CSCI576
Fall 2018
Prof. Parag Havaldar

Assignment - 1

Pavan Athreya Narasimha Murthy
USC ID: 9129210968
E-mail: pavanatn@usc.edu

Note: Please read the README.txt file before running the program

Written Part

Q1:

Lines per frame = 450
Pixels per inch = 520
Frame rate = 25Hz
Color Sub-sampling scheme: 4:2:0
Pixel Ratio: 16:9
Quantization: 8 bits

Bit rate = Sampling(samples/second) *Quantization (bits/sample).
The width and height of the image is 450 and 520 and the frame rate is 25 Hz.
According to the above formula = Sampling = 520*450*25 samples/second.
Quantization can be found according to the 4:2:0 format for the Y, Cr, Cb as the average bits per pixel or sample = (4*8+1*8+1*8)/4 = 12 bits per sample.
Hence Bit rate = 520*450*25*12 = 70.2 Mbps.

If we are to quantize using Y, Cr, Cb 4:2:0 format with only 6 bits for the chrominance channel then average bits per pixel will change.
According to the formula: (4*8+1*6+1*6) / 4 = 11 bits per pixel.
Bit rate would change to = 520*450*25*11 = 64.35 Mbps.
Hence if this made to run for 10 mins (600 seconds) = Disk capacity = 600*64.35 megabits = 4.49 gigabyte.
Hence the storage on the disk = 4.49 Gigabytes.

Q2:

Quantization Interval: [-4,4]
Levels of quantization = 32

Since there are 32 levels present, we have to use 5 bits per sample.
In total we will need 32 * 5 = 160 bits

Quantized Sequence(Signal Values): 1.75, 2.25, 2.25, 3.25 3.25, 3.25, 2.5, 2.75, 2.75, 2.75 ,1.5, 1.0, 1.25, 1.25, 1.75, 2.25, 2.25, 2.25, 2, 2.25, 1.25, 0.25, -1.25, -1.25, -1.75, -1, -2.25, -1.5, -1.5, -0.75, 0, 1.

Quantized Sequence(Level/Bucket Values): 22, 24, 24, 28, 28, 28, 25, 26, 26, 26, 21, 19, 20, 20, 22, 23, 24, 24, 23, 24, 20, 16, 10, 10, 8, 11, 6, 9, 9, 12, 15, 19.

