EE475 Homework #1

Batuhan Tosun, *2017401141*

I. IMAGE READ & WRITE – HISTOGRAM PLOT

The image file "fish.bmp" is stored in an array using MATLAB's imread function. Then, using MATLAB's imwrite function the array is written to the disk in ".png" format. Also, MATLAB's iminfo function is used to se the contents of the written file.

Fig 1. Display of the image "fish.bmp"

A. *Extracting R, G, B Color Components & Analysis*

R, G, B color components of the Fish extracted into three different arrays and then each array is displayed using MATLAB's imshow function.

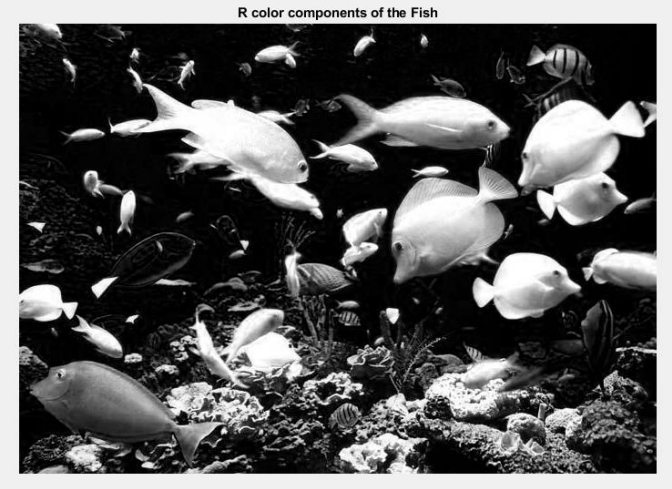

Fig 2. Display of R color components the image "fish.bmp"

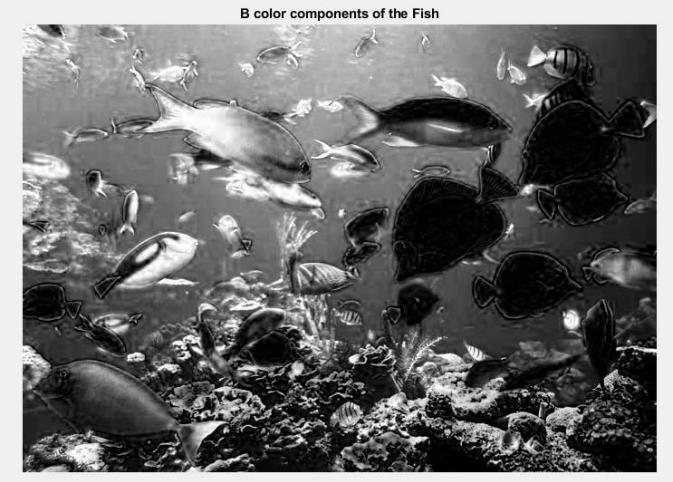

Fig 2. Display of B color components the image "fish.bmp"

Fig 3. Display of G color components the image "fish.bmp"

It is not a surprising situation that R components exist not only in the red regions but also in the image locations where blue, yellow or green colors dominate since many colors that we see in an image is a combination of the three main color components. The answer is the same for the case with B components and G components.

B. *Creating Grayscale Image*

A new image named "Gray_Fish.bmp" is created by taking the average of the color components (R+G+B)/3. The resulting image is displayed using MATLAB's imshow function. Note that to avoid overflow, first the type of the pixel value is converted to double and after the operations converted back to uint8.

Fig 4. Display of the image "Gray_Fish.bmp"

C. *Plotting the Histograms*

Using MATLAB's imhist function, the histograms of the R, G, B, and gray-level components are plotted separately.

