实验2 图像的空域与几何运算

Matlab自带图像所在目录，不同的安装路径前面的目录可能会有差异，请自行查阅安装目录后查找自带图像位置 C:\Program Files\MATLAB\R2016a\toolbox\images\imdata

**实验目的：**

（1）掌握图像的基本算术运算可实现的图像加，减，乘，除效果；

（2）掌握图像的逻辑运算实现前景，背景分离功能；

（3）掌握图像几何变换的基本方法。

**实验要求：**

（1）以自己的学号+姓名创建一个文件夹，并将该文件夹添加到MATLAB路径管理器中；

（2）掌握函数imadd(), imsubtract(), immulitply()，imdivide()等基本函数的调用。

（3）掌握图像按像素&，xor，|等逻辑运算的基本功能和调用；

（4）掌握图像几何变换的基本方法。

**实验内容**：

（1）读入matlab自带彩色图像canoe.tif，将此彩色图像转换为灰度图像；分别在彩色图像和灰度图像上+30，观察增加数值后的图像维度，并显示修改后的图像（注意修改后的图像调整到double[0-1]或uint8[0-255]显示）。

代码：

clc

clear

[im1, map]=imread('F:\matlab\matlab\_r2021a\toolbox\images\imdata\canoe.tif');

subplot(1,3,1)

imshow(im1,map);

title('原图')

subplot(1,3,2)

im2=ind2rgb(im1,map);

im2=imadd(im2,30/255);

imshow(im2);

title('彩色图+30')

subplot(1,3,3)

im3=ind2gray(im1,map);

im3=imadd(im3,30);

imshow(im3);

title('灰度图+30')

结果：



（2）读入matlab自带图像 office\_1.jpg office\_2.jpg, 利用函数imdivide（），用第2副图像除以第一幅图像，第二副图像除0.5，并分别显示这两个操作的结果。（注意修改后的图像调整到double[0-1]或uint8[0-255]显示）

代码：

clear

clc

im1=imread('office\_1.jpg');

im2=imread('office\_2.jpg');

im3=imdivide(im2,im1);

im4=imdivide(im2,0.5);

subplot(2,2,1)

imshow(im1)

title('原图1')

subplot(2,2,2)

imshow(im2)

title('原图2')

subplot(2,2,3)

imshow(im3)

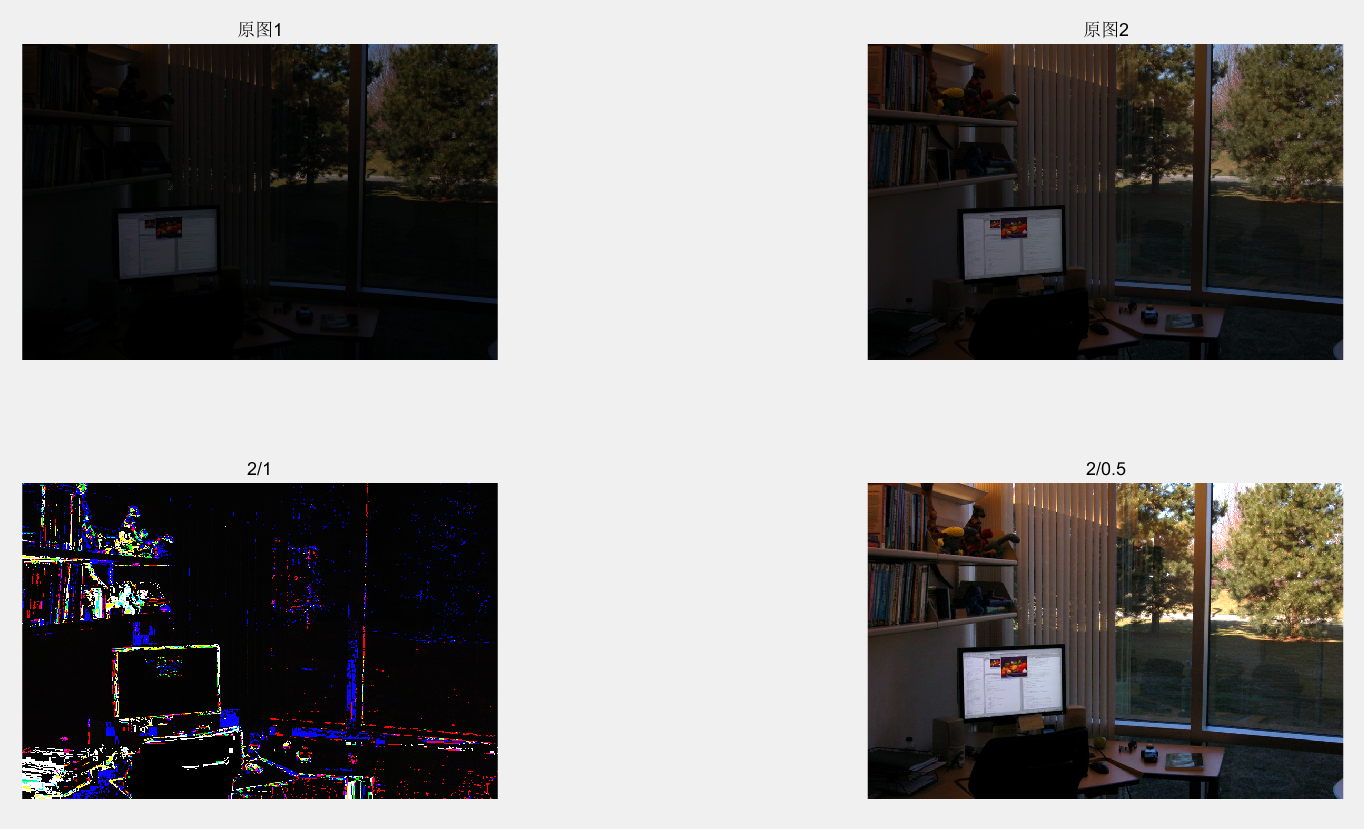
title('2/1')

