

# 浙江大学

## 本科实验报告

直方图计算与均衡化

课程名称： 数字图像处理

姓 名：

学 院： 信息与工程学院

专 业： 信息工程

学 号：

指导老师： 李东晓

2023 年 6 月 4 日

## 一、 实验任务

本次选择的是 PROJECT-03-02 题目。

直方图均衡化

- (1) 编写一个程序计算一个图像的直方图。
- (2) 利用 3.3.1 节讨论的公式执行直方图均衡化操作。
- (3) 对一张图片实现直方图均衡化。

## 二、 算法设计

### 1. 直方图计算

令  $r_k, k = 0, 1, 2, \dots, L - 1$  表示一幅  $L$  级灰度数字图像  $f(x, y)$  的灰度，则  $f$  的非归一化直方图定义为

$$h(r_k) = n_k, k = 0, 1, 2, \dots, L - 1$$

式中， $n_k$  是  $f$  中灰度为  $r_k$  的像素的数量。 $f$  的归一化直方图定义为

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN}$$

式中， $M$  和  $N$  分别表示图像的行数和列数。

编程时，我们可以先统计出图像的灰度取值，然后遍历图像的所有像素，对不同灰度值进行计数，统计出所有灰度值对应的像素个数，画出统计频次对于灰度值的直方图，即可得到非图像的非归一化的直方图。同时可以计算出不同灰度值的频次，画出归一化的直方图。

### 2. 直方图均衡化

对于离散的灰度值，可以用概率求和代替概率密度函数与积分，得到变换后的  $s_k$  为

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p(r_j), \quad k = 0, 1, 2, \dots, L - 1$$

编程时可以利用上式进行直方图均衡化，同时对于计算出的值进行四舍五入取整作为变换后的灰度值。

## 三、 代码实现

本次实验编程语言选择的是 Matlab。

计算图像的直方图信息核心代码如下：

```
1 % 获取灰度直方图的信息 (r, 数量, 概率)
2 [Y,X] = size(img_gray);
3 grayvalue = unique(img_gray);
4 l = 256 - length(grayvalue);
5 b = zeros(l, 1);
6 grayvalue = cat(1,grayvalue,b);
7 imginfo=[];
```

```
8   for i=1:length(grayvalue)
9       [ANSy,ANSx]=find(img_gray==grayvalue(i));
10      imginfo.gray(i)=grayvalue(i);
11      imginfo.count(i)=length(ANSy);
12      imginfo.prob(i)=length(ANSy)/(Y*X);
13  end
```

其中，imginfo 类中的 gray 保存了图像的灰度值，count 保存了各个灰度值的频次统计，prob 保存了各个灰度值的频率统计。

实现直方图均衡化的核心代码如下：

```
1   % 获取量化后的灰度s，数量，概率
2   hist_eq = [];
3   hist_eq.gray = imginfo.gray;
4   hist_eq.s = zeros(1,256);
5   hist_eq.round_s = zeros(1,256);
6   hist_eq.count = zeros(1,256);
7   for k = 1:1:length(grayvalue)
8       for j = 1:1:k
9           hist_eq.s(k) = hist_eq.s(k) + imginfo.prob(j);
10          hist_eq.count(k) = hist_eq.count(k) + imginfo.count(j);
11      end
12      hist_eq.s(k) = (255) * hist_eq.s(k);
13  end
14  hist_eq.round_s = round(hist_eq.s);
```

其中，hist\_eq 类中 gray 保存了均衡化之前的灰度值，s 保存了均衡化之后的灰度值（未四舍五入），round\_s 保存了四舍五入后的 s 值，count 保存了均衡化后的灰度值的频次。

获取均衡化后的图像核心代码如下：

```
1   % 获取均衡化后的图像
2   imgHist = zeros(size(img_gray));
3   for y = 1:1:Y
4       for x = 1:1:X
5           i = img_gray(y,x);
6           imgHist(y,x) = hist_eq.round_s(i+1);
7       end
8   end
9   imgHist = uint8(imgHist);
```

其中 imgHist 为均衡化后的图像二维矩阵。

实验完整代码附录。

## 四、 实验结果

实验测试图像使用如下图像：



图 1: 测试图像

通过自己编写的计算直方图的程序得到的结果如下，同时与 Matlab 自带的计算直方图函数 `imhist()` 比较，可以看出两者几乎没有差别，也看出原始图像的灰度主要集中在灰度低的地方，灰度高的地方几乎没有分布，原图的灰度图偏暗，对比度低。