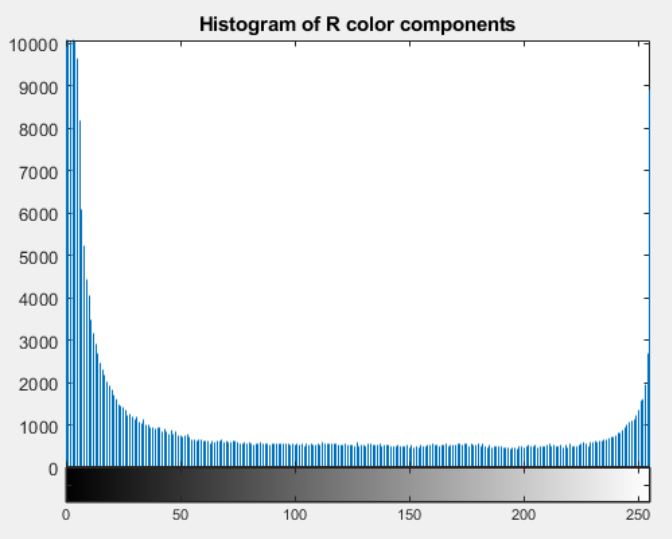

Fig 5. Histogram of R color components

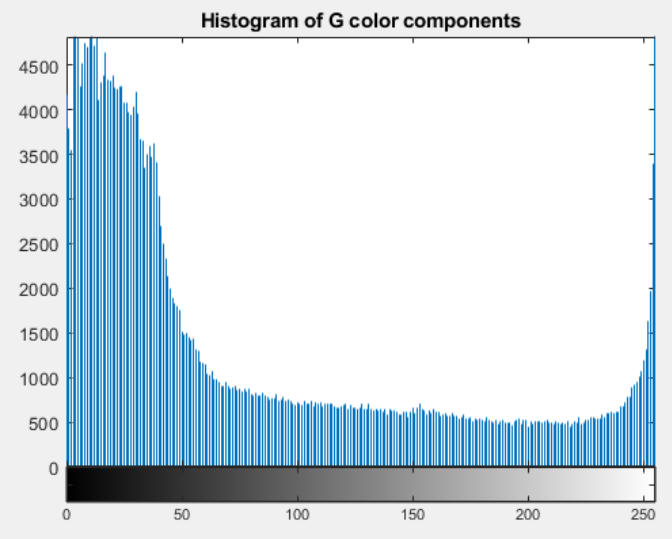

Fig 6. Histogram of G color components

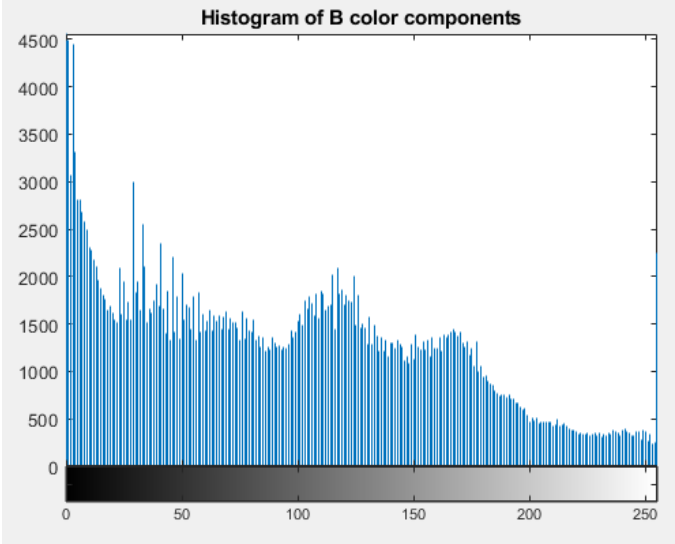

Fig 7. Histogram of B color components

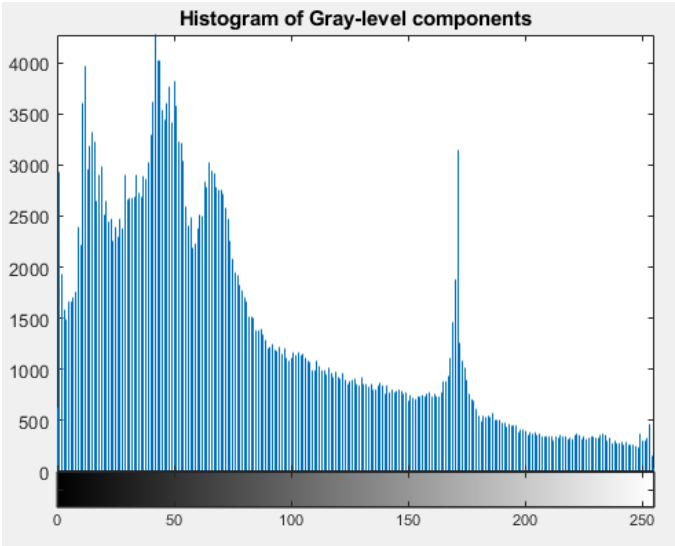

Fig 8. Histogram of Gray-level components

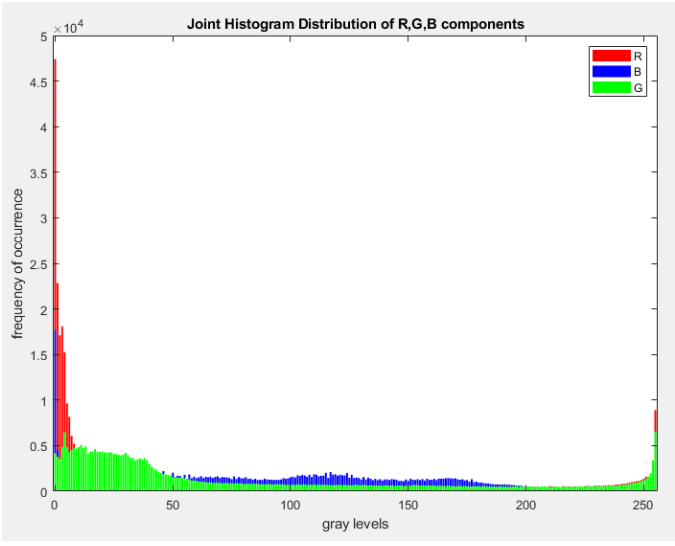

Fig 9. Joint Histogram of R,G,B color components

From the histogram plots, it can be observed that they occupy all of their gray level axes but not evenly. For example, R and G color components are mostly concentrated at lower gray levels whereas B color components are mostly concentrated at the first 200 gray levels. So, their distributions are uneven.

D. Creating Binary Images: R>150, G>130 and B<10

The spatial locations of the pixels that satisfy the R, G, B color value condition R>150, G>130 and B<10 are found and a new binary matrix is created which has 1 in the corresponding spatial locations and otherwise 0. The binary image is displayed after converting the 1's to 255 and keeping 0's as 0.