subplot(2,2,4)

imshow(im4)

title('2/0.5')

结果：



（3）图像的逻辑运算，读入matlab 自带图像pillsetc.png，利用函数im2bw（）函数转为2值图像（自行查阅im2bw()函数的功能），利用此2值图像作为掩模与原始图像进行逻辑乘，取出目标图像；利用2值图像取出图像背景；显示取出的目标图像和背景图像。

代码：

clear

clc

im1=imread('pillsetc.png');

%figure(1)

subplot(2,2,1)

imshow(im1)

title('原图')

im1=double(im1)/255;

%temp=mean(im1(:,:));

%level=mean(temp);

im2=im2bw(im1);

%figure(2)

subplot(2,2,2)

imshow(im2)

title('二值图')

im3(:,:,1)=immultiply(im1(:,:,1),im2);

im3(:,:,2)=immultiply(im1(:,:,2),im2);

im3(:,:,3)=immultiply(im1(:,:,3),im2);

subplot(2,2,3)

imshow(im3)

title('目标图像')

im4(:,:,1)=imsubtract(im1(:,:,1),double(im2));

im4(:,:,2)=imsubtract(im1(:,:,2),double(im2));

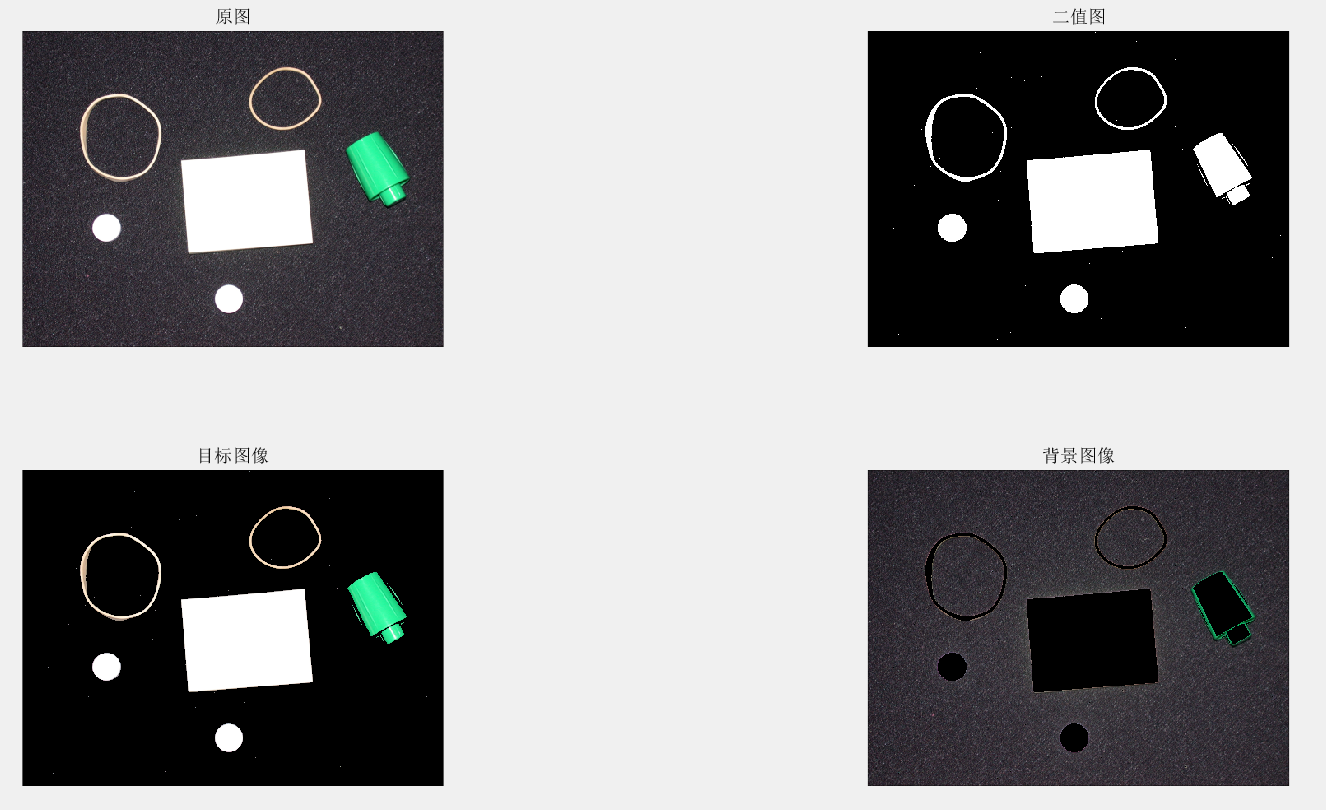
im4(:,:,3)=imsubtract(im1(:,:,3),double(im2));

