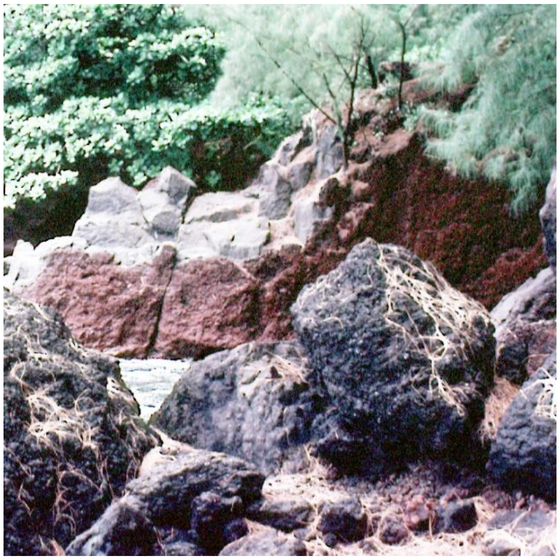
EE475 Fall 2020
HW #7 Color Image Processing **Sezen Perçin**

Histogram Equalization	6.15	Candies	Color edges	Total

1. Color Image Enhancement by Histogram Processing

(a)



```
%reading the image
dark=imread('Dark_stream.tif');
%separating into r-g-b components
R=dark(:,:,1);
G=dark(:,:,2);
B=dark(:,:,3);
%histogram equalization of the components separately
Req=histeq(R);
Geq=histeq(G);
Beq=histeq(B);
%constructing the image again from its components
darkeq=zeros(size(dark));
darkeq(:,:,1)=Req;
darkeq(:,:,2)=Geq;
darkeq(:,:,3)=Beq;
darkeq=uint8(darkeq);
imwrite(darkeq,'dark1a.tiff','TIFF');
```

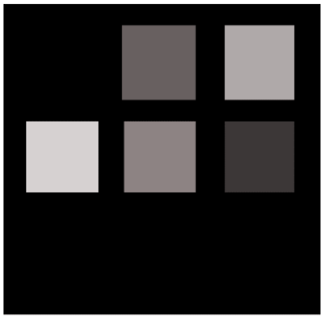
(b)

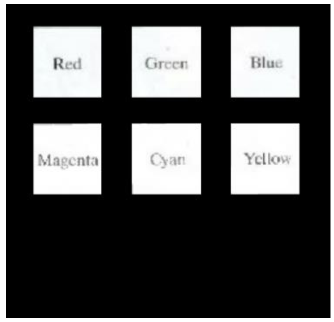
```
%getting the values that r-g-b histograms have
[rv,rbin]=histcounts(R,[0:255]);
[gv,gbin]=histcounts(G,[0:255]);
[bv,bbin]=histcounts(B,[0:255]);
%taking the average
histav=(rv(:,:)+gv(:,:)+bv(:,:))/3;
%applying the average histogram separately
R2=histeq(R,histav);
G2=histeq(G,histav);
B2=histeq(B,histav);
%constructing the second image
darkeq2(:,:,1)=R2;
darkeq2(:,:,2)=G2;
darkeq2(:,:,3)=B2;
%writing the image
imwrite(darkeq2,'dark1b.tiff','TIFF');
```

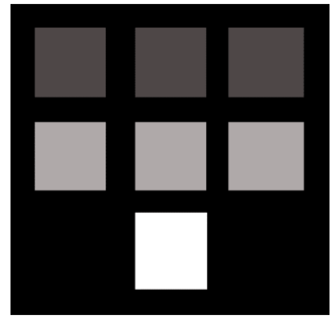
As in the gray scale case here also it is desired the correct the image problems caused by the general distribution of intensity values in the image. This here done by the histogram equalizing. In first step the color components are processed independently, and it caused image to have more erroneous colors. In second step the average of the histograms was taken, and it was applied to the whole image. With this way the necessary information was taken from r-g-b channels while leaving their colors unchanged. Therefore, an average intensity modification resulted in more accurate intensity and saturation wise corrected results.