Fig 10. Binary Image in the case R>150, G>130 and B<10

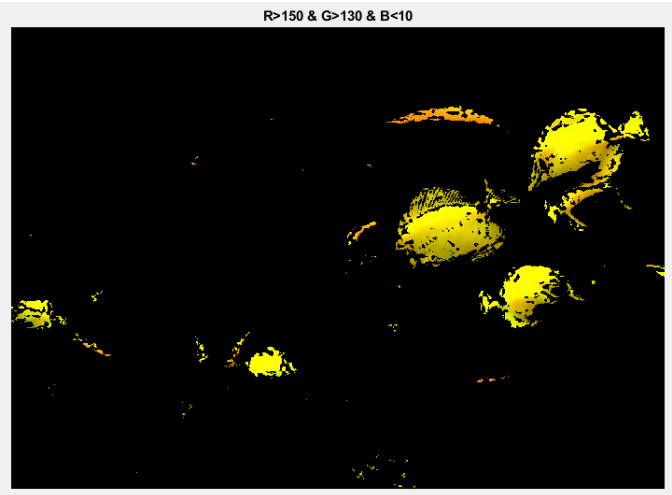

Fig 11. Resulting Image after Using the Binary Image Matrix as a Mask on the Original Image

From the resulting image (with three layer) obtained after the multiplication of the binary image matrix and the original image, we can observe that the pixel with weak blue component corresponds to yellow color. Therefore, yes these "255-valued" patches have yellow content in the image.

E. Creating Binary Images: R<100, G>130 and B>130

The spatial locations of the pixels that satisfy the R, G, B color value conditions R<100, G>130 and B>130 are found and a new binary matrix is created which has 1 in the corresponding spatial locations and otherwise 0. The binary image is displayed after converting the 1's to 255 and keeping 0's as 0.