Height of each level: 8/32 = 0.25

Programming Part

Part 1 - Spatial Resampling and Aliasing

Command to Run the program: $ java MyPart1 360 2.0 1

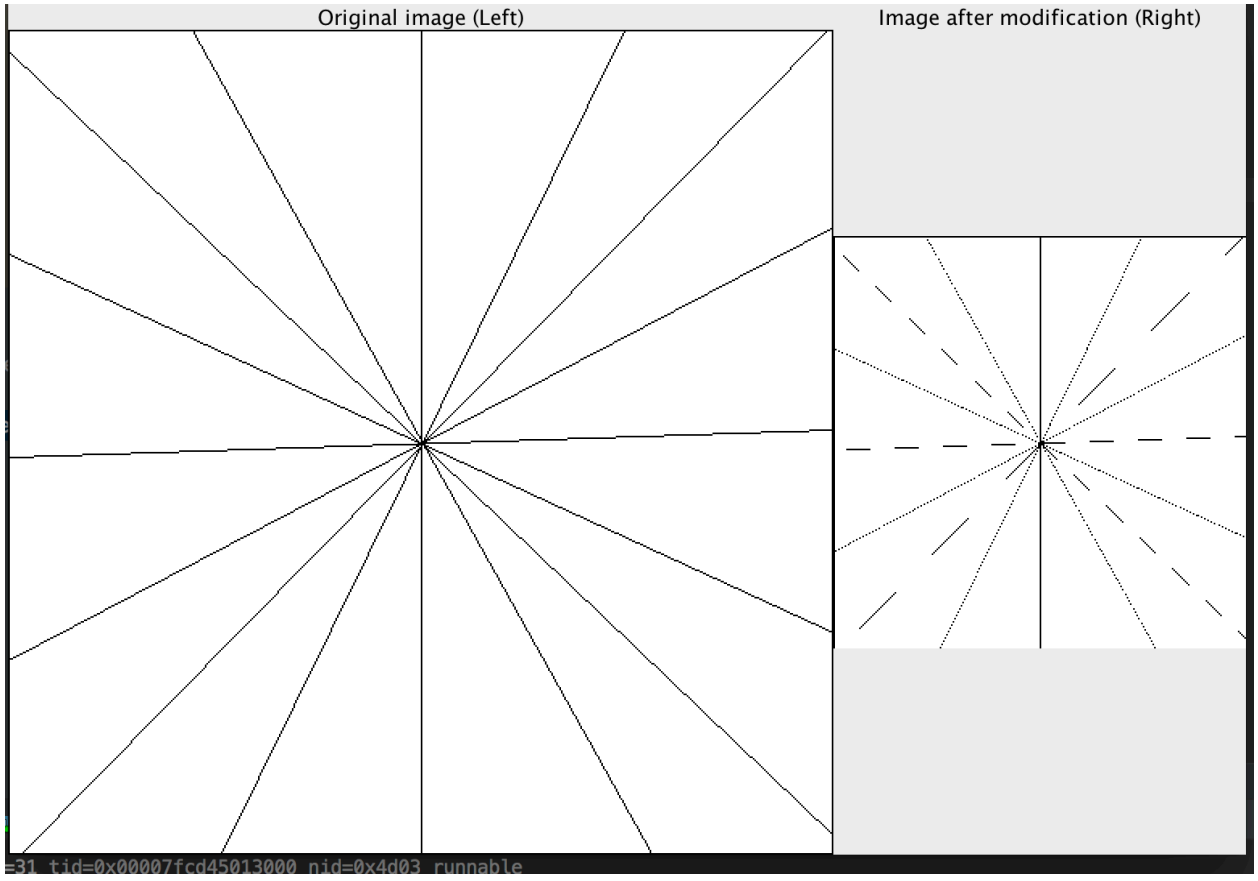
Parameters:

First Parameter - Number of lines to be drawn
Second Parameter - Scaling factor
Third Parameter - Anti-Aliasing

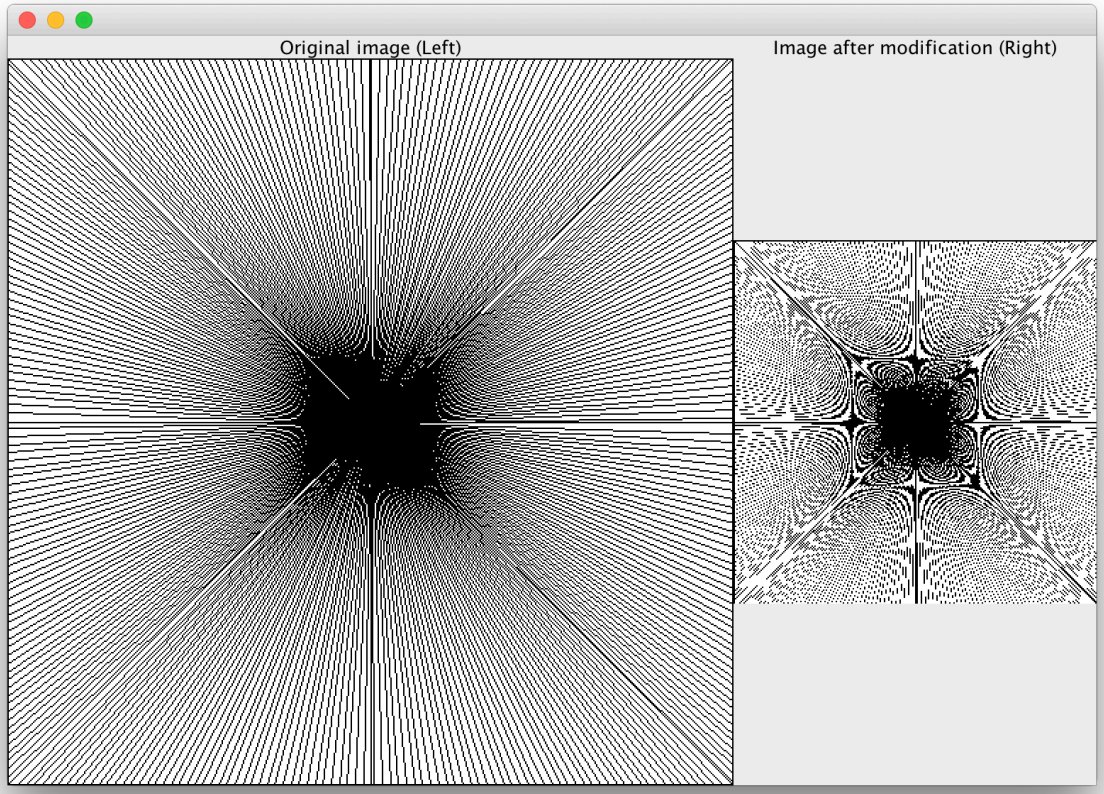
Code files for this part are submitted along with the report

