数字图像处理实验报告

**实验项目名称: 实验3 图像的几何运算**

**姓名: myp 学号 20201202075 班级20级计科一班**

**提交时间: 2023.3.17**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**一、实验目的：**

1、了解图像的几何运算的原理；

2、掌握图像几何运算的Matlab编程。

**二、实验内容和实验原理**

1、编程实现图像的旋转(imrotate)；

2、编程实现图像的按比例缩放(imresize)；

3、编程实现图像的裁剪(imcrop)。

4、编程实现图像的镜像变换(水平、垂直和对角镜像)；

5、编程实现图像的平移。

**三、实验步骤**

1、编程实现图像的旋转(imrotate)；

im = imread('D:\MATLAB\实验素材\slena.jpg');

%旋转30°

%求出旋转矩阵

a=30/180\*pi;

R=[cos(a),-sin(a);sin(a),cos(a)];%旋转矩阵

R=R';%求出旋转矩阵的逆矩阵

%求出图片大小

sz=size(im);

h=sz(1);

w=sz(2);

ch=sz(3);

c1=[h;w]/2;%原图中心

%计算显示完整图像需要的画布大小

hh=floor(w\*sin(a)+h\*cos(a))+1;

ww=floor(w\*cos(a)+h\*sin(a))+1;

c2=[hh,ww]/2;%求新画布中点

%初始化目标画布

im2=uint8(ones(hh,ww,3)\*128);

for k = 1:ch

for i = 1:hh

for j = 1:ww

p=[i;j];%遍历新图像像素点

pp=round(R\*(p-c2)+c1);%计算在原来图像中的位置

%这里仍然使用round函数，但结果会比方案一好得多

%逆向进行像素查找

if(pp(1)>=1&&pp(1)<=h&&pp(2)>=1&&pp(2)<=w)

im2(i,j,k)=im(pp(1),pp(2),k);

end

end

end

end

%显示图像

figure;

imshow(im);

figure;

imshow(im2);

