**School of Information Science and Engineering**

**2019－2020 school year**

Digital Image Processing Experiment Report

Course Name： Digital Image Processing

Title of Experiment： The Indensity Histogram and Equalization

Major and Class Class 3, Communication Engineering

Student ID 201600121116

Studeng Name 张子超

Date 2019/3/16

Table of Contents

[1． Objectives: 3](#_Toc4615292)

[2． Experiment Content: 3](#_Toc4615293)

[3． Experiment Principle: 3](#_Toc4615294)

[4． Experiment Steps: 3](#_Toc4615295)

[1. Display a histogram 3](#_Toc4615296)

[2. use function bar and stem to display a histogram separately. 4](#_Toc4615297)

[3. Display the histogram with function plot 4](#_Toc4615298)

[4. Adjust the contrast of the image 5](#_Toc4615299)

[5. Equalize the histogram and check the changes before and after equalization with imhist 6](#_Toc4615300)

[6. Conduct a brightness equalization with fuzzy algorithm. 7](#_Toc4615301)

[5． Conclusions and Experiences: 8](#_Toc4615302)

[6． Appendix（Code）: 8](#_Toc4615303)

# 1． Objectives:

Get familiar with the conception of grey-level histogram and its computing method.

Get familiar with the computing process of histogram equalization.

Compute and plot a histogram, implement the equalization of histogram.

# 2． Experiment Content:

Learning to use function imhist, histeq, bar, stem, plot, imadjust and tools for image description like title, axis, set and so on.

# 3． Experiment Principle:

The histogram of a picture represents the number of pixels, whose value are same, of every intensity scale. The more pixels biased toward the lighter side of the intensity range, the brighter the image will be. Histogram equalization could distribute intensity more evenly, which makes the image look richer.

# 4． Experiment Steps:

## 1. Display a histogram

Display the image Fig0354(a)(einstein\_orig).tif, label it EINSTEIN, and display its histogram, then adjust the parameter.



Use Matlab function imhist and then the histogram of image data shows on the screen, from which we can be told that the utmost of pixels in this image center around greyscale 90. The title can be put on the graph easily using command title('EINSTEIN');

but this is far from enough, at first the whole value of histogram can’t be shown, which is because of the limited display range. So we can use axis([0,250,0,40000]); to set axis range to 250 for x-axis and 40000 for y-axis.

## 2. use function bar and stem to display a histogram separately.

Use command [h,x]=imhist(y) to get the histogram of image, and h means the number of pixels in each grayscale while x means the greysclae. However, display the image right after this command would not meet the demand. So I sampled the histogram before I display it. The command hdisp=h(1:10:250) extract one in ten element of histogram, then use xdisp=x(1:10:250) to exract x-axis. Use stem(xdisp,hdisp,'diamondr') to draw it on the window, and I still have to set x-axis to modify the index of x-axis. At last, the figure showed the precedent problem again, that is, the axis range is too small. Add an instruction after it: axis([0,255,0,15000]). For the bar graph, use command bar(xdisp,hdisp) to display, an axis command is also required. The width of each bar could also be modified with a single parameter added behind like bar(xdisp,hdisp,0.5), and these figures below are just how they look:



## 3. Display the histogram with function plot

The requirement is the same as the last mission, but function plot draws a continuous curve. As I have obtained the sample data and its index before, I could directly use them. Type the command plot(xdisp,hdisp,'g') then a figure comes out like this:



## 4. Adjust the contrast of the image

The function imadjust maps the input intensity values in input image to new values in output image. However, values below the low-threshould input as well as above the high-threshould will be clipped. We conclude that most of the intensity value center on 90, the vast majority distributes from around 85 to 175. But casting them all isn’t a good choice. Thus, I tried dozens of times with different input and output, finally, I found a pair of parameters quite appropriate, K=imadjust(y,[70/256,170/256],[30/256,1],1.5). To low the intensity down, the gamma value should be above 1. By the way, the axis function still works here.



## 5. Equalize the histogram and check the changes before and after equalization with imhist

The imhist strengthenes the contrast of an image. Its effect is kind of like that the effect of imadjust, while relatively speaking, this one seems easier with less input.