Output

First Command: 16 2.0 0



Second Command: 16 2.0 0

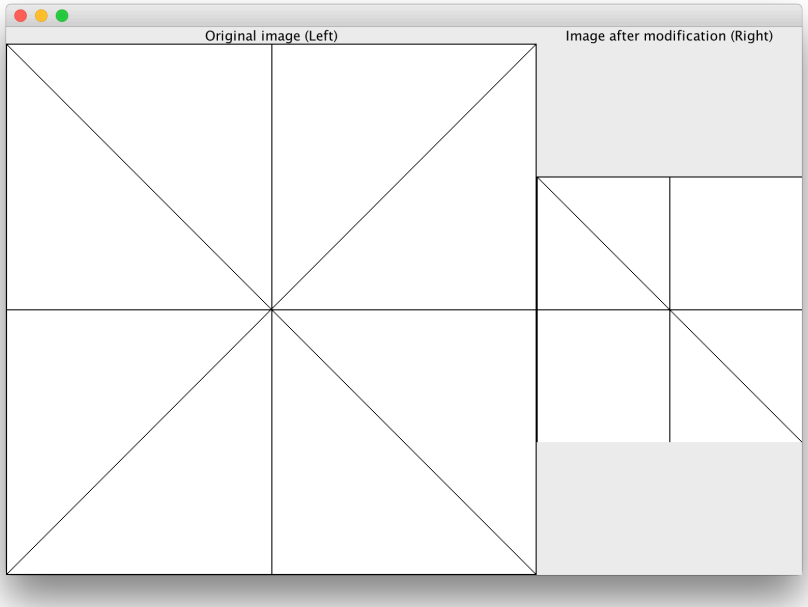


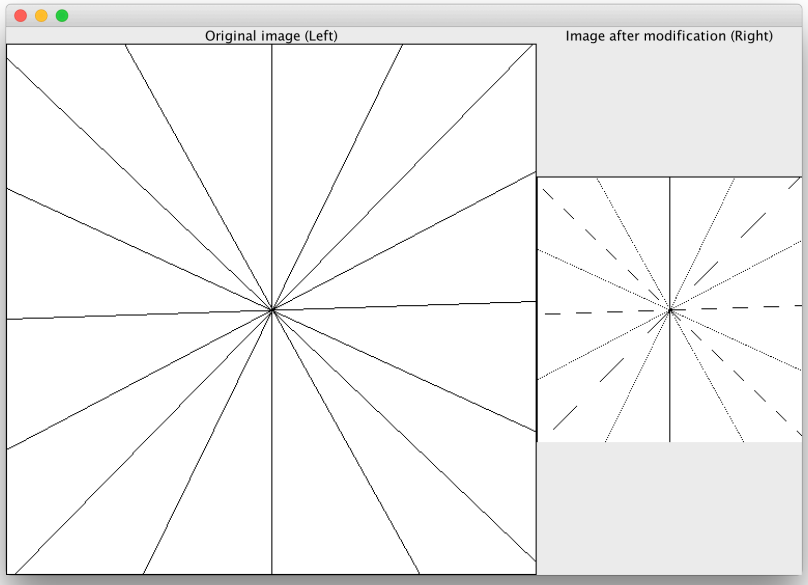
ANALYSIS PART

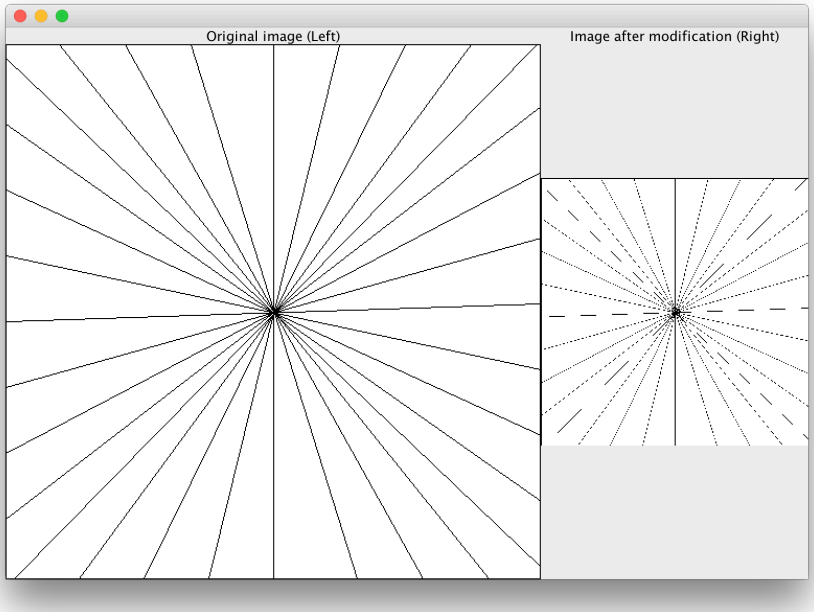
1)

