

云南大学

本科实验报告

课程名称: 图像理解与计算机视觉

实验名称: 实验三. 图像复原实验

学院（系）：_____

专 业: _____

班 级: _____

姓 名: _____

学 号: _____

指导教师: _____

成绩: _____

评语:

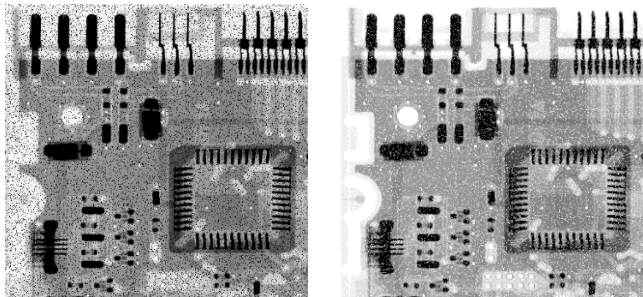
Steven

一. 实验目的

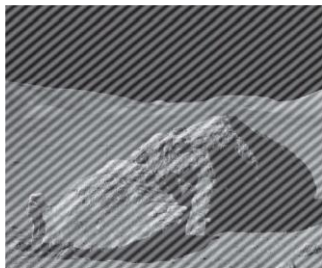
通过编程实现使学生掌握采用空域滤波去除图像全局噪声使图像复原的方法、及采用陷波带阻滤波器去除周期性噪声使图像复原的方法。针对仅存在噪声的图像，学生能够分析空域图像的特点选择恰当的空域滤波器，也能够分析频谱图的特点并构造恰当的陷波带阻滤波器，并取得良好的复原效果。

二. 实验内容

(1) 下图分别是受到胡椒噪声和盐粒噪声干扰的电路板图像，分别为两幅图像选择恰当的反谐波平均滤波器参数，并展示复原效果；



(2) 下图是一幅被周期性正弦噪声污染的图像，根据频谱图的特点构造恰当的陷波带阻滤波器，并展示复原效果。



三. 实验环境

Matlab软件是图像处理领域广泛使用的仿真软件之一。本实验基于Matlab 2022版本完成。

四. 实验代码（详细注释，Times New Roman/宋体 五号字体 单倍行距）

主函数
<pre>%% 使用: 在命令行内调用函数 Exp3(func), func 是不同功能的名字。要将文件和.m 文件放在同一路径下。 %% 每题的 demo 调用格式如下: % 1. 采用合适的反谐波平均滤波器进行图像复原: Exp3("Ques1");</pre>

% 2. 采用陷波带阻滤波器，对受正弦噪声污染的图像进行复原: Exp3("Ques2");

function Exp3(ques)

if ques == "Ques1"

pepper_img = imread("pepper.tif");

pepper_img = pepper_img(:, :, 1); % 不知道为什么，这个灰度图保存成了一个 4 通道的图像，发现第 4 维全为 255，所以格式应该为(R,G,B,alpha)，所以这里取第 1 个通道即为灰度值

pepper_denoise = Ques1(pepper_img, [3, 3], 1.5);

salt_img = imread("salt.jpg");

salt_img = im2gray(salt_img);

salt_denoise = Ques1(salt_img, [3, 3], -100);

subplot(2, 2, 1); imshow(pepper_img); title("原图");

subplot(2, 2, 2); imshow(pepper_denoise); title("胡椒噪声去噪");

subplot(2, 2, 3); imshow(salt_img); title("原图");

subplot(2, 2, 4); imshow(salt_denoise); title("盐粒噪声去噪");

elseif ques == "Ques2"

sineNoise = imread("sineNoise.jpg");

sineNoise = im2gray(sineNoise);

Ques2(sineNoise);

end

end

功能函数1：采用反谐波平均滤波器，对图像进行复原

%% Ques1: 采用反谐波平均滤波器，对图像进行复原

% Input/输入:

% img: 二维矩阵，表示灰度图像

% kernel_sz: 2×1 的向量，分别表示卷积核的宽度、高度

% Q: 标量，阶数

% Output/输出:

% img_new: 2 维灰度矩阵，滤波后的结果

function img_new = Ques1(img, kernel_sz, Q)

img = im2double(img);

[width, height] = size(img);

m = kernel_sz(1); n = kernel_sz(2); % 获取卷积核的尺寸

% 确定要扩展的行列数

len_m = floor(m / 2);

len_n = floor(n / 2);

img_pad = padarray(img, [len_m, len_n], 'symmetric'); % 将原始图像进行扩展

[M, N] = size(img_pad);

img_new = zeros(width, height);

