图像处理作业报告

第一次作业

自动化 66 班 朱子瑜
2160504163

1.

任务：Bmp图像格式简介,以7.bmp为例说明；

思路：用imfinfo函数读取7.bmp的文件信息

运行结果：

```
Filename: 'D:\Program Files\MATLAB\R2016b\bin\imgprocess\firstassignment\7.bmp'
            FileModDate: '03-Jun-2011 11:17:32'
               FileSize: 1134
                 Format: 'bmp'
          FormatVersion: 'Version 3 (Microsoft Windows 3.x)'
                  Width: 7
                 Height: 7
               BitDepth: 8
              ColorType: 'indexed'
        FormatSignature: 'BM'
    NumColormapEntries: 256
               Colormap: [256×3 double]
                RedMask: []
              GreenMask: []
               BlueMask: []
        ImageDataOffset: 1078
      BitmapHeaderSize: 40
             NumPlanes: 1
        CompressionType: 'none'
             BitmapSize: 56
          HorzResolution: 0
          VertResolution: 0
           NumColorsUsed: 0
    NumImportantColors: 0
```

结果分析和说明：

该图片由 7×7 个像素构成，每个像素都是一个 8 位的整型变量。Colormap 是一个 256×3 的双精度浮点数，用来定义图像显示用的颜色查找表。

2.

任务：把 lena 512*512 图像灰度级逐级递减 8-1 显示；

思路：读取 lena 图像，降低像素的位，显示。

运行结果：

1位灰度图像

2位灰度图像

3位灰度图像

4位灰度图像

5位灰度图像

6位灰度图像

7位灰度图像

8位灰度图像

3.

任务：计算 lena 图像的均值方差；

思路：读取图像，通过 mean 和 var 函数直接计算均值和方差。

运行结果：

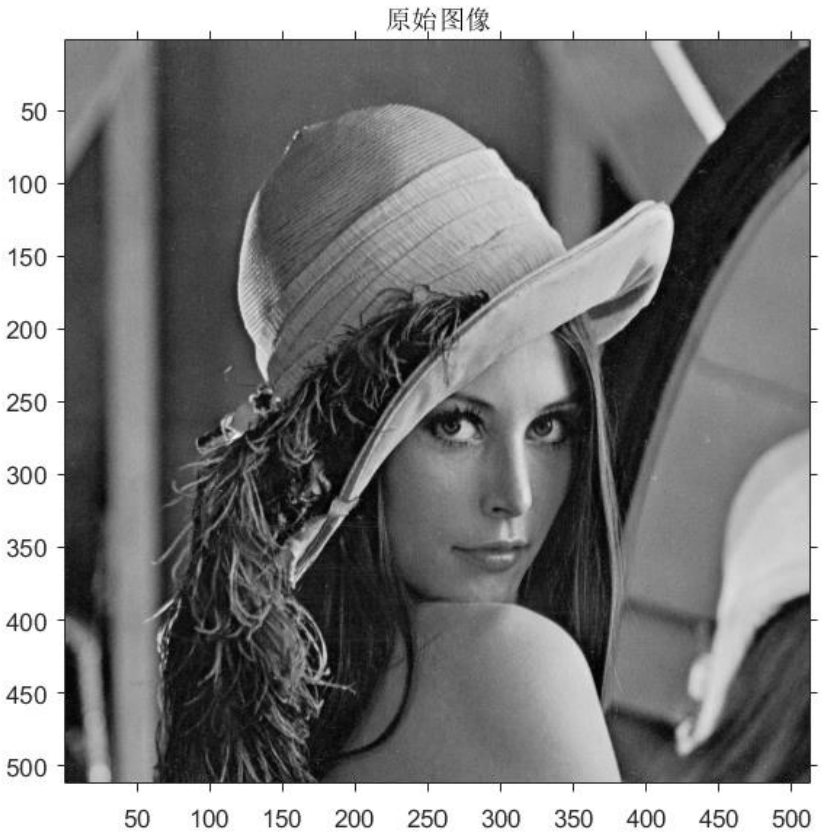
```
i_mean =

    99.0512


s =

    2.7960e+03
```

4.

