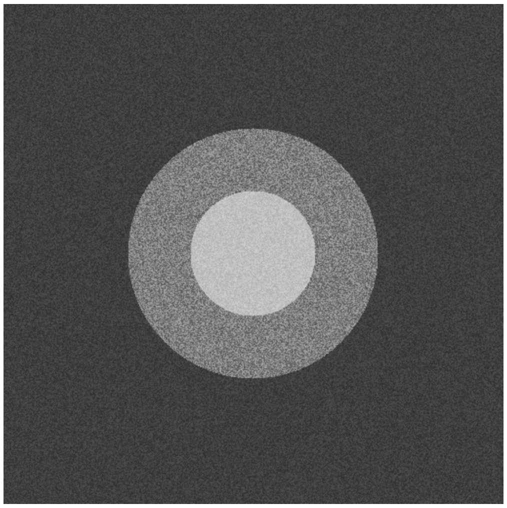
EE475 HW-1

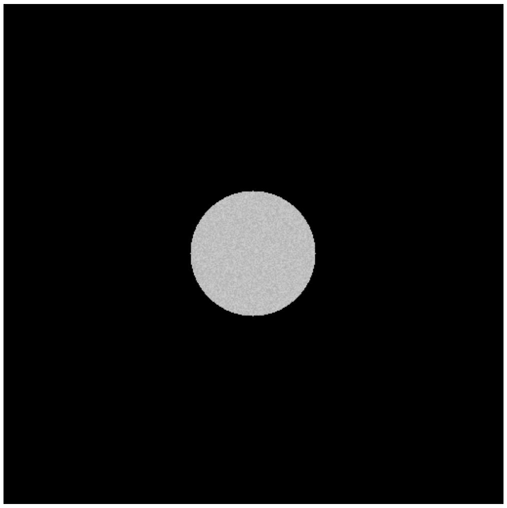
Ibrahim Kahraman 2015401108

23 September 2019

Question 1



(a) *generating_image.tif*



(b) *selection_inner_circle.tif*

To generate an image that composed of two circle, i create a matrix, image_matrix, whose size is 512x512 and i define the center_pixel as (256,256). Then i write a function, distance_to_center, whose inputs are i,ii and center_pixel. This function calculates the euclidean distance between (i,ii) and (center_pixel, center_pixel). Then i start two nested for loop and in the case of distance i give related values to all pixels by using the values in the question and also by using randi function. The i changed the matrix type from int to uint8 and save it as generating_image.tif.

In order to choose inner circle, i use

$$selection\_inner\_matrix = image\_matrix > 175;$$

'selection_inner_matrix' is the boolean matrix that the place that include inner circle is 1, otherwise 0. I multiply each component with the normal values in the image_matrix. Then i changed the type of the matrix to uint8. Then saved this matrix as selection_inner_circle.tif.

The values at the inner circle is greater that 175 and the other pixels are smaller than 175. Therefore, to select inner circle was easy.

Question 2-a

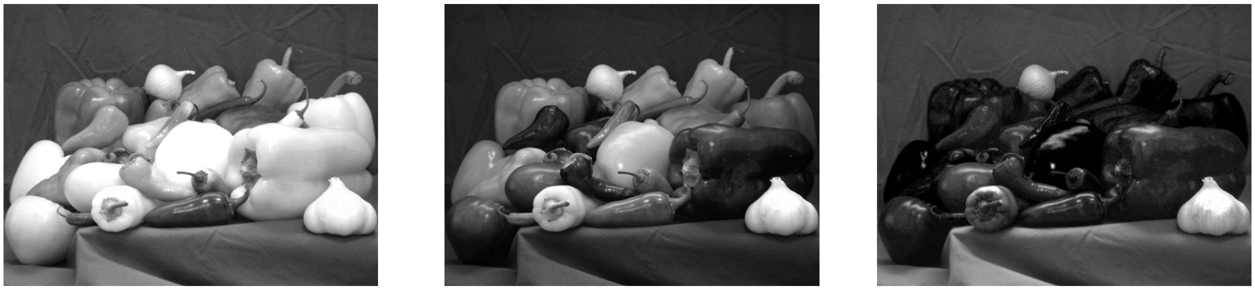


Figure 2: The pictures of the R-G-B components

The first picture is the R component of the 'peppers.png'. This is a gray scale picture that determined by only R component. Therefore, more white places in the first figure has more R value in real picture. So we can conclude that white places in the first figure is either white or red in the real figure. On the other hand, third figure is more black and the first figure is more white. So we can conclude that the red color is the most frequent and blue color is least frequent in the real image.

