Sefa Kayraklık
2015401066

EE475 – Homework 1

Generating Image Patterns

In order to synthetize the asked patterns, an empty matrix array named *img_*pat is formed with the dimension 480 by 480. Then 2 matrixes named as *row* and *col* are formed to be used to determine the different regions in the image such as the inner circle, the outer circle and background. The circle equation $(x - x_{center})^2 + (y - y_{center})^2 = radius^2$ is used to find the boundaries. In the equation, x and y correspond to *row* and *col*, respectively.

After storing the pixels of each region in the logical matrix *circle1px*, *circle2px*, and *background*, they are used to index the image matrix *img_pat*. The pixels' values of the inner circle are determined as 192 added a uniform noise with the range -16 to 16. The pixels' values of the outer circle are determined as 128 added a uniform noise with the range -42 to 42. The description for the background says that it has a mean value of 64 but doesn't say about its noise. So, in order to represent the background smoother, the gaussian distribution with 10 standard deviation is added.

When all the image patterns which are shown in the Figure 1 are formed, they are saved a file named *img_pat* in *tif* format.

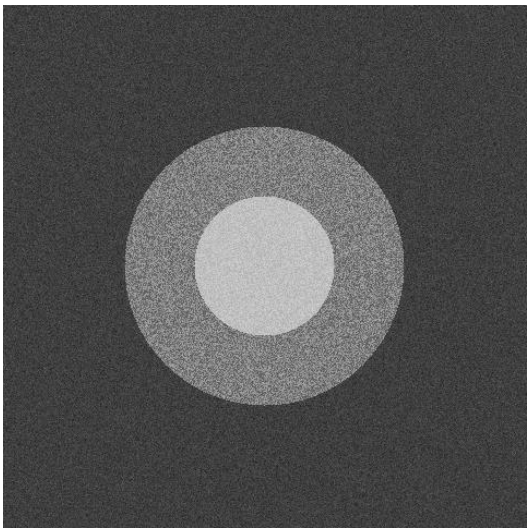


Figure 1: The Synthetic Image Patterns, img_pat

Figure 2: Peppers - Original Image

Image Read & Writes; Histogram Plots

The *peppers.png* is read into a variable *I* and displayed. Since the original image shown in the Figure 2 is in *.png*, it is written to a disk file with the extension *.bmp*. In order to see the content of the written image, imfinfo is used and stored in a variable named as *img_info*. This variable contains the information about file name with the directory, last modification date, size, format, image dimensions, … etc. Namely, a lot of information can be obtained using this function.

Part a)

From the original image, its R, G, and B components are extracted and stored in matrixes named *I_r, I_g,* and *I_b*, respectively. They are shown in the Figure 3.

Figure 3: RGB components of the Original Peppers Image

Part b)

The information about whether the pixels have R, G, B color values R>90, G>10, B<40 or not is stored in a logical matrix which the true (1) corresponds the conditions are hold and false (0) corresponds the conditions aren't hold. When this matrix is displayed using *imshow* function, 1 is plotted as 255 and 0 is plotted as 0. The produced image from the logical matrix is given in the Figure 4.

From the Figure 4, it can be seen that the background pixels are filtered out. However, some pixels in the region of interest are lost such as the white onion and garlic, the white reflection areas on the peppers.

