EE475 **Sezen Perçin**
Homework 1

```
          FileModDate: '02-Nov-2020 22:58:07'
             FileSize: 218756
               Format: 'png'
        FormatVersion: []
                Width: 385
               Height: 272
             BitDepth: 24
            ColorType: 'truecolor'
      FormatSignature: [137 80 78 71 13 10 26 10]
             Colormap: []
            Histogram: []
        InterlaceType: 'none'
         Transparency: 'none'
SimpleTransparencyData: []
      BackgroundColor: []
      RenderingIntent: []
       Chromaticities: []
                Gamma: []
          XResolution: []
          YResolution: []
       ResolutionUnit: []
              XOffset: []
              YOffset: []
           OffsetUnit: []
      SignificantBits: []
         ImageModTime: '2 Nov 2020 19:58:07 +0000'
                Title: []
               Author: []
          Description: []
            Copyright: []
```

1. The information about the written image is displayed as the output with the code seen below.

```
%% Section 1
%%reading the image and storing it in array called 'I'
I=imread('fish.bmp');
figure(1)
imshow(I);
%%creating the png file and calling information function
imwrite(I,'fish.png');
imfinfo('fish.png');
```

```
%% 1.a
R=I(:,:,1);
figure(2)
imshow(R)

G=I(:,:,2);
figure(3)
imshow(G)

B=I(:,:,3);
figure(4)
imshow(B)
```

a. Since the third column of the image matrix consists of the 3 elements, (namely R,G,B) I seperated these elements, hence extracted the RGB components with the code on the left. Here are the outputs in the order of R,G,B:

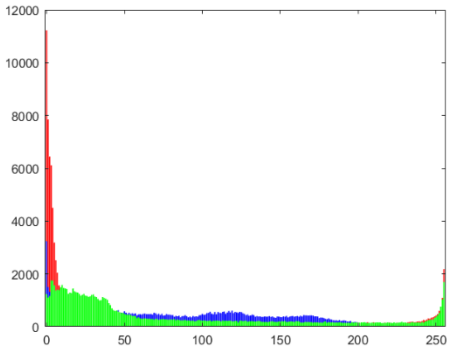


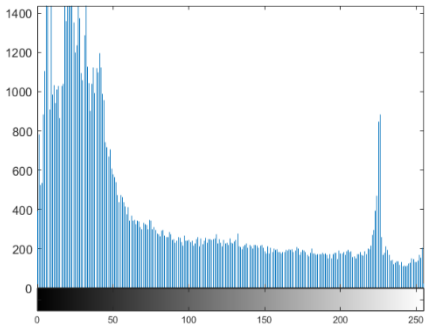
In those images, lighter areas represent the regions where the selected component has the larger values. Together, larger R,G,B values generates lighter colors (e.g. white is represented as the (255,255,255) which is max on this scale), it is not suprising to see all the components in lighter areas even if they are not red, green or blue on the original image. For primary colors to be appear in the image, that color component should have high values while the others should be very low. Therefore these colors can be only visible on the regions that appear lighter in their components plot but ,darker on the other two.

b. By using extracted components and taking the sum of them we create an image with only 1 color component, which varies from 0(black) to 255 (white). In order to prevent data loss, all the operations are performed on double type variables.

```
%% 1.b
figure(5)
I2=(double(R)+double(G)+double(B))./3;
imwrite(uint8(I2),'Gray_Fish.bmp');
imshow(uint8(I2));
```

c.I did plot all the histograms belonging to RGB components w.r.t their own colors. As it can be seen below, components are not distributed evenly on the range of 0 to 255. Second plot shows the histogram of the grayscale image which indicates that image is mainly dark colored, since many elements are closed to 0.





