ECE 637 Lab 4 - Pointwise Operations and Gamma

Yifan Fei

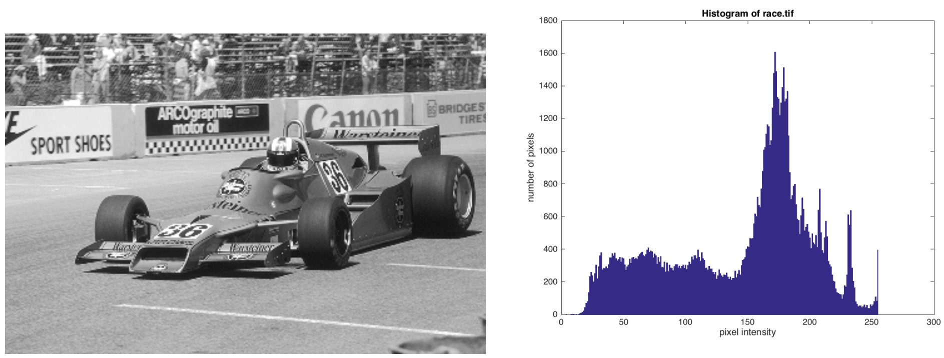
Electrical Engineering
Purdue University

Instructor: Prof. Charles A. Bouman

Spring 2018

Section 1: Histogram of an Image

1: image 'race'

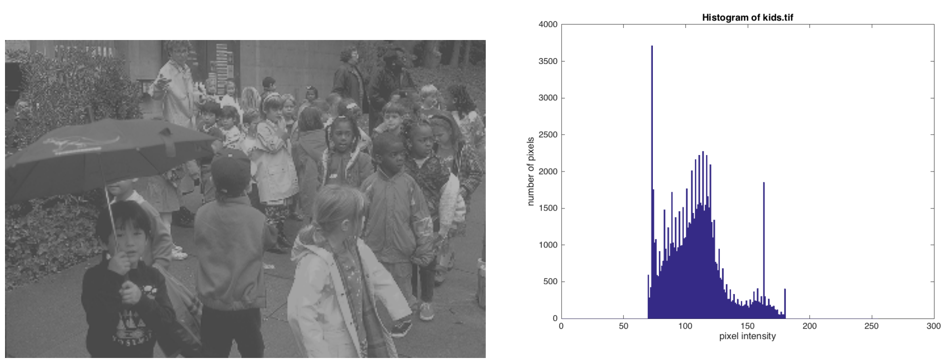


(a) Original image 'race'

(b) Histogram of image 'race'

Figure 1: image 'race'

2: image 'kids'



(a) Original image 'kids'

(b) Histogram of image 'kids'

Figure 2: image 'kids'

Section2: Histogram Equalization

1:function equalize.m

```
% Input Img I
% Output Z
function Z = equalize(I)
n = 256;

[r, c]    = size (I);
I         = mat2gray (I);
[X, map] = gray2ind (I, n);
```

```
% converts the grayscale image I to an indexed image X
[nn, xx] = imhist (I, n);
% returns the histogram counts in nn and the bin locations in xx
Icdf      = 1 / prod (size (I)) * cumsum (nn);
Z         = reshape (Icdf(X + 1), r, c);
```

