Assignment 1

Priyanka kajla 17210073
Deepak 15110040
Sai Rohit 14110126

08/09/2017

Question 1:
Algorithm:

1. Initially we entered all the inputs- Size of image, no of circles, border size, radius range, and foreground and background intensity.
2. Then created a white image of size MxN and created a matrix to store x and y coordinates, radius and intensity radius and intensity
3. Randomly generated coordinates (x, y) within the size of image, radius within the given range and intensity values by using rand () function.
4. We run a infinite while loop it will break when no of circles exceed the given no of Circles (n).
5. Calculated distance between two centers (current center and other centers) and then compared this distance to addition of current circles radius to previous circle's radius. Then put the condition that distance between centers should be greater than the addition of radius of current circle to previous circles radius, to avoid overlapping of circles. If overlapping is not found then updated the new center, radius and intensity value.
6. To draw the circle we run three for loops one from 1 to M 2nd from 1 to N and 3rd from 1 to n, then compared the distance of center to surrounding pixels if this distance is less than equal to radius then fill the intensity value.
7. Then display the final image
8. time taken by processor:0.299 sec

Inputs: No of circles are 10

```
>> assignment1
Enter M250
Enter N500
Enter n10
Enter border5
Enter r15
Enter r210
Enter Vf15
Enter Vb150
   250   500
```

Figure 1:

Size of image 250X500
Radius range [5, 10]
Intensity [Vf, Vb] = [15, 150]
Output:

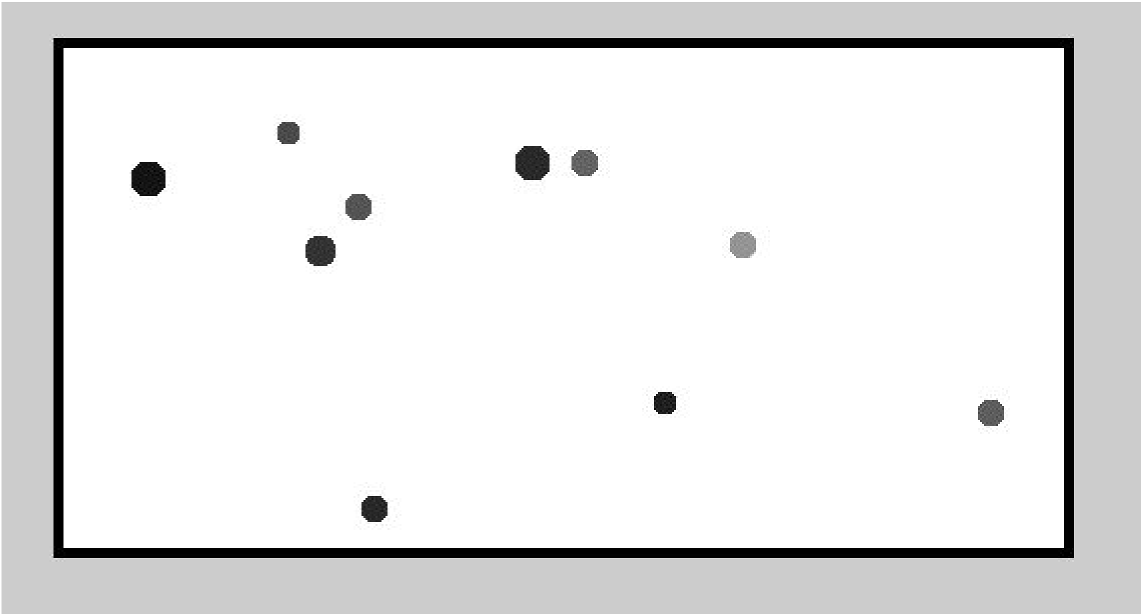


Figure 2:

Question6:

Algorithm:

1. we first created a M X N image which consists of random pixel values in the desired range
2. Given inputs such as two pixels coordinates [(xa,ya),(xb,yb)] between which we should find the mpath distance and the V set
3. Created a column matrix to store the pixel values in V set
4. A dummy matrix of sizeM X N and distance matrix of sizeM X N are created
5. Dummy matrix stores the mpath distance of a pixel in the image from the pixel(xa,ya) whereas distance matrix stores the shortest mpath distance of a pixel from the pixel(xa,ya)
6. We checked if a 4 neighbor pixel of(x,y) is in V and if the condition satisfies we updated the dummy distance value of the 4 neighbor pixel by adding "1' to the dummy distance value of (x,y) If the condition doesnt satisfy we checked if the diagonal pixel is m adjacent and updated updated the dummy distance value of the by adding "1' to the dummy distance value of (x,y) If both conditions doesnt satisfy the dummy distance value remains unchanged
7. If there many dummy distances possible for a pixel(x,y) from (xa,ya) then we choose the minimum dummy distance which gives the shortest mpath distance
8. Final distance matrix (M X N) consists of shortest mpath distance of a pixel(x,y) from (xa,ya)
9. shortest mpath distance of (xb,yb) from (xa,ya) is equal to value of the pixel(xb,yb) in distance matrix

Input1:
Enter X1: 3
Enter Y1: 4
Enter X2: 2
Enter Y2: 5
Enter M: 5
Enter N: 5
Enter no of elements in v: 2
Enter the maximum value of pixel intensity: 6
Enter v set intensity: 1
Enter v set intensity: 2

output1: $Euclideandistance = 1.4142$
$fourpathdistance = 2$
$Eightpathdistance = 1$
The mpath distance 1

Input2:
Enter X1: 4
Enter Y1: 5
Enter X2: 2
Enter Y2: 4
Enter M: 5
Enter N: 5
Enter no of elements in v: 2
Enter the maximum value of pixel intensity:6
Enter v set intensity: 1
Enter v set intensity: 2

output2:
$Euclideandistance = 2.2361$
$fourpathdistance = 3$
$Eightpathdistance = 2$
No path exists between the two given points

Question7:

Applications of Digital Image Processing:

1. Improvement of pictorial information for human perception:

   Digital image processing is very helpful in improvement of pictorial information by using some techniques such as filtering, thresholding, gray level slicing, bit plane slicing logarithmic transformation and histogram processing etc. depending upon the application anyone can be used for better result. Removal of salt and pepper noise from an image using filtering. Here filter of size 3x3 min filter is used that helps in removal of noise from input image

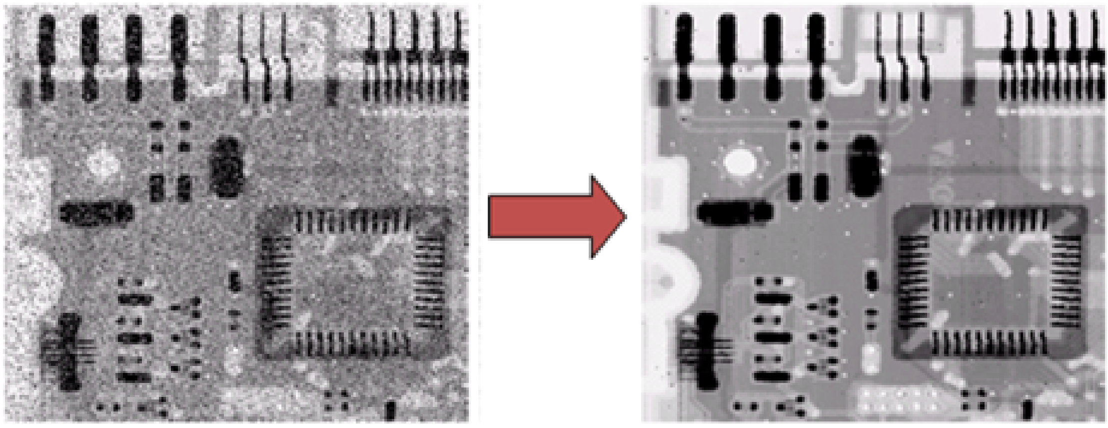


Figure 3:

   Histogram processing is helpful to provide useful information for contrast enhancement; it provides a global description of the appearance of the image.