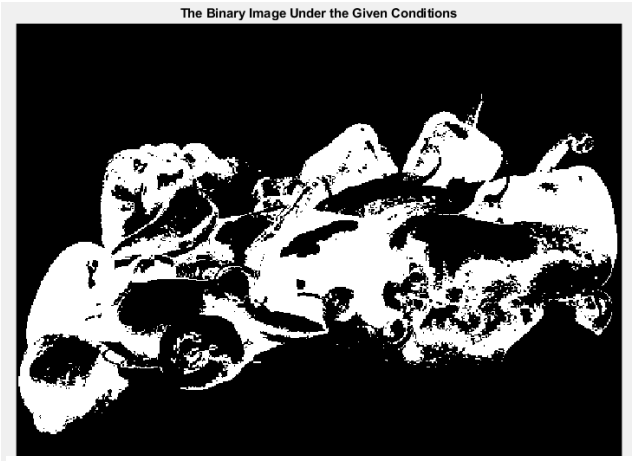


Figure 4: The Binary Image Under Given Conditions

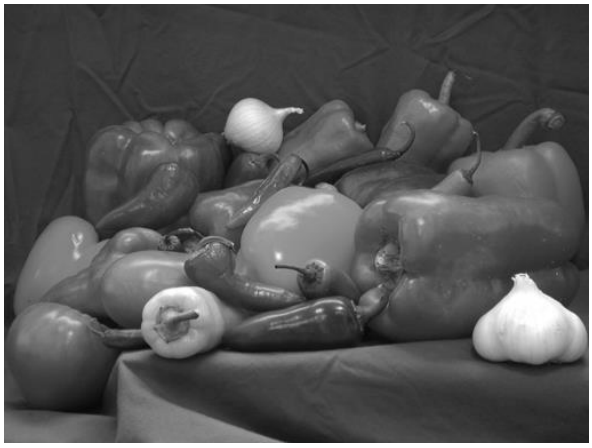


Figure 5: The Gray Version of the Peppers Image

Part c)

By takes the average of the color components of the image, the gray version of the image can be obtained. The gray version of the peppers image is given in the Figure 5.

Part d)

The histogram of the gray version of the peppers image is given in the Figure 6.

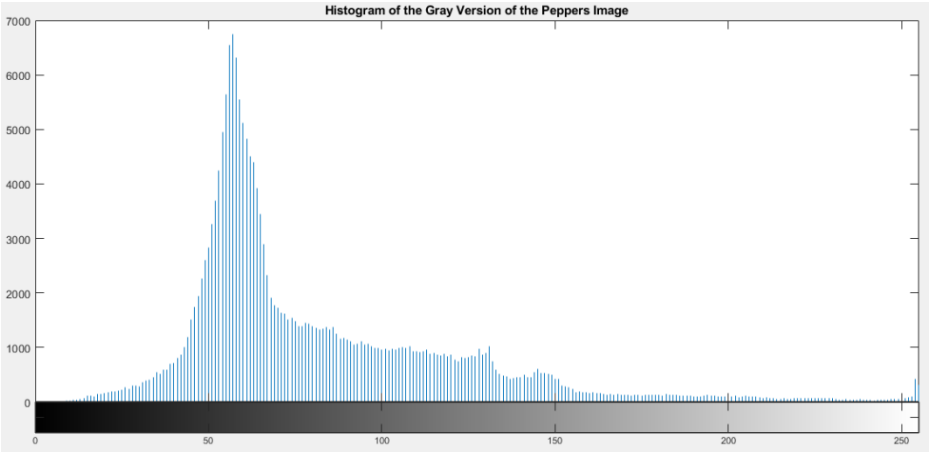


Figure 6: The Histogram of the Gray Version of the Peppers Image

In the histogram, the x axis is the value of the pixels starting from 0 and up to 255. 0 means the darkest pixels and 255 means the brightest pixels. Namely, the x axis is the gray level range. The y axis is the how many pixels there are in the certain pixel value.
It can be seen from the histogram that almost all the gray range is occupied; however, some portions of the gray level from beginning and ending are used rarely according to the other ranges such as from 40 to 70.
It can be seen that there are some values that have the maximum frequencies in the gray range. The first three of them is x=57, 56, and 58, respectively.

Part e)

If the gray scale values of an image are divided by 2 which is shown in the Figure 7, the image becomes darker and occupies only the first part of the gray scale. Therefore, the maximum gray scale value that can be present in the image is 127; namely, there aren't any white or near-white pixels in the image.

