**武汉理工大学**

|  |
| --- |
| **实** |
| **验** |
| **报** |
| **告** |

|  |  |
| --- | --- |
| 课程名称： | 数字图像处理A |
| 学 期： | 2022-2023学年第2学期 |
| 小组成员： | 何江洋 0122111371113  程明超 0122110870608 |
|  | 戴嘉讯 0122114670916  刘琪 0122114671216  黎宇航 0122110870710 |
| 实验名称： | 图像去噪实验 |

2023 年 4 月 20 日

**目 录**

1. 小组成员和分工情况…………………………………………….3
2. 实验环境………………………………………………………….3
3. 实验内容………………………………………………………….3
4. 实验结果………………………………………………………….5
5. 结果分析………………………………………………………….11
6. 实验小结………………………………………………………….12
7. 代码附录………………………………………………………….12
8. **小组成员和分工情况**

何江洋：负责使用Matlab实现整个实验内容。

程明超：负责完成总结分析实验结果，撰写实验报告。

戴嘉讯：负责辅助完成使用Matlab实现整个实验

刘琪： 负责辅助完成使用Matlab实现整个实验

黎宇航：负责辅助完成实验分析总结，撰写实验报告。

1. **实验环境**

由于图像在计算机中的存储形式往往是矩阵的方式，而Matlab在矩阵处理上的功能及其强大，应用非常强。

本次实验采用Matlab实现，Matlab中包含了许多各种应用程序的工具箱，因此利用Matlab来进行数字图像处理十分便利。图像处理工具箱（IPT）：扩展Matlab环境用于解决数字图像处理问题。由于Matlab版本限制，Matlab自带的图像处理函数不全，我们按照实验指导插入了dipum-toolbox包，用于此次降噪实验。

1. **实验内容**

**3.1将原有彩色图像转化为灰度图像**

灰度图像只包含亮度信息，从而减少了计算量和处理复杂性。同时，灰度图像的单通道特性使得噪声类型更加统一，便于模拟和分析噪声，因为噪声只影响图像的一个维度，而不是多个颜色通道。此外，许多图像处理算法和工具箱都是为灰度图像设计的，因此转换为灰度图可以提高算法的通用性，确保它们可以被直接应用。

在视觉效果方面，灰度图像可以更清晰地展示噪声，因为颜色的变化不会干扰噪声的观察，这对于视觉检查和评估噪声处理效果非常有帮助。而且，噪声分析在灰度图像中更为直接，因为可以集中关注亮度变化，而不受颜色变化的干扰，有助于更准确地估计噪声的统计特性。因此，在本次实验中，使用rgb2gray()函数将图像转换为灰度图后再进行加噪与去噪的研究。

**3.2给灰度图添加噪声**

使用imnoise(f,"salt & pepper")可以在图像中添加椒盐噪声。

使用imnoise(f,"gaussian")可以在图像中添加高斯噪声。

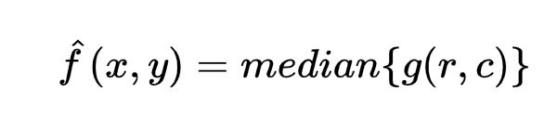
使用imnoise(f,"poisson")可以在图像中添加泊松噪声。

另外我们查询有关资料，对实验进行了拓展，使用imnoise(f,"speckle")函数给图像添加了斑点噪声

**3.3滤波**

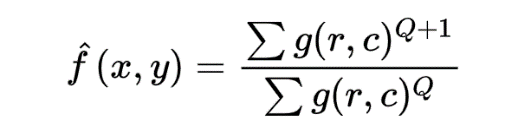
滤波即为降低空间灰度的急剧过度，由于噪声通常是由灰度急剧过度组成，因此滤波在在图像去噪中具有很强的应用性。

**中值滤波：**这是图像处理中最著名的统计排序滤波器，它是利用预定义的邻域中的灰度中值来代替像素的值。



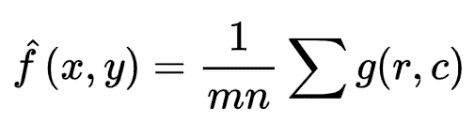
其中r和c分别是预定义区域中的包含像素的行坐标和列坐标。由于在预定义区域中往往选取像素中值作为整个区域的像素值，因此中值滤波能够很好的去除随机噪声，对椒盐噪声的去除效果非常好。

**反调和滤波**：反调和滤波是根据如下表达式复原后得到的图像。



其中Q称为滤波器的阶数，当Q为正数时，该滤波器消除椒噪声，Q为负数时，该滤波器消除盐噪声，当Q为0时，滤波器转变为算术平均滤波器，当Q为-1时，简化为谐波平均滤波器。该种滤波器适合降低或者消除椒盐噪声。

