

数字图像处理实验报告

学院： 计算机学院

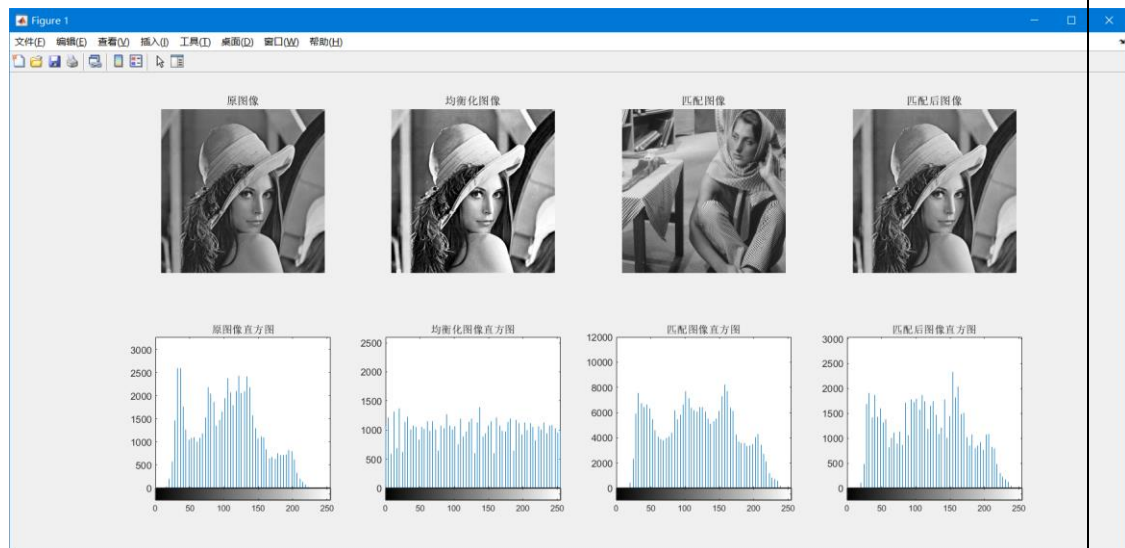
专业： 计算机科学与技术

姓名	///	学号	19//	班级	计科
实验时间	2021	指导教师	//	成绩	
实验目的	掌握灰度变换和图像直方图的概念以及计算方法，了解直方图均衡化和规定化的原理，学会利用 matlab 计算图像的直方图和实现直方图均衡化和规定化。				
实验内容	(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； (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				
实验平台	Matlab R2019a				
实验程序及结果	1.1 内容一的程序代码： <pre> clear; close all; clc; %% 使用matlab自带函数实现直方图的均衡化和规定化 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 set(gcf,'Position',[100,100,1400,600]); 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');</pre>				

```
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');
```

1.2 内容一的程序结果



1.3 内容一的程序思路

读取原始图像→调用matlab提供的histeq、imhist函数

2.1 内容二的程序代码

```
clear;
close all;
clc;
%% 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
```

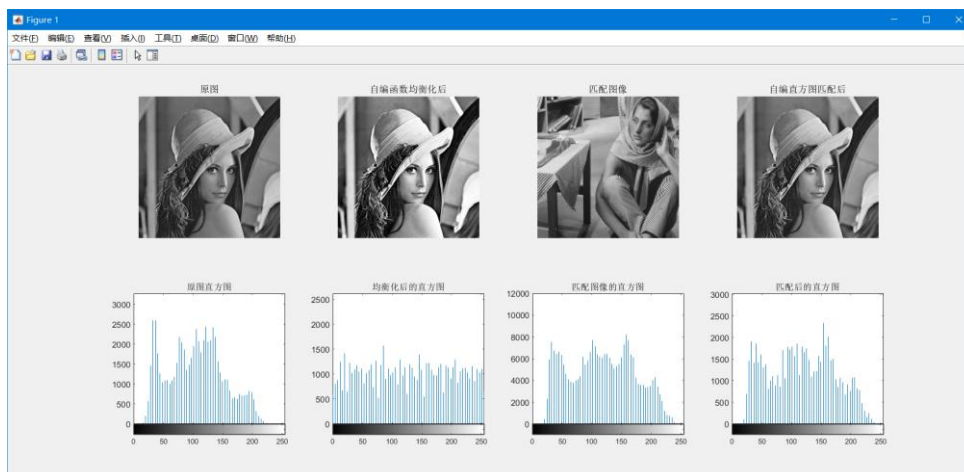
```

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
set(figure(1),'Position',[100,100,1400,600])
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')

%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')