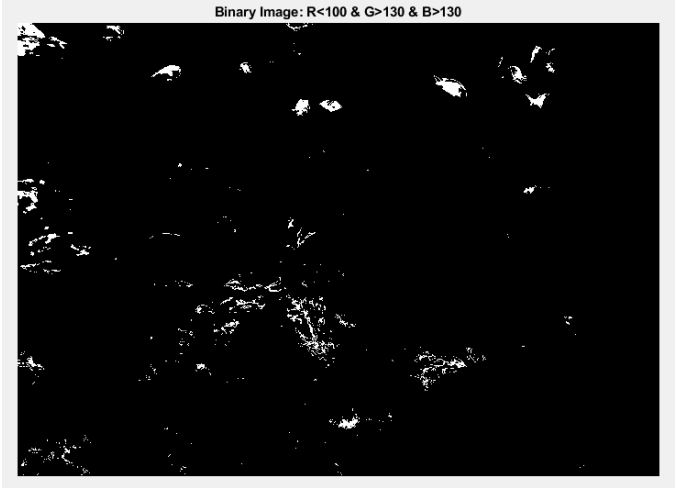

Fig 12. Binary Image in the case R<100, G>130 and B>130

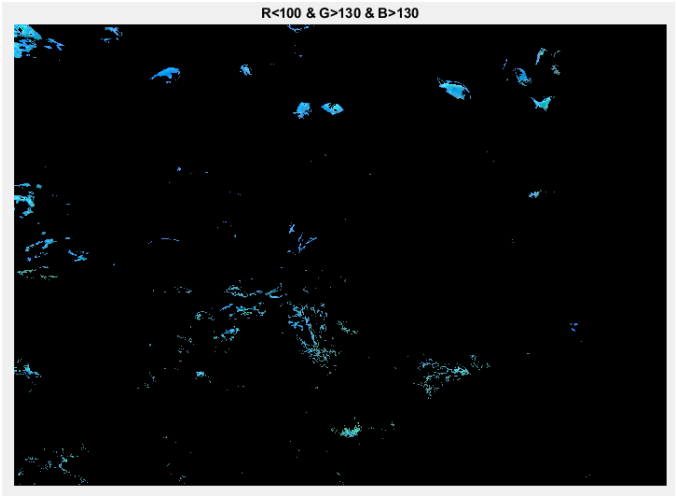

Fig 13. Resulting Image after Using the Binary Image Matrix as a Mask on the Original Image

From the resulting image (with three layer) obtained after the multiplication of the binary image matrix and the original image, we can observe that the pixel with weak red component corresponds to cyan color. Therefore, yes these "255-valued" patches have cyan content in the image.

F. Creating Binary Images: R>130, G<100 and B>130

The spatial locations of the pixels that satisfy the R, G, B color value conditions R>130, G<100 and B>130 are found and a new binary matrix is created which has 1 in the corresponding spatial locations and otherwise 0. The binary image is displayed after converting the 1's to 255 and keeping 0's as 0.

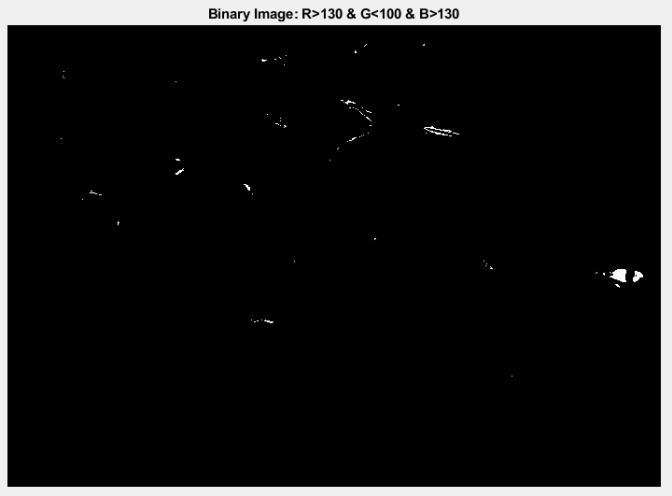

Fig 14. Binary Image in the case R>130, G<100 and B>130

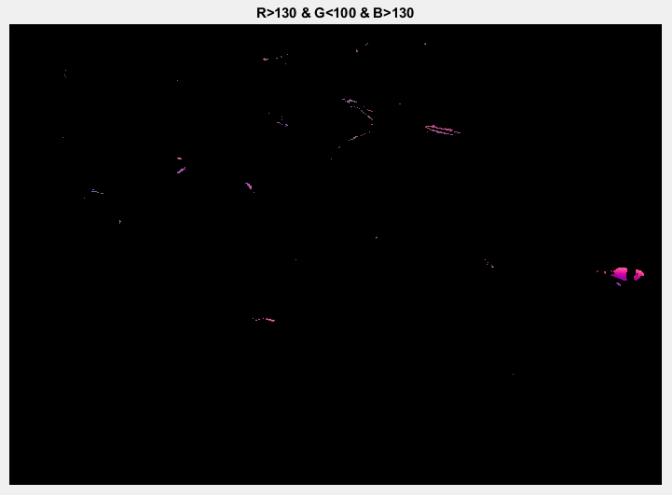

Fig 15. Resulting Image after Using the Binary Image Matrix as a Mask on the Original Image

From the resulting image (with three layer) obtained after the multiplication of the binary image matrix and the original image, we can observe that the pixel with weak green component corresponds to magenta color. Therefore, yes these "255-valued" patches have magenta content in the image.

G. *Finding the Three Reddest Pixels*

I chose my three reddest point coordinates as (325,266), (326,264), (327,264). A figure for the zoomed region is given below.