Gonzalez, 6.13) :

(a) Sketch the hue image: (b) Sketch the saturation image: (c) Sketch the intensity image:





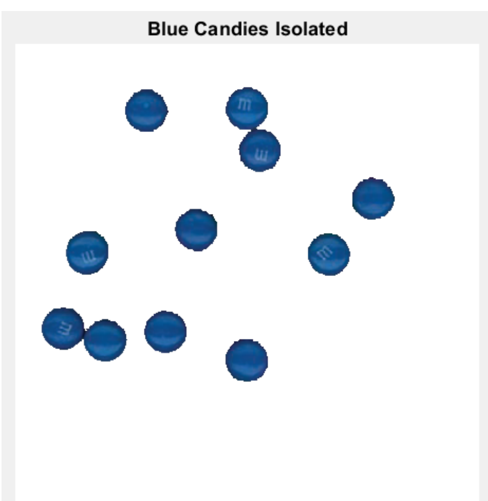


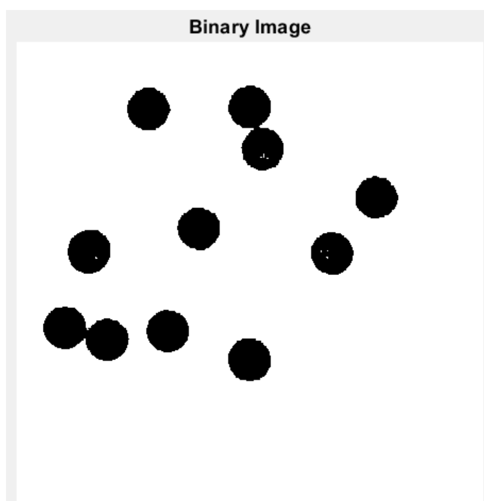
Since the hue represents the color that is perceivable by the eye and since the color are represented as angles, if we map the interval (0-360) to (0-255), all the colors will have an distinct gray value. For example since red is represented by the angle 0 it is represented as 0 on gray scale or green 120 would represented as 120*255/360=85 on gray scale. That was the reason of the color differences.

Since the colors are full spectral ones, their saturations are 255 except white and black since they are on gray scale and their saturation is 0.

R+G+B generates the maximum intensity of 255 on intensity plane. Therefore, color with R+G+B which is white have the max value of 255, then comes the 2 component mixed colors as magenta (R+B), then the single component ones and lastly the black.

Hue Saturation selection:



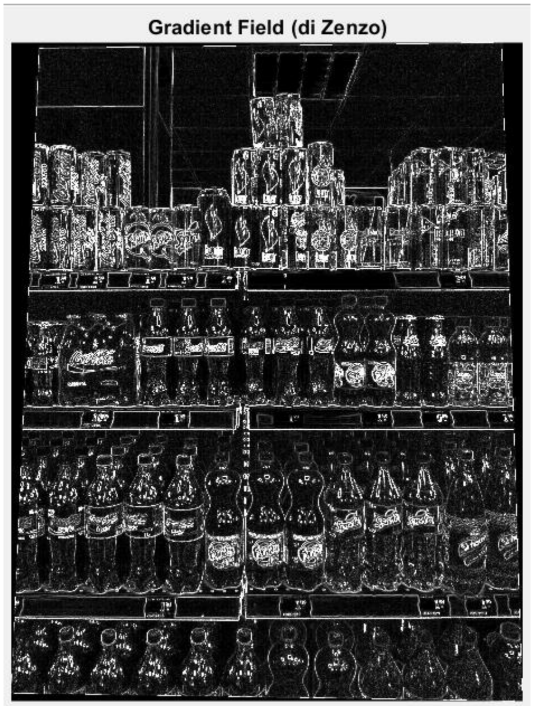


First step of my method was to convert the image from RGB to HSV. Then I threshold the saturation at 120 in order to get rid of noise-like pixels near candies. Then the next step was to determine the areas where the color blue is the color that is perceivable dominantly by the eye (where the angle is between 180 and 260) . Then I used that template (or map) to set the saturation on the areas where blue is not the dominant to 0. With this way, those areas became invisible in saturation sense. Then I converted that map (1-map) and multiplied with 255 to add it to intensity plane. It was because I wanted to set those areas to white color. After the addition converted the resulting image from HSV to RGB and obtained the image on the above left. Final step was to convert this image to binary.