2:a labeled plot of $\hat{F_x}(i)$

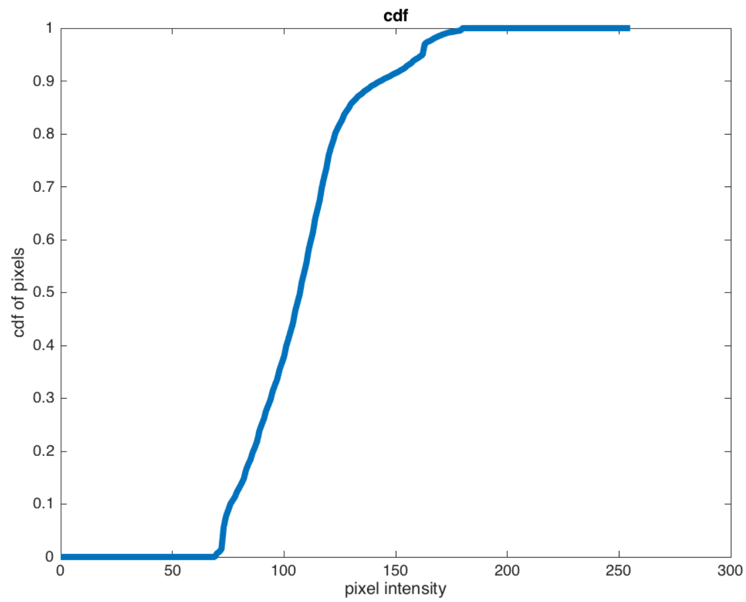


Figure 3: a labeled plot of $\hat{F_x}(i)$

3: a labeled plot of the equalized image’s histogram

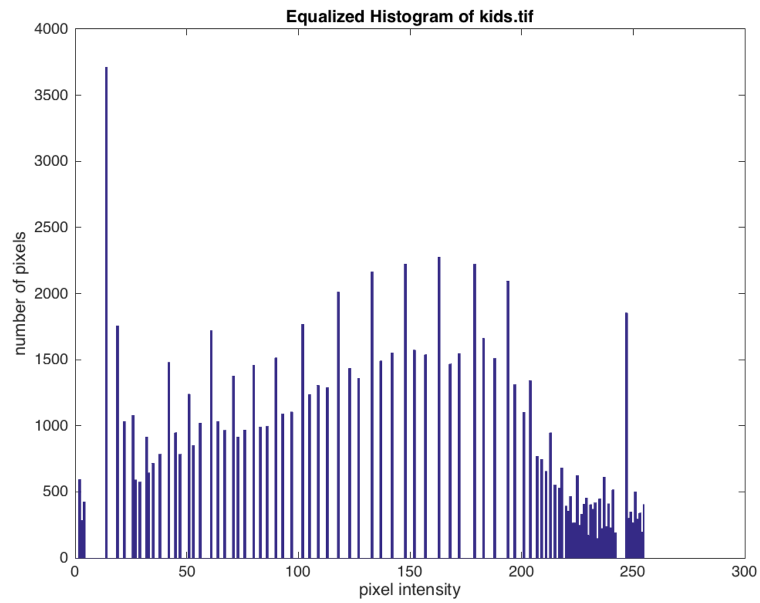


Figure 4: a labeled plot of the of the equalized image’s histogram

4:the equalized image

Figure 5: the equalized image

Section 3:Contrast Stretching

1: code for stretch

```
function output = stretch(input,T1,T2)

if (input <= T1)
    output = 0;
elseif (input>=T2)
    output = 255;
else
    output = round(255/(T2-T1)*(input-T1));
end
```

2:image

In the transformed image, I set T1 as the minx,which is 70, and T2 as the maxx, which is 180, and expand it into the whole range.