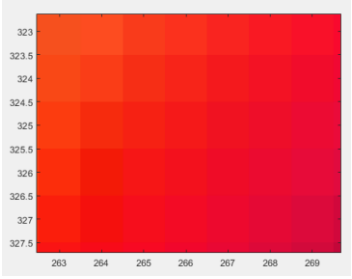

Fig 16. Zoomed Reddest Region in the Image

For a pixel to be considered as the reddest it should have R component around 255 while B and G components around 0. Therefore, first I have found the coordinates of the pixels with R component value of 255. Then, I have set a threshold for B and G components that those pixels should not pass. At the end, the coordinates of the three reddest points are found as: (322, 270), (322, 269), (322,268) with R component value of 255. The points that I chose are actually pretty close to the real reddest points.

II. AVERAGE OPTICAL DENSITY

A. *Finding the Average Optical Density of the Images*

Average Optical Densities (AODs) of the images are found with respect to the given formula.

```
"rsz_05_00_gray.png, AOD => 49.2765"

"rsz_05_05_gray.png, AOD => 89.078"

"rsz_05_11_gray.png, AOD => 114.0295"

"Baby.png, AOD => 66.3749"
```

B. *Plotting the Histograms of the Images*

Using MATLAB's imhist function, the histograms of face images are plotted.

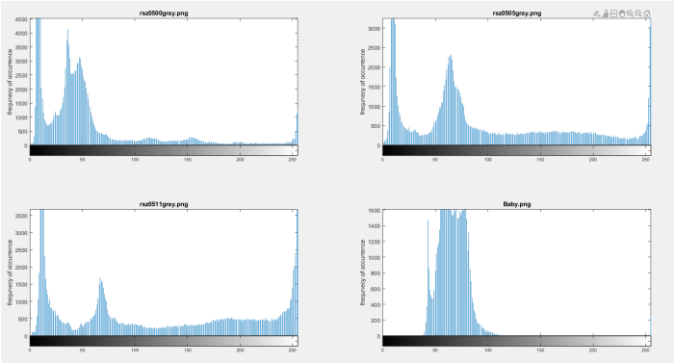

Fig 16. Histogram Plots of Four Images

C. *Setting the AOD Value to 100 by Adding a Constant*

In this part, AOD Values of images are increased/decreased to 100 by adding a different constant c to each image. To find the corresponding c value for each image, the original AOD value of the image is subtracted from 100 and the result is taken as c, value to be added to each pixel of that image. During the operations, to avoid overflow, the type of the pixel values is converted to double and after adding the constant c the resulting pixel values which are below 0 are set to 0 and which are above 255 are set to 255. The previous and resulting images and histograms of the images are given below.

Fig17. Images: Originals in the first, AOD adjusted to 100 ones in the second row

From the resulting images, we can observe that when an original image ("rsz_05_11_gray.png") with AOD value greater than 100 undergoes such operations to set its AOD value to 100 (decrease) its brightness is decreased, and this make sense considering the fact pixel values shifted to lower gray levels to achieve the desired situation. Also, when an original image (other images) with AOD value smaller than 100 undergoes such operations to set its AOD value to 100 (increase) its brightness is increased.