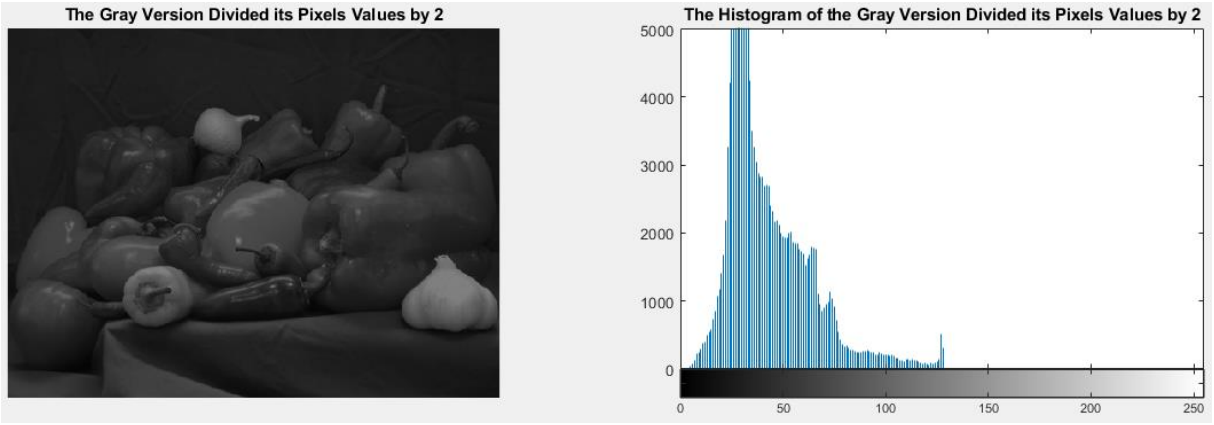


Figure 7: The Gray Version Divided its Pixels Values by 2

If the gray scale values of an image are added 64 which is shown in the Figure 8, the image becomes brighter and occupies only the last three quarters of the gray scale. In the gray image, the part of the gray scale from 0 to 191 is shifted 64 pixels to right and the remaining part of the gray scale becomes 255, namely white. Also, there aren't any black or near-black pixels in the image.

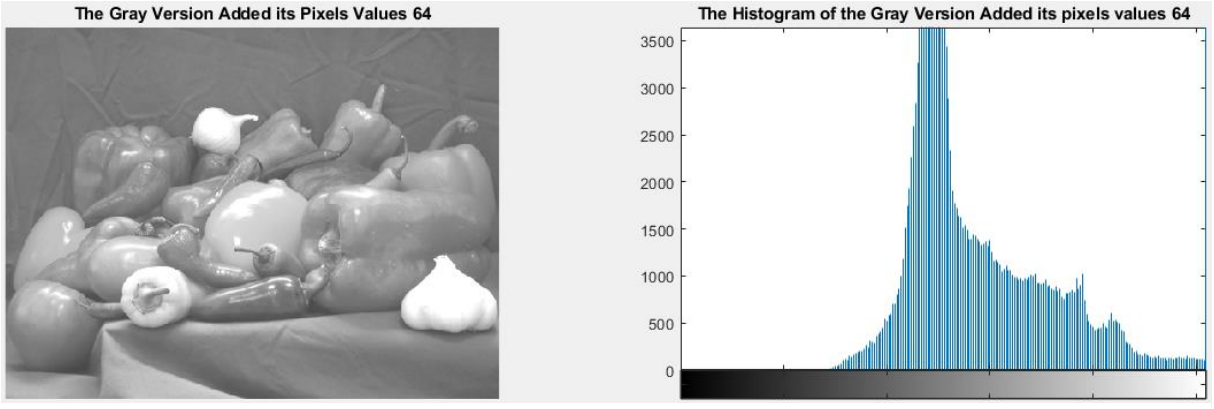


Figure 8: The Gray Version Added its Pixels Values 64

Pixel Varieties

Part a)

There are some methods for determining the coordinates of the brightest points in an image. Two of them are using the gray scale of the original image and using the total RGB values of the original image.

In the first method, the original image is converted to gray scale image. Then, the rows and the columns of the points that emit most amount of light are determined by finding the pixels that have the maximum value.
The second similarly tries to find the pixels that have the maximum value; however, it uses the matrix that is formed by adding RGB components of the pixels.

Since displaying the result coordinates is not very illustrative, the image where only the brightest points are displayed is plotted but the coordinates can be reached in the matlab, they are stored in arrays named *rows1, cols1* and *rows2, cols2* respectively. The plots are shown in the Figure 9.