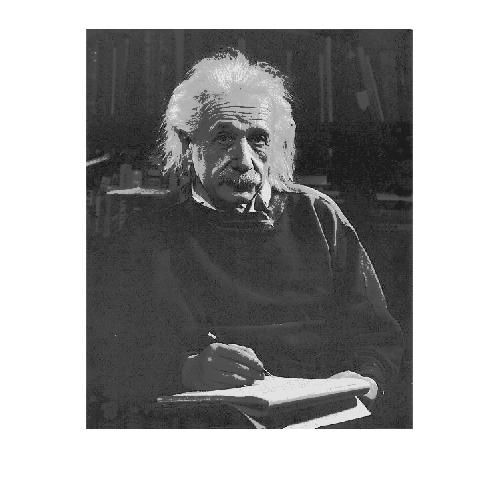
 

## 6. Conduct a brightness equalization with fuzzy algorithm.

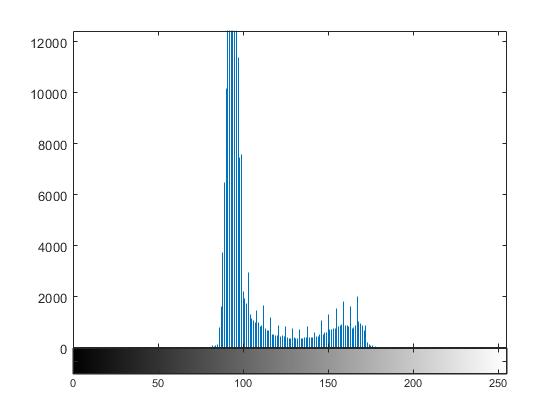
A fuzzy set provides a formalism for dealing with imprecise information. Consider that an element of one set belongs to another set, but its membership is not a “fullmember”, under this circumstance we use a membership function to denote the membership of one set to another one. Similarly, intensity enhancement can be decribed as several simple principles: if a pixel is dark, then make it darker; if a pixel is gray, then make it gray; if a pixel is bright, then make it brighter.

The already-prepared Matlab function can be called directly, though the source code can’t be editted. So we just put those required function in to current folder, and call them in the main function.

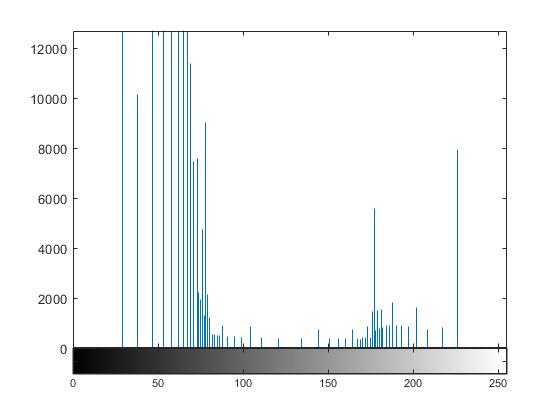
The result goes like this:



Before the intensity enhancement, the histogram of this picture is as follows:



Then after the procedure, the histogram changed



# 5． Conclusions and Experiences:

In this experiment, I learned an important thing, that is, do figure out the principles first. I had thought the histogram represents the composition of one image, it turns out that it has rigid mathematical meaning. So it took me quite a long time to understand how the imadjust can effect histogram. Besides, I’ve learned a lot of functions to avoid too narrow display range and how to set x-axis index. Also, I’ve learned that pictures with wide intensity range makes themselves not so dull, which is what makes the equaliztion so important.

# 6． Appendix（Code）:

[y,mapy]=imread('G:\desktop documents\2018大三下\数字图像处理资料For students\For students\程序与图像\图像库\Fig0354(a)(einstein\_orig).tif'); %导入图像

figure

imshow(y)%显示该图

title('EINSTEIN');%加标题

imhist(y);%显示直方图

figure

title('before the adjustment');

axis([0,250,0,40000]);%扩大显示范围

[h,x]=imhist(y);%取得直方图及其横坐标

hdisp=h(1:10:250);%抽样

xdisp=x(1:10:250);%抽样

figure

stem(xdisp,hdisp,'diamondr')%条形图显示

axis([0,255,0,15000]);

set(gca,'xtick',0:50:255)

figure

bar(xdisp,hdisp)%柱状图显示

axis([0,255,0,15000]);

set(gca,'xtick',0:50:255)

figure

bar(xdisp,hdisp,0.5)%改变柱状图的粗细

axis([0,255,0,15000]);

set(gca,'xtick',0:50:255)

figure

plot(xdisp,hdisp,'g')%使用连续曲线显示

axis([0,255,0,15000]);

set(gca,'xtick',0:50:255)

K=imadjust(y,[70/256,170/256],[30/256,1],1.5);%进行灰度调整

figure

imhist(K)%显示K的直方图，K是经过灰度调整后的图

axis([0,255,0,40000]);

figure

imshow(K)%显示经过灰度调整后的图像进行对比

title('after the adjustment');

eq=histeq(y);%灰度均衡

figure

imshow(y);%显示原图进行比对

figure

imhist(y)

title('before the equalization')

figure

imshow(eq);%显示灰度均衡后的图

figure

imhist(eq);

title('after the equalization');

axis([0,255,0,40000])

udark=@(z)1-sigmamf(z, 0.35, 0.5);

ugray=@(z)triangmf(z, 0.35, 0.5, 0.65);

ubright=@(z)sigmamf(z, 0.5, 0.65);

udarker=@(z)bellmf(z, 0.0, 0.1);

umidgray=@(z)bellmf(z, 0.4, 0.5);

ubrighter=@(z)bellmf(z, 0.8, 0.9);

rules={udark ;ugray; ubright};

outmf={udarker; umidgray; ubrighter};

F=fuzzysysfcn(rules, outmf, [0 1]);

z=linspace(0, 1, 256);

T=F(z);

[y,mapy]=imread('G:\desktop documents\2018大三下\数字图像处理资料For students\For students\程序与图像\图像库\Fig0354(a)(einstein\_orig).tif');

g=intrans(y, 'specified', T);

figure, imshow(g);

figure;

imhist(g)