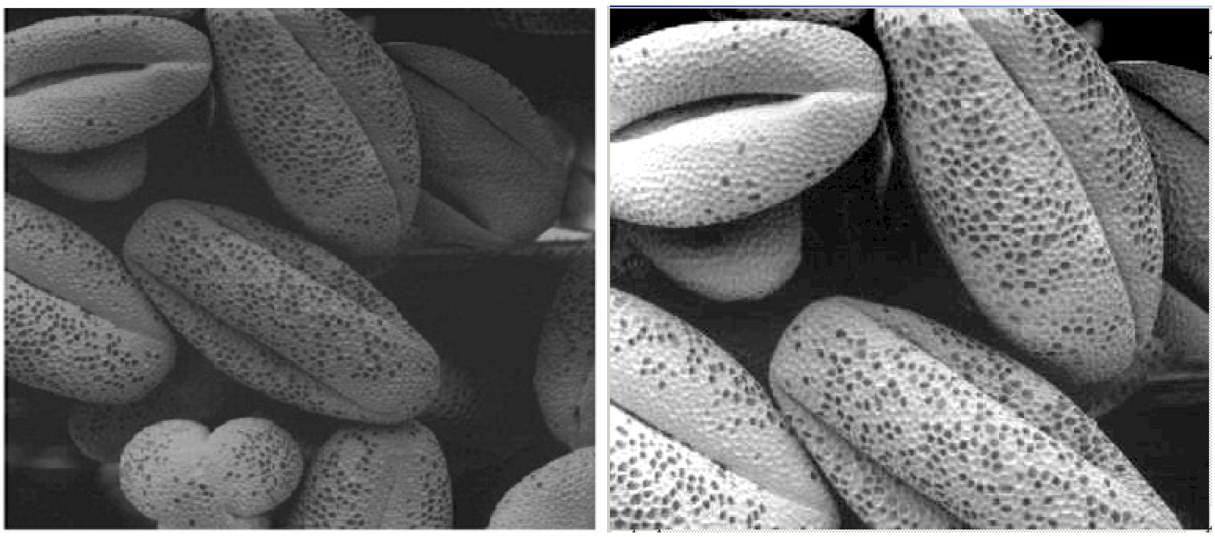


Figure 4: blur image and bright image

Histogram of an image: bright image gives high contrast as compared to blur image.

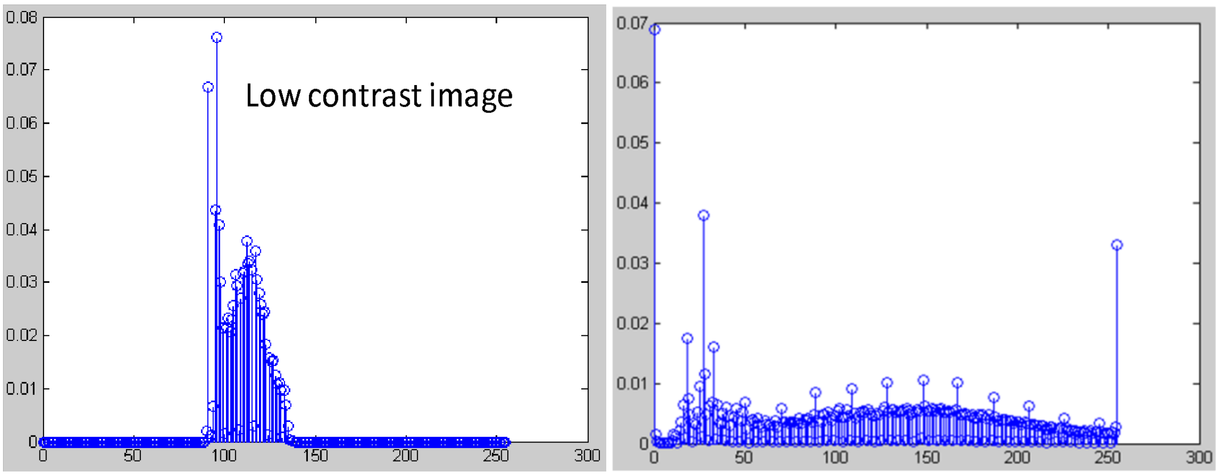


Figure 5:

2. **Autonomous machine vision:** Digital image processing can be helpful to make human computer interfaces more natural.

    - Face recognition
    - Gesture recognition

    The feature of a digital image such as shape, color, texture, topology of object etc. can be used as input for search and retrieval of pictorial information from large image database.

    It can be helpful to face recognition. We can match query face image to the image database which contains several faces.

Figure 6:

Digital image processing is helpful in gesture recognition. If we give a gesture as an input then we can match this with another gesture then we can fine similarities and dissimilarities between gestures

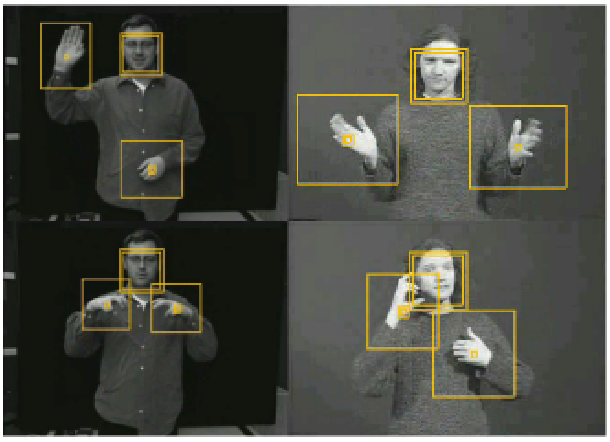


Figure 7:

3. **Efficient storage and Transmission:** In order to process the image, it must be saved on computer. The image output of most sensors is continuous voltage waveform. But computer deals with digital images not with continuous images, thus continuous images should be converted into digital form.