**算术平均滤波器**：算术平均滤波器是最简单的滤波器，我们定义一个m\*n的空间核，在预定义区域中进行均值滤波，计算被污染的图像区域的像素平均值。



运用这种滤波方式，将图像的污染区域的像素值平均，虽然在一定程度上能够降低噪声，但是原始图像也会因为平均而变得模糊。

**最大值和最小值滤波器：**此滤波器是图像处理中最常用的统计排序滤波器，最大值滤波器能够捕捉到图像中的最亮点，同时削弱与明亮区域相邻的暗色区域，最小值滤波器与此恰恰相反。因此可以使用最大值滤波器来降低椒噪声，利用最小值滤波器来降低盐噪声。

**高斯滤波器：**高斯滤波器是我们小组查阅有关资料拓展补充了解的一种滤波器，高斯函数有两个基本性质，高斯函数的乘积和卷积仍然是高斯函数，高斯核的均值为0，这使得卷积结果在滤波中非常重要。

**3.4编程实现**

在**Matlab**中进行编程，实现灰度图像转化、加入噪声、去除噪声。

1. **实验结果**

实验原图：



转化为灰度图像：



添加噪声实验结果如下：

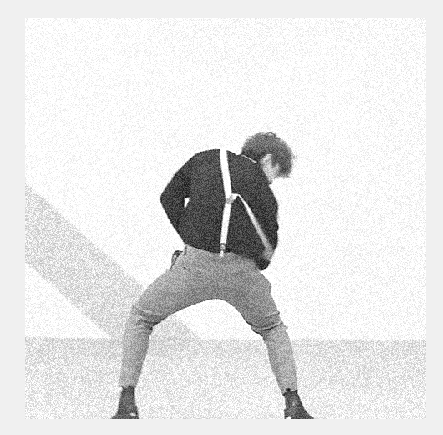
4.1.1椒盐噪声添加如图所示：



4.1.2高斯噪声添加结果如下：



4.1.3均匀噪声添加结果如下：



4.1.4斑点噪声添加如下：



* 1. 不同滤波器去噪情况如下：

**4.2.1中值滤波器：**

Figure 椒盐噪声去噪 Figure高斯噪声去噪

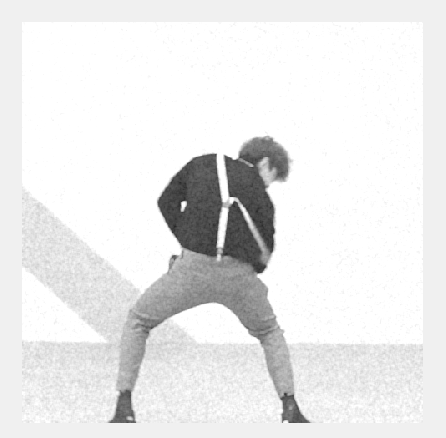
 

Figure均匀噪声去噪 Figure斑点噪声去噪

**4.2.2反调和滤波器：**

Figure椒盐噪声去噪 Figure高斯噪声去噪

Figure均匀噪声去噪 Figure斑点噪声去噪

**4.2.3最大值滤波器：**

Figure 椒盐噪声去噪 Figure高斯噪声去噪

Figure 均匀噪声去噪 Figure 斑点噪声去噪

**4.2.4最小值滤波器：**

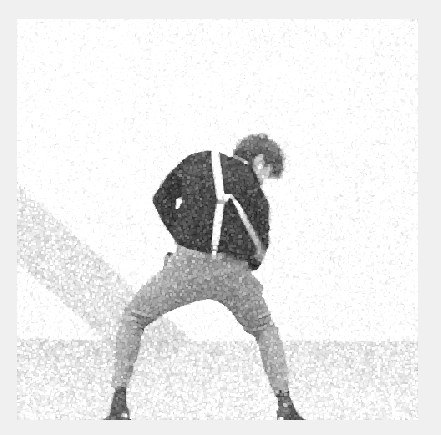
 

Figure 椒盐噪声去噪 Figure 高斯噪声去噪

Figure 均匀噪声去噪 Figure 斑点噪声去噪

**4.2.5算术均值滤波器：**

Figure椒盐噪声去噪 Figure 高斯噪声去噪

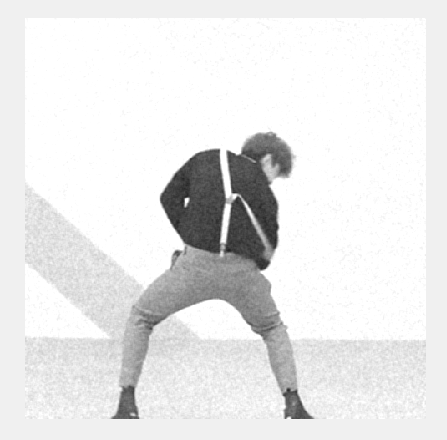
 

Figure 均匀噪声去噪 Figure 斑点噪声去噪

**4.2.6高斯滤波器：**

Figure 椒盐噪声去噪 Figure 高斯噪声去噪

Figure均匀噪声去噪 Figure 斑点噪声去噪

1. **结果分析**

由以上结果可以分析得出一下结论：

5.1中值滤波器对椒盐噪声去除效果最佳。

5.2反调和滤波器去除均匀噪声的效果最佳。

5.3 最大值滤波器能够有效去除椒噪声。

5.4 最小值滤波器能够有效去除盐噪声。

5.5 算数均值滤波器去除均匀噪声的效果最佳。

5.6 高斯滤波器能够有效降噪，对各种噪声降噪程度大致相同。

1. **实验小结**

本次实验很好的掌握了有关图像去噪方面的知识，深入理解了各种滤波器涉及的相关原理，在Matlab环境下进行实验，函数代码相对集成化，代码量不大，实验总体来说相对简单。一方面，我们很好的对课上所学的知识进行了实践，增强了动手能力，另一方面，通过实验，我们也真正理解了数字图像处理有关去噪方面的背后数学原理，取得了很大的收获。

1. **代码附录：**

% f=imread(&apos;homework1.png&apos;);  
% f=im2double(f);  
%   
% FSaltAndPepper=imnoise(f,"salt & pepper"); %椒盐噪声  
% FGaussian=imnoise(f,"gaussian"); %高斯白噪声  
% FPoisson=imnoise(f,"poisson"); %泊松噪声  
% Fspeckle=imnoise(f,"speckle"); %斑点噪声  
