

第五章课堂练习1(第2次课堂作业)

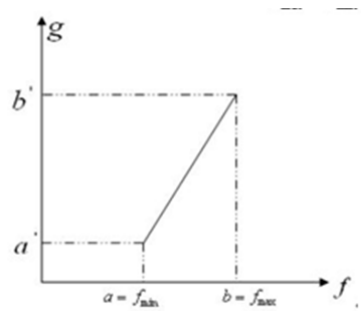
计创18 181002222 连月菡

-目录-

第五章课堂练习1(第2次课堂作业)

- 1.线性变换
- 2.直方图
- 3.自制图像
 - (1)图中有线条
 - (2)图中有正方形
 - (3)任意

1.线性变换



5	0	3	2	1
7	2	5	4	6
0	0	0	1	1
2	2	0	1	1
1	1	0	0	1

(1) 求出线性变换的函数; (a=5,b=10,a'=8,b'=20)

设线性变换函数 $f(x) = kx + t$

由 $a = 5, b = 10, a' = 8, b' = 20$ 得方程组：

$$\begin{aligned} 5k + t &= 8 \\ 10k + t &= 20 \end{aligned}$$

解得： $k = \frac{12}{5}$

$$t = -4$$

线性变换函数为 $f(x) = \frac{12}{5}x - 4$

(2) 对图像进行线性变换；（注意范围！）

在MatlabR2016b环境下,输入以下代码

```

1 a=[5,0,3,2,1;7,2,5,4,6;0,0,0,1,1;2,2,0,1,1;1,1,0,0,1]%输入5*5矩阵
2 fun=12/5*a-4/255 %计算出的线性变换函数
3 imshow(fun) %展示线性变换后的图像

```



a =					fun =				
线性变换前					线性变换后				
5	0	3	2	1	11.9843	-0.0157	7.1843	4.7843	2.3843
7	2	5	4	6	16.7843	4.7843	11.9843	9.5843	14.3843
0	0	0	1	1	-0.0157	-0.0157	-0.0157	2.3843	2.3843
2	2	0	1	1	4.7843	4.7843	-0.0157	2.3843	2.3843
1	1	0	0	1	2.3843	2.3843	-0.0157	-0.0157	2.3843