% 逐点计算子窗口的谐波平均

for i = 1 + len_m:M - len_m

for j = 1 + len_n:N - len_n

<pre> %从扩展图像中取出子图像 block = img_pad(i - len_m:i + len_m, j - len_n:j + len_n); %求子窗口的谐波平均 s1 = sum(sum(block.^(Q + 1))); s2 = sum(sum(block.^Q)); % 为了应对经常出现的 $0^{(Q+1)}/0^Q$ 的情况 if isinf(s1) && isinf(s2) img_new(i - len_m, j - len_n) = 0; else img_new(i - len_m, j - len_n) = s1 / s2; end end end end end </pre>	<p>功能函数2: 采用陷波带阻滤波器, 对受正弦噪声污染的图像进行复原</p>
<pre> %% Ques2: 采用陷波带阻滤波器, 对受正弦噪声污染的图像进行复原 % Input/输入: % img: 二维矩阵, 表示灰度图像 % Output/输出: % img_new: 2 维灰度矩阵, 即去噪结果 function img_new = Ques2(img) f = fftshift(fft2(img)); sz = size(f); f2 = f; % 观察到有两条带状尖峰, 且中心对称, 分别位于列 237 和列 287 上, 应当将其滤掉 kernel = ones(sz); for row = 1:sz(1) kernel(row, 237) = 0; kernel(row, 287) = 0; end f2 = f2 .* kernel; % 观察到两个尖峰分别是[190, 237]、[240, 287], 应当将其一定邻域范围内滤掉, 故仍然使用理想高通的陷波带阻滤波器 kernel = ones(sz); radius = 3; % 邻域半径 for i = 1:sz(1) for j = 1:sz(2) if sqrt((i - 190)^2 + (j - 237)^2) <= radius kernel(i, j) = 0; end if sqrt((i - 240)^2 + (j - 287)^2) <= radius kernel(i, j) = 0; end end end end </pre>	

```

end
end

f2 = f2 .* kernel;
f_origin = log(abs(f) + 1); % 原始频谱图
f_new = log(abs(f2) + 1);
% 逆傅里叶变换，根据频谱图转换原图
img_new = real(ifft2(ifftshift(f2)));
img_new = im2uint8(mat2gray(img_new));
% 绘图
subplot(2, 2, 1); imshow(img); title('原图');
subplot(2, 2, 2); imshow(img_new); title('滤波去噪图');
subplot(2, 2, 3); imshow(f_origin, []); title('原始频谱图');
subplot(2, 2, 4); imshow(f_new, []); title('滤波后频谱图');
end

