

# 数字图像处理第三次报告

学生姓名：任泽华

班级：自动化 71

学号：2171411498

提交日期：2020-3-16

摘要：

本报告主要工作：本次作业对于直方图和空域滤波进行了探究，自编函数实现了直方图绘制，研究了 bmp 图像索引和调色板的关系；实现了直方图均衡，对直方图均衡原理进行了简要分析，对比分析了改善效果；分析了直方图匹配原理，对图片依据原图进行了直方图匹配，同时对不同图片进行了相应的匹配，分析了匹配效果；对四幅图都进行了直方图增强，分析了不同图片应用直方图增强的效果，对比了不同灰度图像的增强效果，比较了用不同尺寸的遮罩进行增强的效果；利用迭代法自动寻找阈值进行了图像分割，同时使用观察法手动寻找阈值进行了分割，对比了二者的差异。

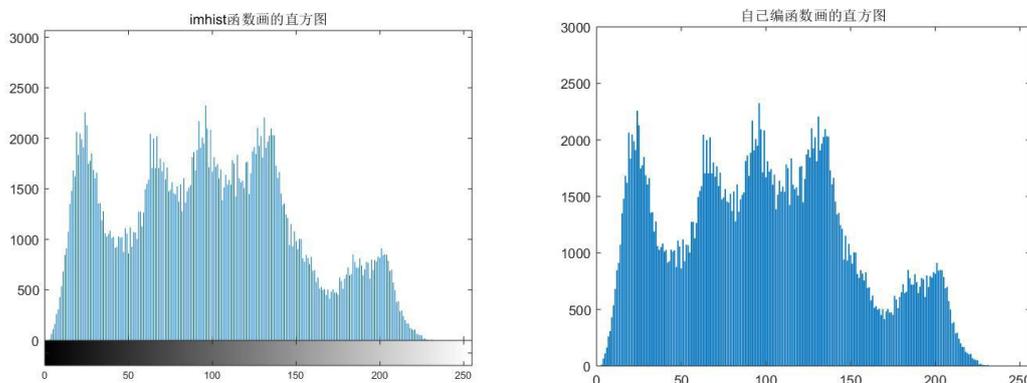
本报告操作平台为 matlab r2018b，所有代码为自己编写。

共 8 幅经变亮或者变暗处理的源图像；

## 一、 把附件图像的直方图画出；

### 1. 自编函数画直方图

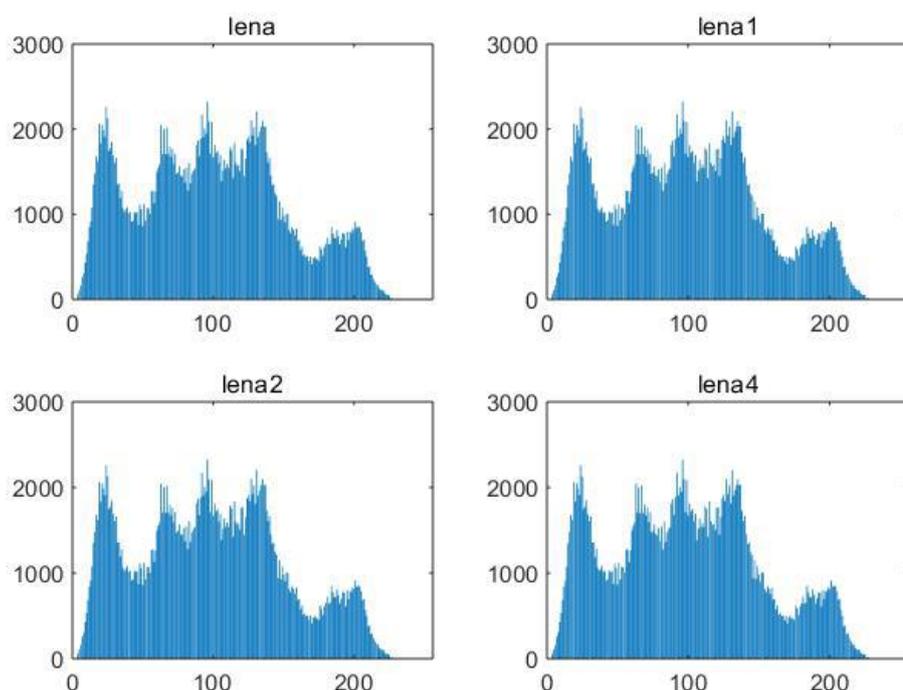
查阅了 matlab 函数发现的确有为图片绘制直方图的函数 `imhist`，直接调用即可画出直方图，但是看起来似乎界面不太友好，而且直方图颜色比较奇怪。为了进一步理解直方图的概念，也为了让直方图看起来更加明显，我自己编写了一个函数绘制直方图，效果果然比内置函数好了一些。下图左侧是 `imhist` 函数绘制的直方图，右侧是自编函数绘制的直方图。可以看出它们的总体形状完全一样，而自编函数绘制的直方图显得更加清晰。



### 2. 画总体直方图

用 `imread` 函数读取四幅 lena 图片，发现结果变得有些奇怪，明明不一样的四幅图在此处直方图显示居然完全一样。

直接读入的lena直方图



经过查阅资料与和同学交流，加上老师上课的讲解，我明白了这和 bmp 文件的存储格式有关。原因是 bmp 格式的文件，其存储主要分四部分，这在第一次实验中也有所提及，而第四部分存储的并不是图像的真实信息，而是图像索引，其值对应的是第三部分的颜色表信息。这样存储的好处之一就是在对图像亮度对比度等信息进行修改时不需要更改全部数据，只需更改颜色表即可。而 matlab 使用 `imread` 函数读入的只是索引信息，而四幅图的索引的相同的，这才造成了这个结果。

512x512 uint8				
	1	2	3	4
1	137	137	137	136
2	137	137	137	136
3	137	137	137	136
4	137	137	137	136
5	137	137	137	136
6	140	140	131	130
7	134	134	141	133
8	133	133	129	132
9	129	129	133	133
10	130	130	133	134

索引值

512x512 uint8				
	1	2	3	4
1	255	255	255	255
2	255	255	255	255
3	255	255	255	255
4	255	255	255	255
5	255	255	255	255
6	255	255	255	255
7	255	255	255	255
8	255	255	255	255
9	255	255	255	255
10	255	255	255	255

按调色板转变完的实际灰度

索引和真实像素值实际并不一样：

512x512 uint8		512x512 uint		调色板		
	1		1			
424	9	425	237	5	0.9294	0.9294
425	5	426	237	6	0.9294	0.9294
426	10	427	237	7	0.9294	0.9294
427	14	428	237	8	0.9294	0.9294
428	10	429	237	9	0.9294	0.9294
429	8	430	237	10	0.9294	0.9294
				11	0.9294	0.9294
				12	0.9294	0.9294
				13	0.9294	0.9294
				14	0.9294	0.9294
				15	0.9294	0.9294

使用[\[a,map\]=imread\('filename'\)](#)即可读入索引和调色板信息，

再使用 [g=ind2gray\(a,map\)](#)函数即可将其转化为灰度信息。

由上图可以看出，灰度值是由索引和调色板的乘积组成的：

$$255 \times 0.9294 = 236.997 \approx 237$$

且 0 值对应调色板第一行，即实际值对应：

$$(\text{索引}+1)\text{行值} \times 255$$

真实图像与索引图像对比



以上是关于索引和真实像素的关系与对比。而在进一步研究之前，我又发现了一个问题：每组图片都有一幅图显示比较诡异，而 matlab 中再次显示又恢复正常。经过研究，我发现因为图片经过了整体的调色，有些偏亮的像素在调色板上面没有对应的值。而普通相册等等软件对这些像素往往自动填充 0，即全为黑色；matlab 对其的处理的全部填充为 255，这就导致了差异，而实际情况应该是填充为 255 比较合适。在 matlab 里面修改这些像素为 0，就可以恢复图形软件查看的效果。

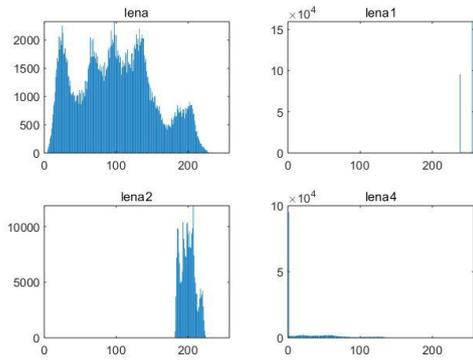
显示不正常图片与实际图片对比：



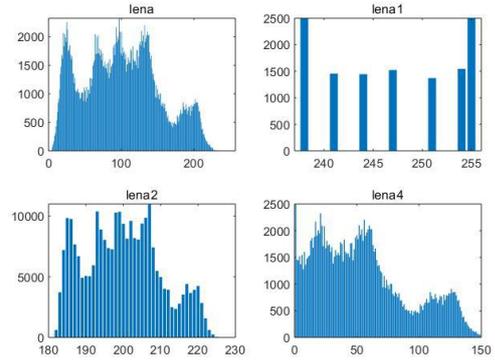
下面就可以画直方图了，由于处理过的图像往往偏黑或者偏白，其中某一像素值会占很大一部分，造成直方图其余部分显示不明显。在左侧图片显示了真实的直方图，右侧显示了经过放大处理以后的直方图。

Lena

修改后的lena直方图

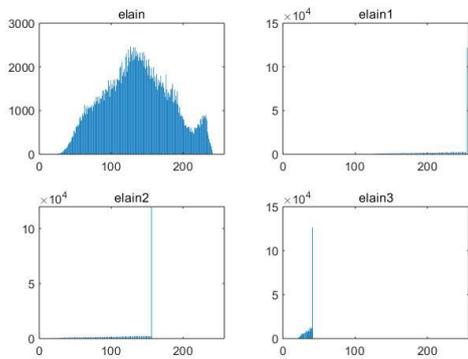


调整后的lena直方图

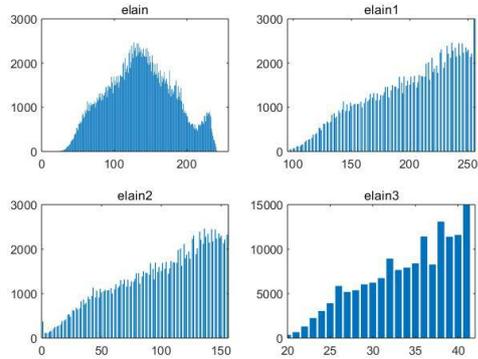


## Elain

修改后的elain直方图

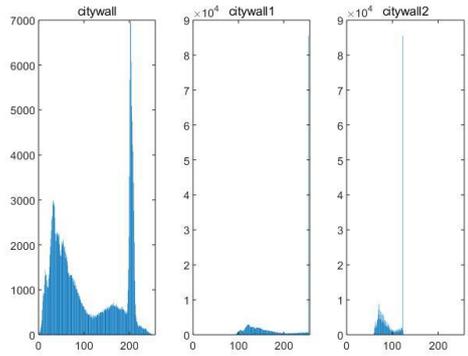


调整后的elain直方图

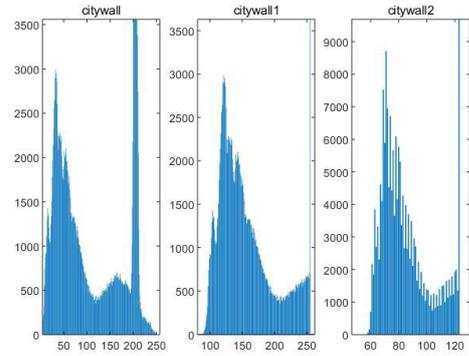


## Citywall

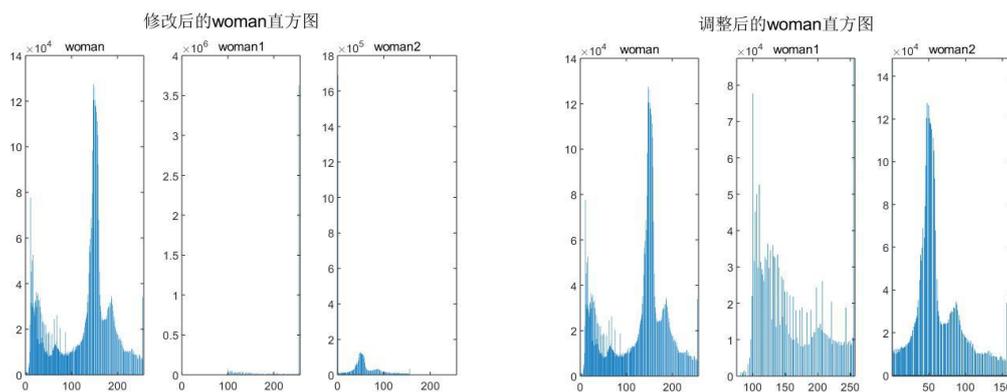
修改后的citywall直方图



调整后的citywall直方图



## Woman



经过观察，同组的几幅图片除了大量特殊值特别大以外，局部直方图的形状和原图都比较近似，这也从侧面说明了这些图片是同一个来源。

## 二、 把所有图像进行直方图均衡；输出均衡后的图像和源图像进行比对；分析改善内容；

### 1. 直方图均衡的原理：

先用变量  $r$  表示待处理的灰度。假设  $r$  的取值区间为  $[0, L-1]$ ，且  $r=0$  表示黑色， $r=L-1$  表示白色。对于输入图像中每个具有  $r$  值的像素产生一个输出灰度值  $s$ 。我们假设

$T(r)$  在区间  $0 \leq r \leq L-1$  上为严格单调递增函数。

当  $0 \leq r \leq L-1$  时， $0 \leq T(r) \leq L-1$ 。

利用变换函数：

$$s = T(r) = (L - 1) \int_0^r p_r(w) dw$$

可以证明无论 $p_r(w)$ 是一个怎样的函数，最后求得的  $s$  都是 $[0,L-1]$ 上面均匀分布的量。

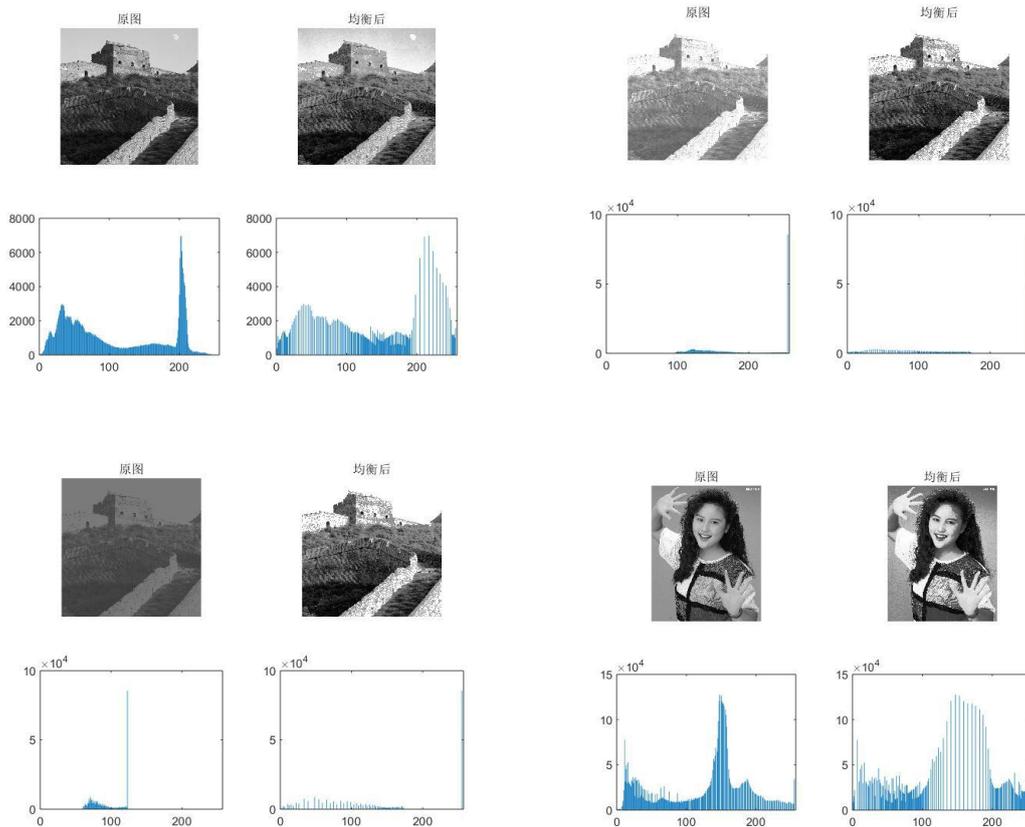
对于离散值，我们处理器直方图与求和来代替处理概率函数与积分。一幅数字图像中灰度级 $r_k$ 出现的概率近似为：

$$p(r_k) = n_k/MN$$

将变换写成离散形式：

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

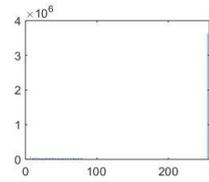
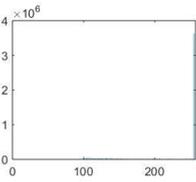
## 2. 编写程序对所有图片进行直方图均衡：



原图



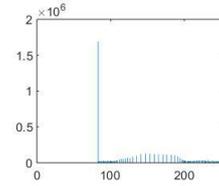
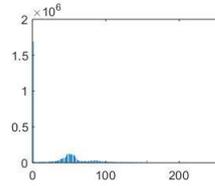
均衡后



原图



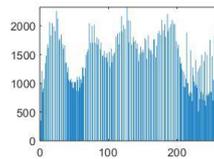
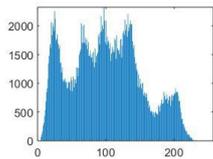
均衡后



原图



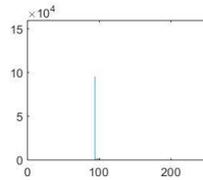
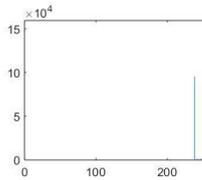
均衡后



原图



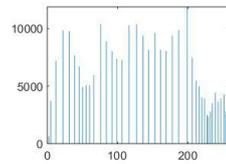
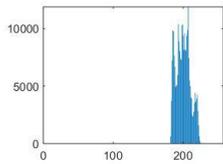
均衡后



原图



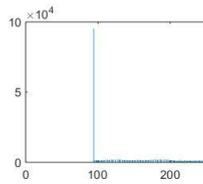
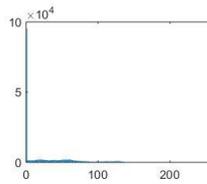
均衡后



原图



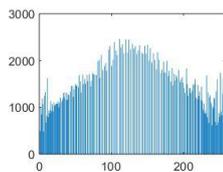
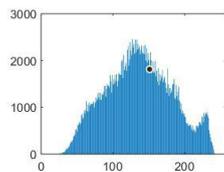
均衡后



原图



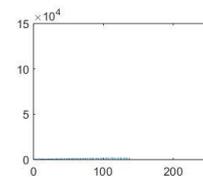
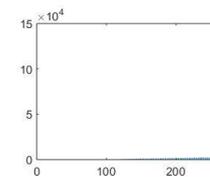
均衡后

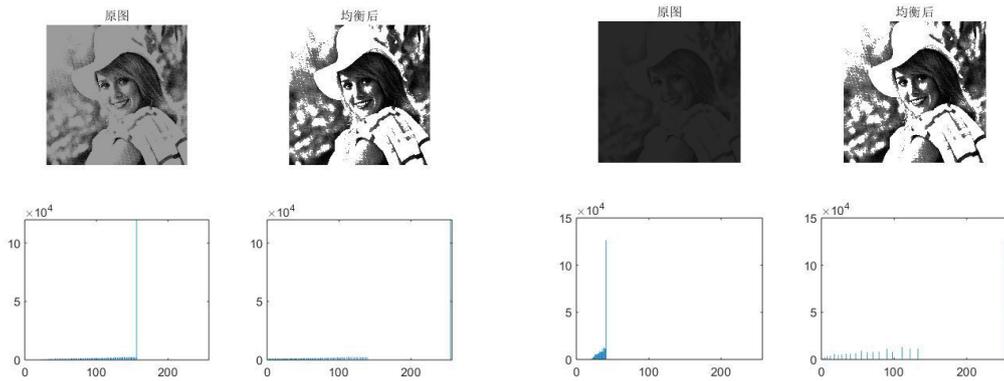


原图



均衡后





### 3. 分析改善内容：

从直方图上看，直方图均衡处理将图片的直方图形状变得更加均匀，之前峰谷不均的形状变得比较平缓，之前分布集中的变得分散均匀。有些本来像素值就比较少，而且比较集中，在均衡后也没办法变得过于均匀，只能尽可能地保证平均值的均衡。代表就是原来值特别大的0或255（黑白）区域变得往中间靠了，原来分布集中的变得散开了。

从图片上来看，直方图均衡带来的最大改变就是图片变得灰度更加均匀，之前偏白的图片整体黑了一些，之前偏黑的图片整体白了一些。而那些灰度相似，即对比度很低的图片也变得黑白分明，对比度增大了。由于只能变换现有的像素，不能创造像素值，所以本来像素值比较接近（像素值少）的图片在均衡后就会显得失真。

## 三、 进一步把图像按照对源图像直方图的观察，各自自行指定不同源图像的直方图，进行直方图匹配，进行图像增强；

### 1. 直方图匹配原理：

直方图匹配指将一幅图像的直方图变成规定形状的直方图而进行的

图像增强方法。其中主要应用的还是直方图均衡，不过不同的是，这次是对原图和模板图都进行直方图均衡，根据均衡结果进行对比，找到最接近的像素取值。

对于原图灰度  $r$  计算  $T$  为最终要得到的映射关系：

$$s = T(r) = (L - 1) \int_0^r p_r(w) dw$$

我们给定一个随机变量，具有如下性质：

$$G(z) = (L - 1) \int_0^z p_z(t) dt = s$$

根据上面两个式子，我们有：

$$G(z) = T(r)$$

那么就必须有：

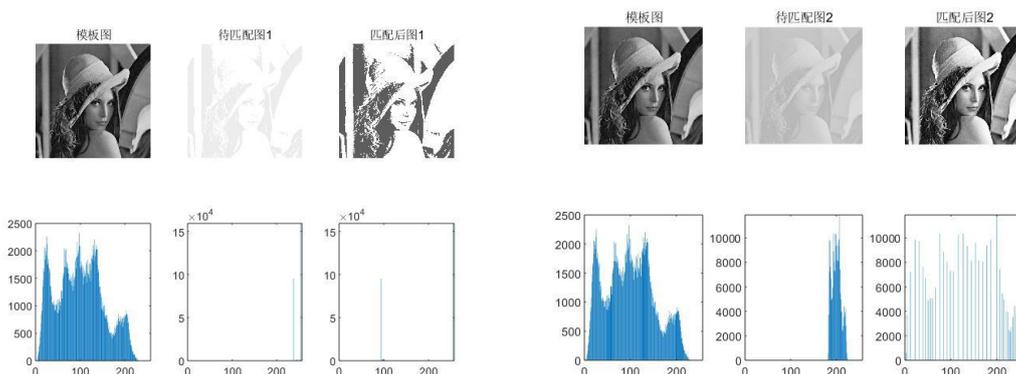
$$z = G^{-1}[T(r)] = G^{-1}(s)$$

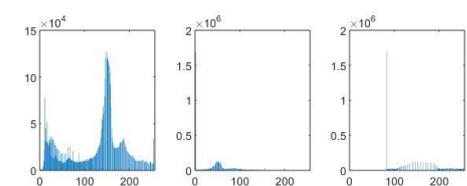
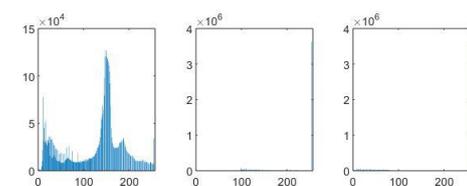
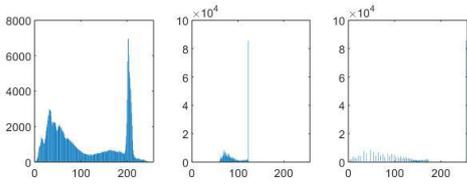
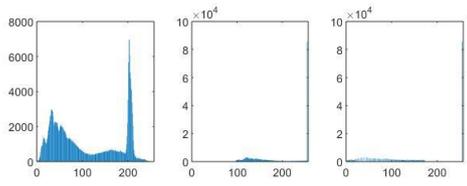
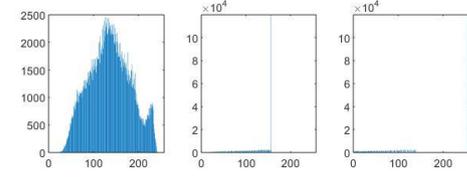
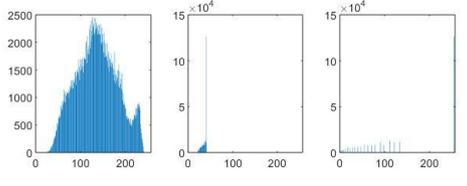
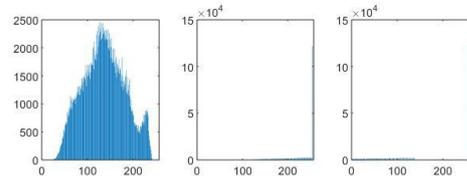
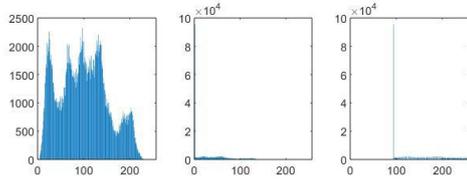
而对于离散数值，寻找  $G^{-1}(s)$  就是一个匹配的过程，要找到最接近的值，我在此次采用了 matlab 里面的语句：

`[~,index]=min(abs(G-s));`

找到最接近  $s$  的  $G$  的值，从而找到合适的变换取值。

## 2. 用各自的原图匹配

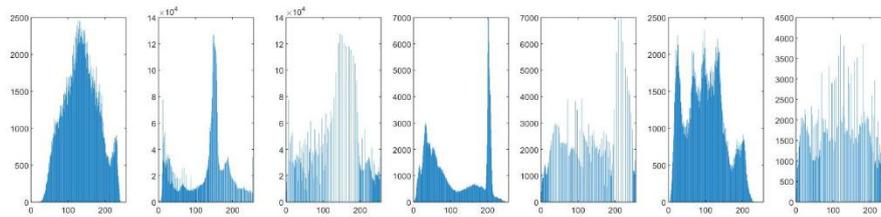
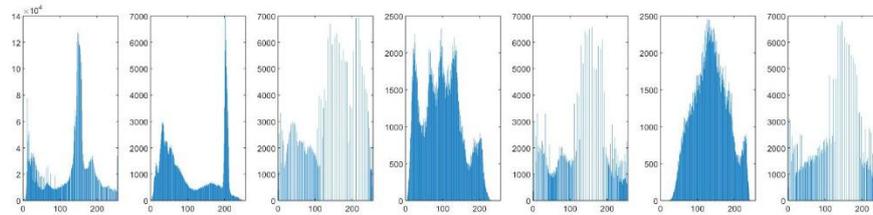




从图像中可以看出，使用原图进行匹配时，最终图片的像素和原图片近似，灰度也整体近似，但对比度差别还是比较大，这是因为待匹配的图片本身由于先前处理的原因，已经丢失了一部分图像的细节信息，这就是的进行匹配恢复的时候无法恢复处原始图片一模一样的信息。

### 3. 用每个原图为模板匹配其他原图





这是一个拓展练习，将不同的原图片作为模板让其他原图取匹配它，得到的效果从直方图上面看起来还不错，但是实际图片的效果就显得比较奇怪，看来每个图片都有其独特的直方图信息，如果改变了这个特征，图片的风格也会发生一定的变化。

## 四、 对 elain 和 lena 图像进行 7\*7 的局部直方图增强；

### 1. 局部直方图增强原理

利用第二问中的直方图均衡原理，使用一个小方块（mask）对图片进行遍历，每次移动一行（或一列），对方块中间的像素值取直方图均衡，而范围仅限与这个方块内。为了保证图片尺寸不发生变化，通常要对图片进行 padding（补零）操作，如果 mask 是  $7 \times 7$  的话，通常上下左右分别要加三行（列）。Mask 也可以取得小一点，但是这样的话最后实现的效果也会相应地变差。

### 2. Lena、 elain



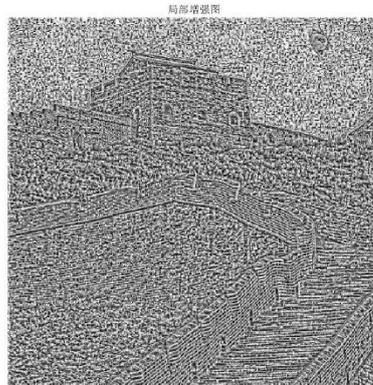
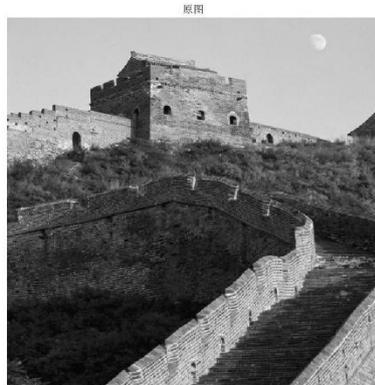
elain 7×7



局部直方图增强的结果就是使得图片细节被保留显示，所有边缘信息都会得到增强，看得更加明显。

### 3. 其他两幅图

citywall 7×7





Citywall 增强后细节显得比较杂乱，因为本身图片细节太多（墙砖、树林等等），而图片本身分辨率不高，这就导致增强后的图片显得杂乱无章。

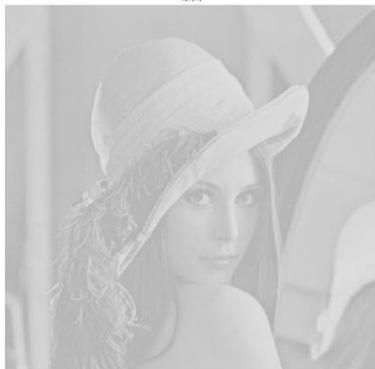
而 woman 由于本身是高清图片，细节内容很多，而整体看来又显得比较简单，所以使用局部直方图增强后反而看不出内容，而其中的细节信息在放大后可以看得比较明显。

#### 4. 对于灰度级别不明显的图像：



lena2 7×7

原图



局部增强图



elain1 7×7

原图



局部增强图



elain2 7×7

原图



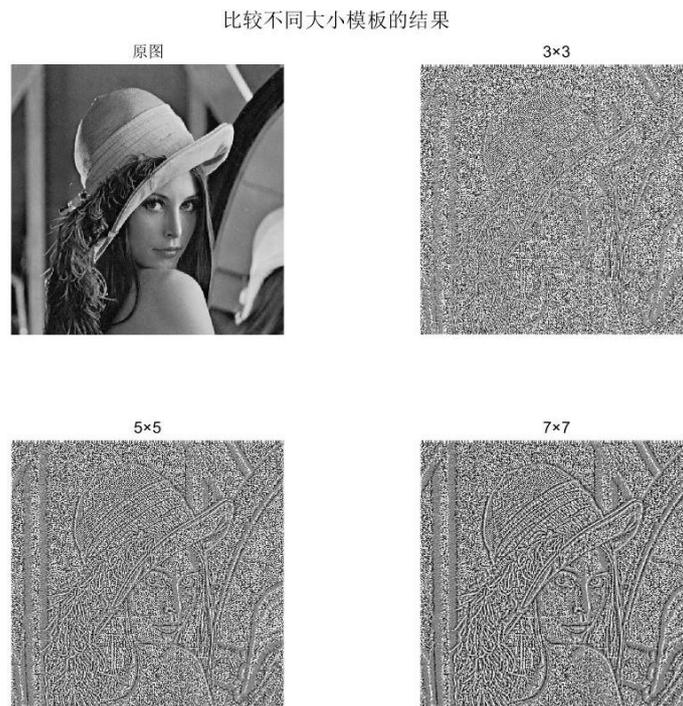
局部增强图



灰度级别不同的图像本身就丢失了许多细节信息，所以在局部直方图增强后没有变化的部分全部呈现纯色块，而且偏亮的图片增强后会偏黑，而偏黑的图片增强后会偏亮。

## 5. 比较不同模板大小的增强效果：

改变 mask 大小，分别使用  $3 \times 3$ 、 $5 \times 5$ 、 $7 \times 7$  的 mask 对 lena 图片进行局部直方图增强，可以发现，mask 尺寸变大后，会保留更多的细节信息。



## 五、 利用直方图对图像 elain 和 woman 进行分割；

### 1. 算法求阈值

主要原理是：

阈值分割方法实际上是输入图像  $f$  到输出图像  $g$  的如下变换：

$$g(i,j) = \begin{cases} 1 & f(i,j) \geq T \\ 0 & f(i,j) < T \end{cases}$$

利用迭代算法找到比较合适的 T 值：

(1) 给定一个初始值  $T$ ;

(2) 利用  $T$  分割原图像，小于的计算均值  $\text{mean1}$ ，大于的计算均值  $\text{mean2}$ ;

(3)  $T' = 1/2 (\text{mean1} + \text{mean2})$ ;

(4)  $T_0 = |T - T'|$ ;

(5)  $T = T'$ ;

(6)  $T_0$  小于阈值，结束，否则返回 (2)。

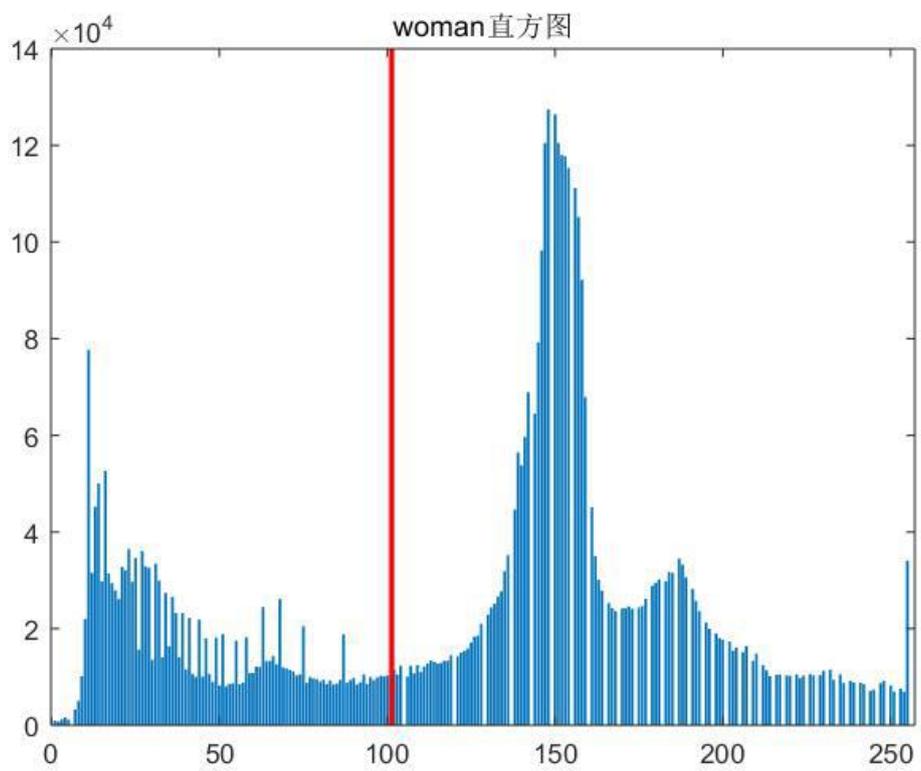
下面是几幅图进行自动分割的结果:

Woman  
 $T = 101.299$

分割区域 (前景为黑)



分割结果 (背景为白)

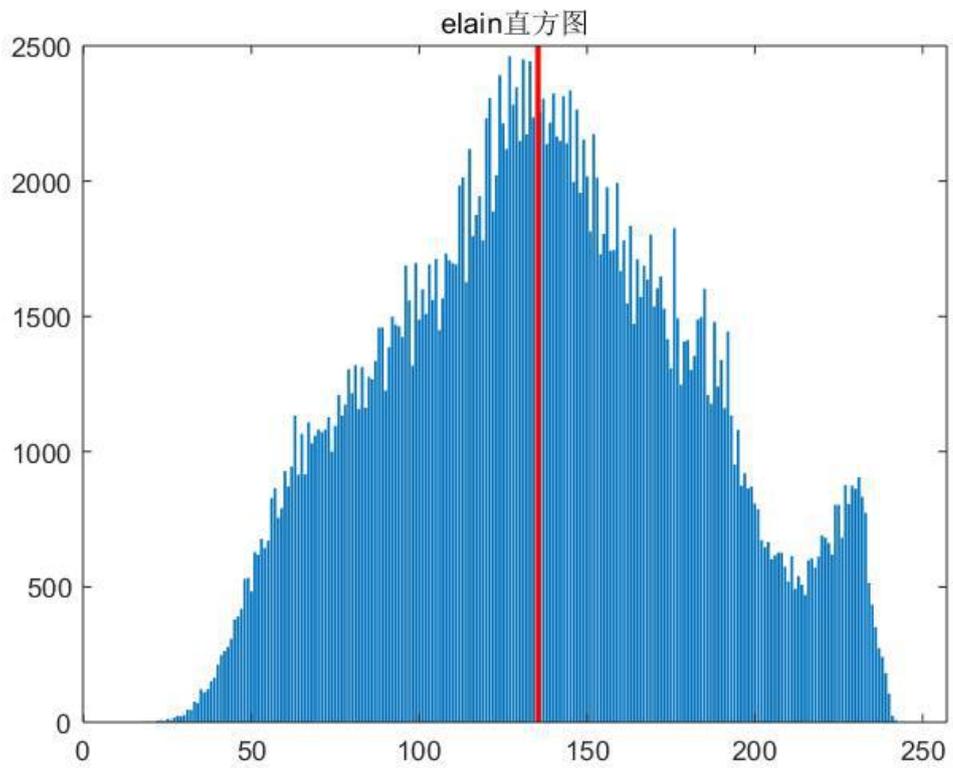


Elain  
T=135.392

分割区域（前景为黑）

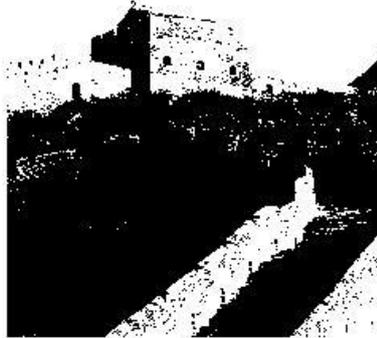


分割结果（背景为白）

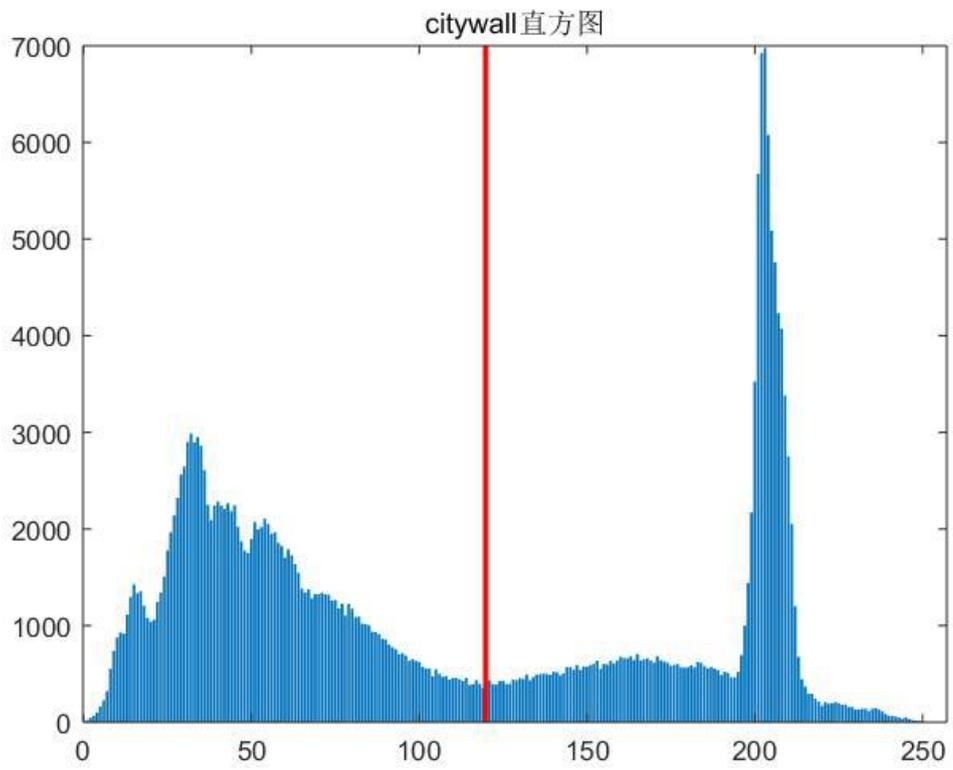
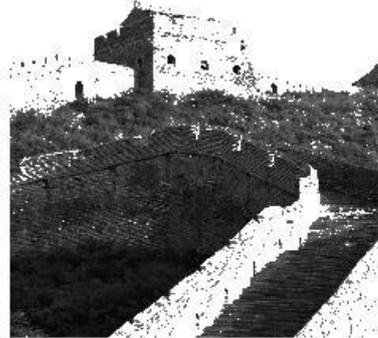


Citywall  
T=119.792

分割区域（前景为黑）

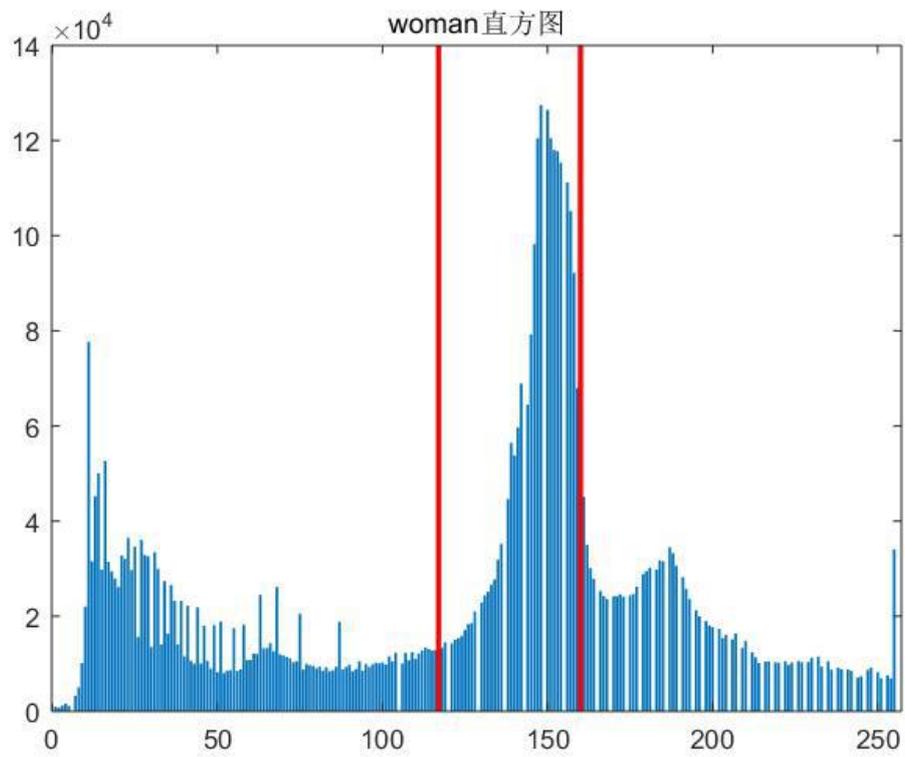
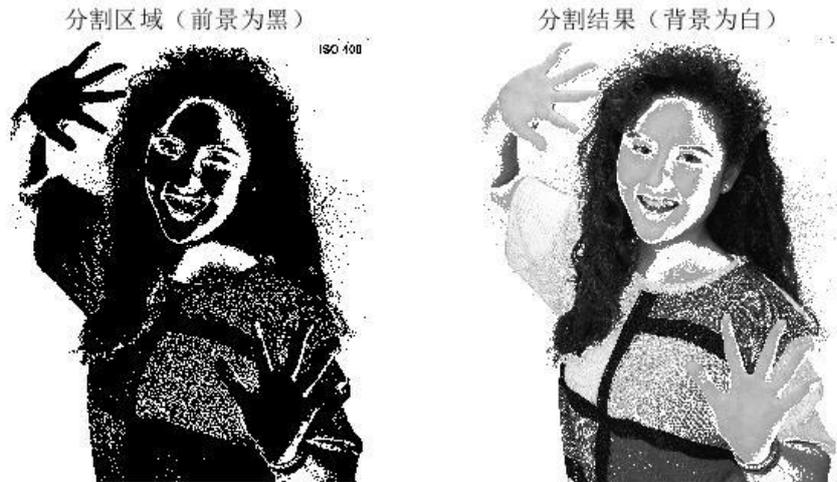


分割结果（背景为白）

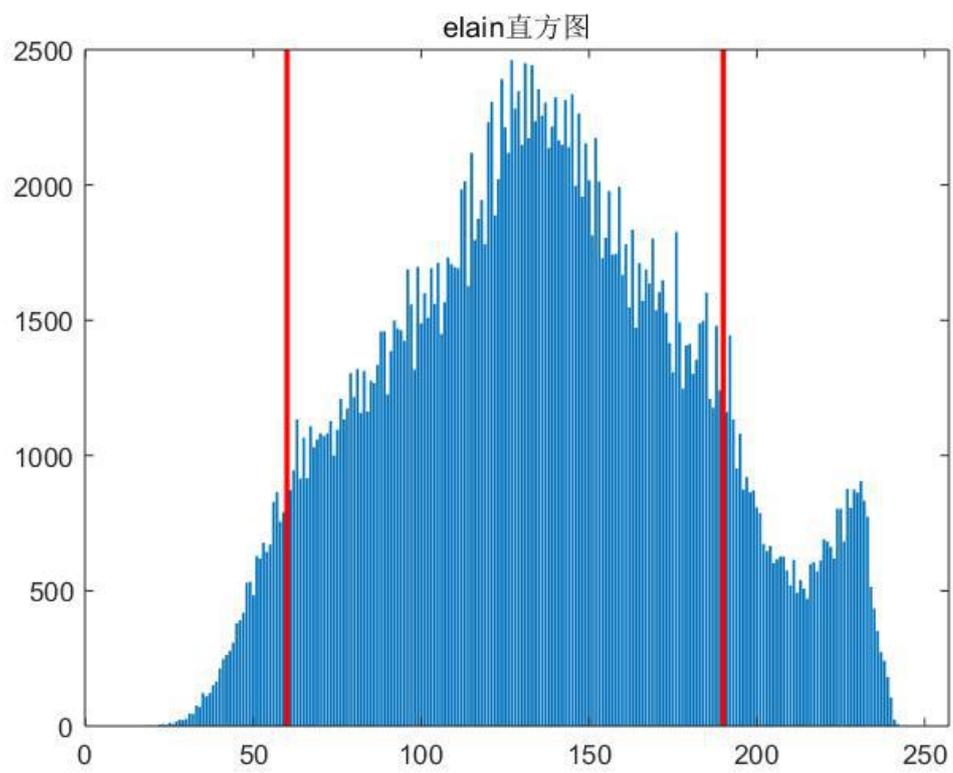


## 2. 自己看图取阈值

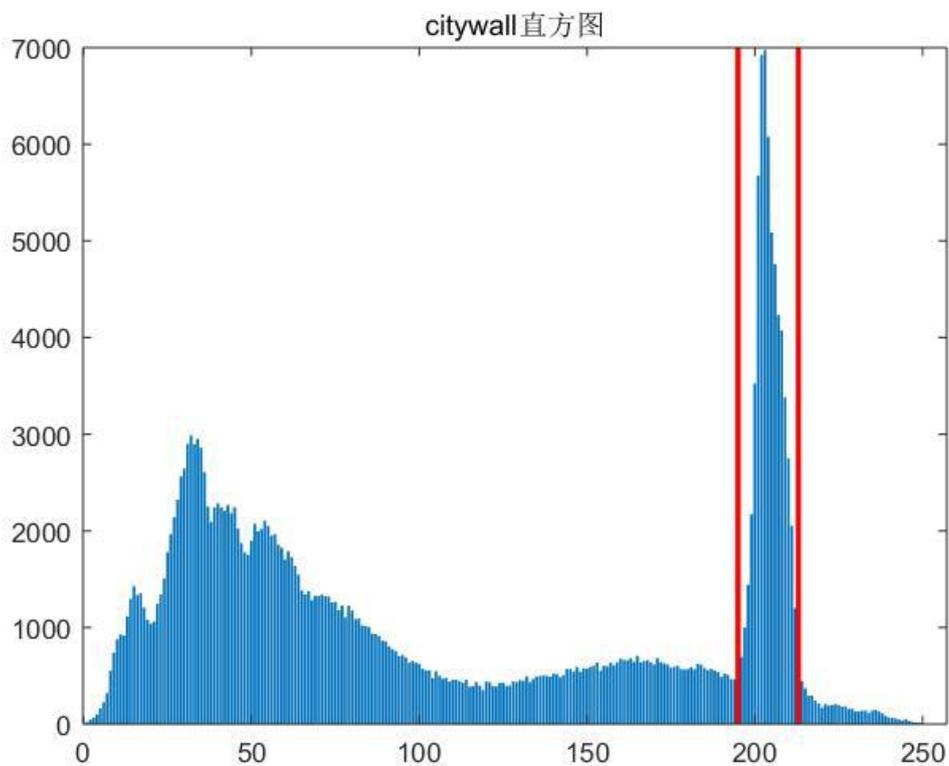
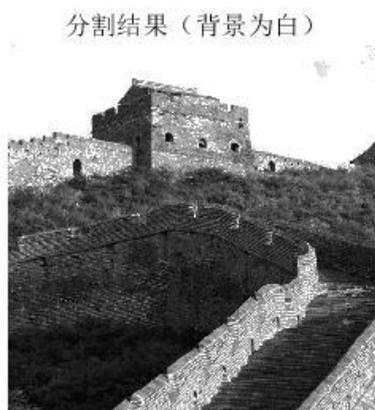
Woman left=117; right=160;



Elain left=60; right=190;



Citywall left=195; right=213;



自己取值区分效果还是可以的，将背景占用最多的像素去掉即可。

# 附录

## 1. 参考文献

[1] Rafael C. Gonzalez (拉斐尔 C. 冈萨雷斯), Richard E. Woods (理查德 E. 伍兹). 数字图像处理(第三版) (英文版). 北京: 电子工业出版社. 2017 年.

[2] 关于 matlab 中的 gray2ind 函数

<https://zhidao.baidu.com/question/500620214.html>

[3] 索引图像 (百度百科)

<https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95%E5%9B%BE%E5%83%8F/2954056?fr=aladdin>

[4] 如何在 matlab 中画出图像灰度直方图

<https://jingyan.baidu.com/article/91f5db1b72564d1c7e05e35c.html>

[5] matlab 中函数 bar 的用法 本文来自: MATLAB 之家, 原地址:

<http://www.matlabhome.cn/post/104.html>

## 2. 源代码

### (1) 第一问

#### 1. 自己编函数画直方图和 imhist 对比

```
[a,map]=imread('lena.bmp');
```

```
imhist(a)
```

```
title('imhist 函数画的直方图')
```

```
num=zeros(256,1);
```

```
[r,c]=size(a);
```

```
for i=1:r
```

```
    for j=1:c
```

```
        num(a(i,j))=num(a(i,j))+1;
```

```
    end
```

```
end
```

```
figure
bar(num);
axis([0,256,0,3000]);
title('自己编函数画的直方图')
```

## 2. 自编统计像素个数函数

```
function num = getnum(a)
num=zeros(256,1);
[r,c]=size(a);
for i=1:r
    for j=1:c
        num(a(i,j))=num(a(i,j))+1;
    end
end
end
```

## 3. 画总体直方图（直接读入）

```
[a,map]=imread('lena.bmp');
num=getnum(a);
subplot(221)
bar(num);
axis([0,256,0,3000]);
title('lena')
[a,map]=imread('lena1.bmp');
num=getnum(a);
subplot(222)
bar(num);
axis([0,256,0,3000]);
title('lena1')
[a,map]=imread('lena2.bmp');
num=getnum(a);
```

```
subplot(223)
bar(num);
axis([0,256,0,3000]);
title('lena2')
[a,map]=imread('lena4.bmp');
num=getnum(a);
subplot(224)
bar(num);
axis([0,256,0,3000]);
title('lena4')

sgtitle('直接读入的 lena 直方图')
```

#### 4.测试索引值和实际灰度的区别

```
[a,map]=imread('woman.bmp');%修改图名即可

set(gcf,'Position',[200,200,1000,500])
subplot(121)
imshow(a)

title('索引读入图')

g=ind2gray(a,map);
subplot(122)
imshow(g)

title('按调色板转为原图')

g1=g;
g1(g1==255)=0;
figure
set(gcf,'Position',[200,200,1000,500])
subplot(121)
imshow(g)

title('按调色板转为原图')
```

```
subplot(122)
imshow(g1)

title('索引超过调色板值改成 0')
```

## 5.绘制最终真实的直方图

%绘制最终真实的直方图 修改名字即可

```
[a,map]=imread('elain.bmp');
g=ind2gray(a,map);
num=getnum(g);
subplot(221)
bar(num);
title('elain')
[a,map]=imread('elain1.bmp');
g=ind2gray(a,map);
num=getnum(g);
subplot(222)
bar(num);
axis([95,256,0,3000]);
title('elain1')
[a,map]=imread('elain2.bmp');
g=ind2gray(a,map);
num=getnum(g);
subplot(223)
bar(num);
axis([0,156,0,3000]);
title('elain2')
[a,map]=imread('elain3.bmp');
g=ind2gray(a,map);
num=getnum(g);
subplot(224)
bar(num);
axis([20,42,0,15000]);
```

```
title('elain3')
sgtitle('调整后的 elain 直方图')
```

## (2) 第二问

```
%直方图均衡

[a,map]=imread('woman2.bmp');
g=ind2gray(a,map);
g1=g;
num=getnum(g);
[c,r]=size(g);
pro=num/c/r;
S=zeros(256,1);

for i=1:256    %求变换对应向量
    s=0;
    for j=1:i
        s=pro(j)+s;
    end
    S(i)=round(s*255);
end

for i=1:c    %进行变换
    for j=1:r
        g1(i,j)=S(g(i,j)+1);
    end
end
num1=getnum(g1);

subplot(221)    %画图

imshow(g);title('原图')

subplot(222)
```

```
imshow(g1);title('均衡后')
```

```
subplot(223)
```

```
bar(num);
```

```
subplot(224)
```

```
bar(num1);
```

### (3) 第三问

#### 1.用各自的原图匹配

```
%直方图匹配
```

```
[a0,map0]=imread('woman.bmp');
```

```
g0=ind2gray(a0,map0);
```

```
num0=getnum(g0);
```

```
[c,r]=size(g0);
```

```
pro0=num0/c/r;
```

```
G=zeros(256,1);
```

```
for i=1:256 %求模板图变换向量
```

```
    s=0;
```

```
    for j=1:i
```

```
        s=pro0(j)+s;
```

```
    end
```

```
    G(i)=round(s*255);
```

```
end
```

```
[a,map]=imread('woman2.bmp');
```

```
g=ind2gray(a,map);
```

```
g1=g;
```

```
num=getnum(g);
```

```
[c,r]=size(g);
```

```
pro=num/c/r;
```

```
S=zeros(256,1);
```

```
for i=1:256 %求待变换图变换向量
```

```

        s=0;
        for j=1:i
            s=pro(j)+s;
        end
        S(i)=round(s*255);
    end
for i=1:c      %进行变换
    for j=1:r
        s=S(g(i,j)+1);
        [~,index]=min(abs(G-s));
        g1(i,j)=G(index);
    end
end
num1=getnum(g1);
subplot(231)  %画图

imshow(g0);title('模板图')
subplot(232)
imshow(g);title('待匹配图 2')
subplot(233)
imshow(g1);title('匹配后图 2')

subplot(234)
bar(num0);
subplot(235)
bar(num);
subplot(236)
bar(num1);

```

## 2.用每个原图为模板匹配其他原图

%直方图匹配 2

```

[a0,map0]=imread('elain.bmp');
[a1,map1]=imread('woman.bmp');
[a2,map2]=imread('citywall.bmp');
[a3,map3]=imread('lena.bmp');

g0=ind2gray(a0,map0);
num0=getnum(g0);
[c0,r0]=size(g0);
pro0=num0/c0/r0;
G=zeros(256,1);

for i=1:256    %求模板图变换向量

    s=0;
    for j=1:i
        s=pro0(j)+s;
    end
    G(i)=round(s*255);
end

g1=ind2gray(a1,map1);
g1_1=g1;
num1=getnum(g1);
[c1,r1]=size(g1);
pro1=num1/c1/r1;
S=zeros(256,1);

for i=1:256    %求待变换图变换向量

    s=0;
    for j=1:i
        s=pro1(j)+s;
    end
    S(i)=round(s*255);
end

for i=1:c1    %进行变换

    for j=1:r1

```

```

        s=S(g1(i,j)+1);
        [~,index]=min(abs(G-s));
        g1_1(i,j)=G(index);
    end
end
num1_1=getnum(g1_1);

g2=ind2gray(a2,map2);
g2_1=g2;
num2=getnum(g2);
[c2,r2]=size(g2);
pro2=num2/c2/r2;
S=zeros(256,1);

for i=1:256    %求待变换图变换向量

    s=0;
    for j=1:i
        s=pro2(j)+s;
    end
    S(i)=round(s*255);
end

for i=1:c2    %进行变换

    for j=1:r2
        s=S(g2(i,j)+1);
        [~,index]=min(abs(G-s));
        g2_1(i,j)=G(index);
    end
end
num2_1=getnum(g2_1);

g3=ind2gray(a3,map3);
g3_1=g3;
num3=getnum(g3);
[c3,r3]=size(g3);
pro3=num3/c3/r3;

```

```

S=zeros(256,1);
for i=1:256    %求待变换图变换向量
    s=0;
    for j=1:i
        s=pro3(j)+s;
    end
    S(i)=round(s*255);
end
for i=1:c3    %进行变换
    for j=1:r3
        s=S(g3(i,j)+1);
        [~,index]=min(abs(G-s));
        g3_1(i,j)=G(index);
    end
end
num3_1=getnum(g3_1);

subplot(271)    %画图

imshow(g0);title('模板图')
subplot(272)
imshow(g1);title('待匹配图 1')
subplot(273)
imshow(g1_1);title('匹配后图 1')
subplot(274)
imshow(g2);title('待匹配图 2')
subplot(275)
imshow(g2_1);title('匹配后图 2')
subplot(276)

```

```
imshow(g3);title('待匹配图 3')
subplot(2,7,7)
imshow(g3_1);title('匹配后图 3')
subplot(2,7,8)
bar(num0);
subplot(2,7,9)
bar(num1);
subplot(2,7,10)
bar(num1_1);
subplot(2,7,11)
bar(num2);
subplot(2,7,12)
bar(num2_1);
subplot(2,7,13)
bar(num3);
subplot(2,7,14)
bar(num3_1);
```

#### (4) 第四问

```
%局部直方图增强
masksize=3;%7×7
[a,map]=imread('elain2.bmp');
g=ind2gray(a,map);
[c,r]=size(g);
padding=floor(masksize/2);
pa=zeros(c+padding*2,r+padding*2);
for i=1:c      %padding 后的矩阵
    for j=1:r
        pa(i+padding,j+padding)=g(i,j);
    end
end
end
```

```

g1=g;
for i=1:c
    for j=1:r
        num=getnum(pa(i:i+masksize-1,j:j+masksize-1));
        pro=num/masksize/masksize;
        S=zeros(256,1);

        for k=1:256    %求变换对应向量

            s=0;
            for m=1:k
                s=pro(m)+s;
            end
            S(k)=round(s*255);
        end

        g1(i,j)=S(g(i,j)+1);%进行变换

    end
end
subplot(121)

imshow(g);title('原图')

subplot(122)

imshow(g1);title('局部增强图')

sgtitle('elain2 7×7')
% sgtitle('lena 5×5')

```

## (5) 第五问

### 1. 算法求阈值

```
% 背景分割 2
```

```
[a,map]=imread('woman.bmp');
```

```

g=ind2gray(a,map);
[r,c]=size(g);
g1=g;
g2=ones(r,c);

T=128; %迭代初值

T0=inf;
while(T0>=1)
    part1=find(g>T);
    part2=find(g<=T);
    sum1=sum(g(part1));
    sum2=sum(g(part2));
    mean1=sum1/numel(part1);
    mean2=sum2/numel(part2);
    T1=(mean1+mean2)/2;
    T0=abs(T-T1);
    T=T1;
end
for i=1:r
    for j=1:c
        if (g(i,j)>=T)
            g1(i,j)=255;
        else
            g2(i,j)=0;
        end
    end
end
end
figure
subplot(121)

imshow(g2);title('分割区域（前景为黑）')

subplot(122)

imshow(g1);title('分割结果（背景为白）')

num=zeros(256,1);
[r,c]=size(g);

```

```

for i=1:r
    for j=1:c
        num(g(i,j)+1)=num(g(i,j)+1)+1;
    end
end
figure
bar(num);

title('woman 直方图')

line([T,T],[0,140000],'color','r','linewidth',2)

```

## 2.自己看图取阈值

```

% 背景分割

left=117;
right=160;
[a,map]=imread('woman.bmp');
g=ind2gray(a,map);
[r,c]=size(g);
g1=g;
g2=ones(r,c);
for i=1:r
    for j=1:c
        if (g(i,j)<=right)&&(g(i,j)>=left)
            g1(i,j)=255;
        else
            g2(i,j)=0;
        end
    end
end
figure
subplot(121)

imshow(g2);title('分割区域（前景为黑）')

```

```
subplot(122)
imshow(g1);title('分割结果（背景为白）')
num=zeros(256,1);
[r,c]=size(g);
for i=1:r
    for j=1:c
        num(g(i,j)+1)=num(g(i,j)+1)+1;
    end
end
figure
bar(num);
title('woman 直方图')
line([left,left],[0,140000],'color','r','linewidth',2)
line([right,right],[0,140000],'color','r','linewidth',2)
```