

直方图均衡实验报告

3190103683 张嘉浩

1. 实验要求

自选一张灰度图像，编程实现直方图均衡处理。

1. 基于累积分布函数实现直方图均衡部分需要自写代码；
2. 以实验报告形式提交结果，报告内容应包括自写源码、直方图均衡处理前后的图像和直方图。

2. 实验步骤

基于累积分布函数 s_k ：

灰度级为离散值的累积分布函数的数学表达式如下：

$$p_r(r_k) = \frac{n_k}{n}$$
$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j)$$

其代码实现如下：

```
# get histogram
hist, bins = np.histogram(img, 256)
# get sum
sum = 0
for i in range(0, 255):
    sum += hist[i]
# calculate s_k
s = np.zeros(256)
s[0] = hist[0]/sum
for i in range(1, 255):
    s[i] = s[i-1] + hist[i]/sum
# image remap
for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        img[i][j] = int(255 * s[img[i][j]] + 0.5)
```

3. 实验结果

处理前后的直方图如图1，图2所示：

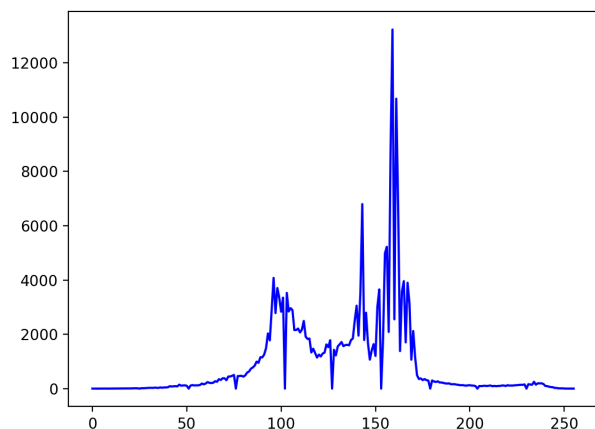


图1. 均衡前直方图

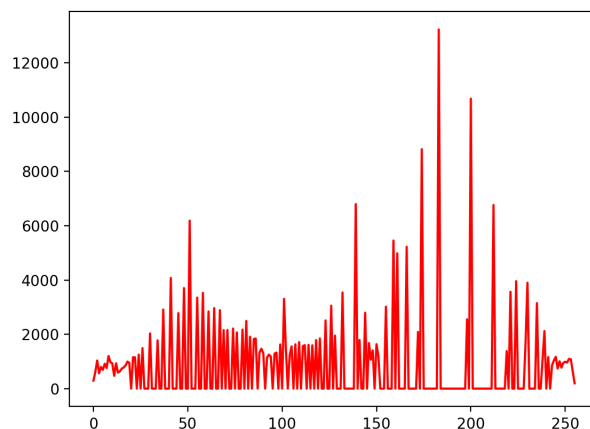


图2. 均衡后直方图

处理前后的图片如图3，图4所示：



图3. 直方图均衡前图片



图4. 直方图均衡后图片

由上面四张图片可以看到，灰度由之前的主要集中分布在直方图中间变成了在区间 $[0, 255]$ 内都有分布，分布更为均匀，同时直方图也变得离散；但从图4中可以看出，通过变换，图像损失了一些细节。

附录：源代码

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

img = cv.imread("./GrayImg.jpg", 0)

# calculate hist
hist, bins = np.histogram(img, 256)
```

```
# plt.plot(hist, 'b')
# print(sum)
s = np.zeros(256)

sum = 0

for i in range(0, 255):
    sum += hist[i]

s[0] = hist[0]/sum

for i in range(1, 255):
    s[i] = s[i-1] + hist[i]/sum
    # print(p[i])

for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        img[i][j] = int(255 * s[img[i][j]] + 0.5)
# calculate hist
hist, bins = np.histogram(img, 256)
# plot hist
plt.plot(hist, 'r')
plt.show()
# cv.imshow("img", img)
cv.waitKey()
```