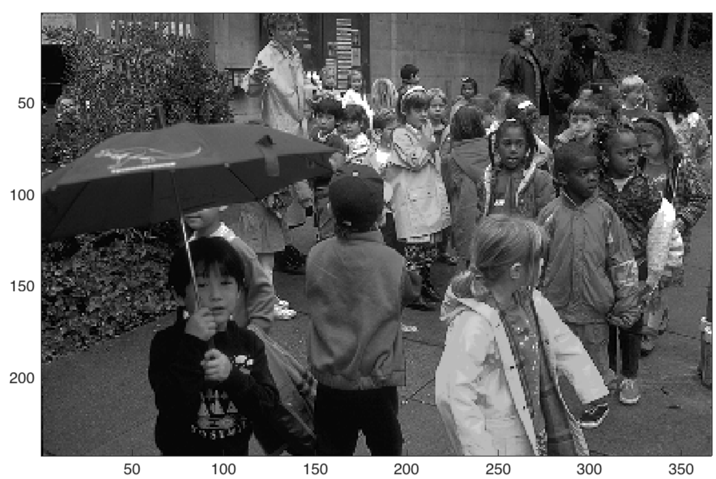


Figure 6: the transformed image

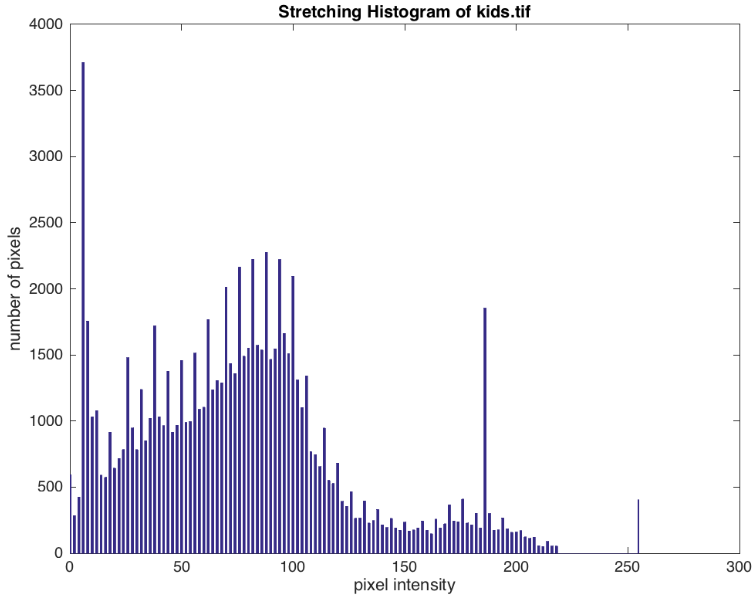


Figure 7: the histogram of the transformed image

4:Gamma

2:Determining the Gamma of Your Computer Monitor

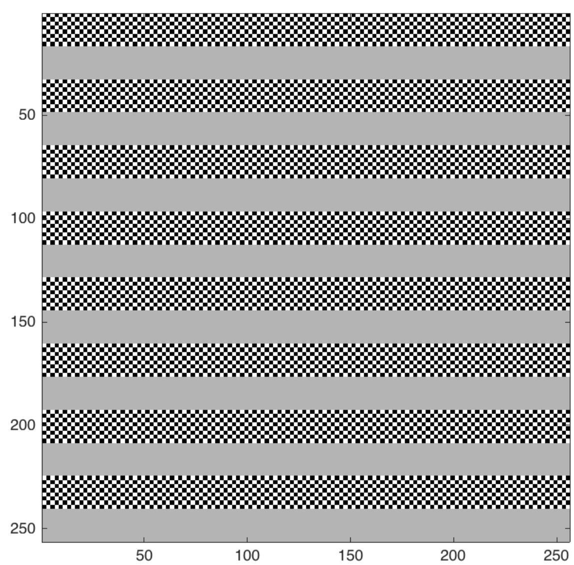


Figure 8: Array Pattern for Determining Gamma, g = 180

3:a derivation of the expression which relates the matching gray level to the value of γ

The perceived intensity of the checkerboard is given by:

$$I_c = I_{255}/2$$

Assuming a standard gamma model for the behavior of the monitor, the gray level g produces a perceived intensity of:

$$I_g = I_{255}(\frac{g}{255})^{\gamma}$$

We can then calculate the value of γ for the monitor by determining the gray level g which makes $I_g = I_c$:

$$1/2 = (\frac{g}{255})^{\gamma}$$
$$\gamma = 1/2ln(255/g)$$

4: the values of the measured gray level and the measured γ

For the test stated before:

$$g = 180$$

So:

$$\gamma = ln(0.5)/ln(180/255)$$
$$\gamma = 1.9900$$

4.3: Gamma Correction

the original and corrected images

Figure 9: original image of 'linear'

Figure 10: corrected image of 'linear' with $\gamma = 1.99$

the formula you used to transform the original image

then the correction process consists of applying the inverse of equation. The original image is with g as the parameter. The transformation shows that:

$$y = 255(\frac{x}{255})^{\gamma}$$

The inverse transformation is thus:

$$x = 255(\frac{y}{255})^{1/\gamma}$$

Use $\gamma = 1.9900$, we get

$$x = 255(\frac{y}{255})^{1/1.99}$$

corrected image for gamma15.tif

Figure 11: original image of 'gamma15'