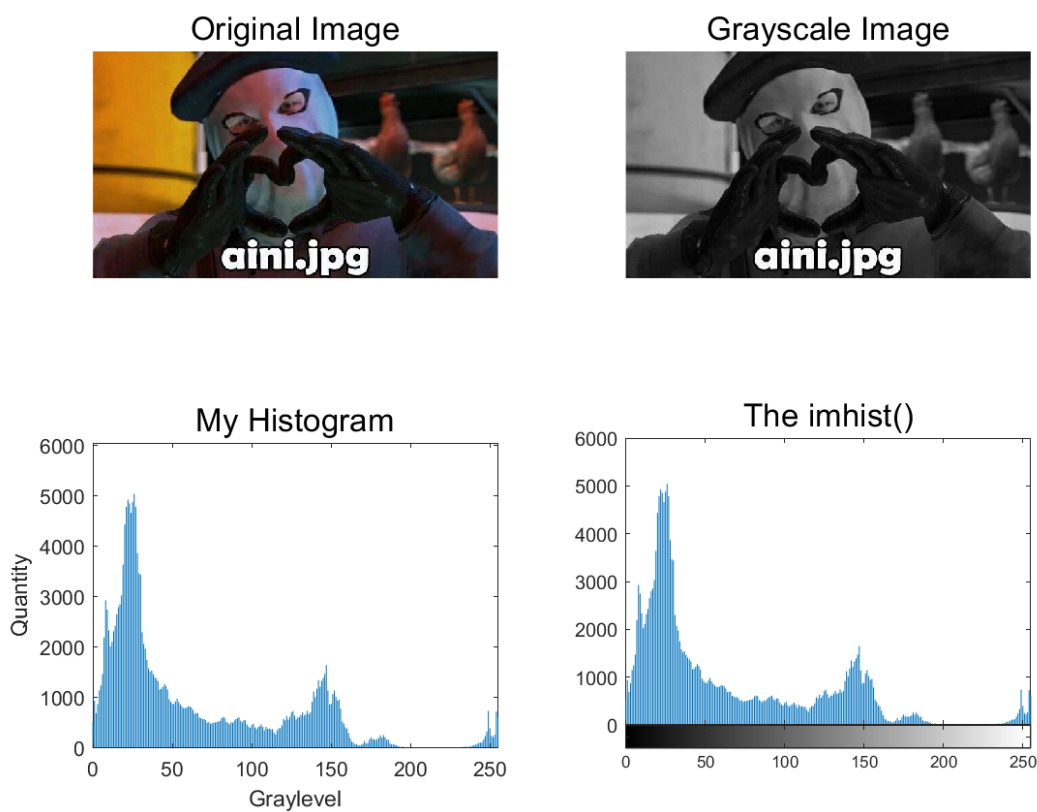


图 2: 直方图计算结果

直方图均衡化后得到的直方图结果以及  $s-r$  函数如下，可以看出经过直方图均衡化后，原本灰度低的地方分布变得分散，而灰度高的地方有了较为均匀的灰度分布。

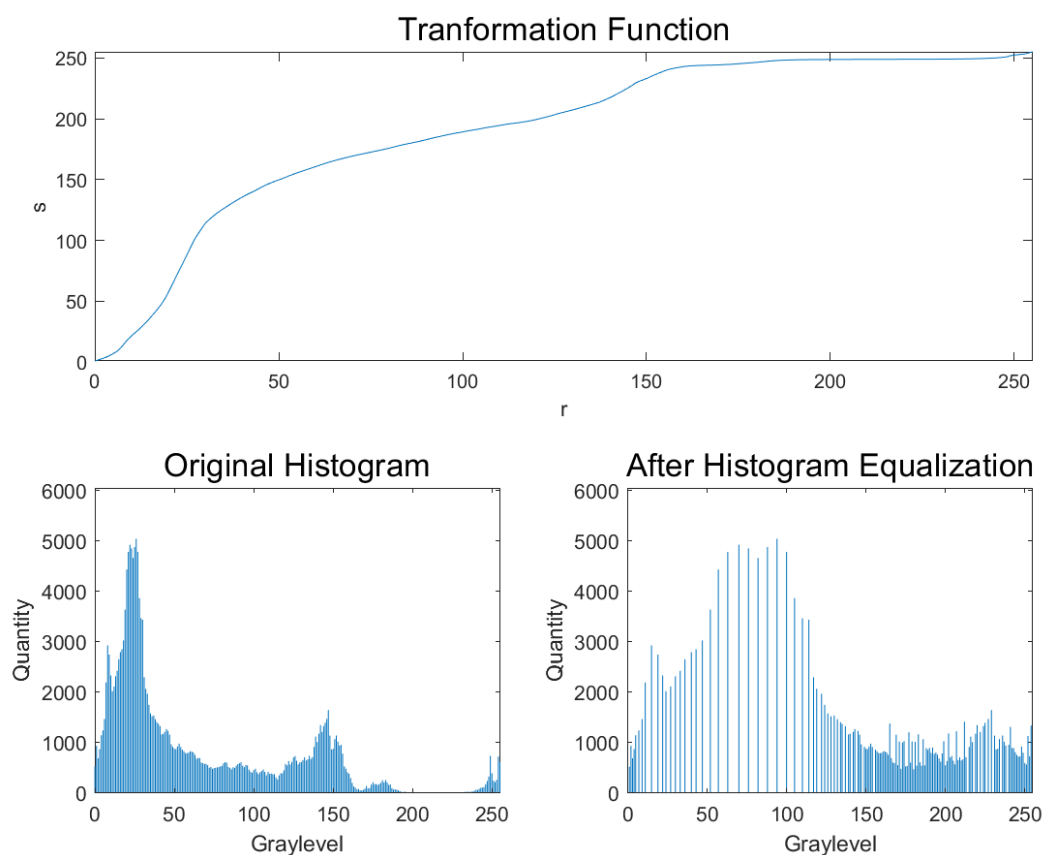


图 3: 均衡化后直方图结果