Constant Scale Factor and variable number of lines

Let us first look at examples along with the images shown above



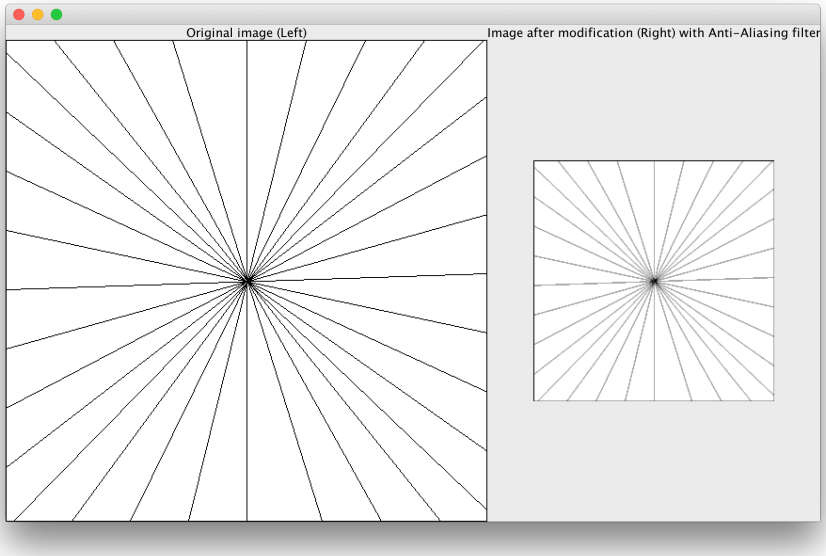




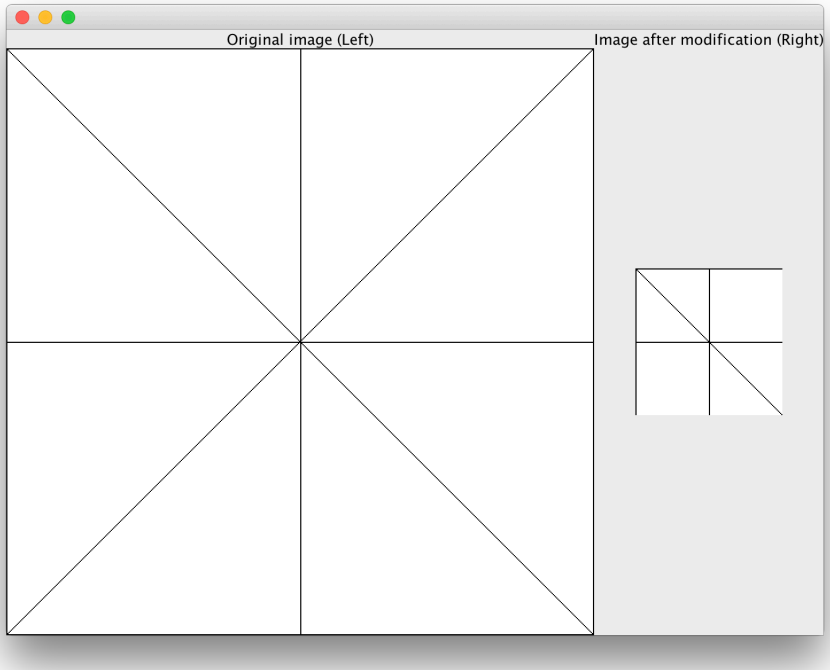
By looking the examples we can clearly see that increasing number of lines increase the aliasing effect when the resampled image is rendered.

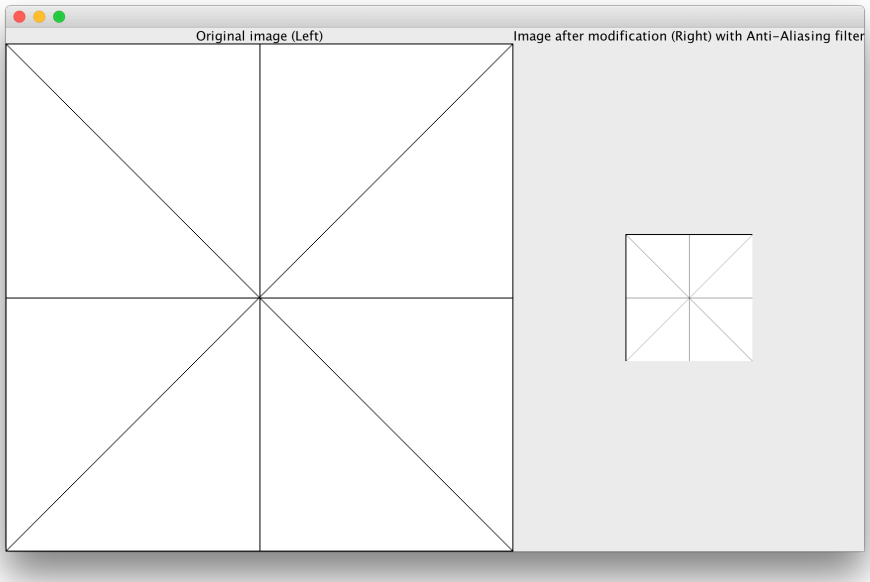
The more the number of lines more the frequency content in the image, resampling this high frequency image would result in loss of data. To prevent this, we need to add anti-aliasing filters such as a low pass filter.