Question 2-b

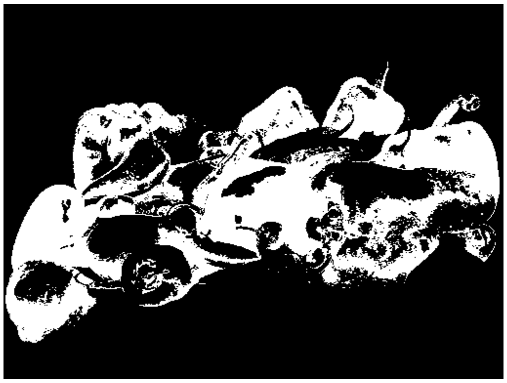


Figure 3: The places that $R > 90$, $G > 10$, $B < 40$

If we just plot the pixels that $R > 90$, we will obtain red pixels plus some other pixels, for instance white pixels have $R > 90$. To eliminate the point like this, we can make some constraint. $B < 40$ eliminates the both white pixels and green pixels. The effect of $G > 10$ is not big, it changes less pixel.

Question 2-c

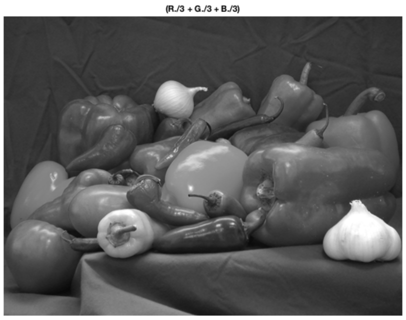


Figure 4: $(R./3 + G./3 + B./3)$

If we sum R,G,B components, there will many pixel that exceeds 255. In order to calculate average, i use the formula

$$new\_image\_matrix = (r\_I./3 + g\_I./3 + b\_I./3);$$

Moreover, we can transform each component to double, then calculate avrage, then transform type to uint8. These two ideas are approximately the same.

Question 2-d

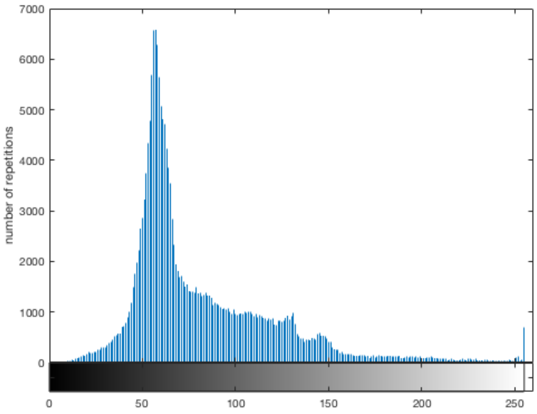
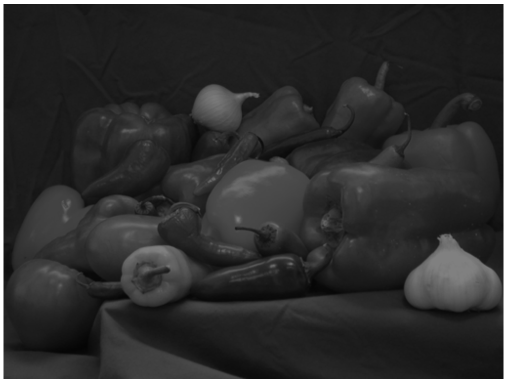


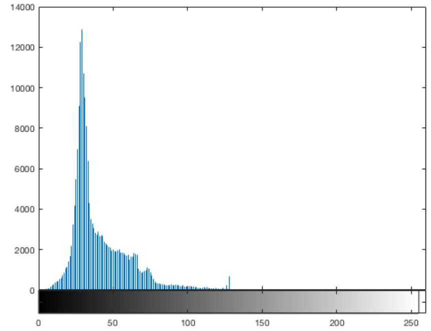
Figure 5: The histogram of 'gray.bmp'

X values represent the average values of R,G,B values of the pixels. So the range of x is [0 255]. Y values represent the number of repetitions per average value. The range of Y is [0 6600]. There is an absolute max at x=57. It means that maximum number of repetitions at average value is 57. Majority of the pixels have average value 57. The right part of the histogram is low. We can understand that the number of white pixels are not much. There is a local max at x=255 because of the white points.