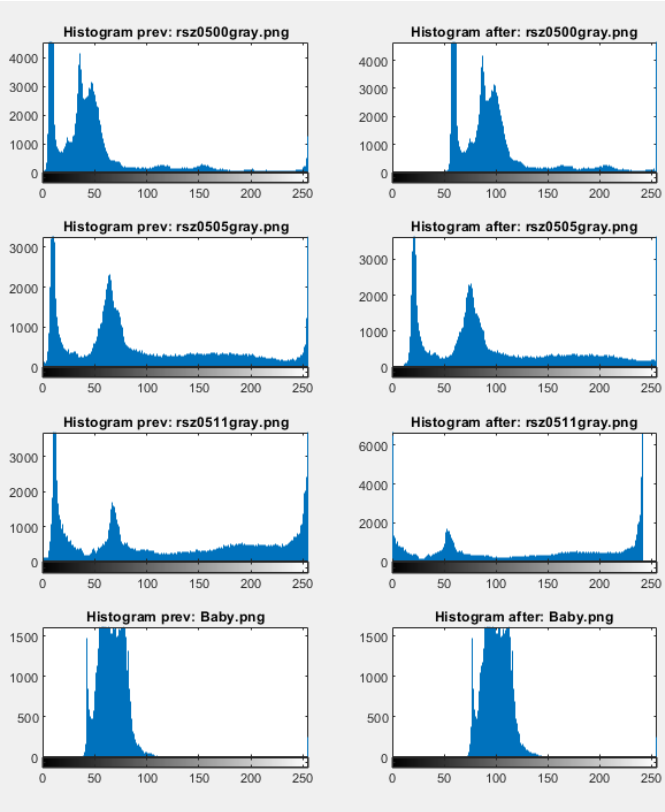

Fig 18. Previous and Resulting Histograms of Images Side-by-side

From the histogram plots, we can observe the shift of the pixel values better. Since normally the image "rsz_05_11_gray.png" has a higher AOD value than 100, its pixel values are shifted to left and resulted an accumulation at 0 gray level. The other images originally have lower AOD value than 100 therefore their pixel values are shifted to right, resulting in accumulation at 255 gray level.

D. Changing the Luminance Range of the Gray Image

Luminance range of the gray images are changed by multiplying each pixel with a certain number a for two different situations (a=0.5 and a=2). Below a figure is given in which the original images are in the first row, the ones multiplied with a = 0.5 are in the second row and the ones multiplied with a = 2 in the last row (keeping their AOD values same).

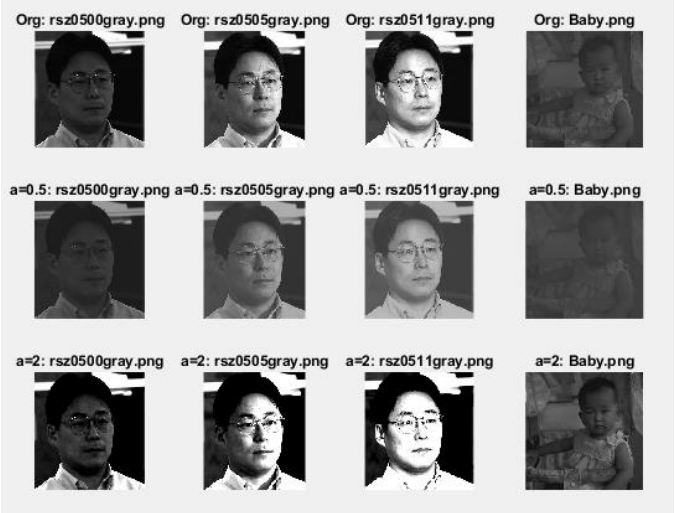

Fig 19. Images: Originals in the first, Luminance Range changed ones with a=0.5 in the second, Luminance Range changed ones with a=2 in the last row

From the figure, we can observe that when the factor we multiply with is smaller than 1, gray level range gets decreased, and therefore, the image gets darker. Whereas when the factor is greater than 1, gray level range gets increased, and therefore, the image gets brighter.

III. PIXEL VARIETIES

A. If an Object is Self-Luminous,

First, to find the coordinates of the point that emits the most amount of light in the image, we need to find the pixel that has the highest R, G, B values (each equal or close to 255) compared to other pixels in the image. Therefore, a binary matrix is created considering the coordinates of the pixels that satisfy the condition of R=255, G=255, and B=255. According to the results, in total, there are 85 pixels in the image that satisfy this situation. The locations/coordinates of those pixels are found using MATLAB's find function and the resulting binary matrix. There are 85 coordinates (x, y) and 5 of them are given below (for all coordinates source code should be run):

```
951    28
250   208
500   425
498   426
500   426
```

Considering the given coordinates above, there is a white point accumulation around the coordinates (500, 425).

Fig 20. Image "Cetus_NGC1052Galaxy.jpg"

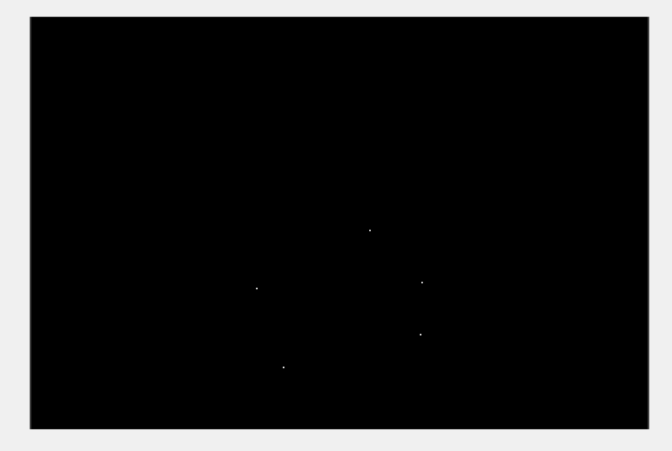

Fig 21. The Coordinates of the Points that emits the most light

B. If an Image is Taken with a Range Camera,

If an image is taken with a range camera, the closest points in the image are the darker ones whereas the farthest points are the whiter ones, which can be observed looking at the example images.