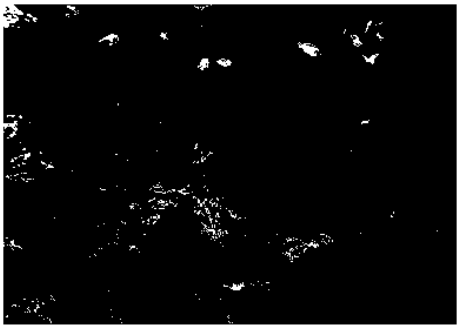

```
%% 1.c
figure(6)
%%in order give the histograms their own value i plotted them as bars
%%then gave them color attributes
[x,y]=imhist(R);
bar(y,x,'FaceColor','r');
hold on;
[x,y]=imhist(B);
bar(y,x,'FaceColor','b');
[x,y]=imhist(G);
bar(y,x,'FaceColor','g');
hold off;
%%histogram of gray scaled image
figure(7)
imhist(rgb2gray(I));
```

```
%% 1.d
%%selecting the pixels that satisfy the given conditions
%%since the resulting matrix has only logical elements
%%it can serve as a binary image when it is provided to imshow()
I_sel = (I(:,:,1) > 150) & (I(:,:,2) > 130) & (I(:,:,3) < 10);
figure(8)
imshow(I_sel);
```

```
%% 1.e
%%same procedure as d
I_sel2 = (I(:,:,1) < 100) & (I(:,:,2) > 130) & (I(:,:,3) > 130);
figure(9)
imshow(I_sel2);
```

```
%% 1.f
%%same procedure as d
I_sel3 = (I(:,:,1) > 130) & (I(:,:,2) < 100) & (I(:,:,3) > 130);
figure(10)
imshow(I_sel3);
```

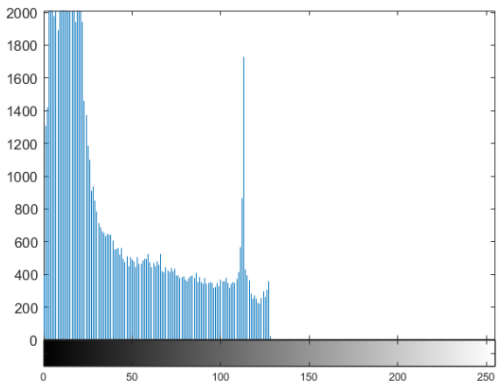
d. As it can be seen below, absence of the color blue responds the regions where its complementary yellow found on the image.



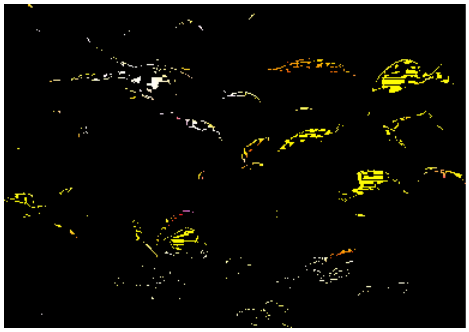
e. Yes, absence of the color red corresponds to regions where the cyan-ish colors can be found (e.g little cyan colored fishes on the background).

f.Yes, absence of the color green respononds to regions where its complementary magenta is present on the picture.

g.Since we divided all the pixel values by 2, their values have decreased. That caused its histogram to be distributed in the half of the whole range and the total profile has shifted to the left. That also caused image to be darker since its color values has approached to zero (black).



h. I chose the point on the right and extraxted the points where R has its max value from the original image with the code that can be below. Mostly maximum R valued regions did not have the color red, but fortunately the point i did choose was pretty close.



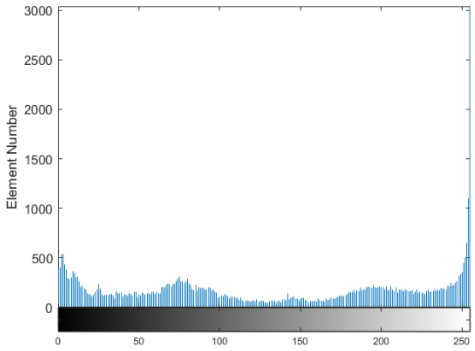
```
%% 1.h
%%finding the max value of the R component present on the image
[Rmax]=max(R,[],'all');
Rmaxmatrix=(R(:,:)==255);
figure(13);
%%extracting points where R is at max value from the original picture
imshow(I.*uint8(Rmaxmatrix));
```

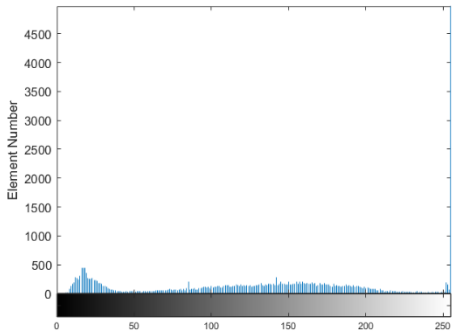
2. **a.**By applying the formula of the average optical density to given images i obtained the AOD values given below (the operation orders are organized with respec to the order of the images in the assignment document):