任务：把 lena 图像用近邻、双线性和双三次插值法 zoom 到 2048*2048；

思路：将放大后的像素点的位置对应到原图的位置，用原图这个位置附近的像素点的值的线性组合算出改该点的值。

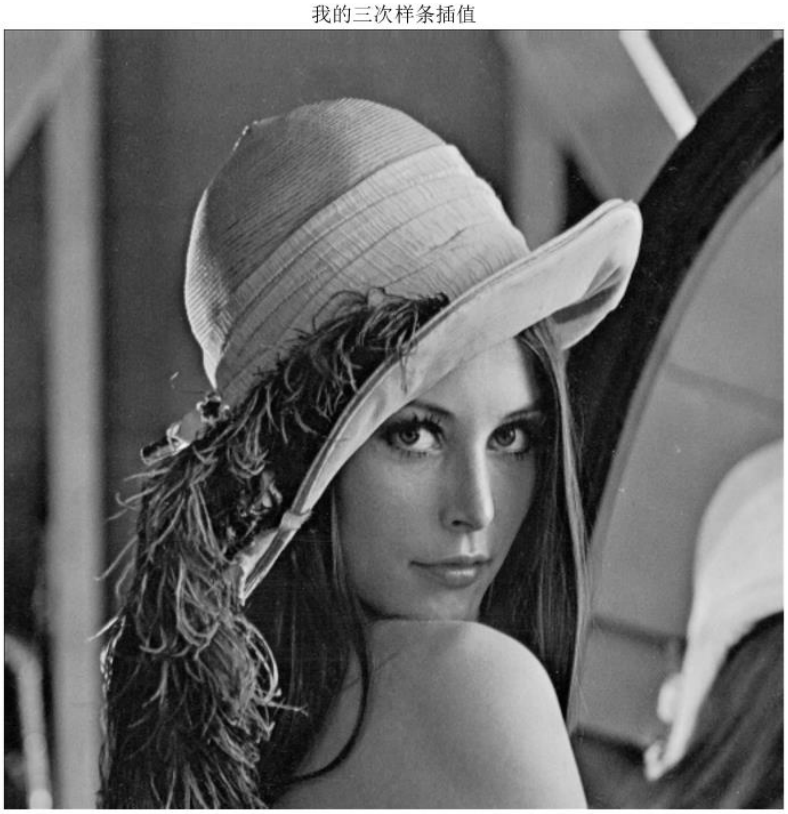
运行结果：




我的最近邻插值
200
400
600
800
1000
1200
1400
1600
1800
2000


我的双线性插值
200
400
600
800
1000
1200
1400
1600
1800
2000

我的三次样条插值



最近邻插值

双线性插值

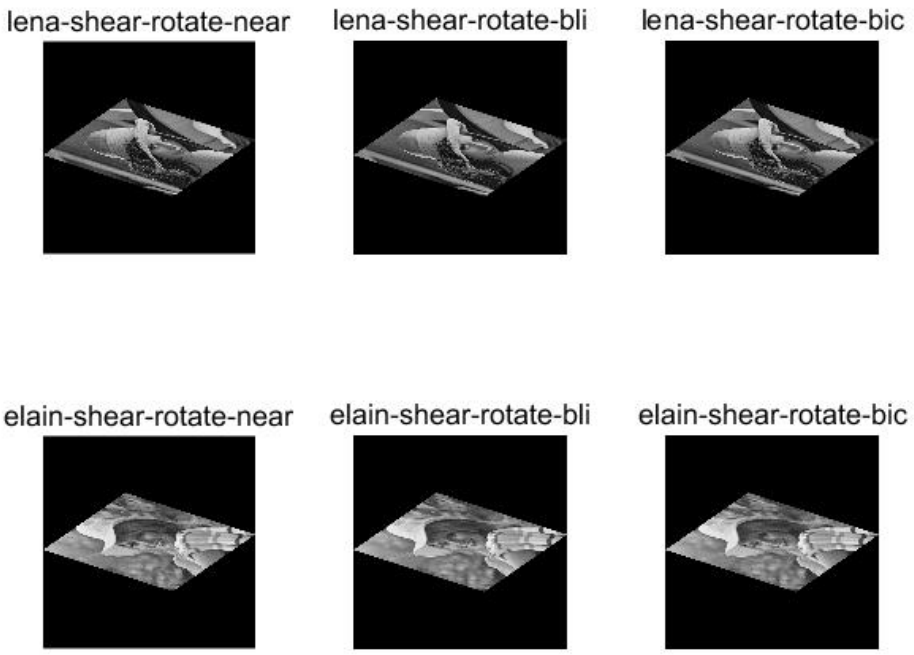
三次样条插值

5.