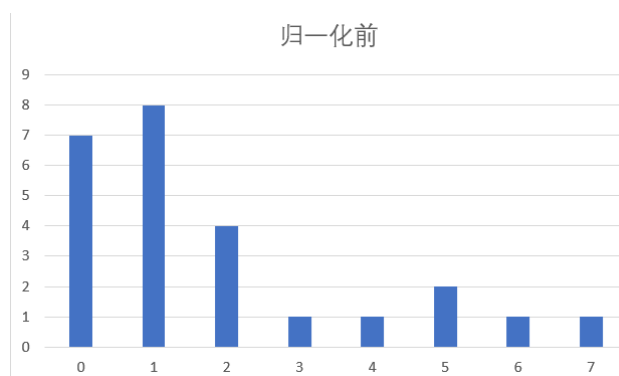
2.直方图

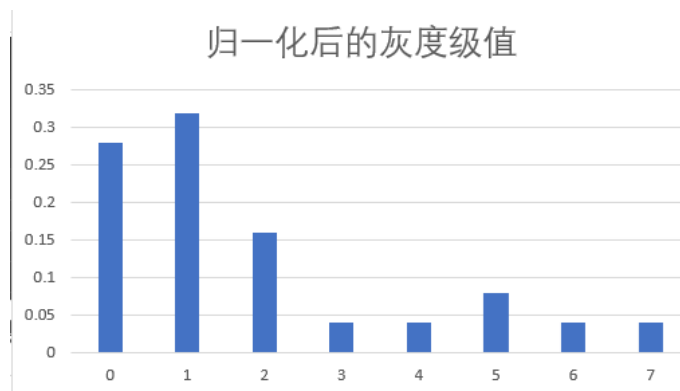
一个5*5的8灰度图像。

5	0	3	2	1
7	2	5	4	6
0	0	0	1	1
2	2	0	1	1
1	1	0	0	1

要求：

1.画出归一化的直方图；





2.进行直方图均衡化运算；

$$s_i = T(r_i) = 7 \sum_{j=0}^i p_r(r_j)$$

代入公式,得 s₀~s₇依次**2, 4.2, 5.32, 5.6, 5.88, 6.44, 6.72, 7。**

3.给出直方图均衡化后的图像；

6.44	2	5.6	5.32	4.2
7	5.32	6.44	5.88	6.72
2	2	2	4.2	4.2
5.32	5.32	2	4.2	4.2
4.2	4.2	2	2	4.2

图片:

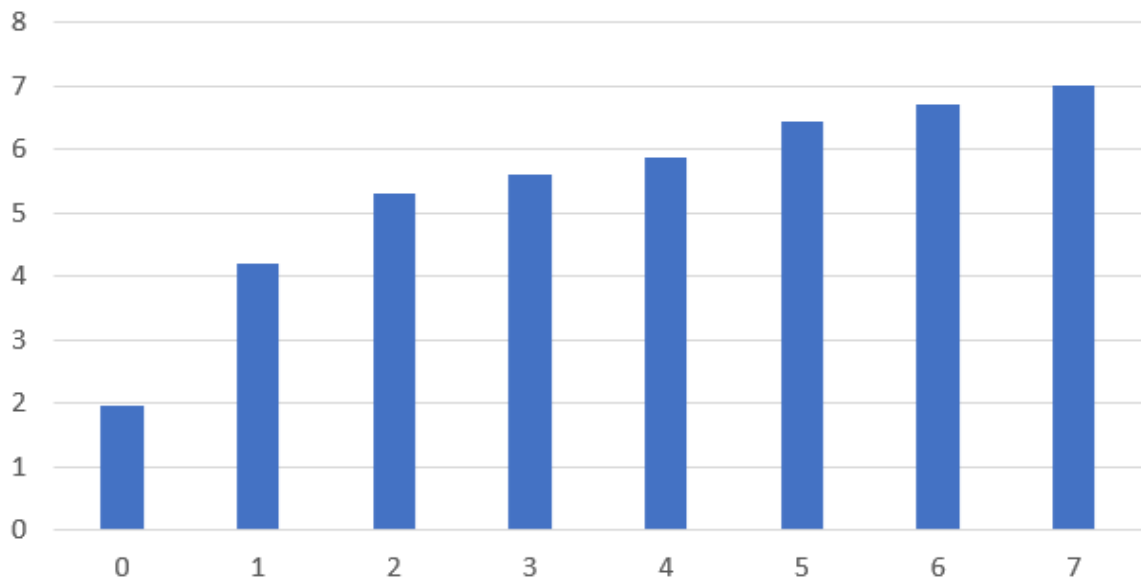
在MatlabR2016b环境下,输入以下代码:

```

1  a=
   [6.44,2,5.6,5.32,4.2;7,5.32,6.44,5.88,6.72;2,2,2,4.2,4.2;5.32,5.32,2,4.2,4.2
   ;4.2,4.2,2,2,4.2]
2  imwrite(a,'homework.bmp') %生成图像,即可得到如上所示的图片
  
```

4.画出直方图均衡化后图像的直方图；

均衡化后



5.对问题进行分析。

我认为,图像均衡化处理的公式, 它是对从0到k的求和。如果一副非常暗的图片, 它的灰度直方图基本集中在了低灰度区, 也就是说它的低灰度的值的频率很高。现在把它均值化处理, 根据公式, 对低灰度的区域求概率相加, 本来频率已经很高了, 累计函数一向加便会更大, 越往后加, 灰度级都很大, 对应的, 灰度级大了。所以直方图就会显示在高灰度区, 相应的图像就会变得很亮。

3.自制图像

自制三幅5*5的图像,灰度级为0~7,要求:

- (1)图中有线条;
- (2)图中有正方形;
- (3)任意;

用sobel算子分别处理,并解释处理的结果.

边界像素的处理: 对原图像进行用0填充.

