计算机与信息工程学院实验报告

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

姓名： 张焱兵 学号： 1603643050 专业：自动化 年级： 2016

课程： 机器视觉 主讲教师： 范明虎 辅导教师： 范明虎

实验时间：2019 年 10 月 28 日 上 午 8 时至 11 时，实验地点 606

实验题目： 实验五： 图像增强

实验目的： （1）熟悉灰度变换的图像增强方法

（2）熟悉常用的锐化滤波方法

实验环境（硬件和软件）： 计算机、Matlab2014a

一、实验内容：

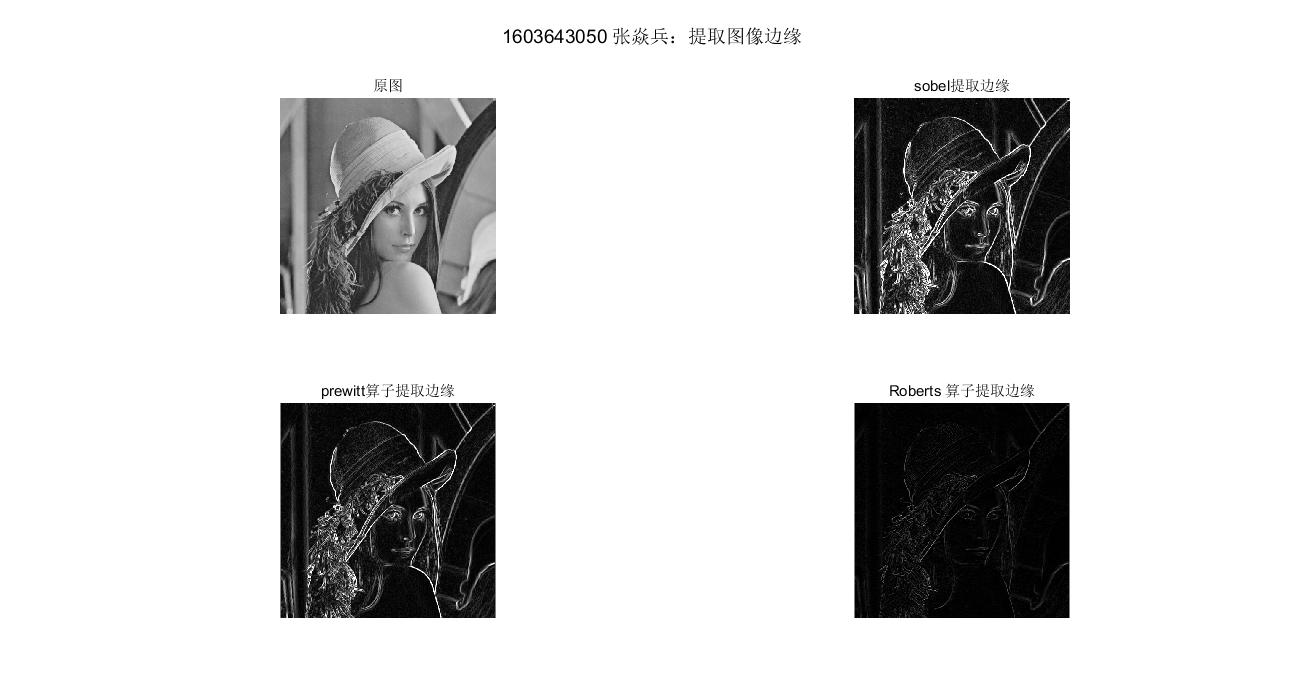
每道题包含以下内容：

1. 实现本次实验内容的程序：

**①主程序：**

clear,clc,close all  
  
img1=imread('figure51.jpg');  
img1 = im2double(img1); *%double类型转换*  
  
figure(1)  
subplot(1,3,1),imshow(img1);  
title('原始图像');  
  
lim = stretchlim(img1); *%灰度变换自适应阈值*  
L =imadjust(img1,lim,[]); *% 增强图像的对比度*  
subplot(1,2,2),imshow(L);  
title('最佳范围调整');  
suptitle('1603643050 张焱兵：灰度变换')  
  
*%% 图像边缘提取*  
img2=imread('lenagray.jpg');  
  
figure(2)  
subplot(2,2,1),imshow(img2);title('原图');  
*% 使用sobel算子*  
img\_edge = SobelOperator(img2);  
subplot(2,2,2);imshow(img\_edge,[]);title('sobel提取边缘');  
  
*% 使用prewitt算子*  
img\_edge = PrewittOperator(img2);  
subplot(2,2,3);imshow(img\_edge,[]);title('prewitt算子提取边缘');  
  
*% 使用roberts算子*  
subplot(2,2,4);  
img\_edge = RobertsOperator(img2);  
imshow(img\_edge);title('Roberts 算子提取边缘');  
  
suptitle('1603643050 张焱兵：提取图像边缘')  
  