subplot(2,2,4)

imshow(im4)

title('背景图像')

结果：



（4）读入matlab自带图像pout.tif, 实现对此图像按比例缩小（纵横向缩小效果一样），不按比例缩小（纵横不同），以及图像成倍放大和不按比例放大效果。

代码：

主函数：

clc

clear

im=imread('pout.tif');

[sm,sn]=size(im);

im1=jihe(sm,sn,im,0.5,0.5);

figure(1)

imshow(im)

title('原图')

figure(2)

imshow(im1)

title('等比例缩放')

im2=jihe(sm,sn,im,0.7,0.5);

figure(3)

imshow(im2)

title('不等比例缩放')

im3=jihe(sm,sn,im,2,2);

figure(4)

imshow(im3)

title('等比例放大')

im4=jihe(sm,sn,im,4,2);

figure(5)

imshow(im4)

title('不等比例放大')

函数代码：jihe.m

function [new\_im] = jihe(sm,sn,im,a,b)

dm=round(sm\*a);

dn=round(sn\*b);

new\_im=uint8(zeros(dm, dn));

for i=1:dm

for j=1:dn

sx=round(i\*sm/dm);

sy=round(j\*sn/dn);

if(sx>=1&&sx<=sm&&sy>=1&&sy<=sn)

new\_im(i,j,:)=im(sx,sy,:);

end

end

end

end

结果：

（5）读入matlab自带图像pout.tif, 选择任何一种方式实现对图像的平移，旋转，尺度缩放，水平镜像，垂直镜像，或者自选任意组合实现对图像的几何变换。并显示几何变换的结果。

代码：

主函数：

clc

clear

im=imread('pout.tif');

figure(1)

imshow(im);

title('原始图像');

[M,N]=size(im);

im1=pingyi(M,N,im,20,20);

figure(2)

imshow(im1);

title('平移后的图像');

im2=imrotate(im,25,'bilinear');

figure(3)

imshow(im2)

title('旋转后的图像')

im3=jihe(M,N,im,0.5,1);

figure(4)

imshow(im3)

title('垂直压缩一倍')

im4=jihe(M,N,im,1,0.5);

figure(5)

imshow(im4)

title('水平压缩一倍')

figure(6);

im5=jingxiang(M,N,im,1);

imshow(im5);

title('水平镜像');

figure(7);

im6=jingxiang(M,N,im,2);

imshow(im6);

title('垂直镜像');

im7=jingxiang(M,N,im,1);

im7=pingyi(M,N,im7,20,20);

figure(8)

imshow(im7)

title('组合（垂直镜像+平移）')

函数文件：

jihe.m

function [new\_im] = jihe(sm,sn,im,a,b)

dm=round(sm\*a);

dn=round(sn\*b);

new\_im=uint8(zeros(dm, dn));

for i=1:dm

for j=1:dn

sx=round(i\*sm/dm);

sy=round(j\*sn/dn);

if(sx>=1&&sx<=sm&&sy>=1&&sy<=sn)

new\_im(i,j,:)=im(sx,sy,:);

end

end

end

end

jingxiang.m

function [im\_new] = jingxiang(M,N,im,mode)

if mode==1

g=zeros(M,N);

for i=1:M

for j=1:N

g(i,j)=im(i,N-j+1);

end

end

else

g=zeros(M,N);

for i=1:M

for j=1:N

g(i,j)=im(M-i+1,j);

end

end

end

im\_new=uint8(g);

end

pingyi.m

function [im\_new] = pingyi(M,N,im,a,b)

im1=zeros(M,N);

for i=1:M

for j=1:N

if(i-a>0)&(i-a<M)&(j-b>0)&(j-b<N)

im1(i,j)=im(i-a,j-b);

else

im1(i,j)=0;

end

end

end

im\_new=uint8(im1);

end

结果：

