

**计算机与信息学院实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验课程： | 数字图像处理实验 | | | | |
| 实验编号： | 实验2 | | | | |
| 实验名称： | 灰度图像的直方图均衡化 | | | | |
| 实验人员： | 学号 | 18111207248 | | | |
| 姓名 | 吴钰 | | | |
| 班级 | 计算机科学与技术（18级创新班） | | | |
| 实验日期： | 2021.5.14 | | | | |
| 实验室： | 2060404 | | | | |
|  |  | | | | |
| 实验评价： |  | | | | |
| 实验成绩： | |  | 评价日期： |  |
|  | 指导教师： | |  | | |

# 灰度图像的直方图均衡化

一、实验目的

1．掌握基于图像统计特征的分析方法；

2．理解直方图均衡化图像增强的原理，熟练使用直方图方法对图像进行增强。

二、实验要求

1．在MATLAB环境下实现灰度直方图均衡化算法；

2．应用直方图均衡算法增强灰度图像。

三、实验内容

1．图像统计特征（频数、均值、密度等）的实现；

2．实现灰度直方图均衡化算法；

3．应用直方图均衡算法增强灰度图像。

设有一幅包含16个灰度级的数字图像，其各灰度级的像素个数如下表所示（对应的图像为mytest16.bmp），要求对其进行直方图均衡化，求出灰度变换关系和变换后的直方图，并应用该算法实现对灰度图像的对比度增强。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 灰度级（***i***） | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 像素个数（***ni***） | 3390 | 0 | 0 | 0 | 0 | 0 | 0 | 16432 | 18601 | 0 | 0 | 0 | 0 | 0 | 0 | 1577 |

四、实验结果及分析

实验素材见实验二附件材料。

1．直方图均衡化算法源码

填写代码

**Hmain.m**

k = zeros(1,16);

k\_change = zeros(1,16);

k\_change\_num = zeros(1,16);

%累计直方图

p = zeros(1,16);

p\_sum = zeros(1,16);

p\_change = zeros(1,16);

Img\_Ori = imread('D:\大三（下）专业课\数字图像处理\Test2\image\mytest16.bmp');

Ima\_Chg = Img\_Ori;

%直方图

for i = 1 : 200

for j = 1:200

k(1,Img\_Ori(i,j)+1)=k(1,Img\_Ori(i,j)+1)+1;

end

end

%累积直方图

x = sum(k);

for i = 1 : 16

p(1,i) = k(1,i)/x;

if i==1

p\_sum(1,i) = p(1,i);

else

p\_sum(1,i) = p(1,i)+p\_sum(1,i-1);

end

end

% 变换后的灰度值

for i = 1 : 16

k\_change(1,i) = round(14\*p\_sum(1,i));

end

disp('变换后的灰度值');

disp(k\_change);

%变换后各灰度级像素个数

for i = 1 : 16

k\_change\_num(k\_change(1,i)) = k\_change\_num(k\_change(1,i))+1;

end

disp('变换后各灰度级像素个数');

disp(k\_change\_num);

% 计算变换后图像的直方图

x2\_sum = sum(k\_change\_num);

for i = 1 : 16

p\_change(1,i) = k\_change\_num(1,i)/x2\_sum;

end

%输出前后图像变换及直方图图像

figure;

subplot(222);

imshow(mat2gray(Img\_Ori));

title('原始图像');

subplot(221);

HEImage\_Show (k);

title('原始图像直方图');

subplot(223);

HEImage\_Show (p\_change);

title('均衡化后的图像直方图');

% 直方图均衡化实现灰度图像增强

for i = 1:200

for j= 1:200

Img\_Cha(i,j) = k\_change(Img\_Ori(i,j)+1);

end

end

subplot(224);

imshow(mat2gray(Img\_Cha));

title('实现灰度图像增强');

**HEImage\_Show.m**

function HEImage\_Show (Image)

%{

Function:

HEImage\_Show: to shouw the histogram of image.

Parameters:

Image: The input image;

Return value:

NULL

%}

figure;

x = 0:1:size(Image,2)-1;

bar(x,Image,0.1);

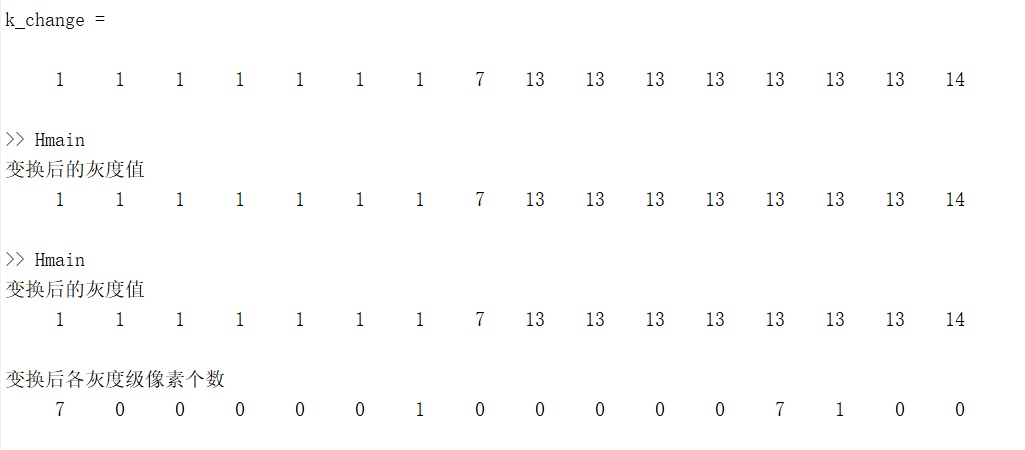
for i=0:length(Image)-1

text(i,Image(i+1),num2str(Image(i+1)),'VerticalAlignment','bottom','HorizontalAlignment','center','FontSize',9,'color','r','FontWeight','bold');

end

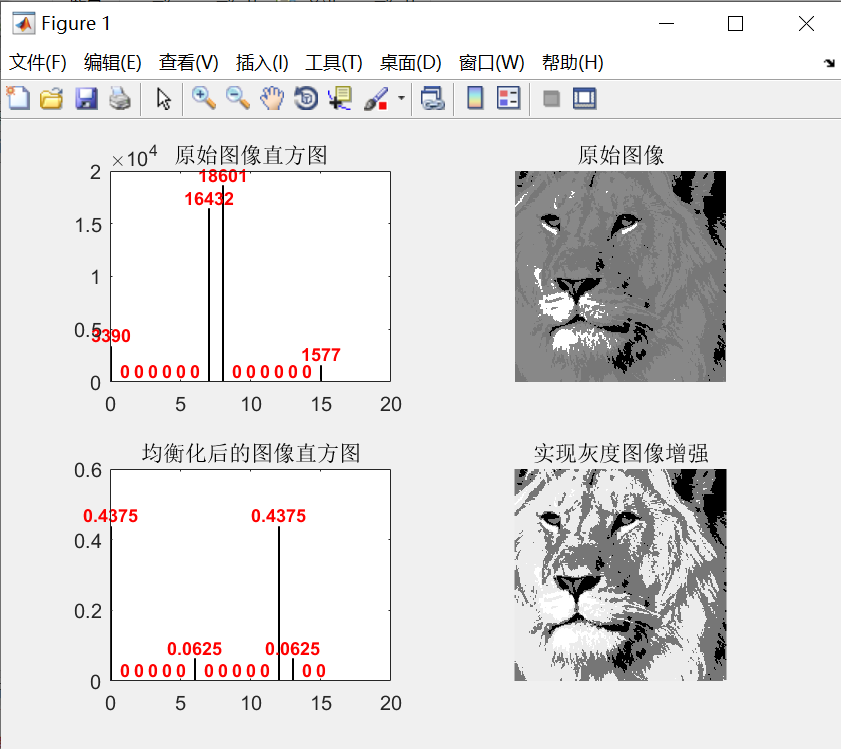
2．16位灰度级数字图像mytest16.bmp的灰度均衡化过程结果（填入下表）

代码输出结果：

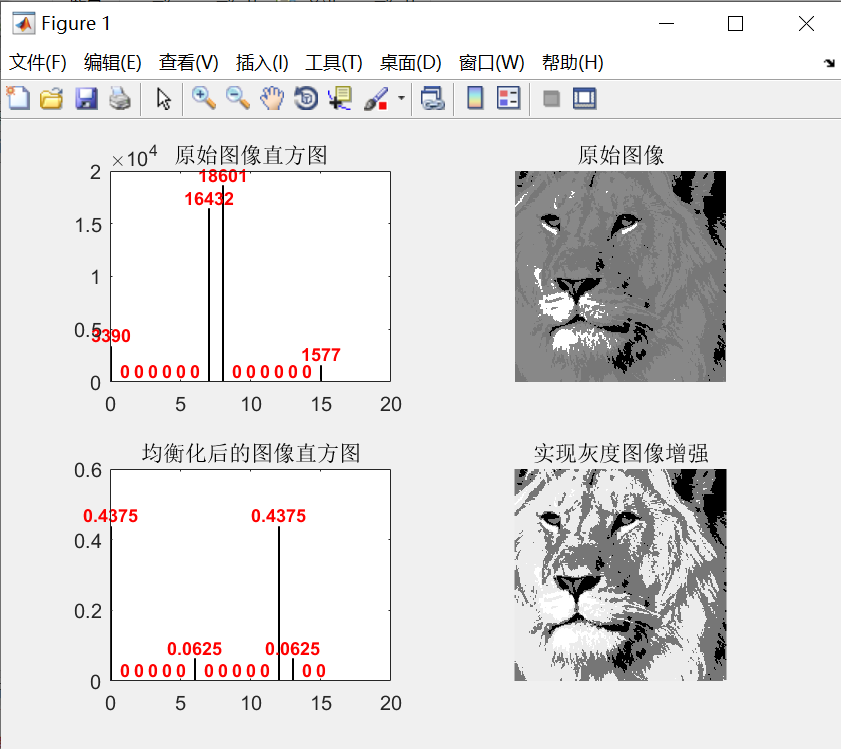


|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 步骤 | 计算方法 | 计算结果 | | | | | | | | | | | | | | | |
| 1 | 灰度级（*i*） | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | 各灰度级像素个数（*ni*） | 3390 | 0 | 0 | 0 | 0 | 0 | 0 | 16432 | 18601 | 0 | 0 | 0 | 0 | 0 | 0 | 1577 |
| 3 | 原始图像灰度直方图（） | 0.08475 | 0 | 0 | 0 | 0 | 0 | 0 | 0.41080 | 0.465025 | 0 | 0 | 0 | 0 | 0 | 0 | 0.039425 |
| 4 | 累积直方图（） | 0.08475 | 同 |  |  |  |  |  | 0.49555 | 0.960575 |  |  |  |  |  |  | 1 |
| 5 | 计算机变换后的灰度值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 14 |
| 6 | 确定灰度变换关系： | 0,1,2,3,4,5,6->1 | | | | | | | 7->7 | 8,9,10,11,12,13,14->13 | | | | | | | 15->14 |
| 7 | 变换后各灰度级像素个数（*nj*） | 7 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 7 | 1 | 0 | 0 |
| 8 | 计算变换后图像的直方图 | 0.4375 |  |  |  |  |  | 0.0625 |  |  |  |  |  | 0.4375 | 0.625 |  |  |

3．给出原始图像直方图和均衡化后的图像直方图



4．给出运用编写的直方图均衡化算法实现灰度图像增强的应用实例



代码：

% 直方图均衡化实现灰度图像增强

for i = 1:200

for j= 1:200

Img\_Cha(i,j) = k\_change(Img\_Ori(i,j)+1);

end

end

subplot(224);

imshow(mat2gray(Img\_Cha));

title('实现灰度图像增强');

subplot(224);

imshow(mat2gray(Img\_Cha));

title('实现灰度图像增强');