任务：把 lena 和 elain 图像分别进行水平 shear（参数可设置为 1.5，或者自行选择）和旋转 30 度，并采用用近邻、双线性和双三次插值法 zoom 到 2048*2048；

思路：shear 只是将原图所有像素的坐标转换到新的坐标系下，像素的值并不改变。旋转使用 matlab 自带的 imrotate 函数。Zoom 采用任务 4 中我自己写的插值函数。

运行结果：



源代码：

1.

task1.m

```
%task1, Bmp图像格式简介,以7.bmp为例说明；[X,map,alpha] =
imread('7.bmp');
info = imfinfo('7.bmp')
```

2.

task2.m

```
% task2将图像的位数降低
I=imread('lena.bmp');
for bit=[1:8];
maxint=uint8(2^bit);

% BWI=imbinarize(I);
% BWI=im2bw(I,int);
% BWI=imadjust(I,[0;1],[0;int]);

iafter=(I/maxint)*maxint;
% subplot(2,4,bit)
figure;
imshow(iafter);
title(sprintf('%d灰度图像',(9-bit)));

end
```

3.

task3.m

```
%task3求图像的均值方差
I=imread('lena.bmp');
i=double(I);
i_mean=mean2(i)%均值
s=var(i(:))%方差
```

4.

task4.m

```
%task4,插值
I=imread('lena.bmp');

figure
% subplot(2,4,1);
imshow(I);
axis on
title('原始图像')

I1=nearzoom(I,4);%最近邻插值
figure
% subplot(2,4,2);
imshow(I1);
axis on
```

```
title('我的最近邻插值')

I2=blizoom(I,4);%双线性插值
figure
% subplot(2,4,3);
imshow(I2);
axis on
title('我的双线性插值')

I3=biczoom(I,4);%三次样条插值
figure
% subplot(2,4,4);
imshow(I3);
title('我的三次样条插值')

I1_2=interp2(double(I),4,'nearest');%最近邻插值
I1_2=uint8(I1_2);
figure
% subplot(2,4,6);
imshow(I1_2);
axis on
title('最近邻插值')

I2_2=interp2(double(I),4,'linear');%双线性插值
```

```
I2_2=uint8(I2_2);
figure
% subplot(2,4,7);
imshow(I2_2);
axis on
title('双线性插值')

I3_2=interp2(double(I),4,'spline');%三次样条插值
I3_2=uint8(I3_2);
figure
% subplot(2,4,8);
imshow(I3_2);
axis on
title('三次样条插值')
```

nearzoom.m

```
%task4_1,实现近邻插值放大图片
function imt=nearzoom(image,xscale,yscale)
%nearzoom(image,scale)
%or nearzoom(image,width,height)
[xs,ys] = size(image);

if nargin == 2
    yscale=xscale;
```

```
else if nargin ==3
      xscale=xscale/xs;
      yscale=yscale/ys;
    end
end
if xscale <= 0
     xscale('缩放倍数 scale的值应该大于0!');
end

imt=zeros(ceil(xs*xscale),ceil(ys*yscale));
imt=uint8(imt);
for i=1:(ceil(xs*xscale))  %amplify th image

    if ceil(i/xscale)>xs  %防止溢出，因为xscale变成double型缩小了，除以XScale之后再进一法取整可能会溢出，下同
        i=i-1;
    end

    for j=1:ceil((ys*yscale))

         if ceil(j/yscale)>ys
        i=i-1;
         end

        imt(i,j) = image(ceil(i/xscale),ceil(j/yscale));
```

```
    end
end

%  figure,imshow(imt);
%  axis on
```

Blizoom.m

```
%task4_2,实现双线性插值放大图片
function imt=blizoom(image,xscale,yscale)
%blizoom(image,scale)
%or blizoom(image,width,height)

[xs,ys]=size(image);

if nargin == 2
    yscale=xscale;
else if nargin ==3
      xscale=xscale/xs;
      yscale=yscale/ys;
    end
end

if xscale<=0||yscale<=0
     sprintf('缩放倍数 scale的值应该大于0!');
```

```
end

imt=zeros(ceil(xs*xscale),ceil(ys*yscale));

image=double(image);
imagez=zeros(xs+2,ys+2);%给原图像周围加0
for i=2:xs+1
    for j=2:ys+1
        imagez(i,j)=image(i-1,j-1);
    end
end

for zi=1:(xs*xscale)  %amplify th image
    for zj=1:(ys*yscale)
        x=zi/xscale;
        y=zj/yscale;%放大后图片对应原图的坐标
        i=floor(x);%对应到原图上的坐标
        u=x-i;
        j=floor(y);
        v=y-j;
        i = i + 1; j = j + 1;
        imt(zi,zj) = (1-u)*(1-v)*imagez(i,j)+ (1-
u)*v*imagez(i,j+1)+u*(1-v)*imagez(i+1,j)+u*v*imagez(i+1,j+1);
```

```
    end
end
imt=uint8(imt);
% figure,imshow(imt);
% axis on
```