Figure 9: The Brightest Points in the Cetus Galaxy

Part b)

Since there is no source image in the zip file, the image from the word is adopted and analyzed.

The first method from the part a is used to determine the closest and the farthest points in the image. The closest point is the pixel that has the maximum gray scale value. And the farthest point is the pixel that has the minimum gray scale value.

The coordinates[1] of the closest point are (156, 236) and the coordinates of the farthest point are (41, 176). (Note that the origin of the coordinate system is the left-top of the image.)

Part c)

Finding the most transparent point is trivial. The most transparent point is the pixel that have the maximum value and the method used in previous parts can also be used here. After using the method, coordinates of the most transparent point for the skull are found as (64,25) and coordinates of the most transparent for the breast are found as (71,220), (73,219), (110,214), (316,220), (318,219), (355,214), (561,220), (563,219), (600, 214).

However, finding the densest point isn't that straightforward since the background has the lowest values. The denser the pixel, the lower the pixel's value. In order to eliminate the background, the pixels that have 0 up to 16 values for the skull and 0 up to 3 values for the breast are converted to white. And by inspecting the resulting image, the possible densest points can be determined. The possible points are determined by selecting the white points in the region of interest since their values are lower than 16 for the skull and 3 for the breast. They are shown in the Figure 10 and 11 with the rectangles.

[1]The coordinates are given as (row#, column#)

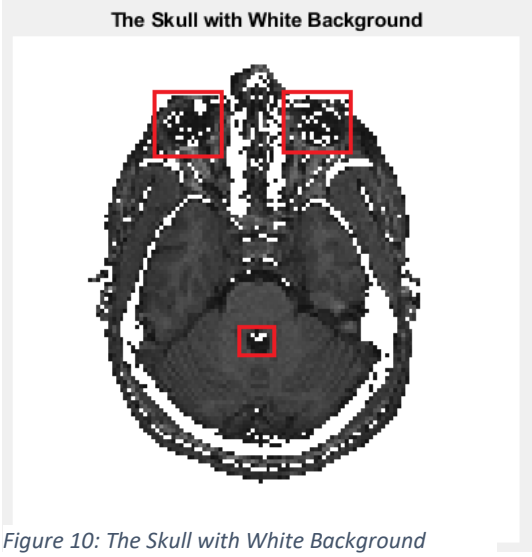


Figure 10: The Skull with White Background

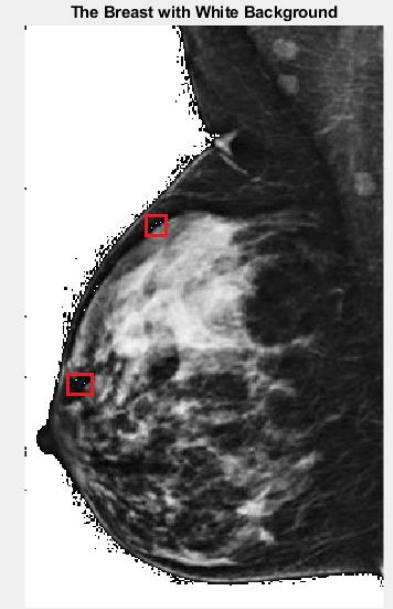


Figure 11: The Breast with White Background

Part d)

Finding the hottest points are same as determining the pixels that have the maximum values. The same method in the previous parts can be used. Since the question asks for the possible points, the range from (maximum value -16) to maximum value can be determined as the hottest points.
The resulting coordinates[2] are (72, 46), (72, 47), (73, 47), (90, 47), (114, 103).

Part e)

I understood the question in 2 ways and tried to find the solution for each way.

The first one is trying to find the fastest moving object according to the 3 sequences. In order to find the change in the sequences, the average of the 3 sequences are taken. If the parts of the resulting image are not changed namely same as the sequences, then it means that these pixels aren't moving. And the blurry regions mean that there are some shifting in the pixels. Therefore, the possible region is in the Figure 12 with rectangles.