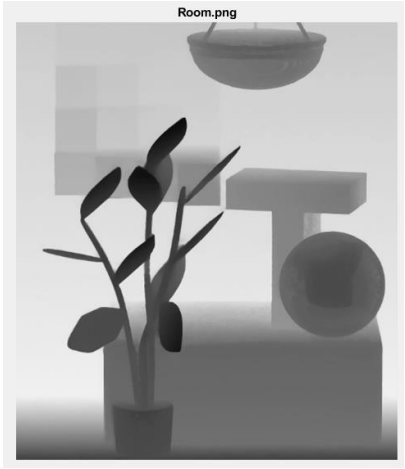

Fig 22. The Image "Room.png"

The coordinates of the closest point in the image is (122,207) with a pixel value of 5. The coordinates of one of the farthest points in the image is (442,448) with a pixel value of 244.

C. If the Image Represents Material Density,

If the image represents material density, the opaquest (densest) points are the whiter areas (with a pixel value around 255) whereas the least opaque (most transparent) points are the darker areas (with a pixel value around 0).

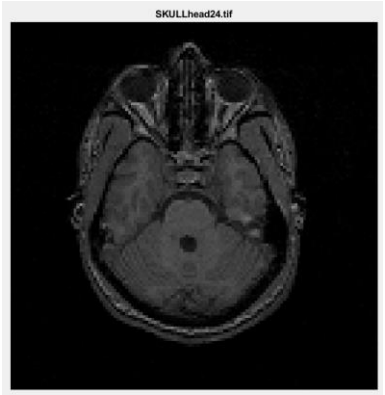

Fig 22. The Image "SKULL_head24.tif"

Fig 23. Original, Only Opaque and Only Transparent Region Images

D. If the Image is Captured with a Thermal Camera

If the image is captured with a thermal camera, the hottest points are the whiter regions (pixels with value close to 255) and the coldest points are the darker regions (pixels with value close to 0). Therefore, I found the coordinates of the pixels with value of 255 to find the hottest points and the ones with the value of 0 to find the coldest points.

There are 27 coordinates with a pixel value of 255, and 272 coordinates (x, y) with a pixel value of 0. Five of the

coordinates of the hottest point are given below:

110	95
123	97
124	97
125	98
175	174

,

Five of the coordinates of the coldest point are given below:

2	2
3	3
4	4
5	5
7	7

I also created a binary matrix for that maps the hottest points in the given image with respect to a threshold value (200). The resulting binary image(matrix) shows the hotter regions with pixel value greater than 200. It can be observed that "humans" are hotter than the surroundings in the image.

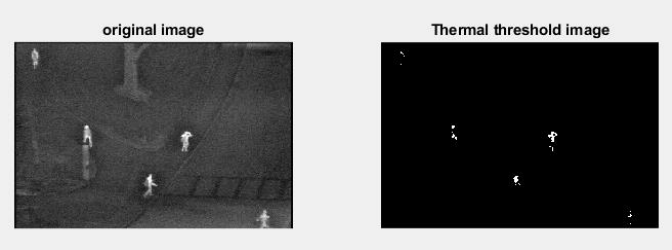

Fig 24. Original Image and Threshold Masked Image

E. Shots from the Hamburg Taxi Sequence

The original images of the three shots from the Hamburg taxi sequence are given below.

Fig 25. Three Shots From the Hamburg Ttaxi Sequence

First, the difference of the images |Taxi2 – Taxi1|, |Taxi3 – Taxi2|, and |Taxi3 – Taxi 1| are found then their gray level inverse is taken for better visibility. The resulting images are given in the right section. Since we have taken the gray level inverse of the (difference) images, the fastest changing points will be the darker points in the resulting images (normally higher magnitude of the difference/change corresponds to higher pixel values but subtracting it from 255 changes the situation).

