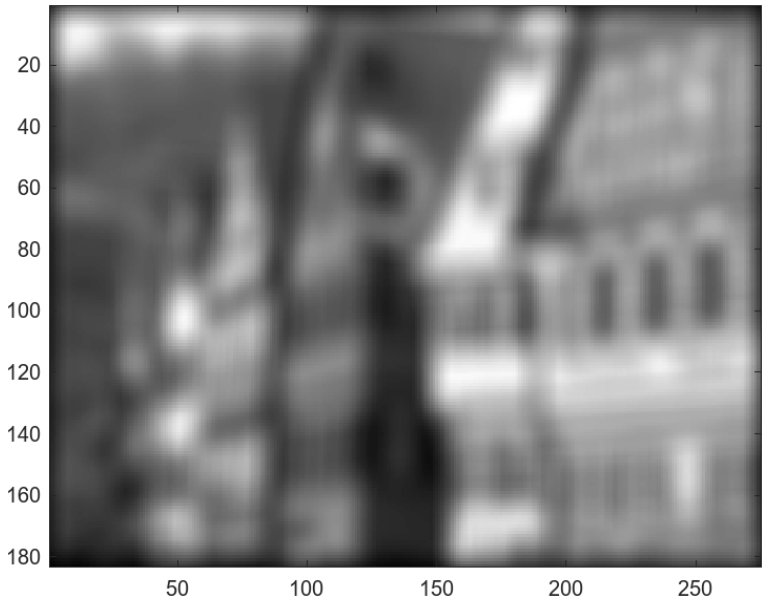


Figure 7: Convolution using 15x15 Gaussian filter

Question 2.2 The following figure shows a convolution of mean filter over a image.

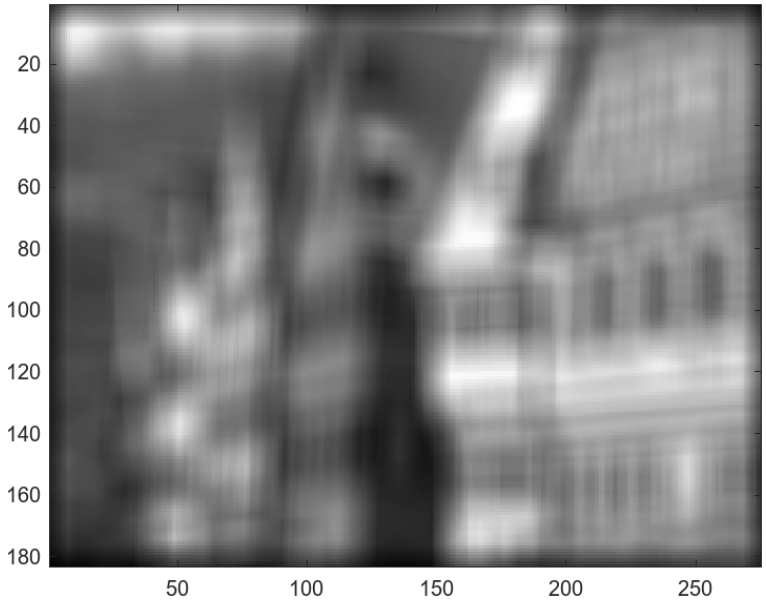


Figure 8: Convolution using 15x15 Mean filter

Question 2.3 The following figure shows a convolution of median filter over a image.

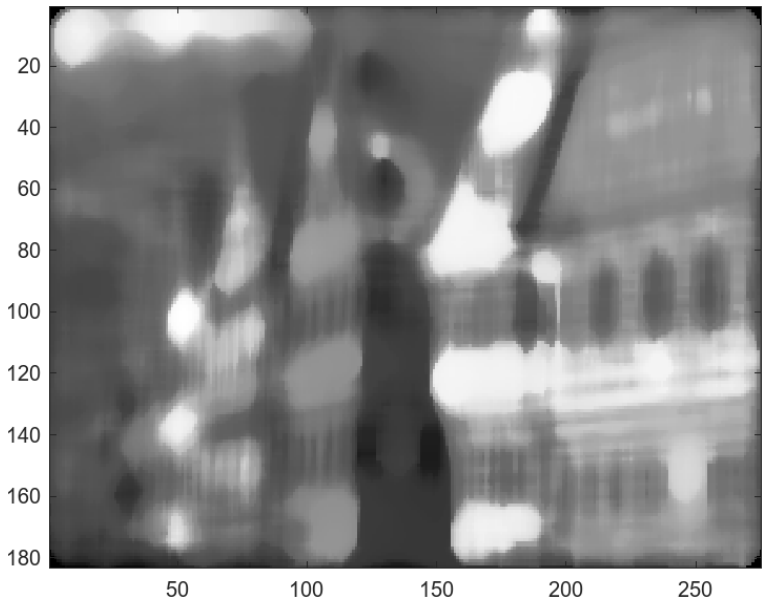


Figure 9: Convolution using 15x15 Median filter)

Question 2.4 The following figure shows a comparison of filters over a image. The filters used is a Gaussian filter *(Mean = 0,std =5,15-by-15)*, Mean filter *(15-by-15)* and a median filter *(13-by-13)*. In this we can see that, for image corrupted with salt and pepper noise the median filter works well compared to others. This might be intuitive already due to the non linear nature of the median filter and the impulse noise. On image corrupted with Gaussian noise, there is no significant difference between the filters. Both the Gaussian and mean filters successfully reduces noise but also significantly diminishes the quality of the image due its blurring effect. In case of Median filter, the noise reduction is less but the edges are well preserved.