对比均衡化之前的灰度图像可以看出，均衡化之前图像偏暗，对比度较低；均衡化之后图像整体变亮，对比度提高。

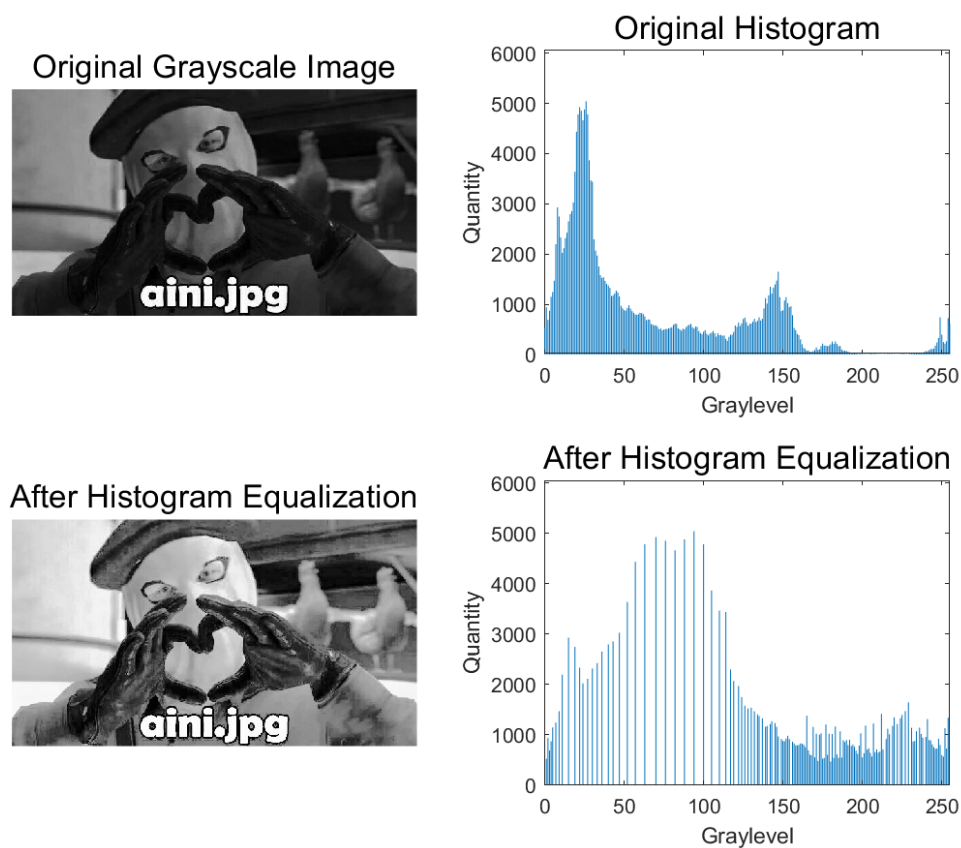


图 4: 均衡化前后灰度图对比

最后将自己的直方图均衡化结果与 Matlab 自带的均衡化函数 `histeq()` 进行对比，从直方图可以看出自己的均衡化效果还不及 Matlab 自带的函数，仍有改进空间，结合已经学内容，可以通过直方图规范化进行一定的改进。