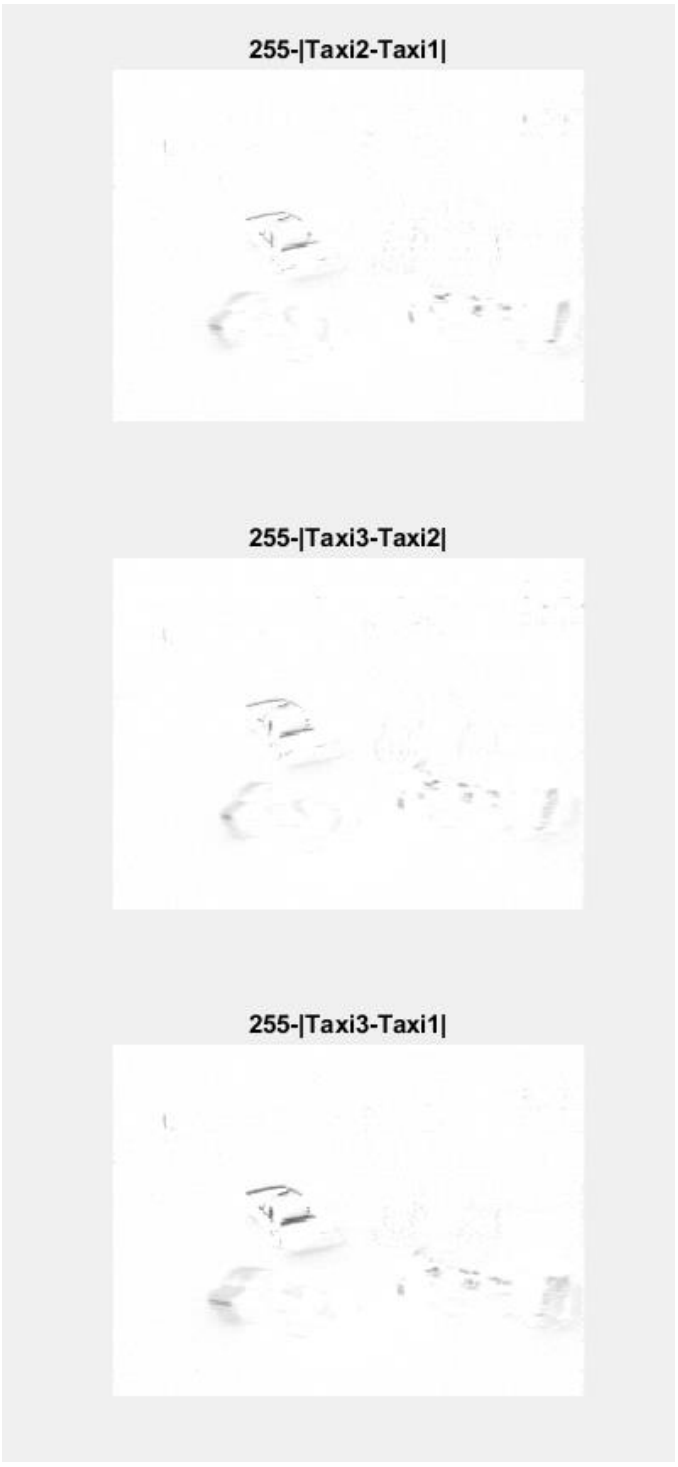

Fig 26. The difference of the images |Taxi2 – Taxi1|, |Taxi3 – Taxi2|, and |Taxi3 – Taxi 1|

IV. CCD CAMERA

It is given that the distance is 50cm (=500mm). Therefore, the target size z can be found by:

z = (14mm/35mm) * 500mm=200 mm.

There are 2048 elements per line. Then, the number line pairs per mm that this camera is able to resolve can be found by:

2048/(200*2) ≈ 5 line pairs/mm

V. IMAGE BIT SLICING

In this part, 8-bit representation of each pixel value is obtained using MATLAB's dec2bin function. Two binary images are created with the use of 8-bit representations. The first one is obtained by considering only the most significant bit (left-most) of each pixel and the second one by considering the least significant bit (right-most). The original, MSB and LSB binary images are given below.

Fig 27. Original, MSB Binary and LSB Binary Image

From the resulting images, it can be observed that when a binary image is formed using only the most significant bit of each pixel value Lena is still perceptible whereas in the case of LSB not. The result is not so surprising since the most significant bit of a pixel value gives us the information whether the pixel value is bigger than 128 or not. Therefore, while creating a binary image by only considering MSB we are assigning high pixel values to highest possible value and low pixel values to lowest possible value, in other terms increasing the contrast in the image. As a result, we still get a reasonable image. However, in the case of LSB, since the lowest significant bit does not give significant information (magnitude) about the pixel value, considering in the creation of a binary image gives us a randomly assigned pixel values. In fact, what we get using LSB looks like a random noise. As a result, we cannot perceive Lena.