

第5章课堂练习2（第3次课堂练习）

181002222_连月菡

-目录-

第5章课堂练习2（第3次课堂练习）

题目描述

练习答案

- 1.均值滤波
- 2.中值滤波
- 3.高斯滤波 (选做)
- 4.对均值滤波进行编程实现 (选做)

题目描述

一个5*5的8灰度图像，用3*3的模板分别进行以下操作，给出操作后的图像数据。

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

(边界上的数据处理: 填充0)

- 1.均值滤波
- 2.中值滤波
- 3.高斯滤波（选做）
- 4.对均值滤波进行编程实现（选做）

练习答案

1.均值滤波

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

 \times

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

 $=$

2.44

将该灰度图像,使用3*3模板,进行均值滤波处理,按照如上图所示的步骤,一一计算。得到**均值滤波的结果**。

	2.44	1.89	2.56	
	2.00	1.67	2.11	
	0.67	0.56	0.56	

2.中值滤波

在MatlabR2016b环境中,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]
2 b=medfilt2(a,[3,3]) %矩阵b即为中值滤波后的图像数据
```

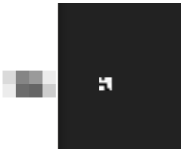
a =	b =	中值滤波的结果
5 0 3 2 1	0 2 2 2 0	
7 2 5 4 6	0 2 2 2 1	
0 0 0 1 1	0 2 1 1 1	
2 2 0 1 1	0 0 0 1 1	
1 1 0 0 1	0 0 0 0 0	

3.高斯滤波（选做）

在MatlabR2016b环境中,输入以下代码。

```
1 w=fspecial('gaussian',[3,3],1); %高斯滤波
2 g=imfilter(a,w,'replicate');%滤波器
```

a =	g =
5 0 3 2 1	4.1778 2.7407 2.4519 2.7735 2.4191
7 2 5 4 6	3.5386 2.4954 2.3608 2.9261 3.0867
0 0 0 1 1	2.0910 1.5470 1.4191 1.8486 2.2201
2 2 0 1 1	1.1778 0.8550 0.5218 0.6021 0.9249
1 1 0 0 1	1.2741 0.9249 0.4243 0.3979 0.8010



```
1  imshow(a)%滤波前图像
2  imshow(g)%高斯滤波后图像
3  %左侧为g(滤波后)  右侧为a(滤波前)
```

即5*5矩阵 g中的数值,为高斯滤波后的图像数据。

4.对均值滤波进行编程实现（选做）

```
1  #include<iostream>
2  #include<windows.h>
3  #include<fstream>
4  #include<cstdlib>
5  #include<cstdio>
6  #include<cmath>
7  #include<iomanip>
8
9  using namespace std;
10 unsigned char* pBmpBuf;//读入图像数据的指针
11
12 int bmpwidth;//图像的宽
13 int bmpHeight;//图像的高
14 RGBQUAD* pColorTable;//颜色表指针
15 int biBitCount;//图像类型，每像素位数
16 //给定一个图像位图数据、宽、高、颜色表指针及每像素所占的位数等信息,将其写到指定文件中
17 bool readBmp(char* bmpName)
18 {
19     FILE* fp = fopen(bmpName, "rb");//二进制读方式打开指定的图像文件
20     if (fp == 0)
21         return 0;
22
23     //跳过位图文件头结构BITMAPFILEHEADER
24     fseek(fp, sizeof(BITMAPFILEHEADER), 0);
25     //定义位图信息头结构变量，读取位图信息头进内存，存放在变量head中
26     BITMAPINFOHEADER head;
27     fread(&head, sizeof(BITMAPINFOHEADER), 1, fp);
28     //获取图像宽、高、每像素所占位数等信息
29     bmpwidth = head.biwidth;
30     bmpHeight = head.biHeight;
31     biBitCount = head.biBitCount;
32     //定义变量，计算图像每行像素所占的字节数（必须是4的倍数）
33     int lineByte = (bmpwidth * biBitCount / 8 + 3) / 4 * 4;
34     //灰度图像有颜色表
35     if (biBitCount == 8){
36         //申请颜色表所需要的空间，读颜色表进内存
37         pColorTable = new RGBQUAD[256];
38         fread(pColorTable, sizeof(RGBQUAD), 256, fp);
39     }
40     //申请位图数据所需要的空间，读位图数据进内存
41     pBmpBuf = new unsigned char[lineByte * bmpHeight];
42     fread(pBmpBuf, 1, lineByte * bmpHeight, fp);
43     fclose(fp);//关闭文件
44     return 1;//读取文件成功
45 }
46
47 //将图像数据保存至image，并进行滤波操作
48 void ImageData(float** image)
```

```

49 {
50     //保存数据
51     for (int i = 0; i < bmpHeight; i++)
52     {
53         for (int j = 0; j < bmpwidth; j++)
54         {
55             image[i][j] = *(pBmpBuf + i * bmpwidth + j);
56         }
57     }
58
59     //滤波操作
60     for (int i = 1; i < bmpHeight - 1; i++)
61     {
62         for (int j = 1; j < bmpwidth - 1; j++)
63         {
64             image[i][j] = (image[i - 1][j - 1] + image[i - 1][j] + image[i
- 1][j + 1] + image[i][j - 1] + image[i][j]
65                 + image[i][j + 1] + image[i + 1][j - 1] + image[i + 1][j] +
image[i + 1][j + 1]) / 9;
66         }
67     }
68
69     //输出图像第一行滤波结果到控制台
70     for (int p = 1; p <= 3; p++)
71     {
72         for (int j = 1; j < bmpwidth; j++)
73         {
74             cout << image[p][j];
75             printf("\t");
76         }
77         cout << endl;
78     }
79 }
80
81
82
83 int main()
84 {
85     char readPath[] = "C:/change/mnmnm/bin/homework.bmp"; //BMP文件路径
86     readBmp(readPath);
87
88     float** Image = new float* [bmpHeight];
89     for (int i = 0; i < bmpwidth; i++){
90         Image[i] = new float[bmpwidth];
91     }
92     ImageData(Image);
93     return 0;
94 }

```

```

113.333 125.926 127.325 255
83.251 78.315 93.5073 0
46.2851 33.4843 79.4785 255

```

在main()函数中输入BMP图片路径,即可获得滤波后的图像数据。