%   
% subplot(1,3,1);  
% imshow(f);  
% subplot(1,3,2);  
% imshow(FSaltAndPepper);  
%   
% ff=medfilt3(FSaltAndPepper,[3 3 3],"symmetric");  
% % ff=medfilt2(FSaltAndPepper,[5 5],"symmetric");  
% subplot(1,3,3);  
% imshow(ff);  
  
f=imread(&apos;homework1.png&apos;);  
f=im2double(f);  
f=rgb2gray(f);  
  
[m,n]=size(f);  
  
FSaltAndPepper=imnoise(f,"salt & pepper"); %椒盐噪声  
  
  
FGaussian=imnoise(f,"gaussian"); %高斯白噪声  
a=0;  
b=0.25;  
I=a+(b-a)\*rand(m,n);%均匀分布噪声  
  
  
% figure(1)  
% imshow(I)  
  
Funiform = I + f;               %均匀噪声  
% Funiform=imnoise(f,"uniform"); %均匀噪声 % 2021b版似乎已经不能用  
  
% imshow(I)  
% figure(2)  
  
  
FPoisson=imnoise(f,"poisson"); %泊松噪声  
Fspeckle=imnoise(f,"speckle"); %斑点噪声  
  
  
mm=2;  
nn=4;  
  
subplot(mm,nn,1);  
imshow(f);  
  
  
% imgset={FSaltAndPepper,FGaussian,Funiform,FPoisson,Fspeckle};  
imgset={FGaussian};  
  
  
  
  
  
for i = [1:1:length(imgset)]  
  
    Fimg = cell2mat(imgset(i));  
  
    figure(i)  
      
    subplot(mm,nn,1);  
    imshow(f);  
    subplot(mm,nn,2);  
    imshow(Fimg);  
  
    Fmed=medfilt2(Fimg,[5 5],"symmetric");  
    subplot(mm,nn,4);  
    imshow(Fmed);  
    Fchmean=spfilt(Fimg,&apos;chmean&apos;,3,3,1.5);  
    subplot(mm,nn,5);  
    imshow(Fchmean);  
  
    Famean=spfilt(Fimg,&apos;amean&apos;,3,3);  
    subplot(mm,nn,6);  
    imshow(Famean)  
    Fmax=spfilt(Fimg,&apos;max&apos;,3,3);  
    subplot(mm,nn,7);  
    imshow(Fmax)  
    Fmin=spfilt(Fimg,&apos;min&apos;,3,3);  
    subplot(mm,nn,8);  
    imshow(Fmin)  
  
    Guass\_kernal=fspecial(&apos;gaussian&apos;,[3,3],1.0);  
    Guass\_kernal  
  
    FGauss=imfilter(Fimg,Guass\_kernal,&apos;replicate&apos;);   %高斯滤波  
    subplot(mm,nn,3);  
    imshow(FGauss)  
  
end