```

2.2 内容二的程序结果



2.3 内容二的程序思路

2.3.1 直方图均衡化

- (1) 统计原始图像直方图。
- (2) 计算新的灰度级。
- (3) 修正灰度级。
- (4) 计算新的直方图。
- (5) 由处理后的新灰度代替处理前的灰度，生成新图像。

2.3.2 直方图规定化（直方图匹配）

- (1) 统计原始图像和匹配图像的直方图。
- (2) 统计两图像的各灰度的概率密度。
- (3) 由概率密度分布函数，建立并计算原图像灰度与匹配图像灰度的映射，找到两图像之间每一个灰度的对应关系。
- (4) 利用映射生成新的图像。

3.1 内容三的程序代码

```
clear;
close all;
clc;
%% 自编程函数实现均衡化和规定化
%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

%均衡化

```
img=imread('lena.bmp');
hist1=Hist(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
set(figure(1),'Position',[100,100,1400,600]);
subplot(241),imshow(img);
title('原图')
subplot(242),imshow(new_img);
title('自编函数均衡化后')
% 绘制直方图，类似imhist，使用64个bin
subplot(245),bar(Hist(img,64),'FaceColor',[0.5.5],'EdgeColor',[0.9.9],'LineWidth',0.1)
title('原图直方图')
subplot(246),bar(Hist(new_img,64),'FaceColor',[0.5.5],'EdgeColor',[0.9.9],'LineWidth',0.1)
title('均衡化后的直方图')

%规定化
img_match=imread('lady.bmp');    %匹配图像
subplot(243),imshow(img_match)
title('匹配图像')
hist2=Hist(img_match);
% 绘制匹配图像的直方图
subplot(247),bar(Hist(img_match,64),'FaceColor',[0.5.5],'EdgeColor',[0.9.9],'LineWidth',0.1)
title('匹配图像的直方图')

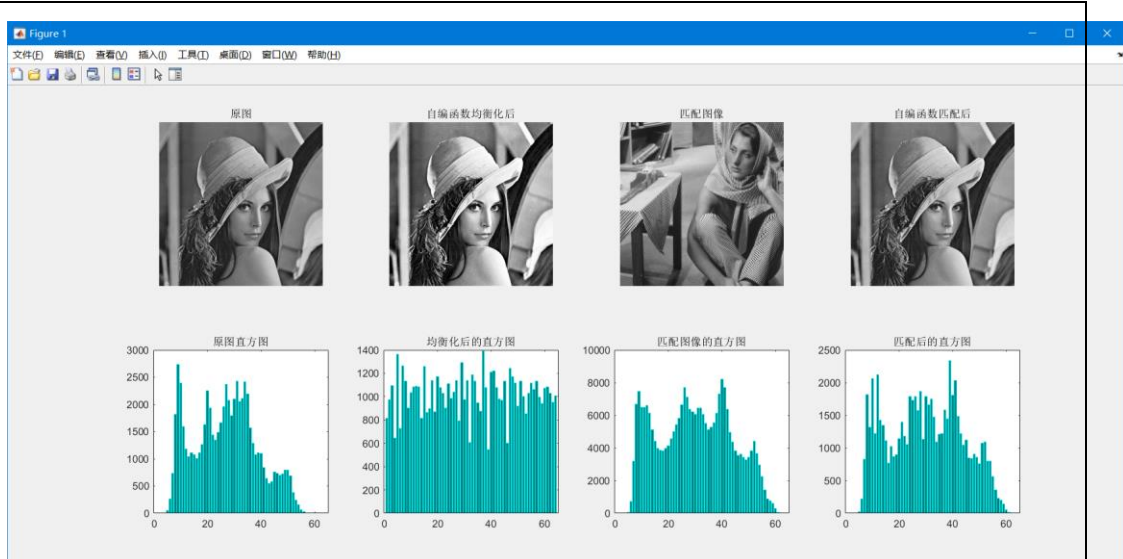
cdf=cumsum(hist1) / numel(img); %计算累积分布函数
cdf2=cumsum(hist2) / numel(img_match);
M=zeros(1,256,'uint8'); %建立映射
for k=1:256
```