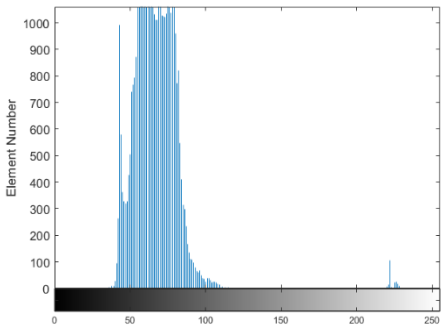
```
%% 2.a
%%since AOD is the average light density (pixel value) of the image
%%i sum all the values of the image matrix and divided it to image size
%%in order to prevent the data loss all operations are performed to double typed variables
AOD1=double(sum(i1,'all'))/(size(i1,1)*size(i1,2));
AOD2=double(sum(i2,'all'))/(size(i2,1)*size(i2,2));
AOD3=double(sum(i3,'all'))/(size(i3,1)*size(i3,2));
```

AOD1	189.8319
AOD2	166.5804
AOD3	66.3717
i1	*220x272 uint8*
i2	*217x286 uint8*
i3	*213x204 uint8*

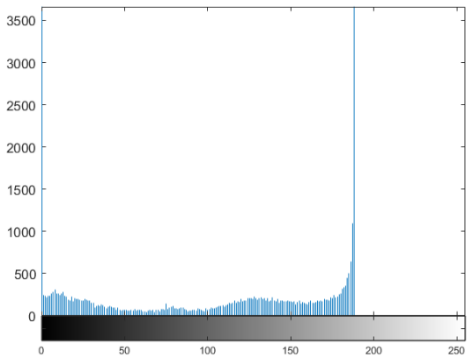
b.Histograms of the images can be seen below in the same order the images were given in the assignment document. It is also clear that an image does not always have to cover the whole 0-255 range and it should not have to. Since has lest contrast, its values are mostly in one area. The first images contain more varying elements on the 0-255 range.

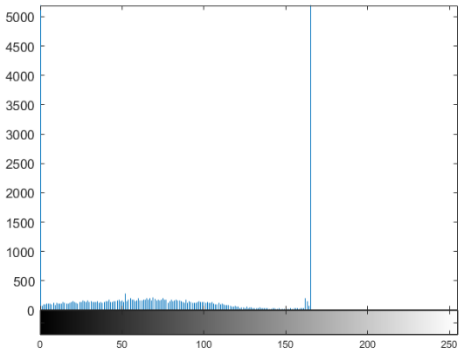


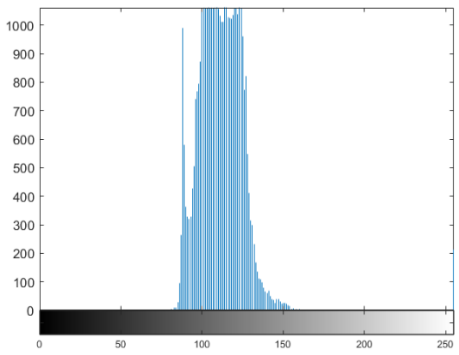




c.As we observe the new histograms below, we see that the values are now more cumulated around the point 100 or they are placed so that the average can be 100.







d.After we multiply the whole image elements by 0.5, the images has darkened but most importantly the contrast levels decreased ,dramatically. By multiplying the image elements by 2, we increased the contrast level of the image. It can be seen that there has been an overall lightening in the images.

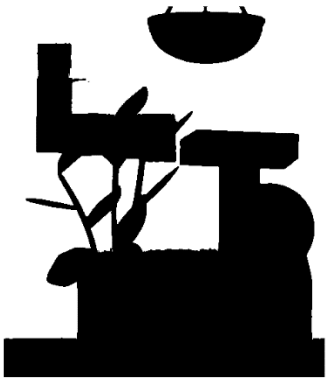





3. In gray scale images the value of the pixel in range 0-255 is very important for us. In most of the subsections of this question i made use of this property.

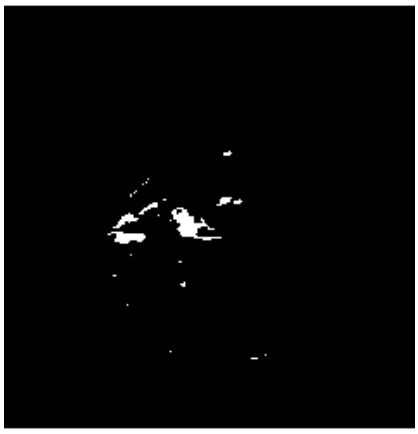
a.The most amount of light would be emitted by the brightest points on the galaxy image. Therefore i found the points that has the maximum value on the image and displayed it as a binary image.

b.Range cameras give the closest element the more darker colors and less values on the range. To be able to display an element of the room rather than farthest or closest point on the image, i made the threshold more flexible. For the closest elements i chose the points that have the values less than the closest (or darkest) element's value + 50. For the farthest element i chose the pixels that have greater values than the maximum value on the image – 50.