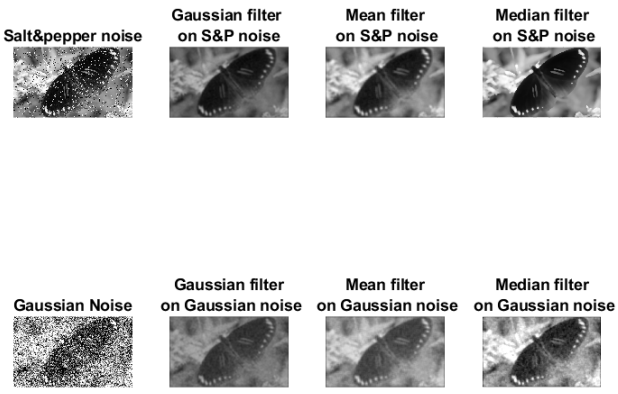


Figure 10: Comparison of effect of different filter on different types of noise

The following case shows the effect of the same set of filters but with a reduced kernel size of *5-by-5*. We can see that although the quality of images produced increases significantly the relative observations we had in earlier case still holds.

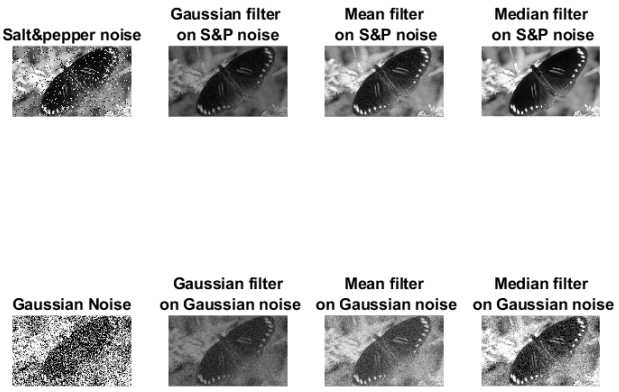


Figure 11: Comparison of effect of different filter with reduced kernel size on different types of noise

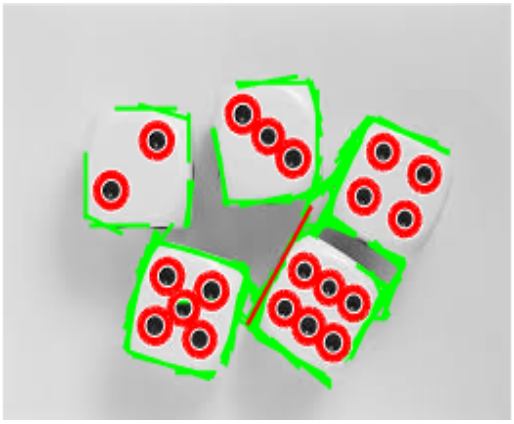
Part 3

For this task, you will have to submit the following file:

- code **sxv234_assignment1_part3.m**

Question 3.1 To obtain the image shown below, a combination of canny edge filter and Roberts edge filter was used. The canny edge filters was used to highlight the edges in image, which included both lines and circles where as the Roberts filter highlighted the circles. A combination of both was used to extract the line edges in the image before applying the Hough tranform. We give the *Rhoresolution* parameter value of the Hough transform as *0.5*. Further we specify the maximum no. of Hough peaks to be discovered as 200, the min. length of the line segments to be 20, and the distance threshold between the line segments to be 5. Any line segment below this value will be merged together.

(a) Original Image



(b) *(Green lines:Line segments,Red line:Longest line segment, Red circles:Circles)*

Part 4

For this task, you will have to submit the following file:

- code **sxv234_assignment1_part4.m**

Question 4.1 The following image shows the detection of 100 features using different algorithms.

Figure 13: Original Image

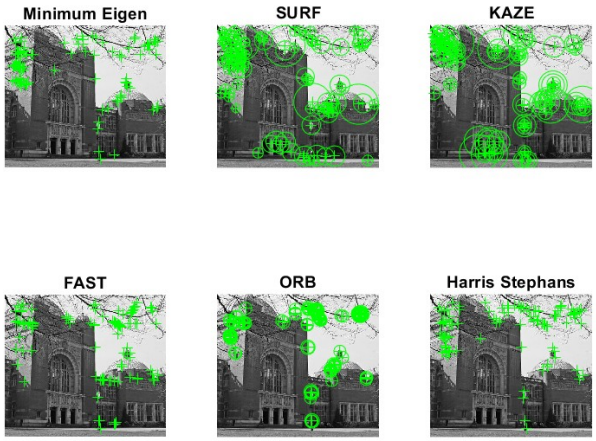


Figure 14: Subplot showing 100 features detected using Minimum eigen value, SURF, KAZE, FAST, ORB, Harris Stephans