# 数字图像处理作业一

16340242 吴宇祺



## 题目

1. 实现river.jpg图像的直方图均衡，将结果和调用histeq()结果作比较。
2. 将图像EightAM.png的直方图匹配为图像LENA.png的直方图，显示前后的图像并绘制LENA的直方图、直方图匹配前后eightam直方图并检查匹配效果

## 算法描述

### 直方图均衡化

**算法描述**：

对于灰度值离散的图像，处理其概率（直方图值）和概率累计函数来替代处理连续情况下的概率密度函数。一副数字图像中，灰度级$r_k$出现的概率近似为

$$p_r(r_k)=\frac{n_k}{MN},k=0,1,2,\ldots,L-1$$

其中MN是像素中枢，n_k是灰度为r_k的像素个数。

概率累积函数为

$$cdf(p_r)=\sum_{j=0}^{k}p_r(r_j)=\frac{1}{MN}\sum_{j=0}^{k}n_j$$

L为图像中可能的灰度值数量，灰度变换函数$s_k$为

$$s_k=T(r_k)=L*cdf(p_r)$$

**伪代码**：

> BEGIN
>
> 输入图像img_river；
>
> 调用imhist() 计算img_river的概率灰度直方图hist_river；

> 计算灰度变换函数cdf (概率累计函数) :
>
> cdf(1) = hist_river(1);
>
> **FOR** i = 2 : 256
>
> cdf(i) = cdf(i - 1) + hist_river(i);
>
> 对于img_river的每个像素点(j, i) :
>
> img_river(j, i) ⟵ L * cdf(img_river(j, i)+1);
>
> END

## 直方图规定化

**算法描述**：

1. 计算原图像直方图$p_r(r)$，均衡化原图像得到$s_k$，（$[0,255]$）
2. 计算变换函数
   $G(z_q)=(L-1)\sum_{i=0}^{q}p_z(z_i),q=0,..,L-1$
3. 对$s_k,k=0,\ldots,L-1$, 用$G(z_q)$找相应$z_q$，使得$G(z_q)$最接近$s_k$（使得$|G(z_m)-s_k|$最小），并存储映射表$map$使得$z_q=map(s_k)$；
4. 均衡输入图像的中间步骤是概念上的，合并两个变换函数$T$、$G^{-1}$ 跳过这一步；

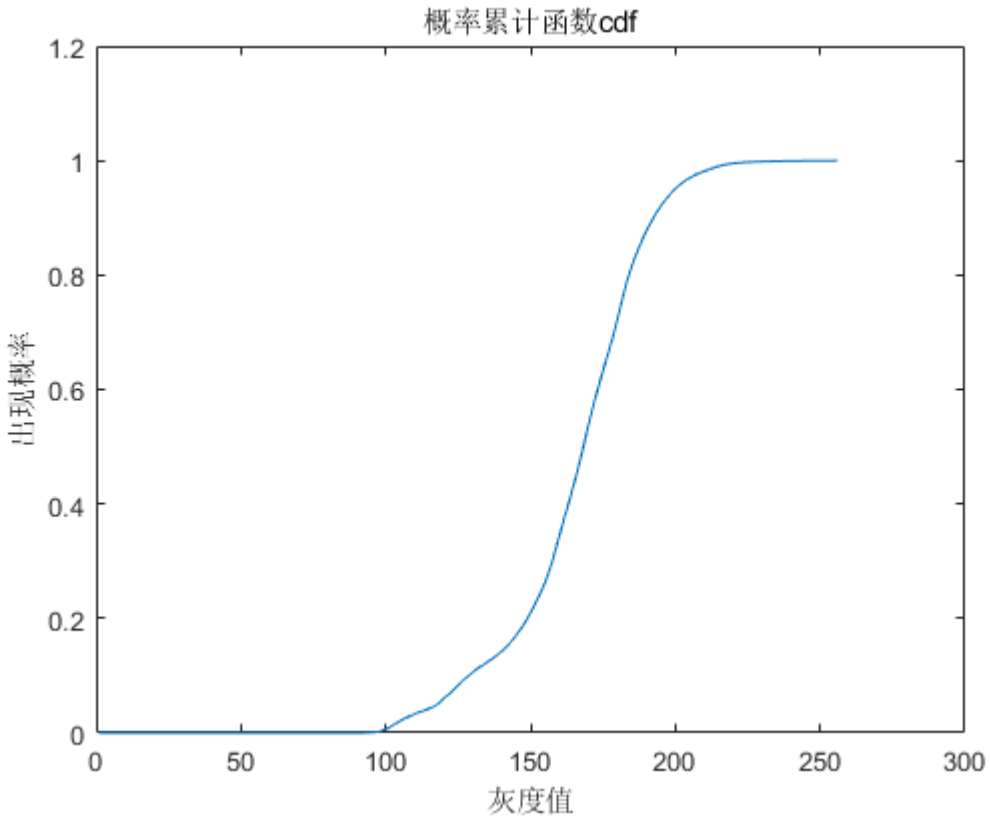
**伪代码**：

> BEGIN
>
> 读入图像img1， img2；
>
> 计算灰度概率直方图hist1和hist2和概率累计函数cdf1和cdf2；
>
> 计算差值矩阵diff：计算每一个cdf1(i)与每一个cdf2(j)的差的绝对值，存储在矩阵diff(j,i)中；
>
> 建立映射表map：遍历差值矩阵，对于每一个$cdf(i)$
>
> 求$index$满足 $diff(index)=min\{diff(1,j),\ diff(2,j),\ldots,diff(256,j)\}$
>
> map(i) = index;
>
> 遍历原图像像素点，用映射表进行灰度变换。
>
> END

# 结果及分析

## 均衡化

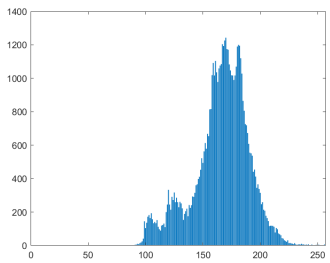
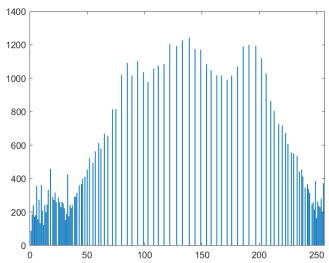
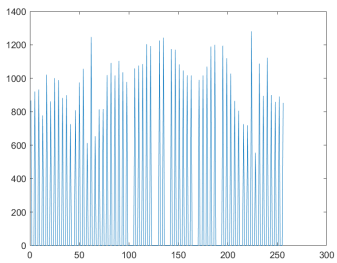
img_river的概率累计函数：




img_river处理前后，调用histeq()图像对比：

| 原图 | 处理后 | histeq处理 |
| --- | --- | --- |
|  |  |  |

img_river处理前后，调用histeq()灰度直方图对比：

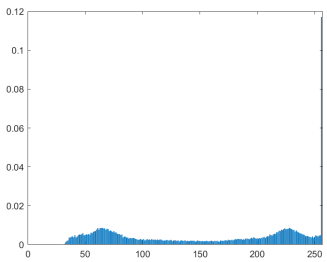
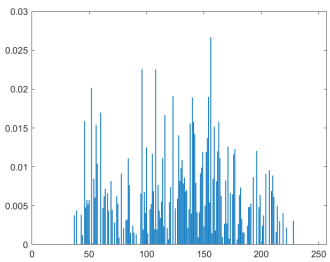
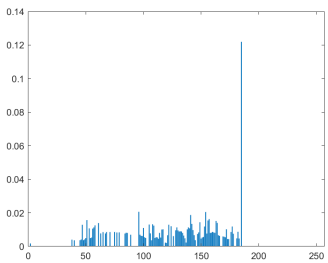
| 原图 | 处理后 | histeq处理 |
| --- | --- | --- |
|  |  |  |

## 规定化

处理前后图像：

| 原图 | 处理后 |
| --- | --- |
|  |  |

直方图：

| 原图 | lena | 处理后 |
| --- | --- | --- |
|  |  |  |

# 源代码

### getCDF.m

```
function [cdf] = getCDF(hist)
cdf = zeros(1, 256);
cdf(1) = hist(1);
for i = 2:256
      cdf(i) = cdf(i-1) + hist(i);
end
```

**task1.m**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Histogram Equalization
%%%%%%%%%%%%%%%%%%%%%%%%%%%

%读入图像
img_river = imread('../src/river', 'JPG');
info = imfinfo('../src/river','JPG');

%计算灰度值（概率）直方图
hist_river = imhist(img_river);
fig1 = bar(hist_river);
saveas(fig1, '../figure/1/hist_origin.png');
%hold on;
hist_river = hist_river/(info.Width*info.Height);

%计算概率累计函数
cdf = getCDF(hist_river);
%plot(cdf);
%hold on;

%变换
L = 256;
for i = 1:info.Width
    for j = 1:info.Height
        img_river(j, i) = L * cdf(img_river(j, i)+1);
    end
end

imwrite(img_river, '../figure/1/processed_river.jpg', 'jpg');

hist = imhist(img_river);
fig_result = bar(hist);
saveas(fig_result, '../figure/1/hist_result.png');

%调用库函数测试
sys = histeq(img_river);
imwrite(sys, '../figure/1/sys.jpg', 'jpg');
test_hist = imhist(sys);
fig_test = plot(test_hist);
saveas(fig_test, '../figure/1/hist_test.png');

```

**task2.m**

```
img1 = imread('../src/EightAM','png');
img2 = imread('../src/LENA','png');
img3 = imread('../src/EightAM','png');

info1 = imfinfo('../src/EightAM','png');
info2 = imfinfo('../src/LENA','png');
```

```
%计算匹配前后直方图(概率)
hist1 = imhist(img1)/(info1.Width*info1.Height);
hist2 = imhist(img2)/(info2.Width*info2.Height);

%计算匹配前后cdf
cdf1 = getCDF(hist1);
cdf2 = getCDF(hist2);

%计算差值
%第i行：对于每个cdf1(s),求与每个cdf2离散值的差
diff = zeros(256, 256);
for i = 1:256
    for j = 1:256
        diff(j, i) = abs(cdf1(i) - cdf2(j));
    end
end

%建立映射表: 输入灰度级1，输出最小差值的（映射）灰度级2
map = zeros(1,256);
for i = 1:256
    min = diff(1, i);
    index = 1;
    for j = 1:256     %找出第i行最小的差值(累计概率)
        if min > diff(j, i)
            min = diff(j, i);
            index = j;
        end
    end
    map(i) = index;
end

%变换
for i = 1:info1.Width
    for j = 1:info1.Height
        img3(j, i) = map(img1(j, i) + 1);
    end
end

imwrite(img3,'../figure/2/processed_EightAM.png','png');
hist_result = imhist(img3)/(info1.Width*info1.Height);

figure(1);
fig1 = bar(hist1);
saveas(fig1, '../figure/2/hist_enghtAM.png');

fig2 = bar(hist2);
saveas(fig2, '../figure/2/hist_lena.png');

fig3 = bar(hist_result);
saveas(fig3, '../figure/2/hist_result.png');

figure(2);
```

```
subplot(1,2,1),imshow(img1);
subplot(1,2,2),imshow(img3);

figure(3)
subplot(1,3,1), bar(hist1);
subplot(1,3,2), bar(hist2);
subplot(1,3,3), bar(hist_result);
```