(1)图中有线条

```
1 b=[
2 5,5,5,5,5;
3 5,5,5,5,5;
4 0,0,0,0,0;
5 5,5,5,5,5;
6 0,0,0,0,0]%像素矩阵
7 imshow(b) %展示图像
```

=

```

2 f=[5,5,5,5,5;5,5,5,5,5;0,0,0,0,0;5,5,5,5,5;0,0,0,0,0];
3 f_s=zeros(5,5);
4 for i=2:4
5     for j=2:4
6         f_s(i,j)=abs(f(i-1,j-1)+2*f(i,j-1)+f(i+1,j-1)-f(i-
7         1,j+1)-2*f(i,j+1)-f(i+1,j+1))+abs(f(i+1,j-1)+2*f(i+1,j)+f(i+1,j+1)-f(i-1,j-
8         1)-2*f(i-1,j)-f(i-1,j+1));
9         if f_s(i,j)<0
10            f_s(i,j)=0;%用0填充
11        else
12            f_s(i,j)=7;
13        end
14    end
15 end
16 imshow(f_s)

```



f_s =

0	0	0	0	0
0	7	7	7	0
0	7	7	7	0
0	7	7	7	0
0	0	0	0	0

(2)图中有正方形

在MatlabR2016b环境下,输入以下代码:

```

1 a=[
2 5,5,5,5,5;
3 5,5,5,5,5;
4 0,0,0,5,5;
5 0,5,0,5,5;
6 0,0,0,5,5]%像素矩阵
7 imshow(a) %展示图像

```



```

1 %用sobel算子处理
2 f=[5,5,5,5,5;5,5,5,5,5;0,0,0,5,5;0,5,0,5,5;0,0,0,5,5];
3 f_s=zeros(5,5);
4 for i=2:4
5     for j=2:4
6         f_s(i,j)=abs(f(i-1,j-1)+2*f(i,j-1)+f(i+1,j-1)-f(i-
7         1,j+1)-2*f(i,j+1)-f(i+1,j+1))+abs(f(i+1,j-1)+2*f(i+1,j)+f(i+1,j+1)-f(i-1,j-
8         1)-2*f(i-1,j)-f(i-1,j+1));
9         if f_s(i,j)<0
10            f_s(i,j)=0;%用0填充
11        else
12            f_s(i,j)=7;
13        end
14    end
15 end
16 imshow(f_s)

```

```

12     end
13 end
14 imshow(f_s)

```



```

f_s =

     0     0     0     0     0
     0     7     7     7     0
     0     7     7     7     0
     0     7     7     7     0
     0     0     0     0     0

```

(3)任意

```

1  c=[
2  2,5,0,3,1;
3  1,4,6,3,5;
4  0,0,5,0,0;
5  0,5,5,0,5;
6  0,1,5,7,2]%像素矩阵
7  imshow(c) %展示图像

```



```

1  %用sobel算子处理
2  f=[2,5,0,3,1;1,4,6,3,5;0,0,5,0,0;0,5,5,0,5; 0,1,5,7,2];
3  f_s=zeros(5,5);
4  for i=2:4
5      for j=2:4
6          f_s(i,j)=abs(f(i-1,j-1)+2*f(i,j-1)+f(i+1,j-1)-f(i-
7          1,j+1)-2*f(i,j+1)-f(i+1,j+1))+abs(f(i+1,j-1)+2*f(i+1,j)+f(i+1,j+1)-f(i-1,j-
8          1)-2*f(i-1,j)-f(i-1,j+1));
9          if f_s(i,j)<0
10             f_s(i,j)=0;%用0填充
11         else
12             f_s(i,j)=7;
13         end
14     end
15 end
16 imshow(f_s)

```



```
f_s =
```

```
0  0  0  0  0
0  7  7  7  0
0  7  7  7  0
0  7  7  7  0
0  0  0  0  0
```

Sobel算子检测方法对灰度渐变和噪声较多的图像处理效果较好，sobel算子对边缘定位不是很准确，图像的边缘不止一个像素,所以得出的结果都是一个正方形。