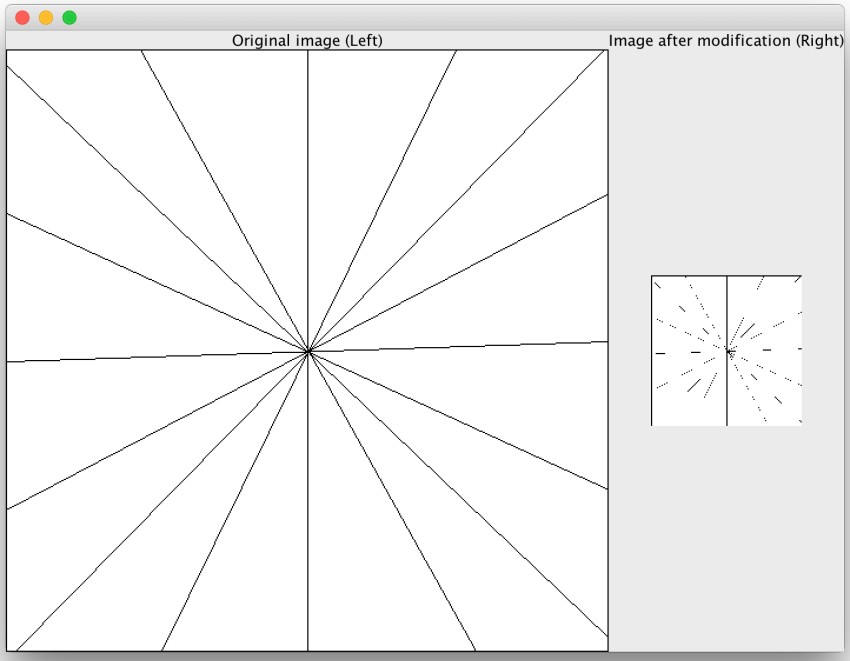
In this project the low pass filter is also designed to provide anti-aliasing features.
Using the filter on the same example above, would result in an image shown below.

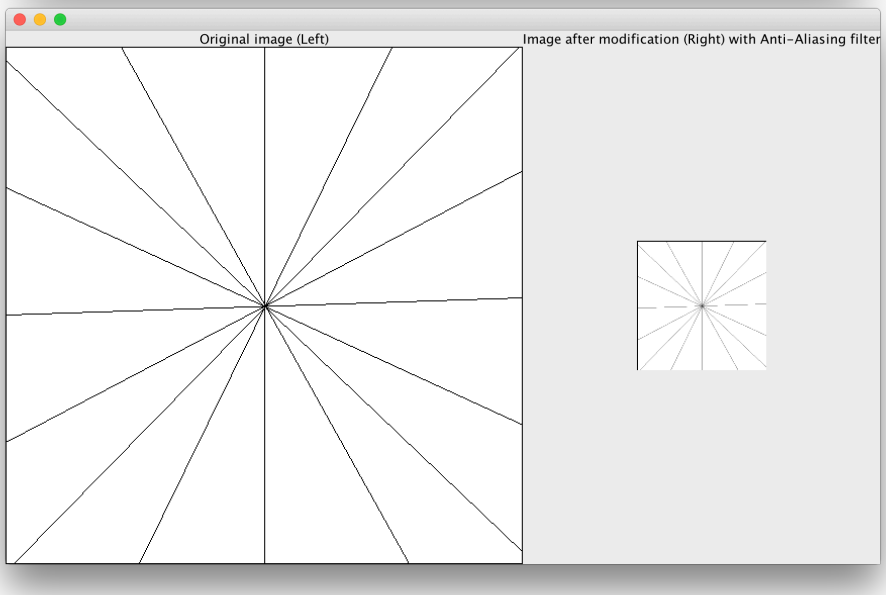


The same effect/result can be see in where the S = 4.0 and variable n as shown below. Here the image before filtering and after filtering are show. We can also see that some of the lines in some images are missing.