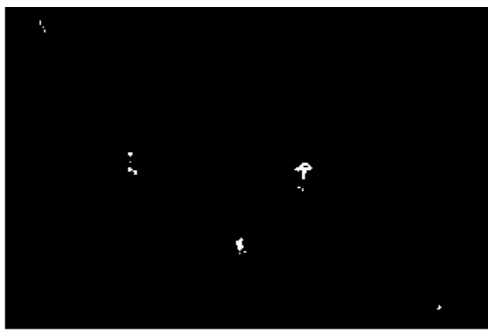
c.I used the same strategy for the Breast and Skull images. I took the brightest points to be the most transparent and the darkest points to be the densest.The resulting images can be seen below:

Darkest (most opaque) (breast and skull):



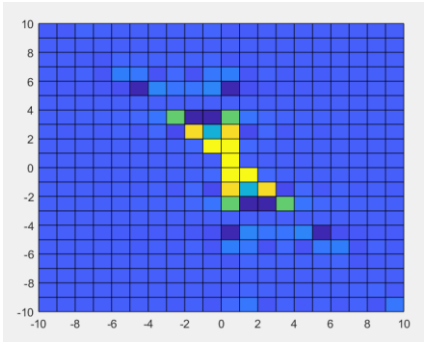
Most transparent:

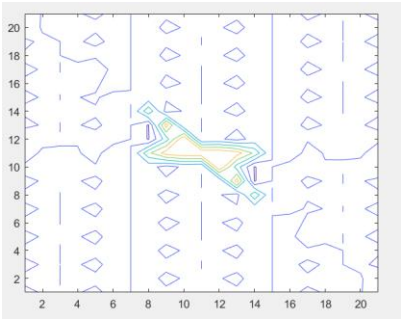
d.In thermal images, the hottest elements are represented, as the brightest points on the images. The colder points have lesser pixel values on the 0-255 range.Again i used a +-50 pixel threshold to obtain elements from images rather than points.

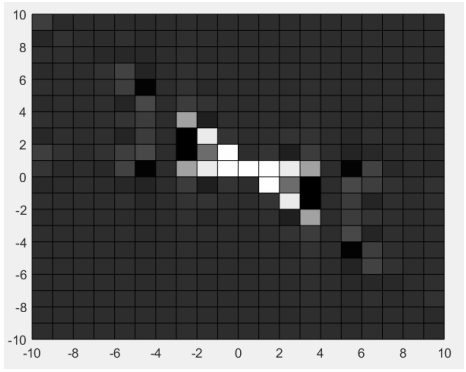


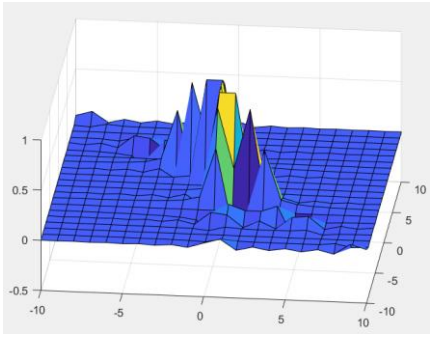
e. If we took the difference between the pixel values of the first image and the second one, the fastest changing points would have the greatest values. Therefore i took the brightest points on the image created by the difference between two pictures.The result can be seen on the right and the brightest points correspond to fastest changing pixels on the image.

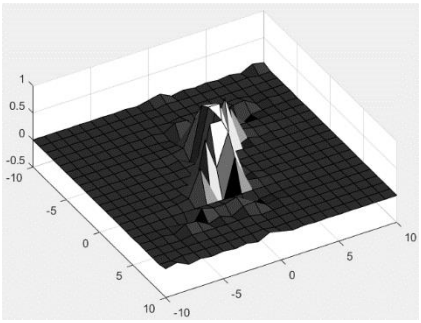
4. a.Isometric b.Contour c.Grayscale











5. Since 14mm/35mm=x/50cm x being the size of the element being pictured, x=20cm. If 20 cm=200mm will be represented by 2048x2048 elements, there are 2048/200mm=10 elements per mm. Then it can be said, that this camera can resolve 10/2=5 line pairs per mm.