```

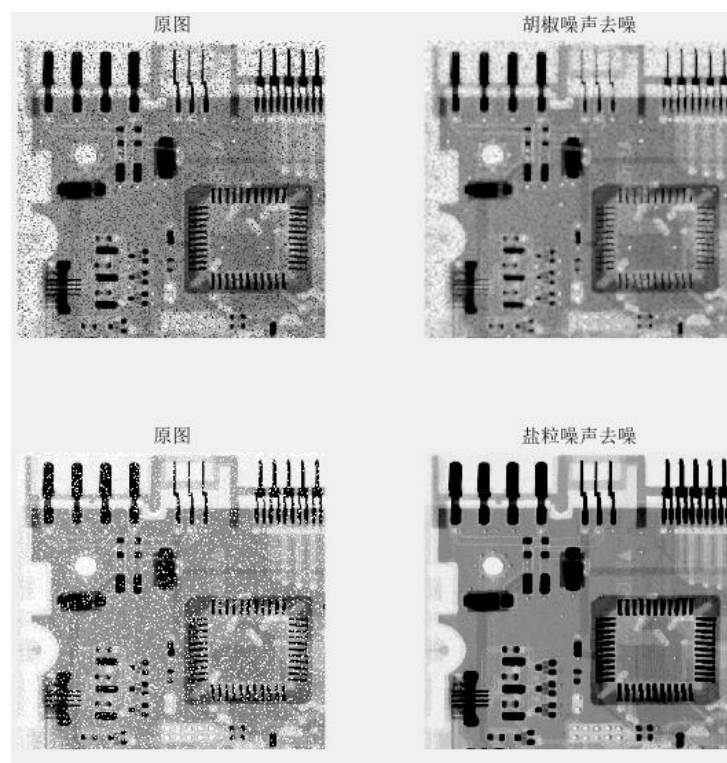


Exp3.m

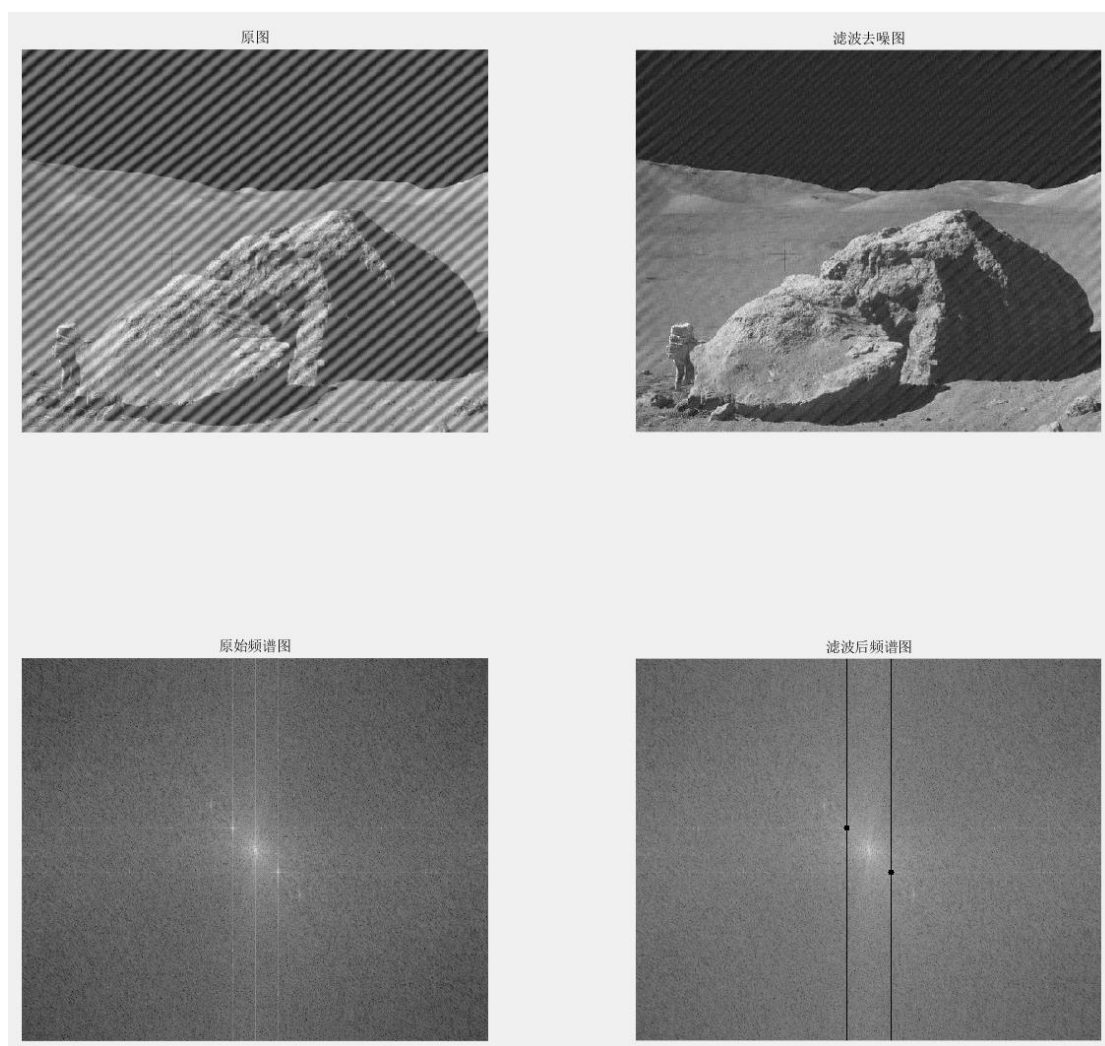
附件（.m文件）：

五. 实验结果

（1）胡椒噪声和盐粒噪声电路板图像的复原结果及对应的反谐波平均滤波器参数



(2) 周期性正弦噪声图像的复原结果



六. 结果分析及体会

在本次实验中遇到了个小问题，便是最初在读取png图片时，读取到了所有4通道的值，而通常来说彩图也只有3通道。经查，png文件可以保存alpha通道，即各像素的透明度。第1题第2张图的png图中，alpha通道全为255，即不透明。所以本题是转换为jpg图像后完成的。

此外还在调试的过程中学到一点技巧：在观察值的时候，直接查看一个很大的矩阵的值往往不太直观，这时候可以在右侧变量位置右键，便可在选项栏中快速建立图像，便于观察，这在我观察卷积核迭代效果时起到了很大作用。