**②Prewitt算子m函数实现：**  
function [ edge ] = PrewittOperator( pic )  
 edge = uint8(zeros(size(pic))); *% 用uint8类矩阵存储提取后的边型缘数据*  
  
 h = size(pic, 1); *% 高*  
 w = size(pic, 2); *% 宽*  
  
 gx = [-1, -1, -1; 0, 0, 0; 1, 1, 1]; *%水平边缘检测Sobel算子（卷积核） gx*  
 gy = gx'; *%水平边缘检测Sobel算子（卷积核） gy*  
  
 for i = 2 : h - 1  
 for j = 2 : w - 1  
 sub = double(pic(i - 1 : i + 1, j - 1 : j + 1)); *% 3\*3像素区域*  
 g1 = abs(sum(sum(sub .\* gx))); *% 水平梯度 Gx*  
 g2 = abs(sum(sum(sub .\* gy))); *% 垂直梯度 Gy*  
 if g1 > g2 *%选取边界轮廓更明显的方向*  
 edge(i, j) = uint8(g1); *%转换类型*  
 else  
 edge(i, j) = uint8(g2);  
 end  
 end  
 end  
end  
  
**②Roberts算子m函数实现：**function [edge] = RobertsOperator(pic)  
*% edge：图像边缘，pic：图像矩阵*  
edge = uint8(zeros(size(pic))); *% 用uint8类型矩阵存储提取后的边缘数据*  
  
 h = size(pic, 1); *% 高*  
 w = size(pic, 2); *% 宽*  
  
 for i = 1 : h-1  
 for j = 1 : w-1  
 *%Roberts算子 G[F(x,y)]*  
 edge(i, j) = uint8(abs(pic(i, j) - pic(i + 1, j + 1)) + abs(pic(i, j + 1) - pic(i + 1, j)));  
 end  
 end  
  
end  
  
**③Sobel算子m函数实现：**  
function [edge] = SobelOperator(pic)  
 edge = uint8(zeros(size(pic))); *% 用uint8类矩阵存储提取后的边型缘数据*  
  
 h = size(pic, 1); *% 高*  
 w = size(pic, 2); *% 宽*  
  
 gx = [-1, -2, -1; 0, 0, 0; 1, 2, 1]; *%水平边缘检测Sobel算子（卷积核） gx*  
 gy = gx'; *%垂直边缘检测Sobel算子（卷积核） gy*  
  
 for i = 2 : h - 1  
 for j = 2 : w - 1  
 sub = double(pic(i-1 : i+1, j-1 : j+1)); *% 3\*3像素区域*  
 g1 = abs(sum(sum(sub .\* gx))); *% 水平梯度 Gx*  
 g2 = abs(sum(sum(sub .\* gy))); *% 垂直梯度 Gy*  
 if g1 > g2 *%选取边界轮廓更明显的方向*  
 edge(i, j) = uint8(g1); *%转换类型*  
 else  
 edge(i, j) = uint8(g2); *%转换类型*  
 end  
 end  
 end  
end

1. 实验数据：



1. 实验结果与分析：

由灰度变换结果可知，通过stretchlim函数得到的灰度变换自适应阈值非常合适，将它用在imadjust函数中，通过增强图像的对比度使图像显得清晰明亮。

锐化滤波里，Roberts算子，Sobel算子，Prewitt算子都属于边缘检测算子，并且都是无方向一阶锐化。将所有的边缘模板逐一作用于图像中的每一个像素，产生最大输出值的边缘模板方向表示该点边缘的方向，如果所有方向上的边缘模板接近于零，该点处没有边缘。

Roberts算子边缘定位准，但是对噪声敏感。适用于边缘明显且噪声较少的图像分割。它是一种利用局部差分算子寻找边缘的算子，图像处理后结果边缘不是很平滑，边缘定位的精度不是很高。

Sobel边缘算子的卷积和如上图所示,图像中的每个像素都用这两个核做卷积。Sobel算子认为邻域的像素对当前像素产生的影响不是等价的，所以距离不同的像素具有不同的权值，对算子结果产生的影响也不同。一般来说，距离越大，产生的影响越小。这两个核分别对垂直边缘和水平边缘响应最大，两个卷积的最大值作为该点的输出位。运算结果是一幅边缘幅度图像。

Prewitt算子在一个方向求微分，而在另一个方向求平均，因而对噪声相对不敏感，有抑制噪声作用。但是像素平均相当于对图像的低通滤波，所以Prewitt算子对边缘的定位不如Roberts算子。

Sobel算法与Priwitt算法的思路相同，属于同一类型，因此处理效果基本相同。Roberts算法的模板为2\*2，提取信息较弱。

二、问题讨论：

本次实验让我对颇为神秘的图像增强有了自己的认识，特别是微分算子提取图像边缘，除上述算子外还有很多其他的算子，它们各有特点和作用。

Robert算子在M语言中没有现成的函数，经过原公式的微分变差分，可以用m函数轻松实现它。