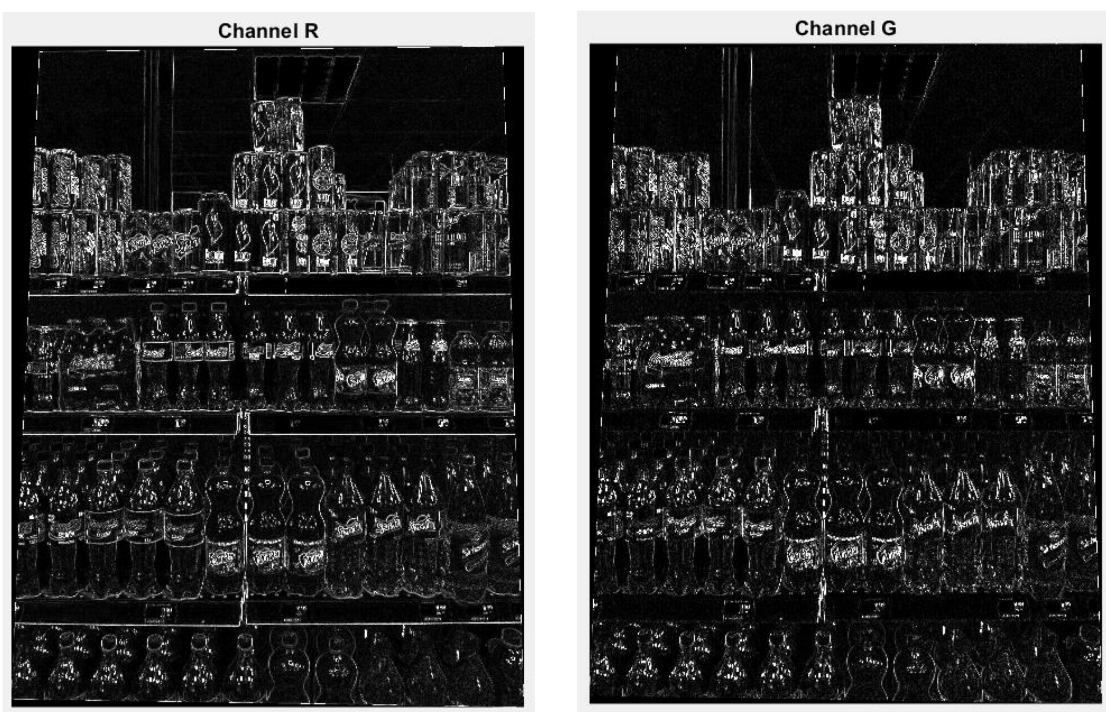
Color edges

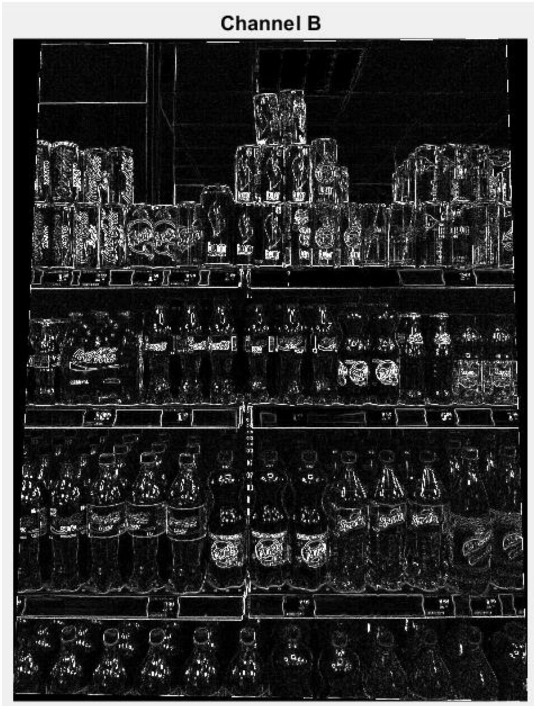
Consider the Shelves image.

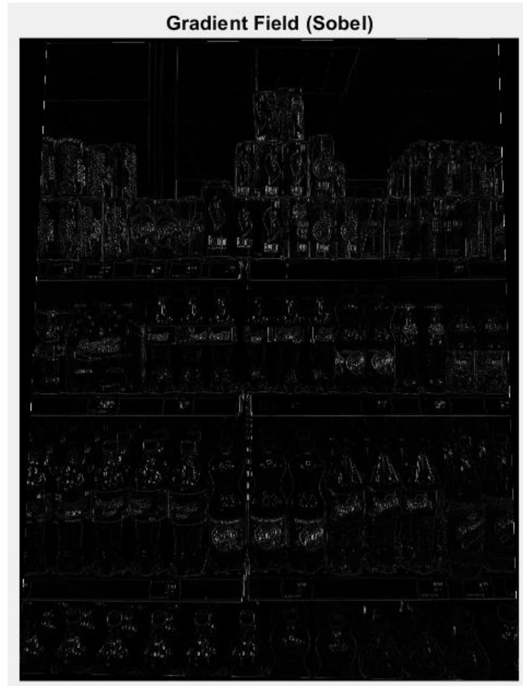
a)



b)







c)