# ECE 637 Digital Image Processing Laboratory: Pointwise Operations and Gamma

Yang WANG

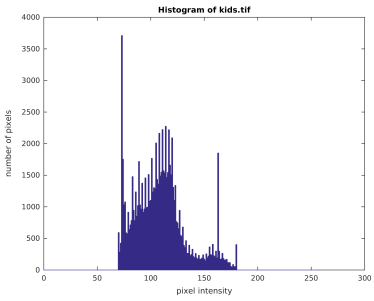
February 12, 2016

## 1 Histogram of an Image

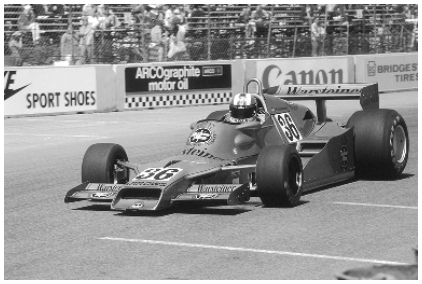
In this section, original sample images and their histograms are plotted.

### 1.1 Plot Sample Images and Its Histograms

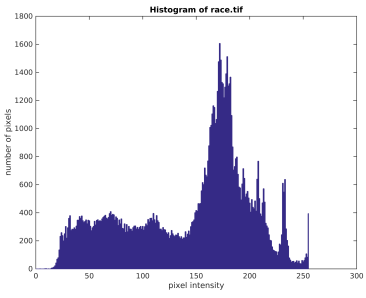
(a) Original kids.tif




(b) Histogram of kids.tif




(a) Original race.tif




(b) Histogram of race.tif

# 2 Histogram Equalization

In this section, histogram equalization is used to enhance the given sample images. A MATLAB function that implements a equalization function that approximates a cumalative distribution function is written.

## 2.1 Plot the CDF $\hat{F}_x(i)$

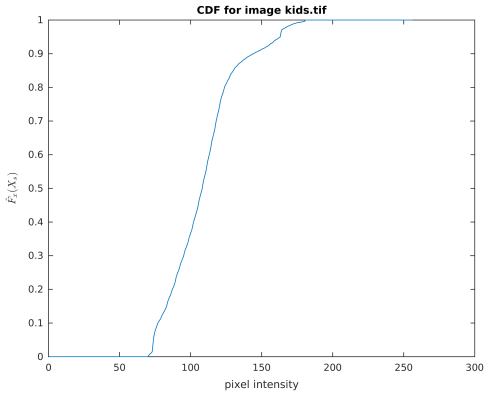



Figure 3: CDF $\hat{F}_x(i)$

## 2.2 Plot the Histogram of the Equalized Image

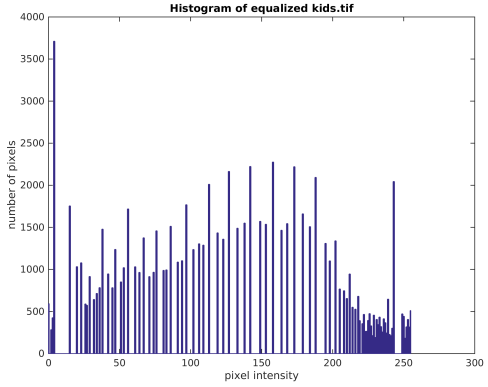



Figure 4: Histogram of Equalized kids.tif

## 2.3 Plot the Equalized Image

It is apparent that the equalized image appears to be brighter than the original image.


Figure 5: Equalized Image of kids.tif

## 2.4 Code Listing

### 2.4.1 equalize.m

```
function [Z Y] = equalize(X)
    N = hist(X(:), [0:255]);
    Y = cumsum(N)/sum(N);

    Ymin = Y(min(X(:)));
    Ymax = Y(max(X(:)));

    Z = uint8(255*((Y(X) - Ymin)/(Ymax - Ymin)));
end
```

### 2.4.2 sec1.m

```
% Section 1
close all;
clear all;

[raceimg] = imread('race.tif');
figure(1);
image(raceimg+1);
axis('image');
```

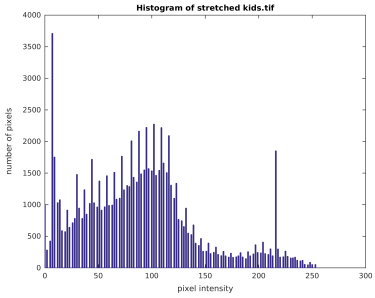
```
graymap = [0:255;0:255;0:255]'/255;
colormap(graymap);

[kidimg] = imread('kids.tif');
figure(2);
image(kidimg+1);
axis('image');
colormap(graymap);

figure(3);
hist(raceimg(:),[0:255]);
xlabel('pixel intensity');
ylabel('number of pixels');
title('Histogram of race.tif');
print('-dpng', '-r300', '../report/race_hist.png');
figure(4);
hist(kidimg(:),[0:255]);
xlabel('pixel intensity');
ylabel('number of pixels');
title('Histogram of kids.tif');
print('-dpng', '-r300', '../report/kids_hist.png');
```

# 3 Contrast Stretching

In this section, another useful image enhancement technique called contrast stretching is explored. The main idea is to define two thresholds, $T1$ and $T2$; $T1$ has a value relatively close to 0 and $T2$ is relatively close to 255. Anything less that $T1$ will be set to 0 and anything greater than $T2$ will be set to 255. Pixel values in between these two values will keep the same.