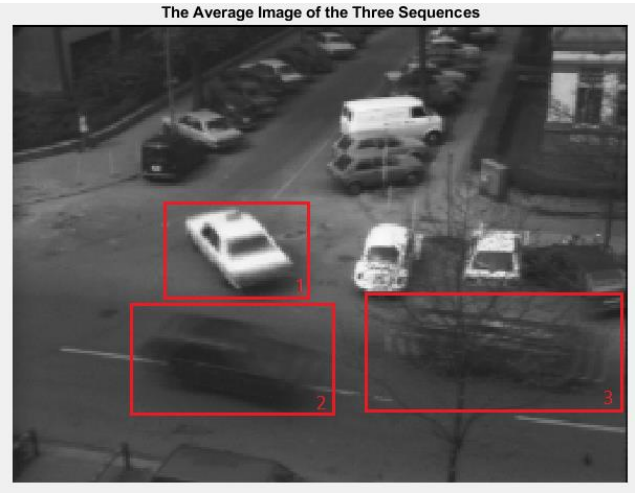


Figure 12: The Average Image of the 3 Sequences

From the figure, it can be seen that there are 3 moving objects. By inspecting the figure, the 1st car's velocity is lower than the others and the 2nd and the 3rd ones almost have the same amount of shift in the pixels.

The second way is trying to determine the fastest change in colors of the pixels. In order to solve the problem, the difference between the sequences can be analyzed. There are 3 difference to consider such as the difference between 2nd sequence and 1st sequence, the difference between 3rd sequence and 2nd sequence, and the difference between 3rd sequence and 1st sequence.

[2] The coordinates are given as (row#, column#)

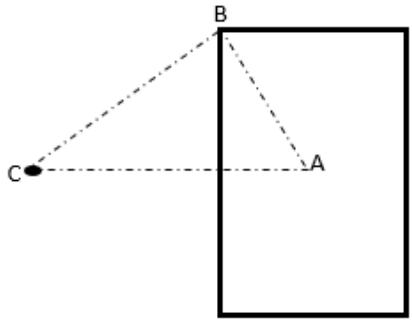
In the difference images, the pixels that have larger values are the possible regions for the fastest changing points. The possible region for each difference is shown in the Figure 13.

Figure 13: The Difference Images

It can be seen from the figure that all the difference images agree with 3 regions that are the candidates for the fastest changing points.

Imaging System Design

Part a)



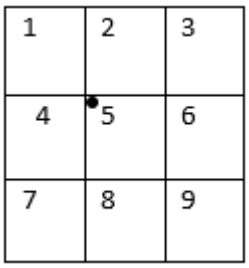
The angle of CAB is 90 degree. And the angle of ACB is equal to half of the field view angle. So, the minimum angle to view the farthest point on the table can be calculated as following:

$$\frac{\alpha}{2} = \tan^{-1}\frac{|AB|}{|AC|} = \tan^{-1}\frac{\sqrt{75^2 + 140^2}cm}{200\ cm} = 38.45°$$

$\alpha_{min} = 76.90°$, Therefore, the only option is 90°.

Part b) a.

It is assumed that the camera spans only the 280*280[3] region including the table. It should be square since the pixels and the resolutions are said to be square.



The ball center is in the 5th pixel; however, due to the +- 1-pixel accuracy, it can be represented in the other 8 pixels according to the 8-connectivity. So, the maximum error can be calculated as following:

$\frac{280\ cm}{p_{min}} * \sqrt{2} = 1\ cm$, p is # of pixels in one side, so 280/p is the size of 1 pixel, $\sqrt{2}$ is the distance between 2 diagonal pixels. (Max error is obtained when the ball is represented on one of the pixels 1, 3, 7, or 9)

$p_{min} = 395.9$, According to this condition, 512*512 is sufficient to use.

Part b) b.

In order to have the ball at least 100 pixels on it, the diameter should have $2 * \sqrt{\frac{100}{\pi}}$ many pixels. (The formula $\pi r^2 = area$ is used.) So, n_{min} = 11.28. It should be integer therefore n is equal to 12.

$\frac{280\ cm}{p_{min}} * 12 = 5.7\ cm, p_{min} = 589.4$, According to this condition, 1024*1024 should be used.

[3]In part a, it says that the most span is 280. But if there is a typo so that it should be must not most, then it can span 400*400. However, the same results are obtained.