biczoom.m

```
%task4_3,实现三次插值放大图片
function imt=biczoom(image,xscale,yscale)
%biczoom(image,scale)
%or biczoom(image,width,height)

[xs,ys]=size(image);

if nargin == 2
    yscale=xscale;
else if nargin ==3

      xscale=xscale/xs;
      yscale=yscale/ys;
    end
end

if xscale<=0
```

```
    sprintf('缩放倍数 scale的值应该大于0!');
end

imt=zeros(ceil(xs*xscale),ceil(ys*yscale));
image=double(image);
imagez=zeros(xs+4,ys+4);%给原图像周围加0
for i=3:xs+2
    for j=3:ys+2
        imagez(i,j)=image(i-2,j-2);
    end
end


for zi=1:(xs*xscale)  %amplify th image
    x=zi/xscale;
    i=floor(x);%i,j为对应到原图上的坐标
     u=x-i;
     i = i + 2;
     A=[sw(1+u) sw(u) sw(1-u) sw(2-u)];
    for zj=1:(ys*yscale)
        y=zj/yscale;%x，y为放大后图片对应原图的坐标
        j=floor(y);
        v=y-j;
```

```
        j = j + 2;
        C=[sw(2-v);sw(1-v);sw(v);sw(1+v)];
        B=[imagez(i-1:i+2,j+2),imagez(i-1:i+2,j+1),imagez(i-1:i+2,j),imagez(i-1:i+2,j-1)];
        imt(zi,zj)=A*B*C;
    end
end
imt=uint8(imt);
% figure,imshow(imt);
% axis on
```

sw.m

```
%三次插值的权重函数
function A=sw(w1)
w=abs(w1);
a=-0.5;
if w<1&&w>=0
  A=1-(a+3)*w^2+(a+2)*w^3;
else if w>=1&&w<2
  A=a*w^3-5*a*w^2+(8*a)*w-4*a;
else
  A=0;
```

```
    end
end
end
```

5.

task5_1.m

```
%task5,把lena和elain图像分别进行水平shear（参数可设置为1.5，或者自行选择）和旋转30度，并采用用近邻、双线性和双三次插值法zoom到2048*2048；
I1=imread('lena.bmp');
I2=imread('elain1.bmp');
shearmat=[1,1;0,1];
I1=shear(I1,shearmat);
I2=shear(I2,shearmat);
I1=imrotate(I1,30,'bicubic','loose');
I1_n=nearzoom(I1,2048,2048);
I1_bli=blizoom(I1,2048,2048);
I1_bic=biczoom(I1,2048,2048);
I2=imrotate(I2,30,'bicubic','loose');
I2_n=nearzoom(I2,2048,2048);
I2_bli=blizoom(I2,2048,2048);
I2_bic=biczoom(I2,2048,2048);
```

```
figure

subplot(231),imshow(I1_n);title('lena-shear-rotate-near');

subplot(232),imshow(I1_bli);title('lena-shear-rotate-bli');

subplot(233),imshow(I1_bic);title('lena-shear-rotate-bic');

subplot(234),imshow(I2_n);title('elain-shear-rotate-near');

subplot(235),imshow(I2_bli);title('elain-shear-rotate-bli');

subplot(236),imshow(I2_bic);title('elain-shear-rotate-bic');
```

shear.m

```
%task5,把lena和elain图像分别进行水平shear（参数可设置为1.5，或者自行选择）和旋转30度，并采用用近邻、双线性和双三次插值法zoom到2048*2048;
function IS=shear(image,shearmat);
%shearmat should be a 2x2 mat:[1,0;m,1]
[xs,ys]=size(image);
IS=zeros(xs+shearmat(2,1)*ys,shearmat(1,2)*xs+ys);
IS=uint8(IS);
for i=1:xs
    for j=1:ys
        IS(i+j*shearmat(2,1),shearmat(1,2)*i+j)=image(i,j);
    end
end
```

```
% figure

% imshow(IS);

% axis on
```