

PROJECT 03-02 [Multiple Uses]

Histogram Equalization

- (a) Write a computer program for computing the histogram of an image.
- (b) Implement the histogram equalization technique discussed in Section 3.3.1.
- (c) Download Fig. 3.8(a) from the book web site and perform histogram equalization on it.

As a minimum, your report should include the original image, a plot of its histogram, a plot of the histogram-equalization transformation function, the enhanced image, and a plot of its histogram. Use this information to explain why the resulting image was enhanced as it was.

一、计算图像直方图的程序

本项目中，采用 matlab 自带 imhist 函数计算图像直方图。

二、程序原理

采用书中 3.3-7 和 3.3-8 的公式进行图像直方图均衡化。

对于离散值，我们处理其概率（直方图值）与求和来替代处理概率密度函数与积分。正如前面提到的那样，一副数字图像中 r_k 出现的概率近似为：

$$p_r(r_k) = \frac{n_k}{MN}, k = 0, 1, 2, \dots, L-1 \quad (3.3-7)$$

其中，MN 是图像中像素的总数， n_k 是灰度为 r_k 的像素个数，L 是图像中可能的灰度级的数量。与 r_k 相对的 $p_r(r_k)$ 图形称为直方图。

$$s_k = T(r_k) = (L-1) \sum_{j=0}^k p_r(r_j) = \frac{L-1}{MN} \sum_{j=0}^k n_j, k = 0, 1, 2, \dots, L-1 \quad (3.3-8)$$

在这个公式中，变换 $T(r_k)$ 称为直方图均衡或直方图线性变换。

三、程序源代码

```
1      %输入要处理的图像
2 -    I = imread('3.8(a).png');
3 -    [height,width] = size(I);
4 -    figure
5 -    subplot(221)
6 -    imshow(I)%显示原始图像
7 -    subplot(222)
8 -    imhist(I)%显示原始图像直方图

10     %进行像素灰度统计:
11 -    NumPixel = zeros(1,256);%统计各灰度数目，共256个灰度级
12 -    for i = 1:height
13 -        for j = 1: width
14 -            NumPixel(I(i,j) + 1) = NumPixel(I(i,j) + 1) + 1;%对应灰度值像素点数量增加一
15 -        end
16 -    end

18     %计算灰度分布密度
19 -    ProbPixel = zeros(1,256);
20 -    for i = 1:256
21 -        ProbPixel(i) = NumPixel(i) / (height * width * 1.0);
22 -    end

24     %计算累计直方图分布
25 -    CumuPixel = zeros(1,256);
26 -    for i = 1:256
27 -        if i == 1
28 -            CumuPixel(i) = ProbPixel(i);
29 -        else
30 -            CumuPixel(i) = CumuPixel(i - 1) + ProbPixel(i);
31 -        end
32 -    end

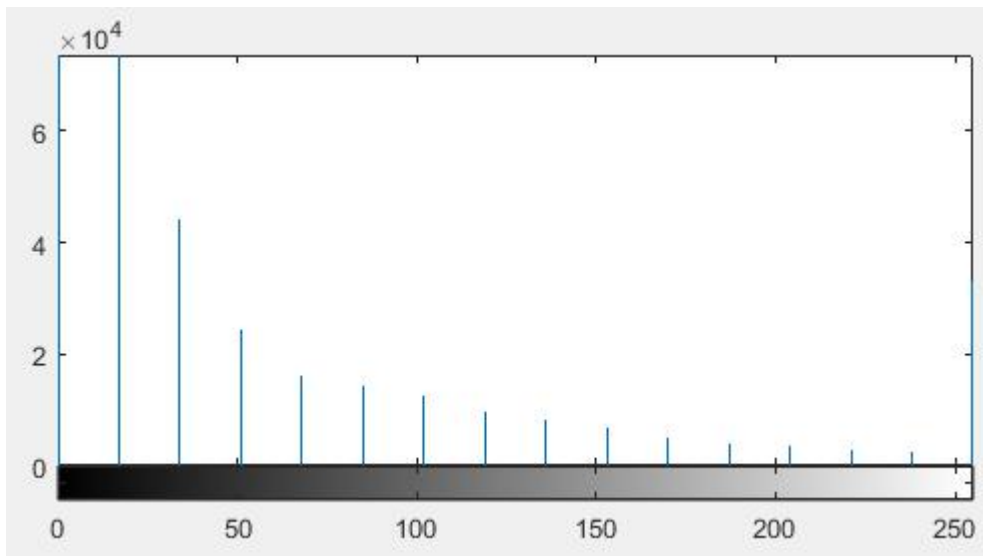
34     %累计分布取整
35 -    CumuPixel = uint8(255 .* CumuPixel);
36     %对灰度值进行均衡化
37 -    for i = 1:height
38 -        for j = 1: width
39 -            I(i,j) = CumuPixel(I(i,j)+1);
40 -        end
41 -    end

42
43 -    subplot(223)
44 -    imshow(I)%显示处理的图像
45 -    subplot(224)
46 -    imhist(I)%显示处理的图像直方图
```

