专业：_自动化（电气）_
姓名：_____潘盛琪_____
学号：___3170105737___
日期：_____3.21_____
地点：**生工食品学院机房**

浙江大学实验报告

课程名称：**计算机图像处理与机器视觉** 指导老师：**饶秀勤** 成绩：_____

实验名称：**图像平滑** 实验类型：**设计型**

一、实验目的和要求

1. 实现图像平滑的功能
2. 实现伪彩色变换

二、计算机配置与软件处理平台

硬件：

Windows 版本

Windows 10 家庭中文版

© 2018 Microsoft Corporation。保留所有权利。

系统

处理器:	AMD Ryzen 7 PRO 2700U w/ Radeon Vega Mobile Gfx 2.20 GHz
已安装的内存(RAM):	8.00 GB (6.93 GB 可用)
系统类型:	64 位操作系统，基于 x64 的处理器
笔和触控:	没有可用于此显示器的笔或触控输入

软件：基于 matlab

三、算法描述

> 二、均值平滑算子：

2、阈值平滑算子：

- 由于颗粒噪声通常比图像细节有更高的空间频率，即颗粒噪声处的灰度突变明显不同于它的邻域，而图像细节处的灰度通常具有渐变特征，该处像素的灰度与其邻域的平均灰度差异较小。
- 阈值平滑算子：

$$g(i,j)=\begin{cases}\bar{f}_s(i,j) & \left|f(i,j)-\bar{f}_s(i,j)\right|\geq T\\ f(i,j) & \left|f(i,j)-\bar{f}_s(i,j)\right|<T\end{cases}$$

装 订 线

三、空域低通滤波：

- 噪声具有高频特性，采用低通滤波可阻断高频分量通过，达到抑制噪声的效果。
- 低通滤波的卷积模板：

$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$
$\frac{1}{10}$	$\frac{1}{5}$	$\frac{1}{10}$
$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$

L_1

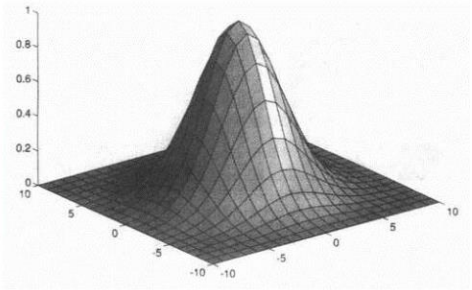
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$

L_2

选择滤波权值应使得滤波器只有一个峰值，称之为主瓣，并且在水平和垂直方向上是对称的。

四、高斯平滑滤波

$$g[i,j]=e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$



高斯滤波器是一类根据高斯函数的形状来选择权值的线性平滑滤波器．高斯平滑滤波器对去除服从正态分布的噪声是很有效的．

其中，高斯分布参数 σ 决定了高斯滤波器的宽度

四、结果与讨论

4.1 输入图像

4.2 输出图像

1. 阈值平滑

   调节阈值与窗口大小观察输出图像的变化

2. 空域低通滤波

3. 高斯滤波器

4. 中值滤波

4.3 讨论

可以看到在阈值平滑算法中，不断改变阈值和窗口大小，图像的模糊程度也会改变，通过输出的图像可以明显地看到，设定的阈值越小输出的图像越模糊，窗口越大输出图像越模

糊。此外，由于我对边缘没有进行任何处理，采取直接保留，因此当窗口变大之后边缘没有被模糊。

空域低通滤波和高斯滤波都有不错的效果。

中值滤波对于消除噪声非常有效。

五、结论

阈值平滑、空域低通滤波、高斯滤波、中值滤波都可以起到较好的图像平滑作用。一般而言，窗口越大，处理后的图片越模糊。

六、源程序

1. 阈值平滑

```
%%%%阈值平滑%%%%
img = imread('test.png');
img = rgb2gray(img);
img = double(img);
[row, col] = size(img);     %确定图像大小
imgAvg = sum(sum(img)) / (row * col);   %求均值


subplot(2,1,1);
imshow(uint8(img));
title('原图像');
subplot(2,1,2);
imgTh = pinghuaTh(img, 20, 3);
imshow(uint8(imgTh));
title('阈值平滑后的图像');
for Th = 0 : 10 : 20
    for window = 3 : 2 : 7
        imgTh = pinghuaTh(img, Th, window);
        i = ((window - 1) / 2 - 1) * 3 + Th / 10 + 1;%当前是第 i 张图
        subplot(3,3,i);
        imshow(uint8(imgTh));
        title(['Th = ', num2str(Th), ' window = ', num2str(window)]);
    end
end

%%%%实现阈值平滑
%%%%函数的第一个参数为图像，第二个参数为阈值，第三个参数为窗口大小
function imgTh = pinghuaTh(img, Th, window)
    imgTh = img;
    [row, col] = size(imgTh);
    for i = (window - 1)/2 + 1 : (col - (window - 1)/2)%遍历列数
        for j = (window - 1)/2 + 1 : (row - (window - 1)/2)%遍历行数
```

```
            imgThAvg = sum(reshape(imgTh(j - (window - 1) / 2 : j + (window - 1) /
2, i - (window - 1)/2 : i + (window - 1)/2), [1,window * window])) / (window *
window);
            if abs(imgTh(j, i) - imgThAvg) > Th
                imgTh(j, i) = imgThAvg;
            end
        end
    end
end
%%%%实现空域低通滤波
```

