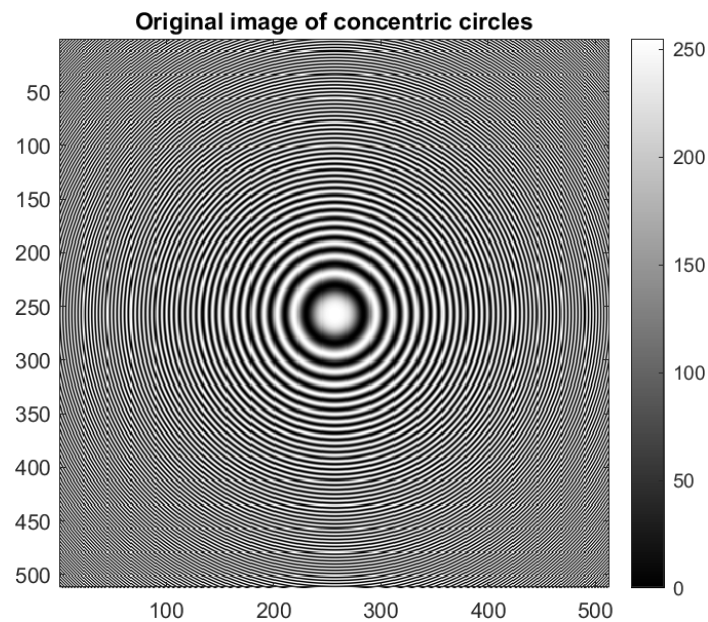


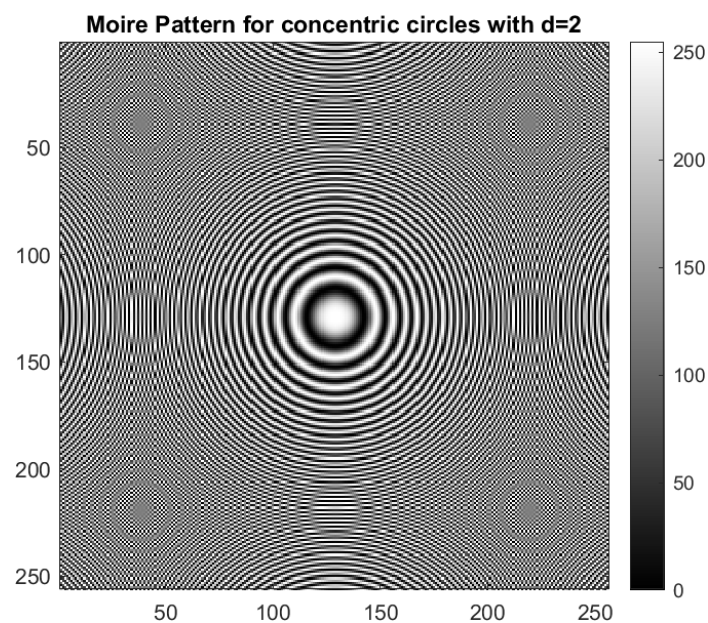
## Question 1

### Part A

Moire Patterns occur when we undersample an image. Below you can find the striking difference between original image and the undersampled images. Here undersampling factor is denoted by  $d$ .



**Figure 1:** Concentric Circles



**Figure 2**

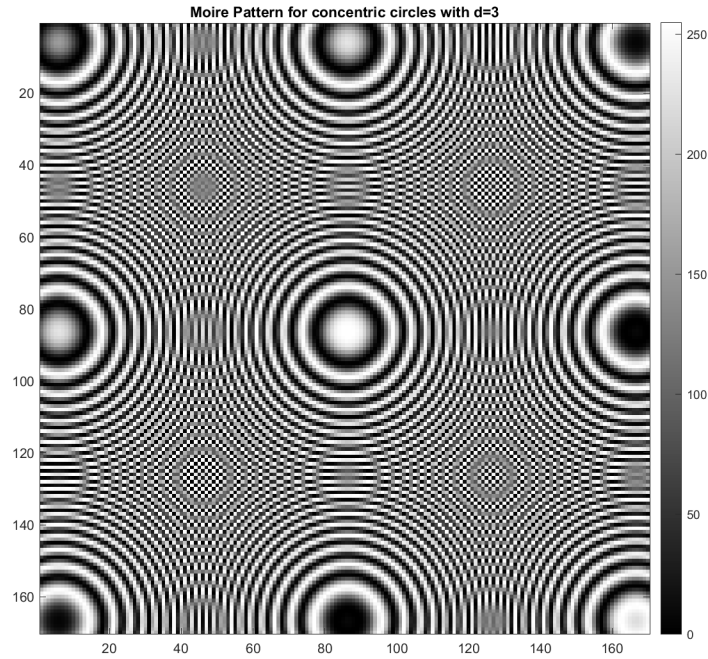


Figure 3

## Part B

Here the original image is an  $N \times N$  pixel dimensional image, where  $N=103$ . We want to enlarge the image by increasing the image pixel dimensions. The increment is as follows:

$$new\_rows = 3 * M - 2, \text{ where } M = \text{number of rows}$$

$$new\_cols = 2 * M - 1, \text{ where } N = \text{number of columns}$$

Here  $M=N$ .

