PURDUE UNIVERSITY

ECE 637 Laboratory Exercise 4
Pointwise Operations and Gamma

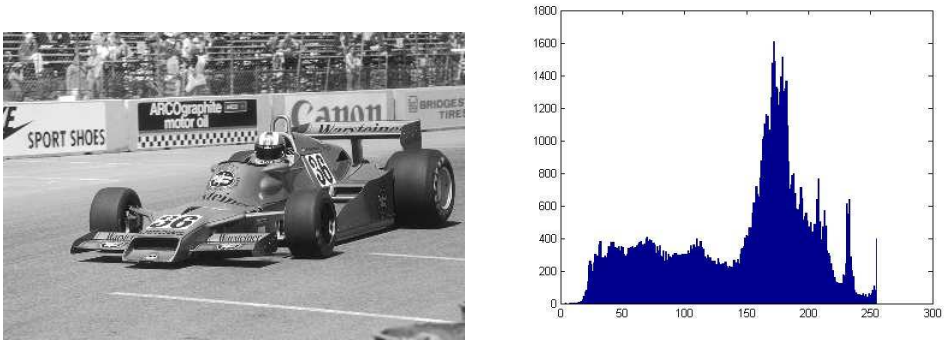
Tong Shen

February 7, 2017

1 HISTOGRAM OF AN IMAGE

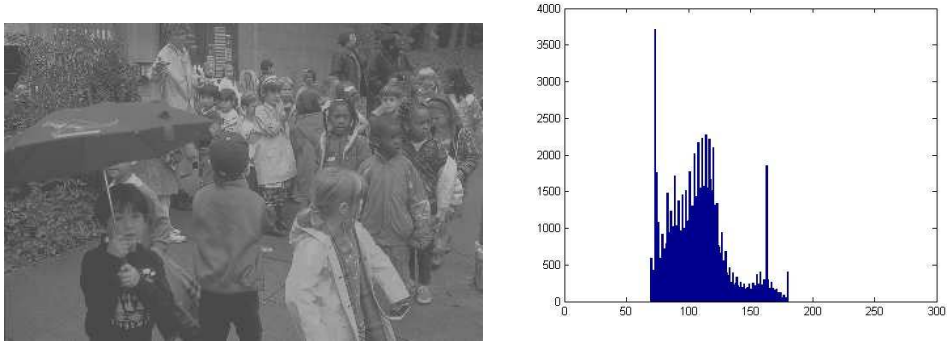
The histogram of a digital image shows the distribution of the of its pixel intensities. Also, an image enhancement method is introduced based on this.

1.1 HISTOGRAM OF AN IMAGE



(a) *race.tif* (b) Histogram of *race.tif*

Figure 1.1: *race.tif* and Its Histogram



(a) *race.tif* (b) Histogram of *kids.tif*

Figure 1.2: *kids.tif* and Its Histogram

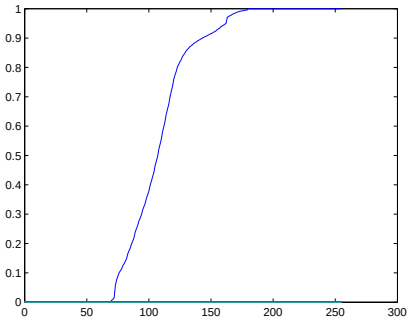
2 HISTOGRAM EQUALIZATION

2.1 RESULT DISPLAY

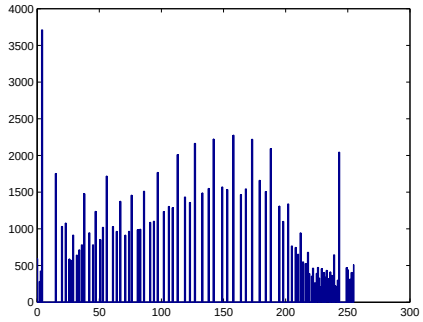
Histogram is a common image enhancement technique. This center principle is to spread the pixel value over the full range of the pixel values. We will increase the contradiction of the image in this way.

(a) *kids.tif*

(b) Histogram Result of *kids.tif*



(c) Plot of $\hat{F}_X(i)$



(d) Histogram of the Equalized image

Figure 2.1: Histogram Equalization Process of *kids.tif*

From the image and histogram, we can easily see that the pixel values are spread more uniformly over the full range.

2.2 Code Listing

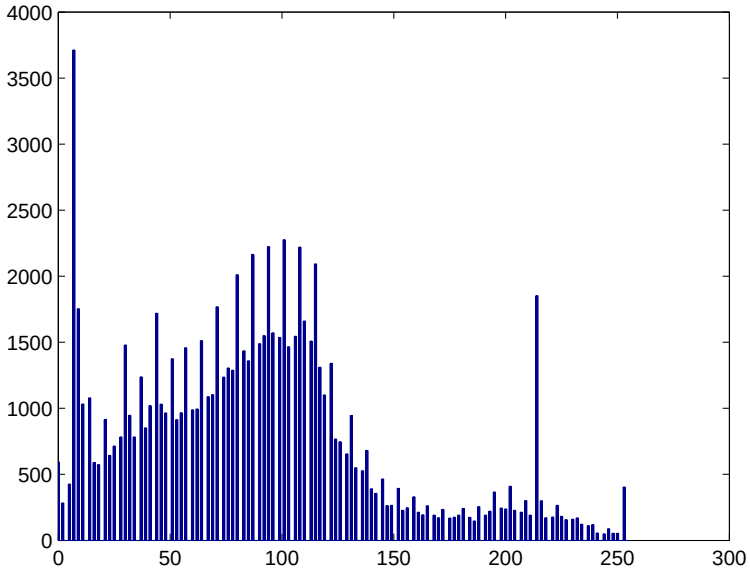
```
1
2  function Z = equalize(x)
3  histogram = hist(x(:),0:255);
4  sum_h = sum(histogram);
5  F_hist = zeros(256);
6  for i = 1:256
7  temp = 0;
8  for j = 1:i
9  temp = temp + histogram(j);
10 F_hist(i) = temp/sum_h;
11 end
12 end
13 Y_max = F_hist(max(x(:)));
14 Y_min = F_hist(min(x(:)));
15 Z = uint8(255*(F_hist(x) - Y_min)/(Y_max - Y_min));
16 end
```

3 Contrast Stretching

Another useful image enhancement technique is contrast stretching. In this method, we will simply stretch the pixel values over the full range and get the result. This is method is more convenient and sometime more efficient that histogram equalization.