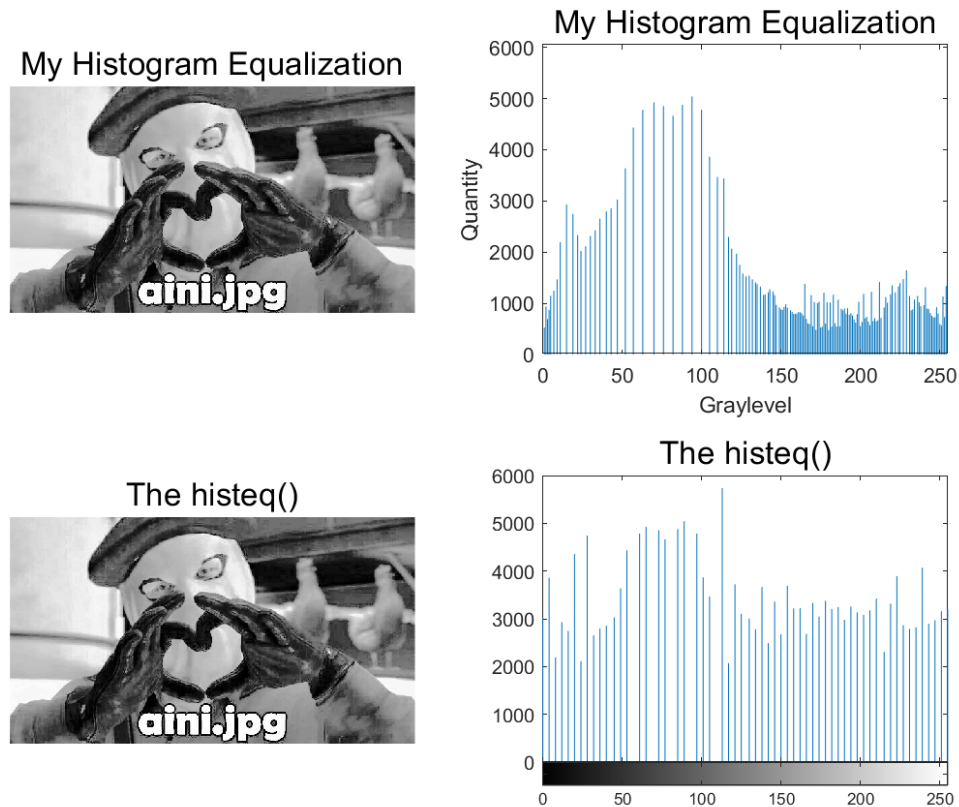


图 5: 直方图计算结果

## 五、 总结

本次实验主要是通过 Matlab 编程语言实现了课程中所讲的灰度图的直方图计算绘制以及直方图均衡化。

在实验的过程中，我对于课程中所学的直方图以及均衡化的理论有了更加深刻的理解，同时也意识到课程中所讲的内容与实际应用还有一定的差别，需要更多的理解与学习。

## 六、 附录：实验完整代码

```

1 close all;
2 clear;
3 clc;
4 img = imread('aini.jpg');
5 img_gray = rgb2gray(img);
6
7 % 获取灰度直方图的信息 (r, 数量, 概率)
8 [Y,X] = size(img_gray);
9 grayvalue = unique(img_gray);
10 l = 256 - length(grayvalue);
11 b = zeros(l, 1);