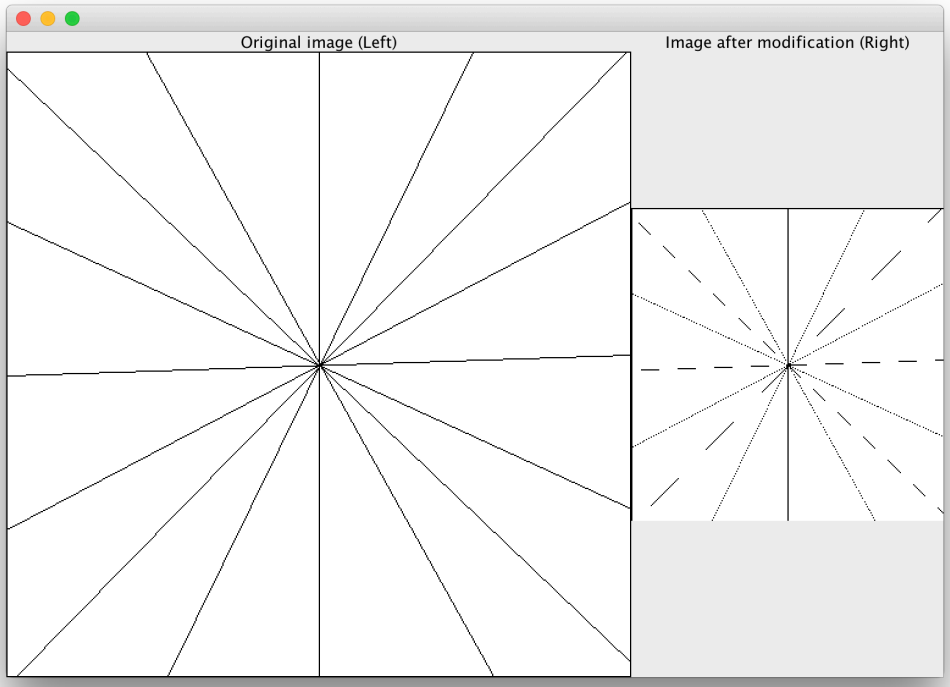

Original image (Left)
Image after modification (Right) with Anti-Aliasing filter

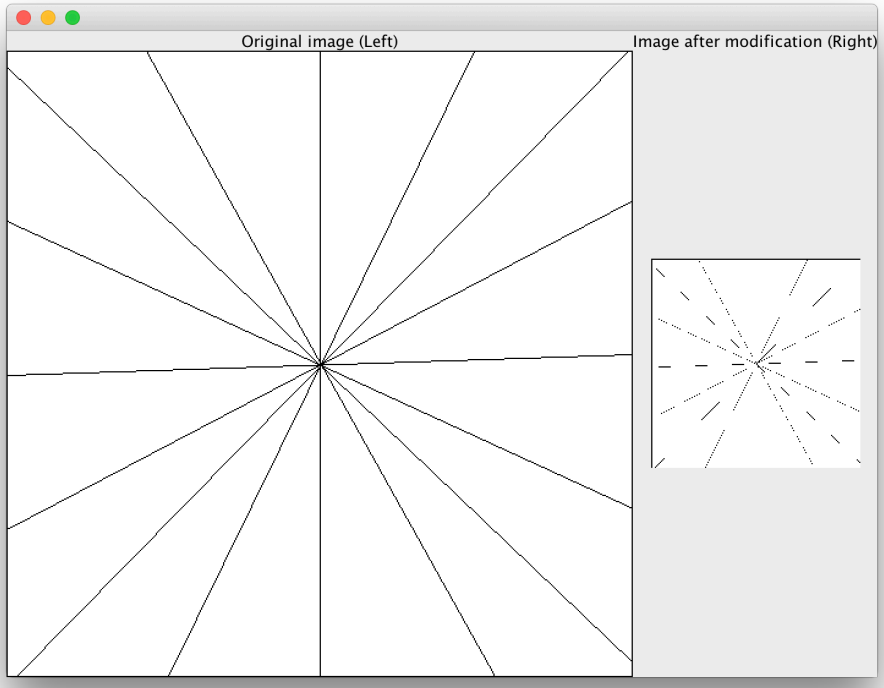

Original image (Left)
Image after modification (Right)