3.1 Result Display

(a) Stretched Image of *kids.tif*



(b) Histogram of the Stretched image

Figure 3.1: Stretch Process of *kids.tif*

3.2 Code List

3.2.1 Stretch.m

```
1
2  function output = stretch(img,T1,T2)
3  [height,width] = size(img);
4  img = double(img);
5  output = zeros(height,width);
6  for i = 1:height
7  for j = 1:width
8  if(img(i,j) < T2 && img(i,j) > T1)
9  output(i,j) = 255*(img(i,j) - T1)/(T2 - T1);
10
11 end
12 end
13 end
14 output = uint8(output);
15 end
```

4 Gamma Transform

In this section, we will learn the properties of Gamma transform of an image and analyze the gamma transform of out monitor.

4.1 Image Corresponding to Matching Gray Level

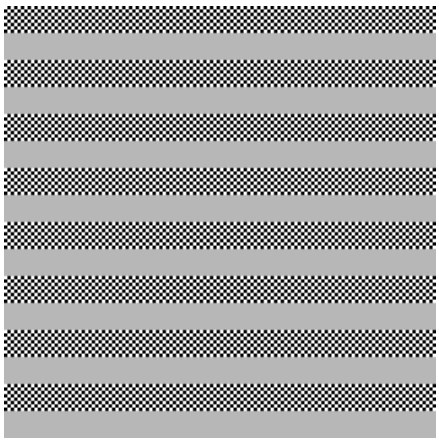


Figure 4.1: Image with the Matching Gray Level

In my experiment, the matching gray level is 183.

4.2 A Derivation of the Expression of the Value of Gamma

First of all, we know:

$$I_g = I_{255}(\frac{g}{255})^{\gamma} \tag{4.1}$$

when $I_g = 255/2$, and g = 183 from last experiment, we have :

$$ln(\frac{I_g}{255}) = \gamma ln(\frac{g}{255}) \tag{4.2}$$

Accordingly,

$$\gamma = \frac{ln(\frac{I_g}{255})}{ln(\frac{g}{255})} = 2.0892 \tag{4.3}$$

In conclusion, the measured matching gray value is 183 and the calculated Gamma is 2.0892.

5 Gamma Correction

5.1 Result Display

(a) The original image *linear.tif* (b) The Image after Gamma Correction

Figure 5.1: Comparison between the Original and Corrected Image

5.2 Formula Used in This Coorection

According to the last experiment, the Gamma transform index γ is 2.0892. So, the formula used should be as follows:

$$\begin{aligned} \gamma &= 2.0982 \\ I_{correction} &= (\frac{I}{255})^{\frac{1}{\gamma}} \end{aligned} \tag{5.1}$$

6 Gamma Correction 2

6.1 Result Display

(a) The original image *gamma15.tif*

(b) The Image after Gamma Correction

Figure 6.1: Comparison between the Original and Corrected Image

6.2 Formula Used in This Coorection

The original image already have a gamma transform with a $\gamma = 1.5$, so, the formula should be as follows:

$$\begin{aligned} I_{original} &= (\frac{I}{255})^{\frac{1}{1.5}} \\ I_{correction} &= (\frac{I}{255})^{\frac{1}{\gamma}} \end{aligned} \tag{6.1}$$

In this way, the final formula should be:

$$I_{correction} = (\frac{I_{original}}{255})^{\frac{1.5}{\gamma}} \tag{6.2}$$

7 MATLAB CODE LIST

```
1
2  clc
3  clear
4
5  img_race = imread('race.tif');
6  figure(1)
7  hist(img_race(:),0:255);
8
9  img_kids = imread('kids.tif');
10 figure(2)
11 hist(img_kids(:),0:255);
12
13 hist_before = hist(img_kids(:),0:255);
14
15 y = equalize(img_kids);
16 figure(3)
17 imshow(y)
18
19 figure(4)
20 hist(y(:),0:255);
21
22
23 sum_h = sum(hist_before);
24 F_hist_before = zeros(256);
25 for i = 1:256
26 temp = 0;
27 for j = 1:i
28 temp = temp + hist_before(j);
29 F_hist_before(i) = temp/sum_h;
30 end
31 end
32 figure(5)
33 t = 0:255;
34 plot(t,F_hist_before);
35
36 figure(6)
37 output = stretch(img_kids,70,181);
38 imshow(output)
39
40 figure(7)
41 hist(output(:),0:255);
```

```
42
43 imshow(uint8(output))
44 g = 183;
45 figure(8)
46 checkboard = checkerboard(g);
47 imshow(uint8(checkboard))
48
49 x = log(0.5)/log(g/255);  %r
50
51 img_l= im2double(imread('linear.tif'));
52
53 linear = 255*(img_l).^(1/x);
54 figure(9)
55 imshow(uint8(linear))
56
57 figure(10)
58 img_g = im2double(imread('gamma15.tif'));
59
60 linear1 = 255*(img_g).^(1.5*1/x);
61 imshow(uint8(linear1))
```