```

```
12 grayvalue = cat(1,grayvalue,b);
13 imginfo=[];
14 for i=1:length(grayvalue)
15     [ANSy,ANSx]=find(img_gray==grayvalue(i));
16     imginfo.gray(i)=grayvalue(i);
17     imginfo.count(i)=length(ANSy);
18     imginfo.prob(i)=length(ANSy)/(Y*X);
19 end
20
21 % 获取量化后的灰度s, 数量, 概率
22 hist_eq = [];
23 hist_eq.gray = imginfo.gray;
24 hist_eq.s = zeros(1,256);
25 hist_eq.round_s = zeros(1,256);
26 hist_eq.count = zeros(1,256);
27 for k = 1:1:length(grayvalue)
28     for j = 1:1:k
29         hist_eq.s(k) = hist_eq.s(k) + imginfo.prob(j);
30         hist_eq.count(k) = hist_eq.count(k) + imginfo.count(j);
31     end
32     hist_eq.s(k) = (255) * hist_eq.s(k);
33 end
34 hist_eq.round_s = round(hist_eq.s);
35
36 % 获取均衡化后的图像
37 imgHist = zeros(size(img_gray));
38 for y = 1:1:Y
39     for x = 1:1:X
40         i = img_gray(y,x);
41         imgHist(y,x) = hist_eq.round_s(i+1);
42     end
43 end
44 imgHist = uint8(imgHist);
45
46 % 计算均衡化后的直方图信息
47 [Y1,X1] = size(imgHist);
48 hist_eq_grayvalue = unique(imgHist);
49 hist_eq.count_new = zeros(1, 256);
50 hist_eq.prob = zeros(1, 256);
51 for i=1:length(grayvalue)
52     [ANSy,ANSx]=find(imgHist==grayvalue(i));
53     % imginfo.position(i) = [ANSy, ANSy];
54     hist_eq.count_new(i)=length(ANSy);
55     hist_eq.prob(i)=length(ANSy)/(Y1*X1);
56 end
57
58 % 输出图像
59 f1 = figure(1);
60 set(f1,'position',[0 0 800 600]);
61 subplot(2,2,1);
```



```
62     imshow(img);
63     title('Original Image', fontsize = 16);
64 subplot(2,2,2);
65     imshow(img_gray);
66     title('Grayscale Image', fontsize = 16)
67 subplot(2,2,3);
68     stem(imginfo.gray,imginfo.count,'Marker','none');
69     xlabel('Graylevel');
70     ylabel('Quantity');
71     axis([0 255 0 1.2*max(imginfo.count)]);
72     title('My Histogram', fontsize = 16);
73 subplot(2,2,4);
74     imhist(img_gray);
75     ylim('auto');
76     title('The imhist()', fontsize = 16)
77     saveas(f1, 'hw1-1.png');
78 f2 = figure(2);
79     set(f2,'position',[0 0 800 600]);
80 subplot(2,2,[1,2]);
81     plot(hist_eq.gray, hist_eq.s);
82     xlabel('r');
83     ylabel('s');
84     axis([0 255 0 255]);
85     title('Tranformation Function', fontsize = 16)
86 subplot(2,2,3);
87     stem(imginfo.gray,imginfo.count,'Marker','none');
88     xlabel('Graylevel');
89     ylabel('Quantity');
90     axis([0 255 0 1.2*max(imginfo.count)]);
91     title('Original Histogram', fontsize = 16);
92 subplot(2,2,4);
93     stem(hist_eq.gray,hist_eq.count_new,'Marker','none');
94     xlabel('Graylevel');
95     ylabel('Quantity');
96     axis([0 255 0 1.2*max(hist_eq.count_new)]);
97     title('After Histogram Equalization', fontsize = 16);
98     saveas(f2, 'hw1-2.png');
99 f3 = figure(3);
100 set(f3,'position',[0 0 800 600]);
101 subplot(2,2,1);
102     imshow(img_gray);
103     title('Original Grayscale Image', fontsize = 16);
104 subplot(2,2,2);
105     stem(imginfo.gray,imginfo.count,'Marker','none');
106     xlabel('Graylevel');
107     ylabel('Quantity');
108     axis([0 255 0 1.2*max(imginfo.count)]);
109     title('Original Histogram', fontsize = 16);
110 subplot(2,2,3);
111     imshow(imgHist);
```

```
112     title('After Histogram Equalization', fontsize = 16);
113     subplot(2,2,4);
114     stem(hist_eq.gray,hist_eq.count_new,'Marker', 'none');
115     xlabel('Graylevel');
116     ylabel('Quantity');
117     axis([0 255 0 1.2*max(hist_eq.count_new)]);
118     title('After Histogram Equalization', fontsize = 16);
119     saveas(f3, 'hw1-3.png');
120 f4 = figure(4);
121 set(f4,'position',[0 0 800 600]);
122 subplot(2,2,1);
123 imshow(imgHist);
124 title('My Histogram Equalization', fontsize = 16);
125 subplot(2,2,2);
126 stem(hist_eq.gray,hist_eq.count_new,'Marker', 'none');
127 xlabel('Graylevel');
128 ylabel('Quantity');
129 axis([0 255 0 1.2*max(hist_eq.count_new)]);
130 title('My Histogram Equalization', fontsize = 16);
131 subplot(2,2,3);
132 I = histeq(img_gray);
133 imshow(I)
134 title('The histeq()', fontsize = 16);
135 subplot(2,2,4);
136 imhist(I);
137 ylim('auto');
138 title('The histeq()', fontsize = 16);
139 saveas(f4, 'hw1-4.png');
```