## 3.1 Plot the Transformed Image and its Histogram


(a) "Stretched" kids.tif




(b) Histogram of "stretched" kids.tif

## 3.2 Code Listing

### 3.2.1 stretch.m

```
function [output] = stretch(input, T1, T2)
    output = zeros(size(input));

    index = find(input > T1 & input < T2);
    output(index) = (input(index) - T1)*(255/(T2-T1));
    output(input > T2) = 255;

    output = uint8(output);
```

### 3.2.2 sec3.m

```
% Section 3
close all;
clear all;

img = imread('kids.tif');
strImg = stretch(img, 70, 180);
figure(1);
image(strImg+1);
axis('image');
graymap = [0:255;0:255;0:255]'/255;
colormap(graymap);
print('-dpng', '-r300', '../report/kids_str.png');
figure(2);
hist(strImg(:),[0:255]);
xlabel('pixel intensity');
ylabel('number of pixels');
title('Histogram of stretched kids.tif');
print('-dpng', '-r300', '../report/kids_str_hist.png');
```

# 4 Gamma ($\gamma$)

In this section, the topic of gamma, gamma correction and its relation with gray level is discussed.

## 4.1 Setting the Black Level and Picture of Your Monitor

Values are set. Nothing due for report.

## 4.2 Determining the Gamma of Your Computer

### 4.2.1 Plot the Checkboard to the Matching Gray Level

It is determined from my screen that with a gray level of 195, it produces the best intensity match between the stripes.

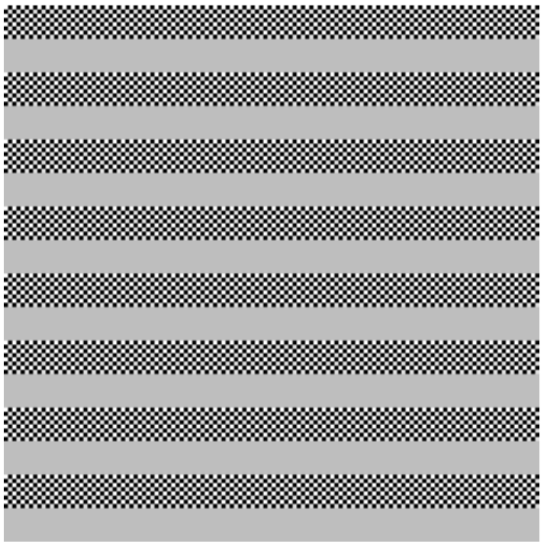


Figure 7: Checkerboard Image with gray level 195

### 4.2.2 Derive an Expression that Relates the Matching Gray Level to the Value of $\gamma$

We are given:

$$I_c = \frac{I_{255}}{2}$$

and:

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

For matching gray level, we have:

$$I_c = I_g$$
$$\frac{I_{255}}{2} = I_{255}(\frac{g}{255})^\gamma$$

Hence,

$$\gamma = -\frac{\log 2}{\log \frac{g}{255}} \tag{1}$$

### 4.2.3 Calculate the Value of Gray Level and $\gamma$

Using equation (1), $g = 195$, we have:

$$\begin{aligned}\gamma &= -\frac{\log 2}{\log \frac{195}{255}} \\ &= 2.583\end{aligned}$$

## 4.3 Gamma Correction

### 4.3.1 Plot the Original and the Corrected Images

(a) Original linear.tif

(b) Corrected linear.tif with $\gamma = 2.583$

### 4.3.2 Derive the Formula for Transforming the Image

We have:

$$\begin{aligned}y &= 255(\frac{x}{255})^{\gamma} \\ \frac{y}{255} &= (\frac{x}{255})^{\gamma} \\ \frac{x}{255} &= (\frac{y}{255})^{\gamma}\end{aligned}$$

Hence,

$$x = 255(\frac{y}{255})^{\gamma} \tag{2}$$

## 4.4 Gamma Correction on Gamma Corrected Image

### 4.4.1 Plot the Original and the Corrected Images

(a) Original gamma.tif

(b) Corrected gamma.tif with $\gamma=\frac{1.5}{2.583}$

### 4.4.2 Derive the Formula for Transforming the Image

Let's say we already have a image with gamma correction $\gamma_1$:

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

And we want to do another gamma correction on top of that with $\gamma_2$,

$$z = 255(\frac{y}{255})^{\gamma_2}$$

Then the result image will be:

$$z = 255(\frac{x}{255})^{\frac{\gamma_1}{\gamma_2}} \tag{3}$$

Hence, for given $\gamma_1 = 1.5$, the resulting $\gamma$ will be:

$$\gamma = \frac{1.5}{2.583}$$

## 4.5 Code Listing

### 4.5.1 sec3.m

```
%Section 4
linearImg = imread('linear.tif');
corLinearImg = 255*(double(linearImg)/255).^(double(1/2.583));
figure(1);
image(uint8(corLinearImg)+1);
axis('image');
graymap = [0:255;0:255;0:255]'/255;
colormap(graymap);
print('-dpng', '-r300', '../report/linear_gamma.png');

gammaImg = imread('gamma15.tif');
corGammaImg = 255*(double(gammaImg)/255).^(double(1.5/2.583));
figure(2);
image(uint8(corGammaImg)+1);
axis('image');
colormap(graymap);
print('-dpng', '-r300', '../report/gamma15_gamma.png');
```