Question 2-e

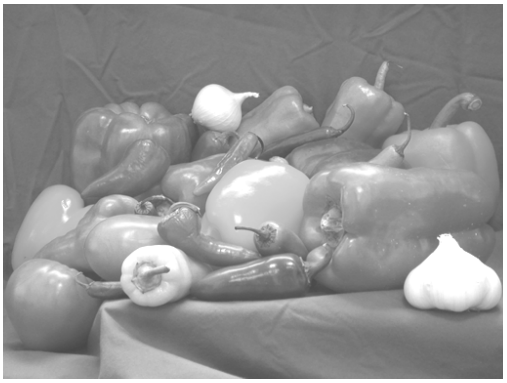


(a) grayscale./2 figure

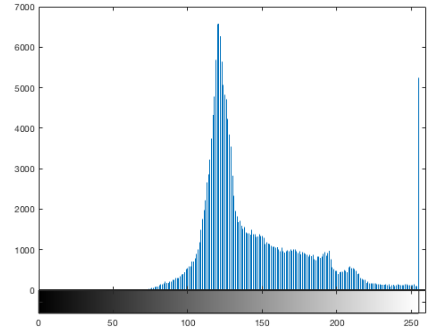


(b) grayscale./2 histogram

We compressed the histogram. The max x-value of the histogram is now 128. Moreover the maximum average repetitions is now at x=29. On the other hand, the figure is now more black than the before.



(a) grayscale./2 figure



(b) grayscale./2 histogram

By adding 64 to each pixel shifts the histogram 64 unit right. However, there are many point that exceeds 255, so they will be 255 still. Therefore, the y value at 255 increases most. The figure is now more white. The yellow parts of the figure is now white.

Question 3-a

To find the pixels that emits maximum of light, i find the max values of the R,G,B values. All these are 255. Then i find the pixels that [R G B]=[255 255 255]. As a result, there are 85 points like that. Some of them are:

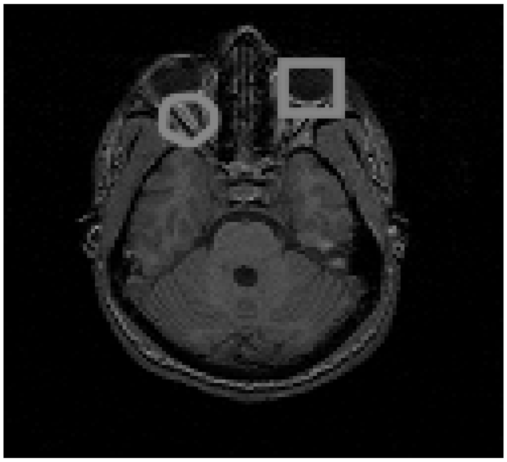
(18,1383) (52,782) (90,1338) (119,1192) (180,732) (270,787) (500,428) (595,1095) (777,988) (951,28)

Question 3-b

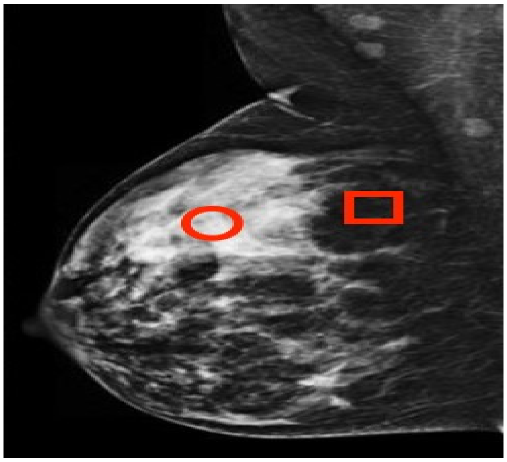
At the range camera, the closer points are more white and the farther points are more black. Max pixel value is 223 and min pixel value is 67. The closest points are:

(570,817), (570,818), (571,817), (571,818), (572,817),(572,818)
On the matlab code, 'closest_coordinates' parameters gives the closest poibts and some of the farthest points are:
(147,638), (198,573),(238,585)

Question 3-c



(a) *generating_image.tif*



(b) *selection_inner_circle.tif*

For skull:
The more white pixel color represents more transparent points, The only pixel that has the grayscale value 146,which is the max grayscale value, is the coordinate (64,25). I also draw this location with a circle. The desest points determined by hand and the rectangle define these points at the region of interest.
For breast:
I transform the 'Breast_density.jpg' from RGB to grayscale by using the matlab function gray2gray. Then find the max value and it is 246. The coordinates that has a pixel value 246 are (214,110), (219,73), (220,71). I also draw these places in a red circle. The desest points determined by hand and the rectangle define these points at the region of interest.

Question 3-d

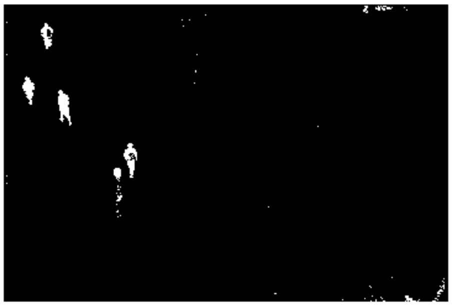
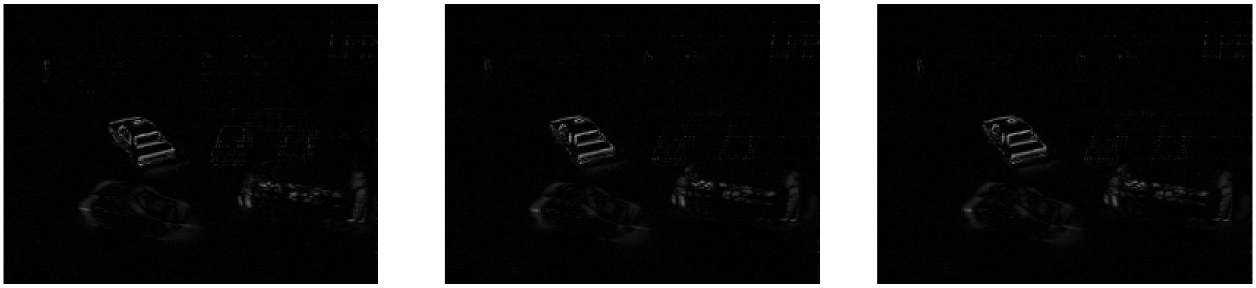


Figure 9: The points that the pixel value> 110

The termal camera clasify the the points by thir termal emission. So the hottest points will be the the places that contain human in this picture. I binarized the picture by putting 1 when pixel value is greater than 110 and otherwise 0. Change boolean to uint8 and multiply it by 255. Then i obtained the Figure 8.

Question 3-e



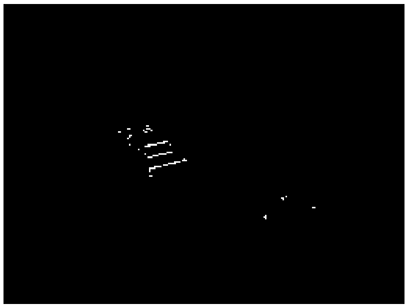
36 to 38 - 38 to 40 - 36 to 40

To find fastest changing points, i changed the type of the matrix from uint8 to double, then i found the difference between matrix and take the absolute values. Then i changed the type uint8 and figure it. The fastest changing points are belong the the white car.
To find maximum changing points, first i binarized the difference image like: (for the change between taxi36 to taxi38)

$$temp1\_2 = max(max(diff1\_2));$$

$$new\_matrix\_1 = diff1\_2 > temp1\_2/2;$$

then i changed type of new_matrix_1 from boolean to uint8 and multiply it by 255. The resulting figure is:



Binarize version of changing

In this figure, we can easily understand that the fastest changing points belong to white car.