We used different interpolation techniques. This part is for Bilinear Interpolation. Results are shown in next page:

## Part C

This part has has the same requirements as needed previously only we interpolated the function using nearest neighbor. Results are shown in next page:

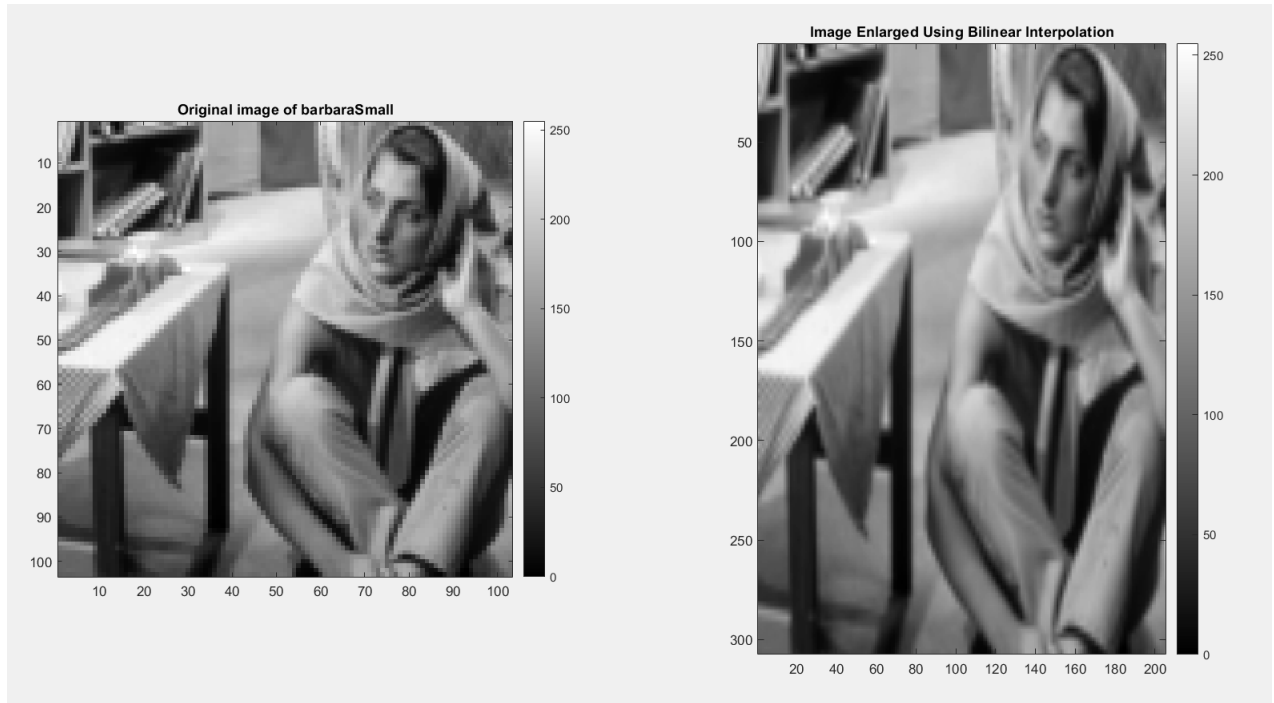


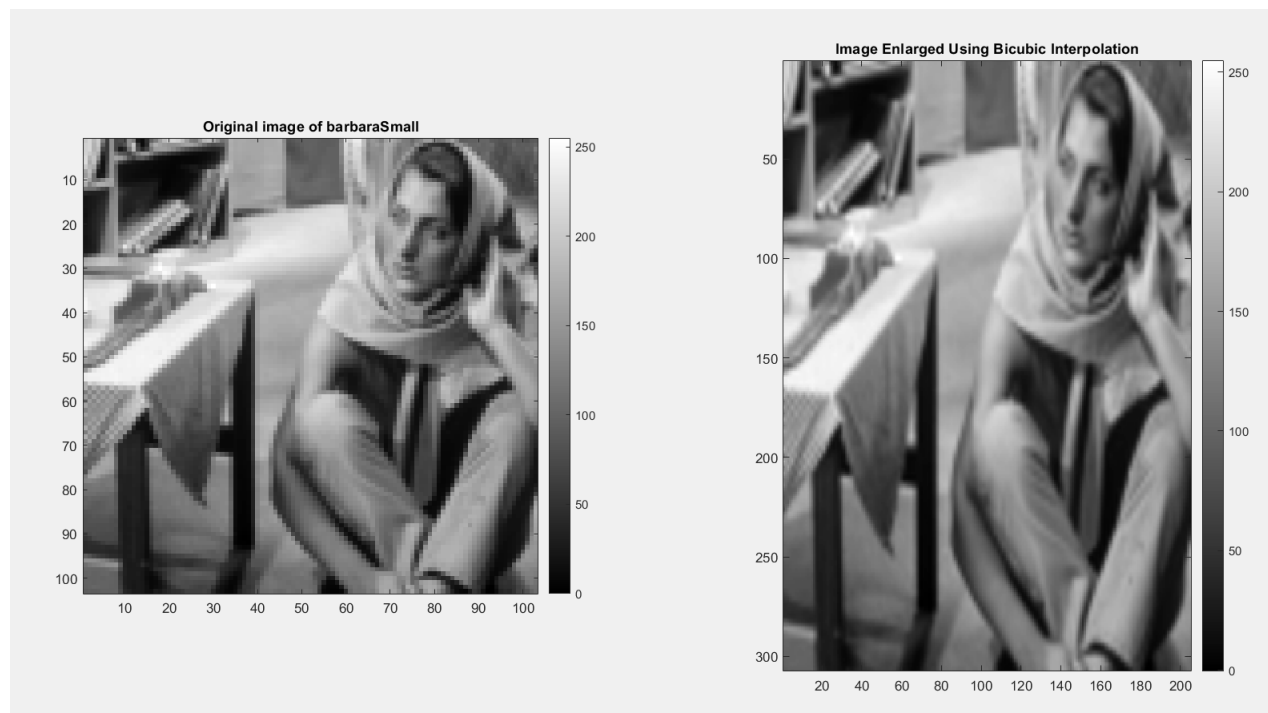
Figure 4: Bilinear Interpolation



Figure 5: Nearest Neighbor Interpolation

## Part D

Again, this part also has has the same requirements as needed previously only we interpolated the function using bicubic interpolation. Results are shown below:



**Figure 6:** Bicubic Interpolation

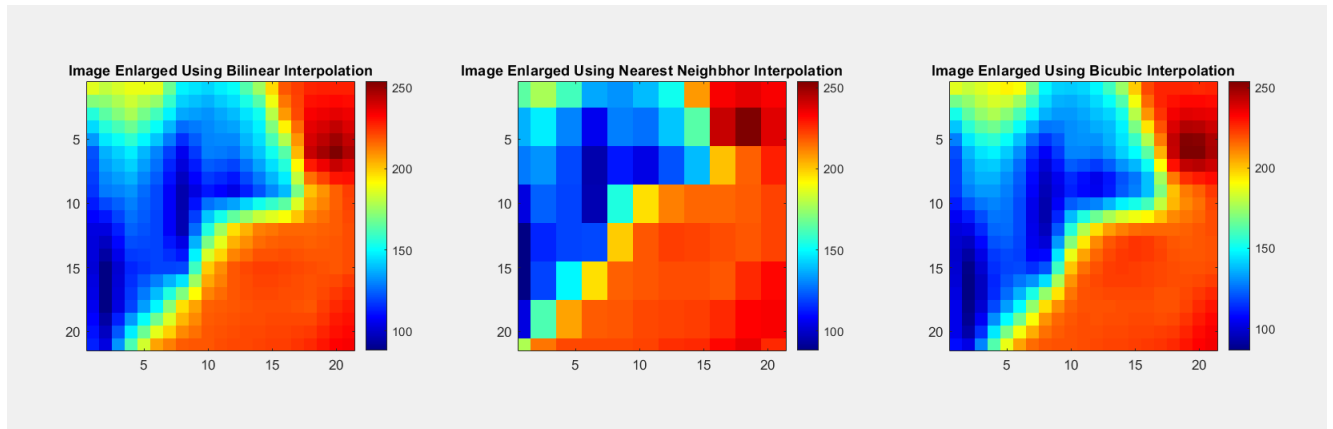
## Part E

The coarsest image obtained was from nearest neighbor interpolation. This interpolation method just looks at the nearest function value from a required point and assigns it the same value thus leads to very pixelated image.

Bilinear interpolation approximates the function using linear approximation thus leads to better image. Though the function value is continuous everywhere the derivative is discontinuous at given function points.