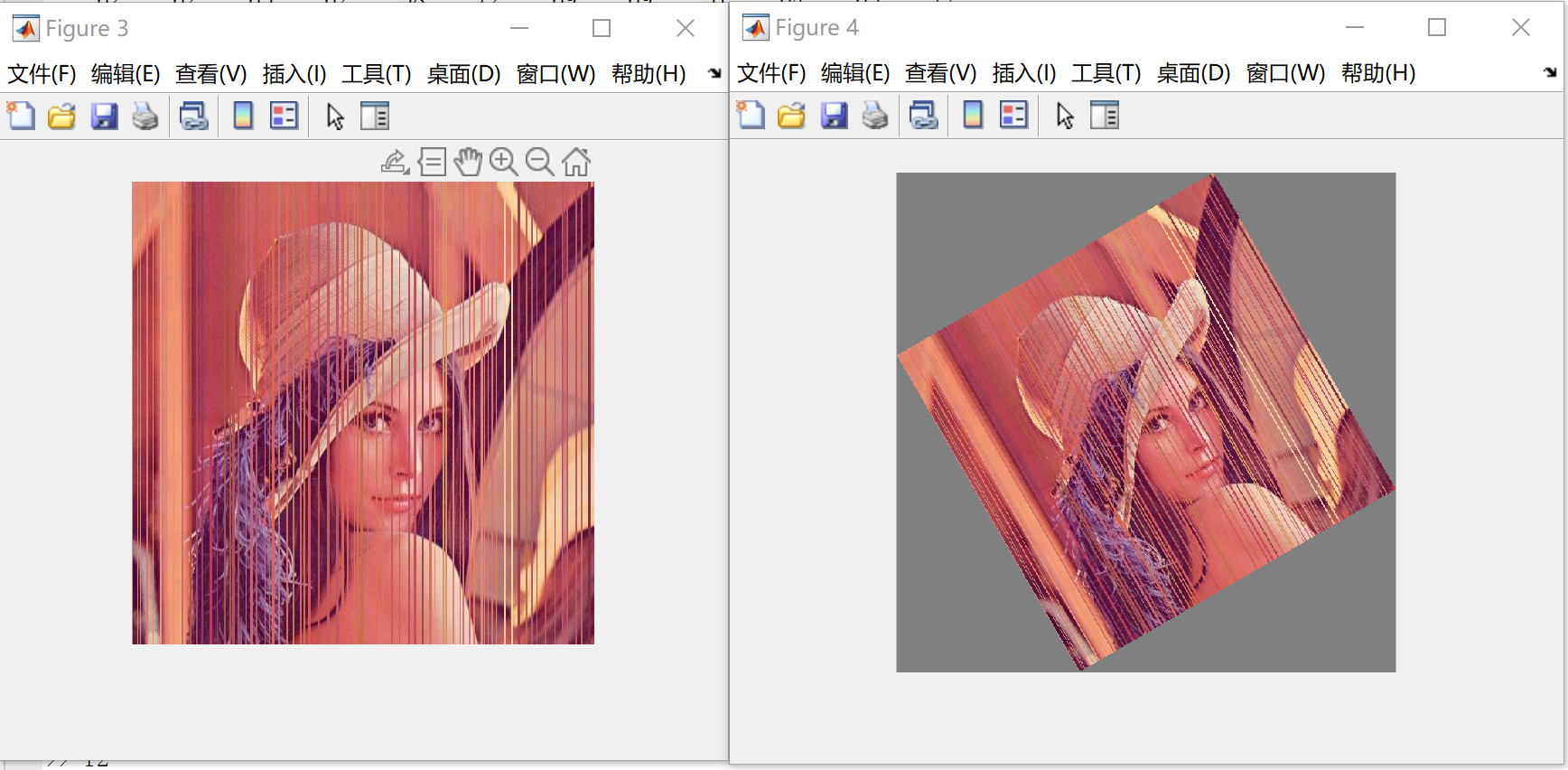


图 1 旋转前后的图像

2、编程实现图像的按比例缩放(imresize)；

t=imread('D:\MATLAB\实验素材\slena.jpg');

t = rgb2gray(t);

[m,n]=size(t);

%自定义长 宽

rm= 1/2\*m

rn= 1/2\*n

%构造新矩阵 存储收缩后的图片

rt=zeros(rm,rn);

for i=1:rm

for j=1:rn

%坐标转换

x=i\*m/rm;

y=j\*n/rn;

%求出偏移量

u=x-floor(x);

v=y-floor(y);

%边缘处理

if x<1

x=1;

end

if y<1

y=1;

end

%双线性插值

rt(i,j)=t(floor(x),floor(y),:)\*(1-u)\*(1-v)+t(floor(x),ceil(y),:)\*(1-u)\*v+t(ceil(x),ceil(y),:)\*(u)\*(1-v)+t(ceil(x),ceil(y),:)\*u\*v;

end

end

imshow(t);title('原图');

figure;imshow(uint8(rt));title('1/2倍的图片');