Figure 12: corrected image of 'gamma15' with $\gamma = 1.99$

the formula you used to transform gamma15.tif

$$x_{linear} = 255(\frac{x}{255})^{\gamma_1}$$

$$x_{corrected} = 255(\frac{x_{linear}}{255})^{1/\gamma_2}$$

where $\gamma_1 = 1.5, \gamma_2 = 1.99$

So

$$x_{corrected} = 255(\frac{x}{255})^{\frac{\gamma_1}{\gamma_2}}$$

$$x_{corrected} = 255(\frac{x}{255})^{\frac{1.5}{1.99}}$$

Conclusion

In this laboratory, various types of point operations on images was discussed and applied. In Section 1, we learn histogram and in section 2, we know how to do histogram equalization. In section 3, we learn how to do Contrast Stretching and expand the picture. In section 4, we learn the nonlinear behavior and gamma correction.

Attachments: code

Code for Q1

```
clear all;
clc;

figure;
x1=imread('race.tif');
hist(x1(:),[0:255]);
xlabel('pixel intensity');
ylabel('number of pixels');
title('Histogram of race.tif')

figure;
x2=imread('kids.tif');
hist(x2(:),[0:255]);
xlabel('pixel intensity');
ylabel('number of pixels');
title('Histogram of kids.tif')
```

Code for Q2

```
clear all;
clc;

L = 256;
x=imread('kids.tif');
h = hist(x(:),[0:L-1]);

Output=h/numel(x);
%Calculate the Cumlative sum
hSum=cumsum(Output);
figure;
plot(0:L-1,hSum,'LineWidth',4)
```

```
xlabel('pixel intensity');
ylabel('cdf of pixels');
title('cdf')

%Perform the transformation S=T(R) where S and R in the range [0 1]
Z=hSum(x+1);
%Convert the image into uint8
Z=uint8(Z*(L-1));
figure,imshow(Z);

figure(10);
image(Z+1);
axis('image');
graymap = [0:255; 0:255; 0:255]'/255;
colormap(graymap);


figure;
hist(Z(:),[0:255]);
xlabel('pixel intensity');
ylabel('number of pixels');
title('Equalized Histogram of kids.tif')
```

Code for Q3

```
clear all;
clc;

L = 256;
x=imread('kids.tif');
xmin = min(x(:))
xmax = max(x(:))
for i = 1:size(x,1)
    for j = 1:size(x,2)
        y(i,j) = strecth(x(i,j),xmin,xmax);
    end
end

figure(10);
image(y+1);
axis('image');
graymap = [0:255; 0:255; 0:255]'/255;
colormap(graymap);

figure(11);
hist(y(:),0:255);
xlabel('pixel intensity');
ylabel('number of pixels');
title('Stretching Histogram of kids.tif')
```

Code for Q4

```
gray_level = 180;
x_gray = gray_level*ones(16,256);
```

```
unit = [255 255 0 0;
    255 255 0 0;
    0 0 255 255
    0 0 255 255];
x_check = repmat(unit,4,4*16);
x_unit = [x_check;x_gray];
x = repmat(x_unit,8,1);
figure(12);
image(x+1);
axis('image');
graymap = [0:255; 0:255; 0:255]'/255;
colormap(graymap);


clear all;
clc;

figure;
x=imread('linear.tif');
x = double(x);
figure(14);
image(x+1);
axis('image');
graymap = [0:255; 0:255; 0:255]'/255;
colormap(graymap);


gamma = 1.99;
for i = 1:size(x,1)
    for j = 1:size(x,2)
%           y(i,j) = 255*(x(i,j)/255)^gamma;
%           y(i,j) = 255*exp(log(x(i,j)/255)/gamma);
        y(i,j) = 255*(x(i,j)/255)^(1/gamma);
    end
end
figure(15);
image(y+1);
axis('image');
graymap = [0:255; 0:255; 0:255]'/255;
colormap(graymap);


%% gamma 1.5
gamma = 1.5;
x15=imread('gamma15.tif');
x15 = double(x15);
figure(16);
image(x15+1);
axis('image');
graymap = [0:255; 0:255; 0:255]'/255;
colormap(graymap);

x_linear = 255*(x15./255).^(1.5);
x_corrected = 255*(x_linear./255).^(1/1.99);
```

```
figure(17);
image(x_corrected+1);
axis('image');
graymap = [0:255; 0:255; 0:255]'/255;
colormap(graymap);
```