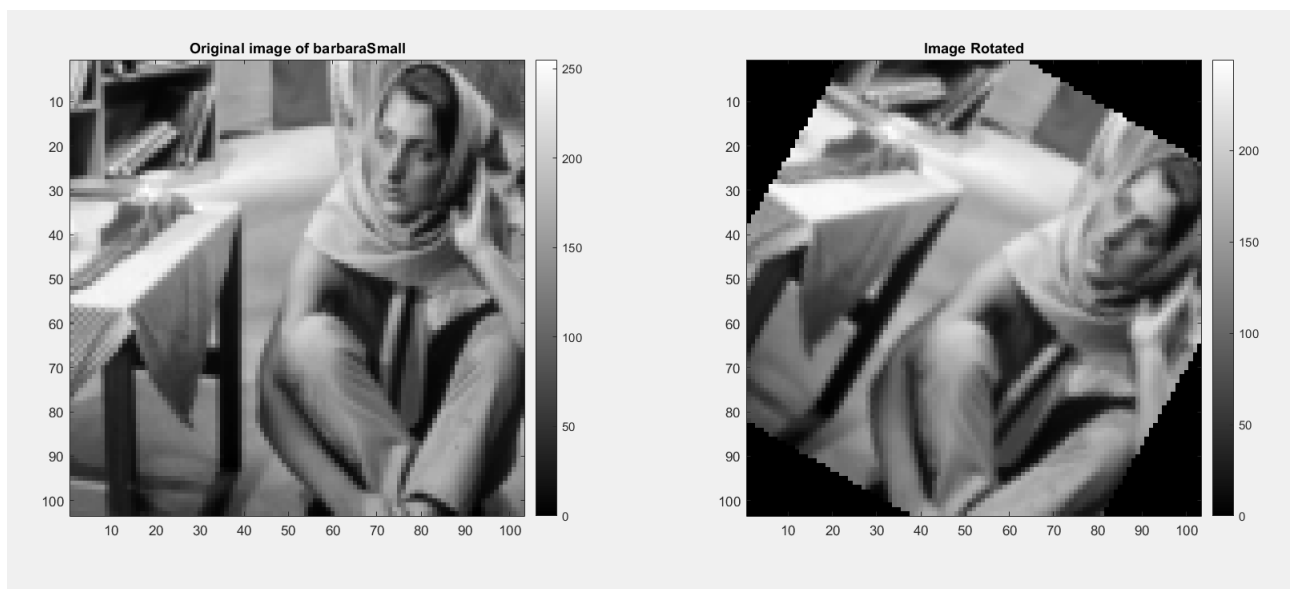
Bicubic interpolation approximates the function using cubic approximation thus leads to even better image. The interpolated function generated is continuous as well as its first derivative is also continuous at every point. The image is smoothest generated by this method.

We took a small patch of image and used colormap to generate the differences. Though bilinear and bicubic give almost same results but we find smoother transition from darker to lighter pixel intensities in bicubic technique (especially in red and blue regions). in the image are Results are shown in next page.

**Figure 7:** Three interpolation methods

## Part F

Our last task in this question was to rotate the image by  $30^\circ$  in the clockwise direction. Here we employed the bilinear interpolation method to find function values at the new image grid. Since we generated the image having same size as the original image edges were cut off. Result is shown below.

**Figure 8:** Image Rotation by  $30^\circ$

## Part G

The main file myMainScript.m is divided into 6 parts.

- Part A (Image Shrinking)- It contains a simple function, myShrinkImageByFactorD, which just takes the function (image matrix) and shrinking factor d. We shrunk the image size by a factor of d along each dimension using image sub-sampling by sampling / selecting every d-th pixel along the rows and columns.
- Part B (Bilinear Interpolation)- It contains the function myBilinearInterpolation which takes four original image points, a point and area and then calculates the function at the given point using the image points.
- Part C (Nearest Neighbor Interpolation)- It contains the function myNearestNeighborInterpolation which takes four original image points, a point and then calculates the image point nearest to the given point and assigns it this value.
- Part D (Bicubic Interpolation)- We first calculate the first derivatives w.r.t. x and y and mixed second derivative for the image grid using difference method. Now we pass these matrices along with the image value, the point at which function value is required and the matrix A\_inv(a constant matrix). It then calculates the value at the given point using the bicubic formula.
- Part E (Interpolation Comparison)- It uses the previously defined function to generate a small patch of image to compare the differences in the interpolation techniques.
- Part F (Image Rotation)- It contains the function myImageRotation which takes four original image points, a point and then bilinearly interpolates to get the new image grid.