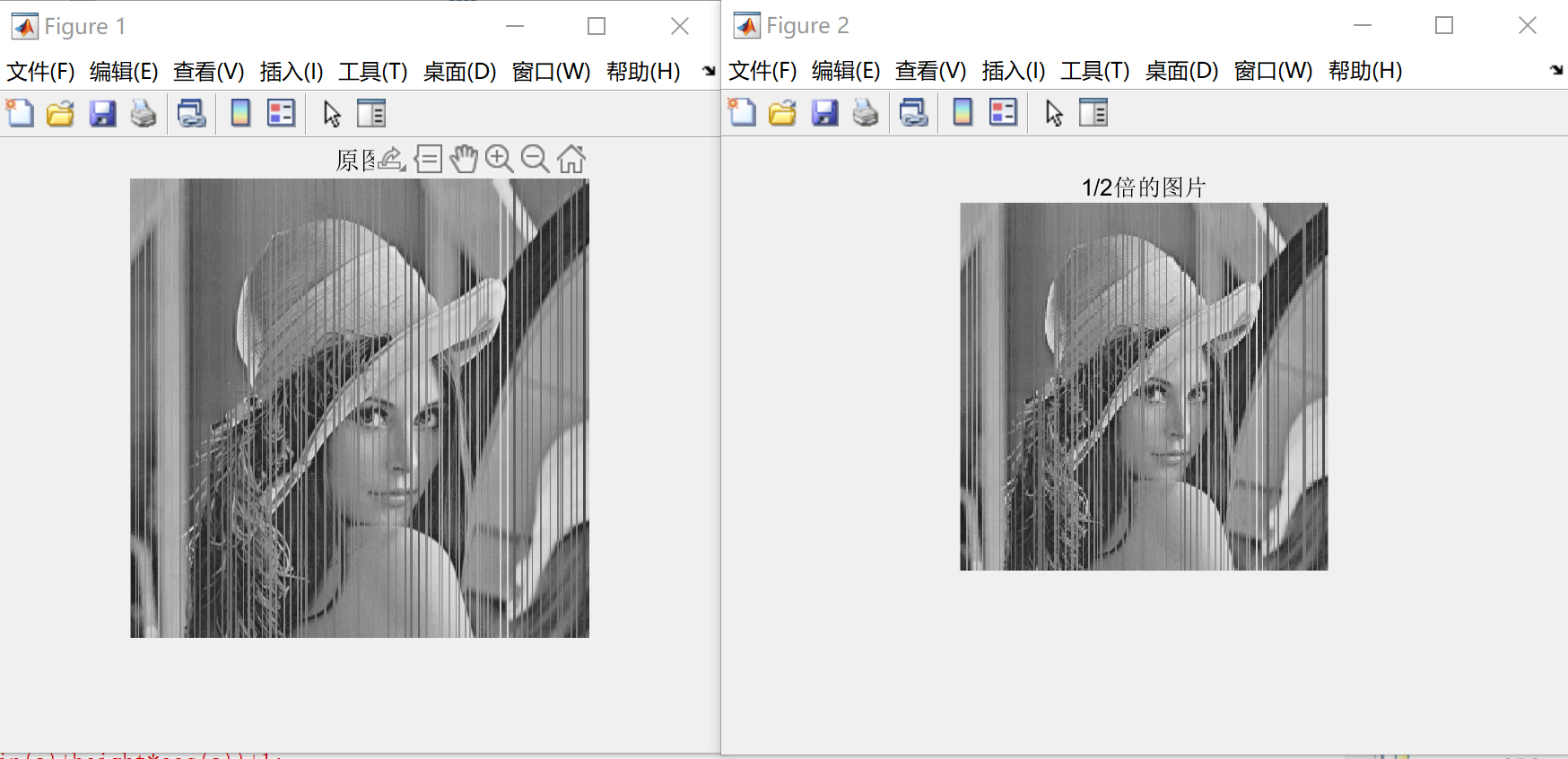


图 2 放缩后的图像

1. 编程实现图像的裁剪(imcrop)。

rt=t(200:440,100:end);

imshow(rt);title('裁剪后的图片结果');



1. 编程实现图像的镜像变换(水平、垂直和对角镜像)；

t=imread('D:\MATLAB\实验素材\slena.jpg');

t = rgb2gray(t);

[m,n]=size(t);

t4=t;

t5=t;

t6=t;

for i=1:m

for j=1:n

t4(i,j)=t(i,n-j+1);

end

end

for i=1:m

for j=1:n

t5(i,j)=t(m-i+1,j);

end

end

for i=1:m

for j=1:n

t6(i,j)=t5(i,n-j+1);