四、程序运行结果



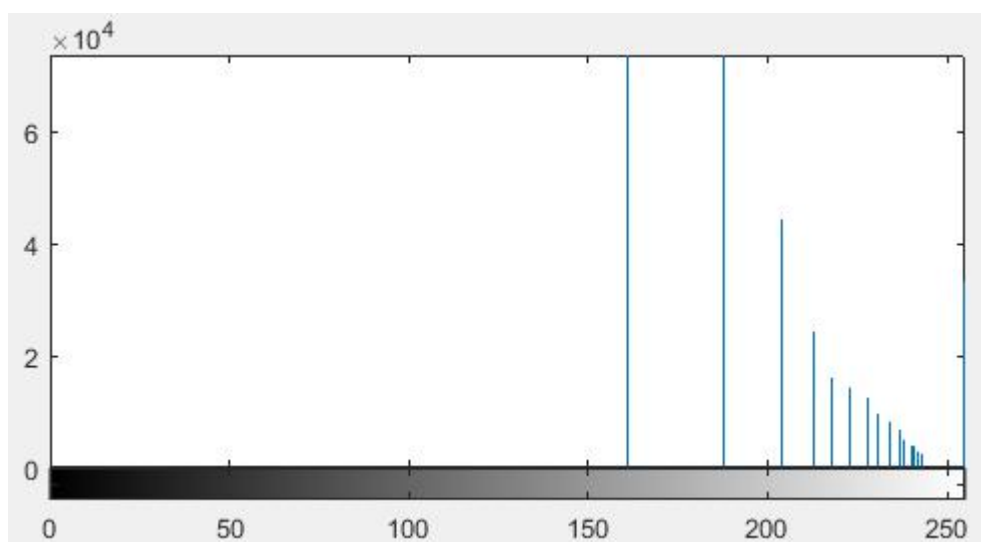
图一为均衡化之前的原图。



图二是原图的灰度直方图。



图三是均衡化之后的原图。



图四是均衡化之后的直方图。

五、结果分析

从图像中可以看出，经过直方图均衡化之后，图像中的更多的细节变得可见了，背景中的细节得到了进一步的增强，但是对比度降低，图像有一点“冲淡”的外观。

直方图均衡化处理的“中心思想”是把原始图像的灰度直方图从比较集中的某个灰度区间变成在全部灰度范围内的均匀分布。直方图均衡化就是对图像进行非线性拉伸，重新分配图像像素值，使一定灰度范围内的像素数量大致相同。直方图均衡化就是把给定图像的直方图分布改变成“均匀”分布直方图分布。

但是在本项目之中，均衡化之后直方图变得更密集，直方图的一些高峰，经处理之后变得不正常的高，所以图像对比度降低，图像外观有所冲淡。