```

[~,ind]=min(abs(cdf(k) - cdf2)); %找到两图像之间每一个灰度的对应关系
M(k)=ind-1;
end
imMatch = M(img+1);
subplot(244),imshow(imMatch)
title('自编函数匹配后')
subplot(248),bar(Hist(imMatch,64),'FaceColor',[0.5 .5],'EdgeColor',[0.9 .9],'LineWidth',0.1)
title('匹配后的直方图')
%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

```

3.2 内容三的程序结果



3.2 内容三的程序思路

3.3.1 直方图均衡化

- (1) 统计原始图像直方图。
- (2) 计算新的灰度级。
- (3) 修正灰度级。
- (4) 计算新的直方图。
- (5) 由处理后的新灰度代替处理前的灰度，生成新图像。

3.3.2 直方图规定化（直方图匹配）

- (1) 统计原始图像和匹配图像的直方图。
- (2) 统计两图像的各灰度的概率密度。
- (3) 由概率密度分布函数，建立并计算原图像灰度与匹配图像灰度的映射，找到两图像之间每一个灰度的对应关系。
- (4) 利用映射生成新的图像。

3.3.3 自编直方图统计函数 Hist.m

- (1) 参考matlab提供的直方图统计函数`imhist`的用法，设计Hist函数的输入为一个图像矩阵或者一个图像矩阵和数字n。n表示统计的直方图中bin的个数，由于本程序主要处理的是灰度图像，因此n的默认值为256。输出是n*1的矩阵。
- (2) 遍历图像矩阵每一个灰度值，统计每一个灰度值的频数。若输入的bin的数量少于256，则对图像的灰度级数进行压缩处理。

实验总结	<p>通过这次实验，我学到了对于灰度图像的均衡化和规定化的相关知识，认识到了对于灰度图像的多种灰度处理方式并尝试编写了相关程序代码，了解到了对于灰度图像处理的算法实现。同时，这次实验加强了我对于 matlab 的各类函数的掌握程度和认知水平，从实验操作中我认识到了 Matlab 对于图像处理的强大优势，初步了解到了图像处理的思想。在实验初期，由于是第一次接触图像处理，感觉无从下手。在经过小组讨论和查阅 Matlab 和图像处理相关的资料后，逐步有了思路。在实验过程中，也遇到一些关于图像矩阵处理过程数据类型的问题，例如无符号整型 8 位数 uint8 和双精度浮点数 double 两种数据类型在图像处理时的差异。通过小组讨论和反复尝试，最终解决了问题。</p> <p>总的来说，这次虽然是第一次数学图像处理的实验，但在实验过程中，通过反复尝试和查阅相关资料，我学到了很多。数字图像处理是一门理论和实际联系非常紧密的学科，在以后的学习中，一定要将理论学习和实践结合起来，才能对它有更深刻的认识。</p>
指导教师意见	<div></div> <div style="text-align: right;"> 签名： 年 月 日 </div>