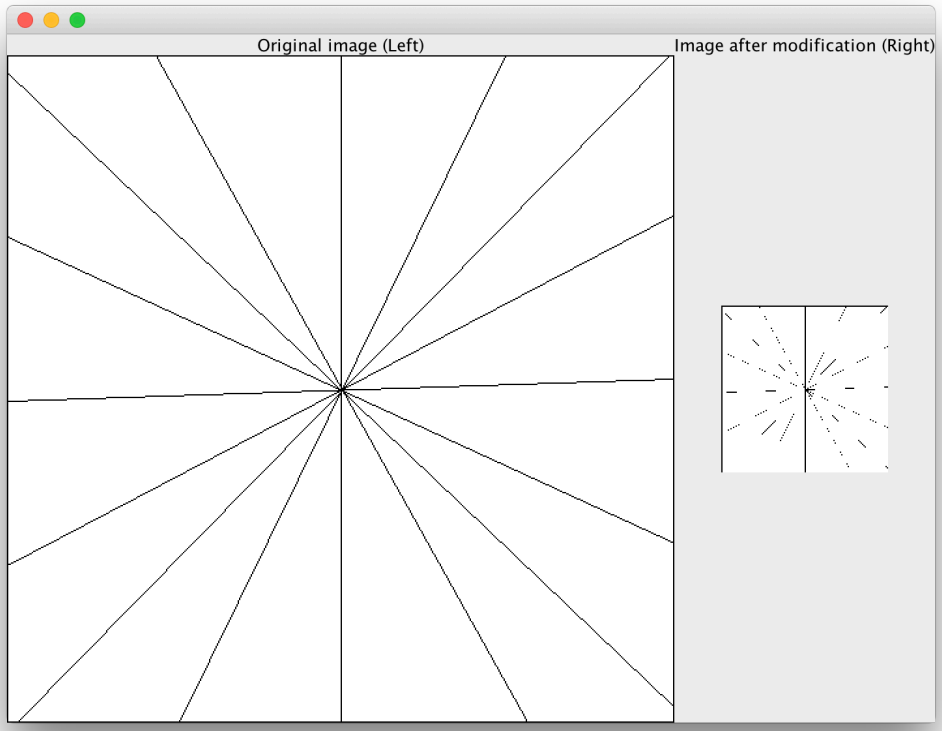

Original image (Left)
Image after modification (Right) with Anti-Aliasing filter

2)

Keeping the number of lines constant, we now change the scaling factor to observe that some lines go missing in resampling. This can also be fixed by the anti-aliasing filter in this assignment. Screenshots are provided below.







Part 2 - Temporal Aliasing

Command to Run the program: $ java MyPart2 64 4.0 12.0

Parameters:

First Parameter - Number of lines to be drawn
Second Parameter - Speed of Rotation
Third Parameter - Frames per second

Code files for this part are submitted along with the report

Since the videos cannot be rendered not he report, they have to be viewed by running the program.

ANALYSIS
1)
Formula relating speed of rotation, observed speed of rotations and frames per second

$$\text{Observed Speed} = \frac{\text{degree of rotation} * \text{FPS}}{360}$$

2)

Keeping S = 10

FPS 25: Observed Speed = 90

FPS 16: Observed Speed = 57.6

FPS 10: Observed Speed = 36

FPS 8: Observed Speed = 28.8

Part 3 - Extra Credit

Command to Run the program: $ java MyExtraCredit 64 4.0 7.0 2.0 1

Parameters:

First Parameter - Number of lines to be drawn
Second Parameter - Speed of rotation
Third Parameter - Frames per second
Fourth Parameter - Scaling Factor
Fifth Parameter - Anti-Aliasing

Code files for this part are submitted along with the report