2. 空域低通滤波

```
%%%实现空域低通滤波%%%
img = imread('test.png');
img = rgb2gray(img);
img = double(img);
[row, col] = size(img);%确定图像大小
kernel = [1/10, 1/10, 1/10; 1/10, 1/5, 1/10; 1/10, 1/10, 1/10];%输入卷积核
imgout = convolve(img, kernel);


subplot(2,1,1);
imshow(uint8(img));
title('原图像');
subplot(2,1,2);
imshow(uint8(imgout));
title('空域低通滤波后的图像');

%%%%实现空域低通滤波
%%%%直接用卷积函数,传入的第一个参数为输入图像，第二个参数为卷积核
function imgout = convolve(imgin, kernel)
    [row, col] = size( imgin );
    [~, kernelsize] = size( kernel );
    r = (kernelsize - 1) / 2;
    rowout = row - 2 * r;  %输出图像大小
    colout = col - 2 * r;  %输出图像大小
    if rowout <= 0 || colout <= 0
        return;
    end
    imgout = zeros(rowout, colout);%初始化输出数组
    for i = 1 : rowout
        for j = 1 : colout
            imgout(i, j) =  max(0, sum(sum(imgin(i : i + 2 * r, j : j + 2 *
r).*(kernel))));
```

```
        end
    end
end
```

3. 高斯滤波器

```
%%%实现空域低通滤波%%%
img = imread('test.png');
img = rgb2gray(img);
img = double(img);
[row, col] = size(img);%确定图像大小
kernel = [0.11, 0.2, 0.29, 0.32, 0.29, 0.2, 0.11; ...
          0.2, 0.37, 0.54, 0.61, 0.54, 0.37, 0.2; ...
          0.29, 0.54, 0.78, 0.88, 0.78, 0.54, 0.29; ...
          0.32, 0.61, 0.88, 1, 0.88, 0.61, 0.32; ...
          0.29, 0.54, 0.78, 0.88, 0.78, 0.54, 0.29; ...
          0.2, 0.37, 0.54, 0.61, 0.54, 0.37, 0.2; ...
          0.11, 0.2, 0.29, 0.32, 0.29, 0.2, 0.11]/21.52;%输入卷积核
imgout = convolve(img, kernel);


subplot(2,1,1);
imshow(uint8(img));
title('原图像');
subplot(2,1,2);
imshow(uint8(imgout));
title('高斯滤波后的图像');

%%%%实现高斯滤波
%%%%直接用卷积函数,传入的第一个参数为输入图像，第二个参数为卷积核
function imgout = convolve(imgin, kernel)
    [row, col] = size( imgin );
    [~, kernelsize] = size( kernel );
    r = (kernelsize - 1) / 2;
    rowout = row - 2 * r;  %输出图像大小
    colout = col - 2 * r;  %输出图像大小
    if rowout <= 0 || colout <= 0
        return;
    end
    imgout = zeros(rowout, colout);%初始化输出数组
    for i = 1 : rowout
        for j = 1 : colout
            imgout(i, j) =  max(0, sum(sum(imgin(i : i + 2 * r, j : j + 2 *
r).*(kernel))));
        end
    end
```

```
end
```

4. 中值滤波

```
%%%中值处理
img = imread('zaosheng.png');
img = rgb2gray(img);
[row, col] = size(img);%确定图像大小
window = 3;
result = zeros(row-(window-1), col-(window-1));
for i = 2:(col - (window - 1)/2)%遍历列数
    for j = 2:(row - (window - 1)/2)%遍历行数
        result(j-1,i-1) = min(reshape(img(j-1:j+1, i-1:i+1), [1,9]));
    end
end
subplot(2,1,1)
imshow(img);
title('原图像');
subplot(2,1,2)
imshow(uint8(result));
title('中值滤波图像');
```