end

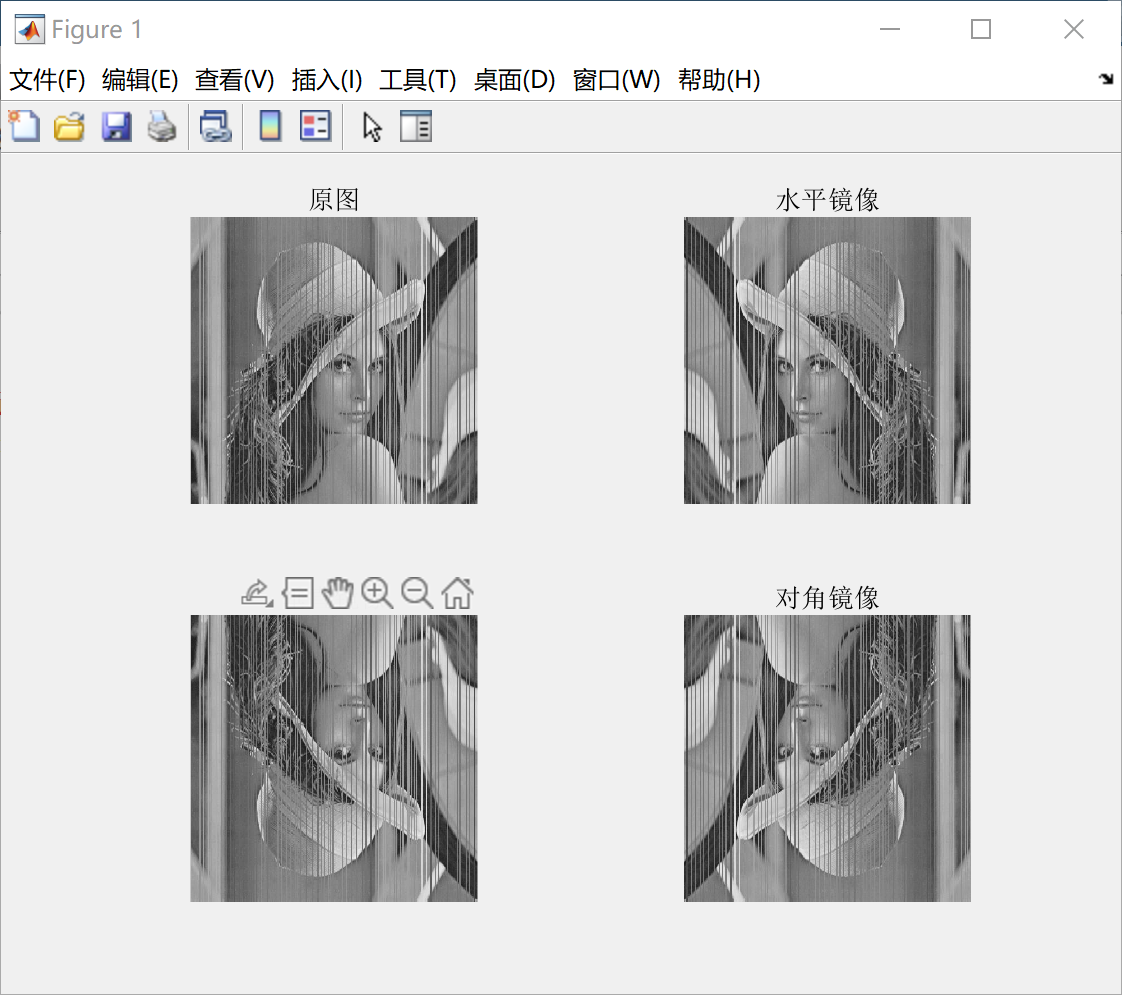
end

subplot(2,2,1),imshow(t),title('原图')

subplot(2,2,2),imshow(t4),title('水平镜像')

subplot(2,2,3),imshow(t5),title('垂直镜像')

subplot(2,2,4),imshow(t6),title('对角镜像')



5、编程实现图像的平移。

t=imread('D:\MATLAB\实验素材\slena.jpg');

t = rgb2gray(t);

[R, C] = size(t); % 获取图像大小

res = zeros(R, C); % 构造结果矩阵。每个像素点默认初始化为0（黑色）

delX = 200; % 平移量X

delY = 100; % 平移量Y

tras = [1 0 delX; 0 1 delY; 0 0 1]; % 平移的变换矩阵

for i = 1 : R

for j = 1 : C

temp = [i; j; 1];

temp = tras \* temp; % 矩阵乘法

x = temp(1, 1);

y = temp(2, 1); % 变换后的位置判断是否越界

if (x <= R) && (y <= C) && (x >= 1) && (y >= 1)

res(x, y) = t(i, j);