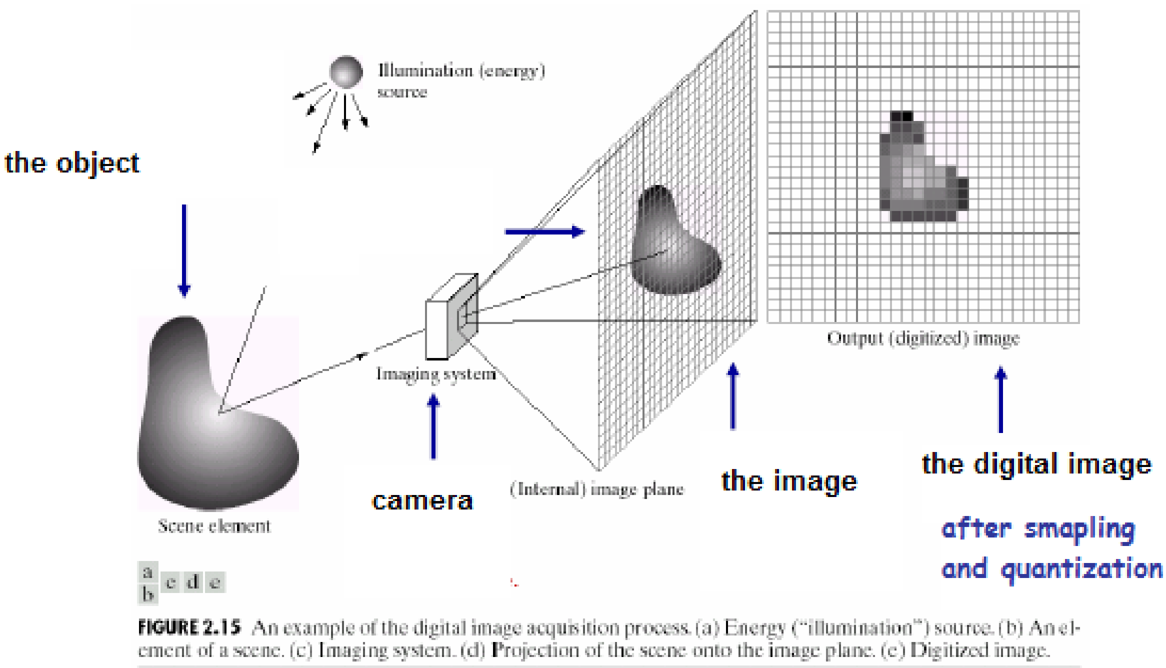


Figure 8:

Continuous image (in real life) digital (computer) We can convert continuous image into digital form by using two process i.e sampling and quantization.

Sampling: sampling is digitization of coordinate value Quantization: quantization is digitization of intensity value

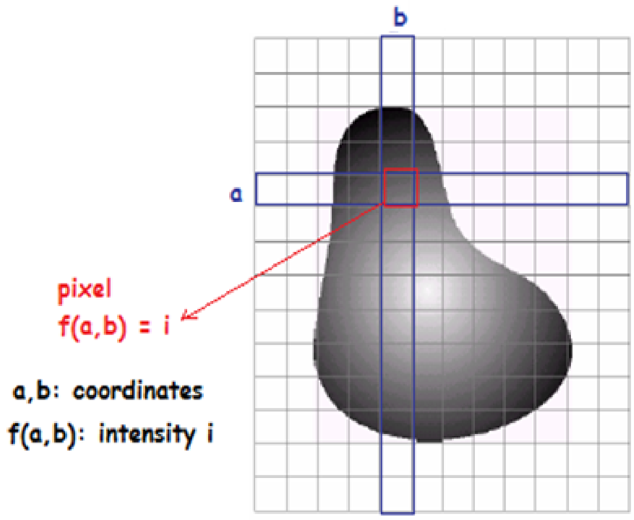
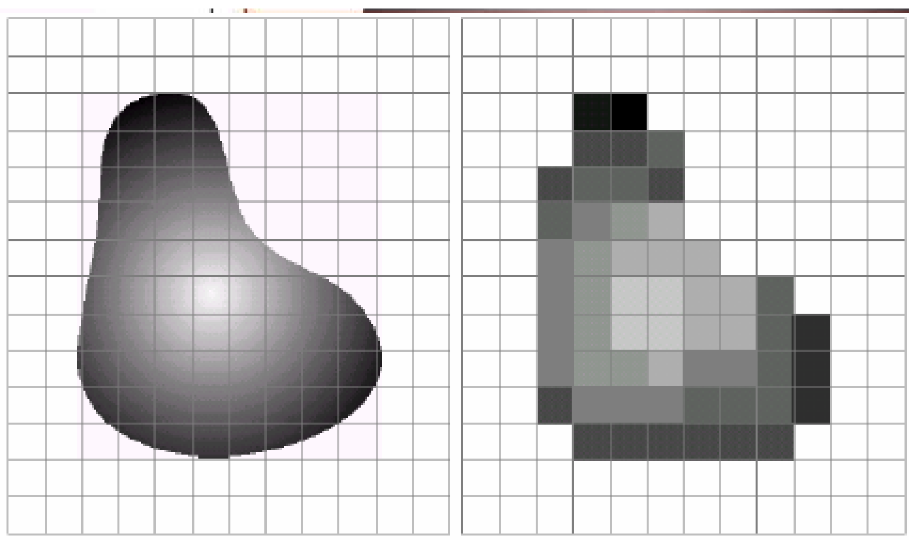


Figure 9:

Input image: image is a function f(x, y)
x and y are coordinate value of pixels
f: intensity value (Amplitude)

Result of digital image after sampling and quantization:



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

Figure 10: