

——直方图均衡化和规定化

实验内容

- 必做内容：

- (1) 调用matlab函数完成直方图均衡化和规定化；

- (2) 调用Matlab关于直方图的相关函数完成直方图均衡化和规定化函数自编；

- 选做内容：

- (1) 在Matlab平台下，不利用直方图相关函数自编函数实现直方图均衡化和规定化算法。

实验平台： Windows10 + Matlab

CONTENT 目录

PART 01 直 方 图

PART 02 均 衡 化

PART 03 规 定 化

PART 04 程 序 代 码





Part 1

灰度级直方图

1.1 灰度级直方图：

▣灰度级的直方图是灰度级的函数，是反应一幅图像中的灰度级与出现这种灰度的概率之间的图形。灰度直方图反映了数字图像中的每一个灰度级与其出现频率或频数之间的关系，它能描述该图像的概貌。通常用横坐标表示灰度级，纵坐标表示频数或频率。通过修改直方图的方法增强图像是一种实用且有效的处理技术。

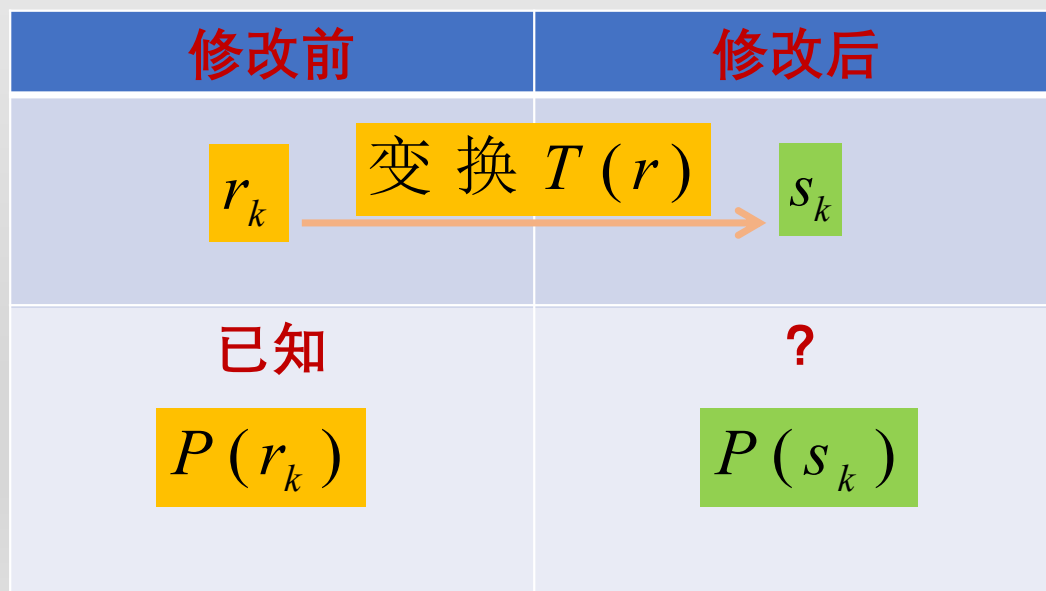
▣设图像的灰度范围为 $[a,b]$ ， r 为此灰度范围内的任一灰度级， $p(r)$ 为这幅图像中灰度级为 r 的像素出现的频率，可以看出 $p(r)$ 是 r 的函数。该函数的图形称为这幅图像的直方图。

$$p(r) = \frac{\text{灰度为}r\text{的像素数}}{\text{图像上的总像素数}}$$

$$\sum_{i=1}^n p(r_i) = 1$$

1.2 直方图处理的思想：

大多数自然图像，其灰度分布集中在较窄的区间，引起图像细节不够清晰，采用直方图修正后可使图像的灰度间距拉开或使灰度分布均匀，从而增大反差，使图像细节清晰，达到增强的目的。例如一幅过曝光的图片，其灰度级都集中在高亮度范围内，而曝光不足的图片，其灰度级集中在低亮度范围内，具有这样直方图的图片其可视效果比较差。



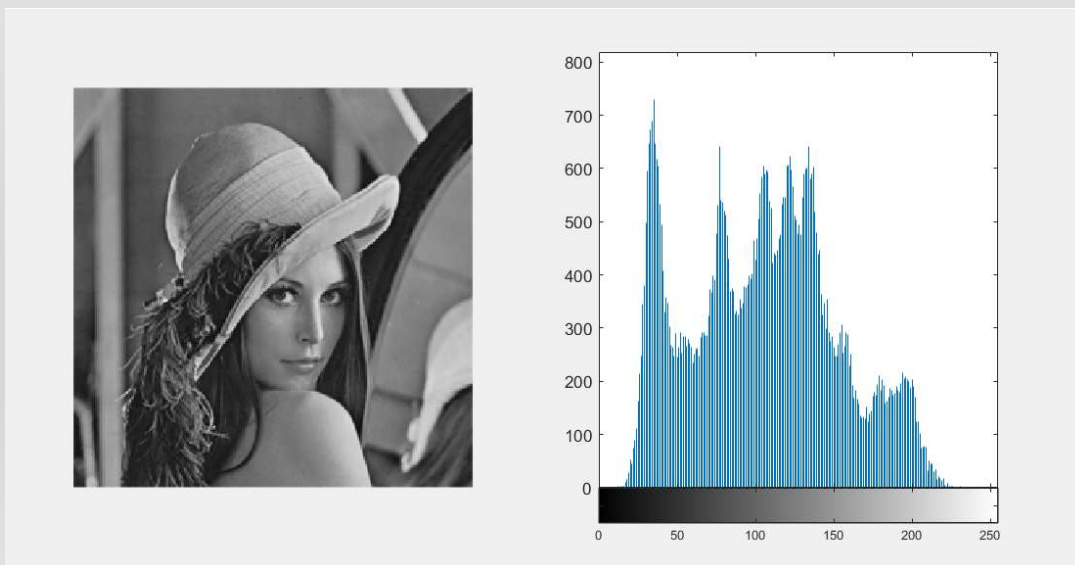
1.3 灰度直方图统计/绘制的步骤:

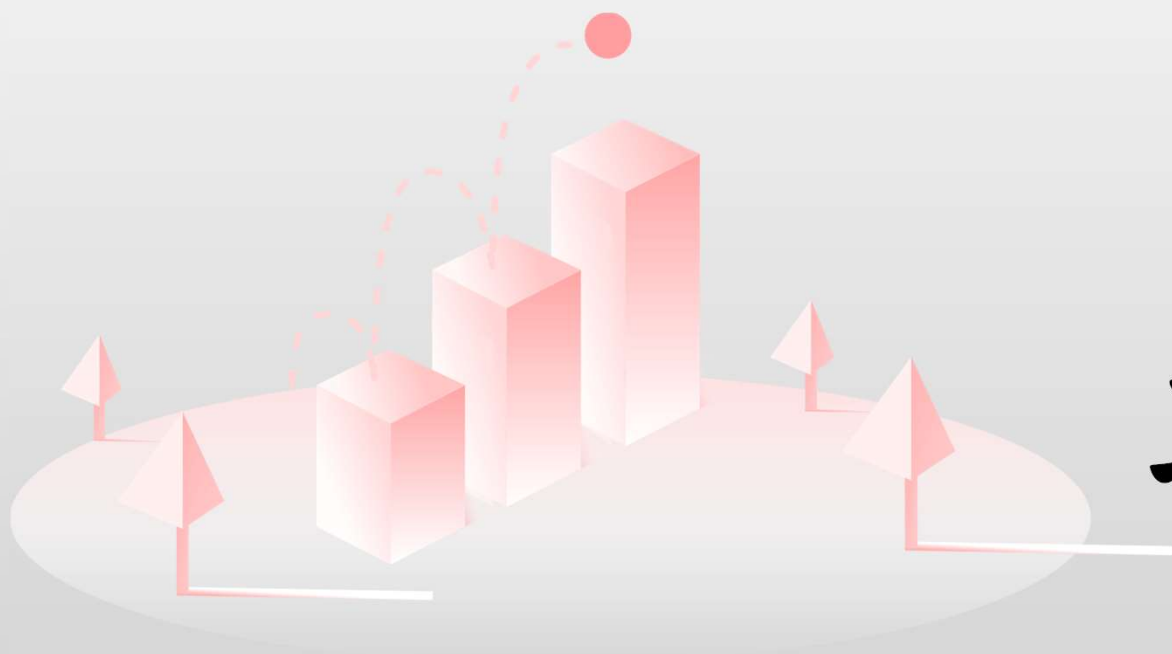
①程序读入灰度图像，得到图像矩阵img

②遍历图像矩阵img的每一个元素，统计每一个灰度值的出现次数，得到灰度值矩阵counts。

对于灰度图像，灰度值的范围为0到255，灰度值矩阵counts是256×1的矩阵。

③根据灰度值矩阵，绘制出相应的直方图。其结果如下所示。





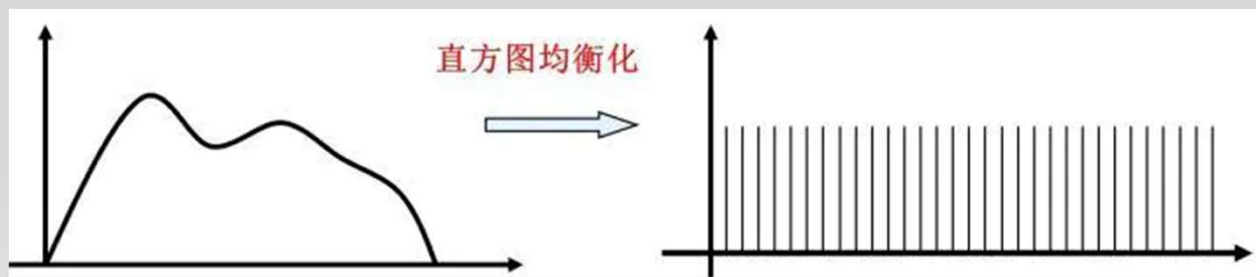
Part 2

直方图 均衡化

2.1 直方图均衡化：

直方图均衡化是一种简单有效的图像增强技术，通过**改变图像的直方图**来改变图像中**各像素的灰度**，主要用于增强动态范围偏小的图像的对比度。原始图像由于其灰度分布可能集中在较窄的区间，造成图像不够清晰。例如，过曝光图像的灰度级集中在高亮度范围内，而曝光不足将使图像灰度级集中在低亮度范围内。采用直方图均衡化，可以把原始图像的直方图变换为均匀分布（均衡）的形式，这样就增加了像素之间灰度值差别的动态范围，从而达到增强图像整体对比度的效果。

换言之，直方图均衡化的基本原理是：对在图像中像素个数多的灰度值（即对画面起主要作用的灰度值）进行展宽，而对像素个数少的灰度值（即对画面不起主要作用的灰度值）进行**归并**，从而增大对比度，使图像清晰，达到增强的目的。



2.2 直方图均衡化的步骤:

1 确定图像的灰度级

在实际情况下,如果我们的图像是彩色,需要将其转换为灰度图像,其中的灰度级一般是0-255。

2 计算原始直方图的概率

统计每一个灰度在原始图像上的像素所占总体的比例,记为 P_i

3 计算直方图概率的累加值 $S(i)$

直到最后一个灰度级,总和为1

由公式:

$$s_k = T(r_k) = \sum_{j=0}^k P_r(r_j) = \sum_{j=0}^k \frac{n_j}{N}$$

可以得到:

$$s_0 = T(r_0) = \sum_{j=0}^0 P_r(r_j)$$

$$s_1 = T(r_1) = \sum_{j=0}^1 P_r(r_j) = P_r(r_0) + P_r(r_1)$$

4 根据公式求取像素映射关系

(最大灰度级-最小灰度级)*累加概率+0.5后取整数

5 灰度映射,得到新的像素灰度

找到原图像和均衡化图像灰度的对应关系,对原图进行操作,将每个像素映射成新的像素

$$s_s(i) = \text{int}(\max(\text{pix}) - \min(\text{pix})) * S(i) + 0.5$$

2.3 直方图均衡化实验结果：

原始图像的直方图和直方图均衡化处理后的图像直方图显示结果如右图所示。可以看出，变换后的新直方图比原图像的直方图要平坦得多。理想情况下，经过直方图均衡化处理的图像直方图是十分均匀平坦的，但实际情况并非如此，和理论分析有所差异，这是由于图像在直方图均衡化处理过程中，灰度级作“近似简并”引起的结果。

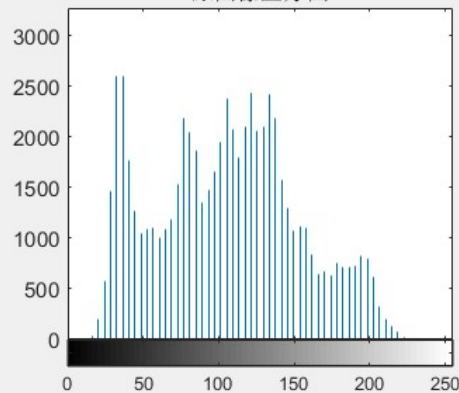
原图像



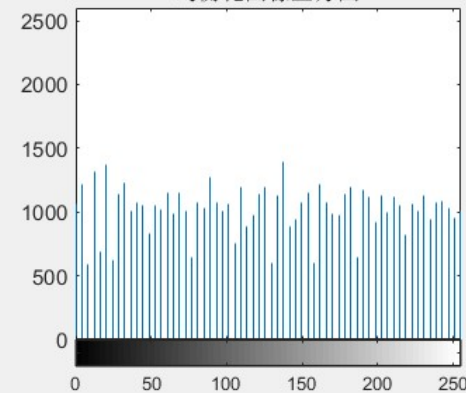
均衡化图像



原图像直方图



均衡化图像直方图





Part 3

直方图 规定化

3.1 直方图规定化/匹配:

直方图规定化, 也叫做直方图匹配, 用于将图像变换为某一特定的灰度分布, 也就是其目的的灰度直方图是已知的。这其实和均衡化很类似, 均衡化后的灰度直方图也是已知的, 是一个均匀分布的直方图; 而规定化后的直方图可以随意的指定, 也就是在执行规定化操作时, 首先要知道变换后的灰度直方图, 这样才能确定变换函数。规定化操作能够有目的的增强某个灰度区间, 相比于, 均衡化操作, 规定化多了一个输入, 但是其变换后的结果也更灵活。

在某些情况下, 并不一定需要具有均匀直方图的图像, 有时需要具有特定的直方图的图像, 以便能够增强图像中某些灰度级。直方图规定化方法就是针对上述思想提出来的。直方图规定化是使原图像灰度直方图变成规定形状的直方图而对图像作修正的增强方法。可见, 它是对直方图均衡化处理的一种有效的扩展, 直方图均衡化处理是直方图规定化的一个特例, 直方图的均衡化自动的确定了变换函数, 而直方图规定化的变换函数由其他方式决定。

3.2 直方图规定化/匹配的步骤:

- ①统计原图像的直方图。
- ②统计规定图像的直方图。
- ③计算两直方图各灰度的累积分布函数 CDF。

我们将第一张图像的CDF表示为 F_1 ，而第二张图像的CDF为 F_2 。因此，将 $F_1(x)$ 表示第一幅图像的强度为x的CDF值。

- ④根据两直方图差值建立灰度级的映射。

一旦计算出每个图像的CDF，就需要计算一个映射，该映射将第一个图像的一个强度转换为与第二个图像的强度分布一致。要做到这一点，对于第一幅图像中的每个强度 G_1 （在 $[0,255]$ 范围内），我们必须在第二幅图像（也在 $[0,255]$ 范围内）找到强度 G_2 。

有可能是，我们不会得到正是相等的情况下，因此我们需要做的是找到 $F_1(G_1)$ 和 $F_2(G_2)$ 之间的最小差值的绝对值。换句话说，对于一个映射 M ， G_1 为每个条目，我们必须找到一个强度 G_2 ：

$$M(G_1) = \arg \min_{G_2 \in [0, 255]} |F_1(G_1) - F_2(G_2)| \quad \forall G_1 \in [0, 255]$$

计算256个值的映射结果，即可建立新的灰度级映射。

⑤由映射生成每个像素新的灰度值，生成新的图像。

3.3 直方图规定化/匹配的实验结果：

原图像



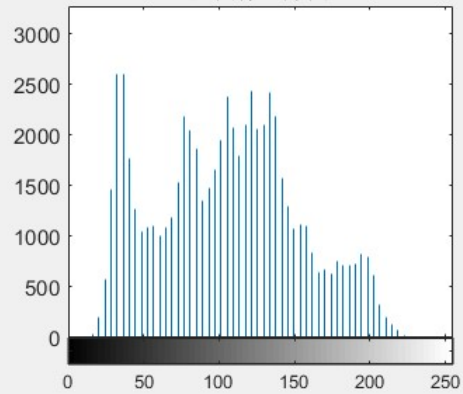
匹配图像



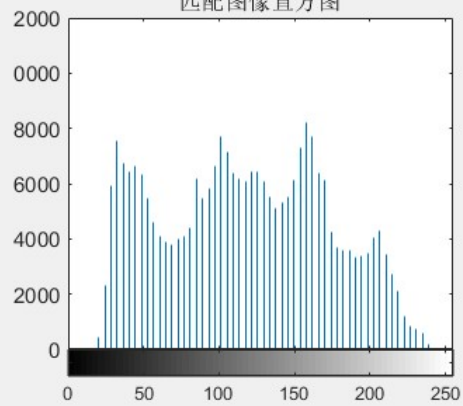
匹配后图像



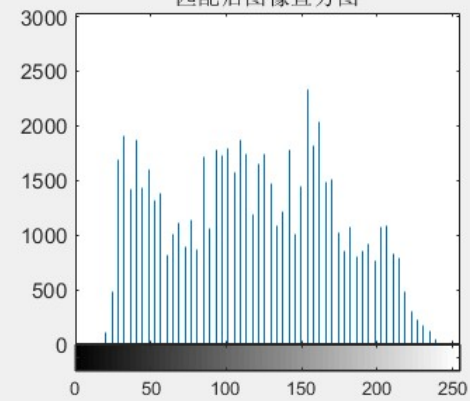
原图像直方图



匹配图像直方图



匹配后图像直方图





Part 4

程序设计

4.1 调用`matlab`函数完成直方图均衡化和规定化

```
G=imread('lena.bmp');
g1=imread('lady.bmp');
g2=imhist(g1);
z1=histeq(G);%直方图均衡化
z2=histeq(G,g2);%直方图规定化
figure,subplot(2,4,1);imshow(G);title('原图像');
subplot(2,4,2);imshow(z1);title('均衡化图像');
subplot(2,4,3);imshow(g1);title('匹配图像');
subplot(2,4,4);imshow(z2);title('匹配后图像');

subplot(2,4,5);imhist(G,64);title('原图像直方图');
subplot(2,4,6);imhist(z1,64);title('均衡化图像直方图');
subplot(2,4,7);imhist(g1,64);title('匹配图像直方图');
subplot(2,4,8);imhist(z2,64);title('匹配后图像直方图');
```

调用matlab函数实验结果

原图像



均衡化图像



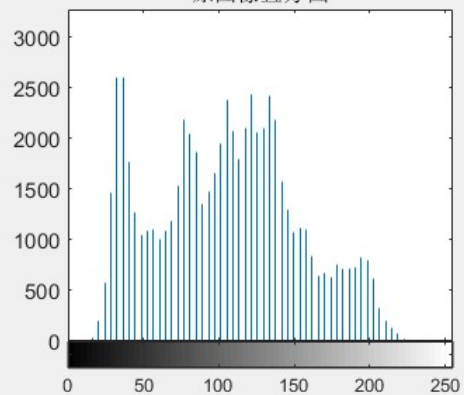
匹配图像



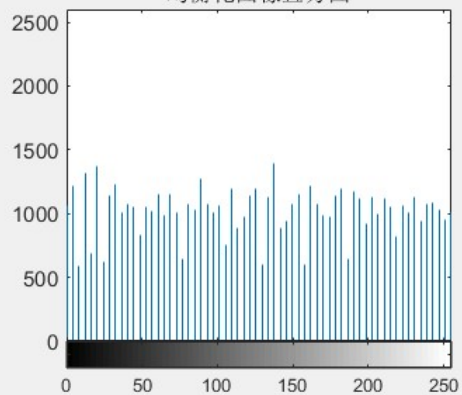
匹配后图像



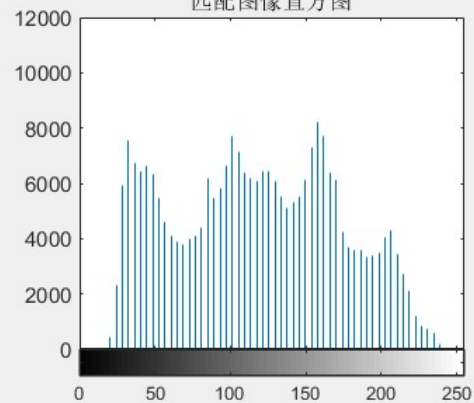
原图像直方图



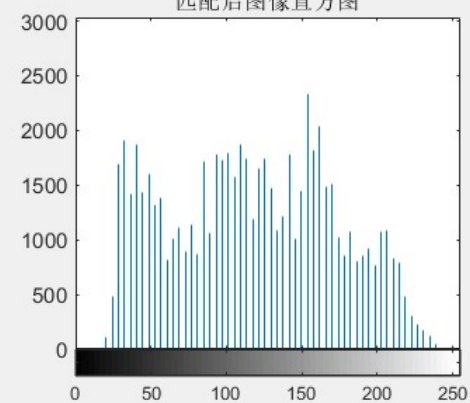
均衡化图像直方图



匹配图像直方图



匹配后图像直方图



4.2 调用 *Matlab* 关于直方图的相关函数完成直方图均衡化和规定化函数自编

%% 自编函数实现均衡化和规定化

%不使用histeq函数，使用imhist函数对图像直方图的各灰度值进行统计和绘制图像的直方图

%均衡化

img=imread('lena.bmp');

hist1=imhist(img); %统计图像直方图每个灰度值的个数

[h,w]=size(img);

new_img=zeros(h,w);

s=zeros(1,256);

s(1)=hist1(1);

for t=2:256

s(t)=s(t-1)+hist1(t); % 计算新的灰度值

end

for i=1:w

for j=1:h

new_img(j,i)=s(img(j,i)+1)/(w*h); % 生成新图像

end

end

new_img=im2uint8(new_img); %上一步生成新图像运算过程中是double类型，将double转换为无符号8位整型

figure,subplot(241),imshow(img);

title('原图')

subplot(242),imshow(new_img);

title('自编函数均衡化后')

subplot(245),imhist(img,64);

title('原图直方图')

subplot(246),imhist(new_img,64);

title('均衡化后的直方图')

4.2 调用 *Matlab* 关于直方图的相关函数完成直方图均衡化和规定化函数自编

%% 自编函数实现均衡化和规定化

%不使用histeq函数，使用imhist函数对图像直方图的各灰度值进行统计和绘制图像的直方图

%均衡化

img=imread('lena.bmp');

hist1=imhist(img); %统计图像直方图每个灰度值的个数

[h,w]=size(img);

new_img=zeros(h,w);

s=zeros(1,256);

s(1)=hist1(1);

for t=2:256

s(t)=s(t-1)+hist1(t); % 计算新的灰度值

end

for i=1:w

for j=1:h

new_img(j,i)=s(img(j,i)+1)/(w*h); % 生成新图像

end

end

new_img=im2uint8(new_img); %上一步生成新图像运算过程中是double类型，将double转换为无符号8位整型

figure,subplot(241),imshow(img);

title('原图')

subplot(242),imshow(new_img);

title('自编函数均衡化后')

subplot(245),imhist(img,64);

title('原图直方图')

subplot(246),imhist(new_img,64);

title('均衡化后的直方图')

4.2 调用 *Matlab* 关于直方图的相关函数完成直方图均衡化和规定化函数自编

%规定化

img_match=imread('lady.bmp'); %匹配图像

subplot(243),imshow(img_match)

title('匹配图像')

subplot(247),imhist(img_match,64)

title('匹配图像的直方图')

hist2=imhist(img_match);

%建立映射

cdf=cumsum(hist1) / numel(img);

cdfRef=cumsum(hist2) / numel(img_match);

M=zeros(1,256,'uint8');

for k = 1 : 256

[tmp,ind] = min(abs(cdf(k) - cdfRef)); %找到两图像之间每一个灰度的对应关系

M(k) = ind-1;

end

%利用映射生成新的图像

imMatch = M(img+1);

subplot(244),imshow(imMatch)

title('自编直方图匹配后')

subplot(248),imhist(imMatch,64);

title('匹配后的直方图')

调用 *Matlab* 关于直方图的相关函数实验结果

原图



自编函数均衡化后



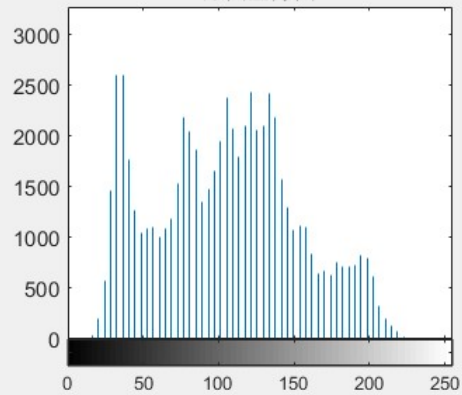
匹配图像



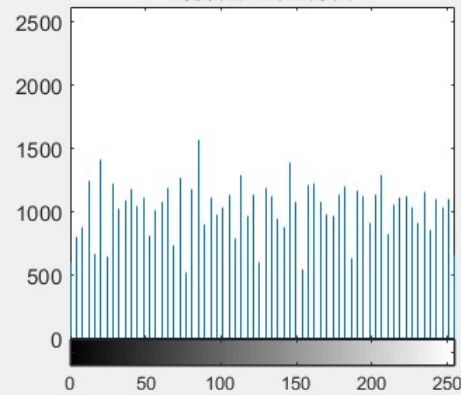
自编直方图匹配后



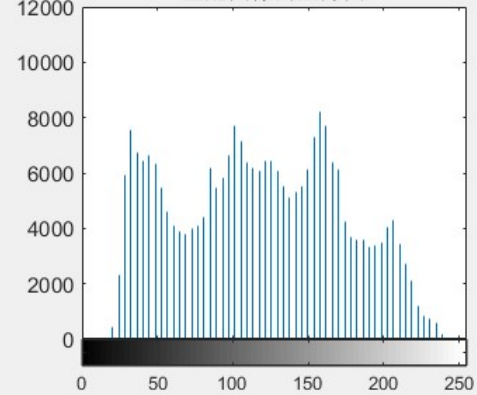
原图直方图



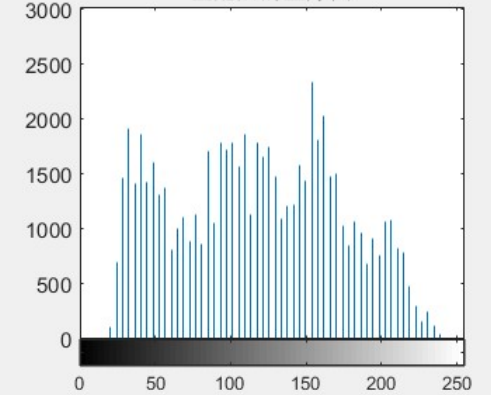
均衡化后的直方图



匹配图像的直方图



匹配后的直方图



4.3 在Matlab平台下，不利用直方图相关函数自编函数实现直方图均衡化和规定化算法

```
%Hist.m
% 对图像直方图进行统计，其中counts是每个灰度值的个数，x代表灰度值的范围，默认为1:256
% 输入的a是bin的个数，类似imhist函数的用法
function [counts,x]=Hist(img,a)
if(~exist('a','var'))
    a=256;      %调用该函数时若未给a赋值，则a的默认值为256
end
x=1:a;
img=double(img);    %返回的counts矩阵也是double类型
[m,n]=size(img);
counts=zeros(a,1); %灰度数矩阵
for i=1:m
    for j=1:n
        if(a==256)
            counts(img(i,j)+1)=counts(img(i,j)+1)+1;
        else
            t=floor(img(i,j)/(256/a))+1;
            counts(t)=counts(t)+1;
        end
    end
end
end
end
```

不使用histeq函数和imhist函数，自编Hist函数对图像直方图的各灰度值进行统计,使用bar函数绘制直方图。主体程序部分与上一步算法基本一致，仅将imhist替换为Hist，并使用bar函数绘图。

自编函数函数实验结果

原图



自编函数均衡化后



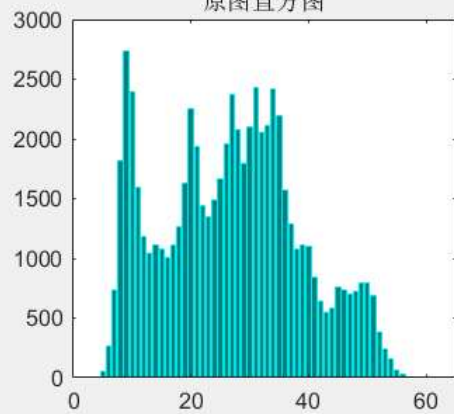
匹配图像



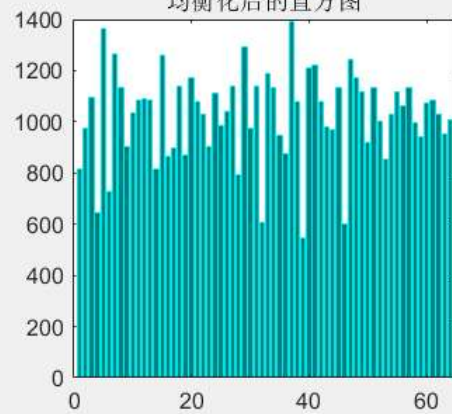
自编函数匹配后



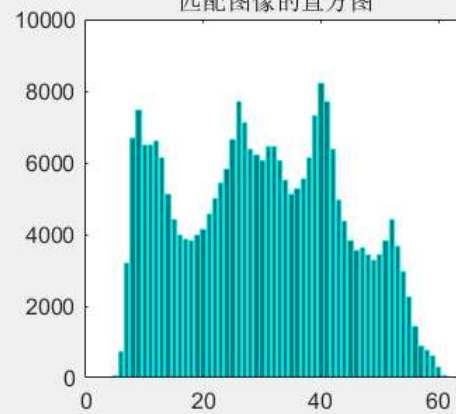
原图直方图



均衡化后的直方图



匹配图像的直方图



匹配后的直方图