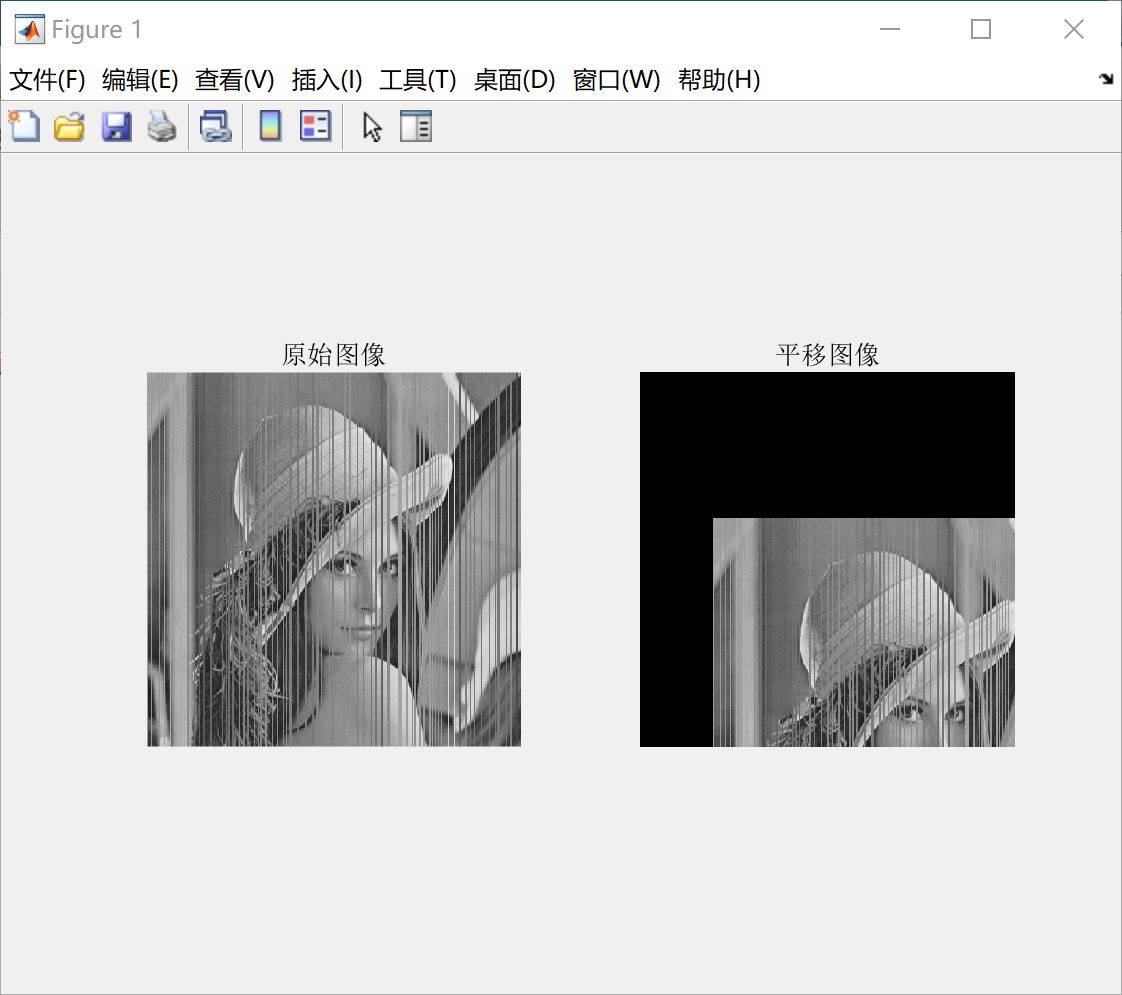
end

end

end

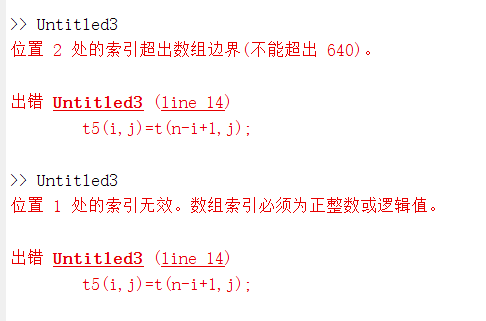
subplot(1,2,1),imshow(t),title('原始图像')

subplot(1,2,2),imshow(uint8(res)),title('平移图像')



**四、实验心得体会**

实验过程中遇到问题：



一开始以为设置的越界了，后来发现是位置搞反了，m,n位置错位了。