数字图像处理第一次作业

自动化 63 边策 2160504062

2019-3-3

摘要：本次实验主要是基于 bmp 图像格式对图像进行像素级的处理，在 matlab 环境下完成。通过对灰度值进行变化，做到降低分度级，图片放大后的插值以及对图片进行变换这几项工作。

报告正文

问题讨论

第一问：典型的 BMP 图像文件由四部分组成：

1．位图头文件数据结构，它包含 BMP 图像文件的类型、显示内容等信息；

2．位图信息数据结构，它包含有 BMP 图像的宽、高、压缩方法，以及定义颜色等信息；

3．调色板，这个部分是可选的，有些位图需要调色板，有些位图，比如真彩色图（24 位的 BMP）就不需要调色板；

4．位图数据，这部分的内容根据 BMP 位图使用的位数不同而不同，在 24 位图中直接使用 RGB，而其他的小于 24 位的使用调色板中颜色索引值。

针对 7.bmp 来说，图像位图数据是一个 7*7 的矩阵，矩阵的数值就代表着该点的灰度值，这是图片的主要信息。

第二问：将 lena 图像灰度级逐级递减并打印。因为图像内容是无符号整形（uint8）所以可以直接对图像矩阵 tuxiang 进行处理。利用 matlab 独特的矩阵处理方法 tuxiang. / (2^i) 通过对矩阵数值压缩 2 的 i 次幂，进行级数的递减，但是这会让图像变黑，所以之后还要. *(2^i)就可以实现极度递减，并输出图像一样了。

第三问：通过概率所学方差计算公式：

$$\sigma = \sqrt{\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}(f(x,y)-\varepsilon)^2}$$

可以利用 matlab 矩阵的运算和自带的 sum()函数，可以使得程序简单。

第四问：对 lena 图像用三种方法进行插值。对于这个问题首先要解决的一个问题就是要将 512*512 的矩阵扩大为 2048*2048 的矩阵，矩阵放大后，对应的点位不变而扩充的待填充的点为-1。解决第二个问题就是要怎样定义像素的距离，我通过城市间距来定义距离即

$$\text{distance} = |s - x| + |t - y|$$

第三个问题是如何寻找最近的距离，我是这么做的：建立一个计算距离的函数 dis（）参数为1.展开的图片2.当前像素点在展开图片的坐标。通过展开图片可知最近的点一定在当前点的9*9邻域里（有的时候当前点展开不了9*9）建立距离矩阵`ones(min(x+4,9),min(y+4,9))*100;`计算灰度值非-1的点到当前点的距离，并依次输入到矩阵中。返回矩阵的值和当前点在距离矩阵的位置x0 y0

第四个问题就是计算最近的1（4或9）的值，如果是最近邻就最近的值是多少就填充多少。如果是双线性内插则就解四元方程得到结果。双三次插值同理，但是因为计算时间太长我用了系统自带的函数

第五问：这一问我用的是matlab自带的函数包`maketform()`和`imtransform()`

来进行转换，设置好转换矩阵就可以将矩阵再扩充至`2048*2048`的图像，同第

四问既可以解决

反思与疑问

1. 在本次实验中第二问我们按理来说图像矩阵除以 128 就可以得到灰度级为 1 的图像，但是图像有三种灰度值：0，128，255，除以 256 才可以得到灰度级为 1 的图像。对这个问题一直没有得到很好的解决。
2. 在本次实验中，第四问的双线性插值，常常在求解四元方程时调用 solve 函数得不到解析解从而导致程序出错。想知道为什么。
3. 如果第五问用自己编的函数，旋转后的图像如何确定大小，如何确定旋转后坐标非整数的点的坐标。

实验结果

第二问







第三问

average = 99.0512

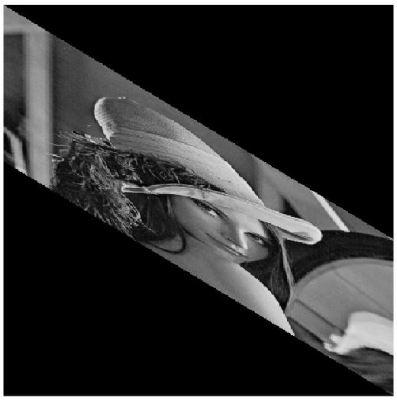
D = 52.8775

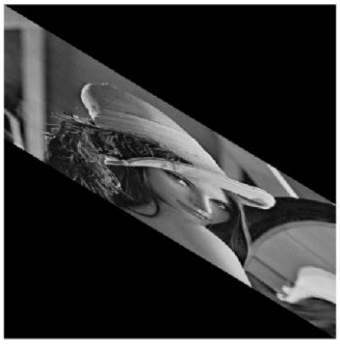
第四问

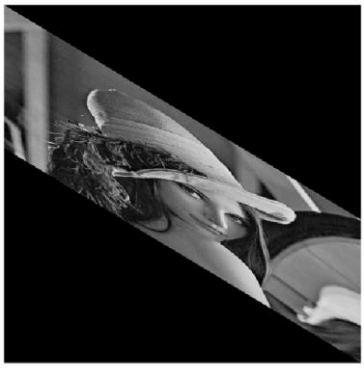
依次为最近邻/双线性插值/双三次插值


第五题第一问

分别为 lena 进行水平 shera 和旋转 30 的结果，并附上三种插值

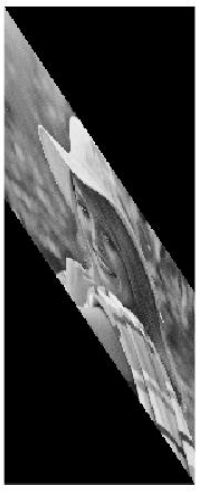


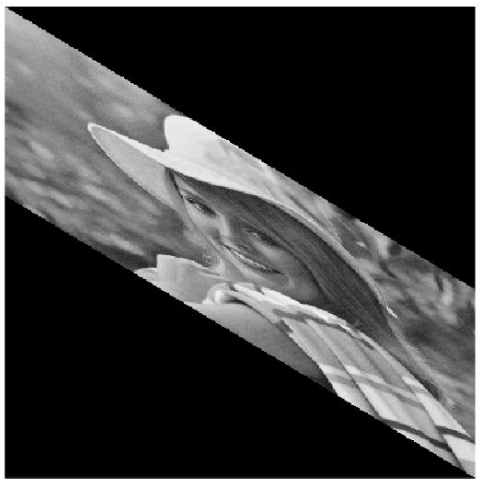


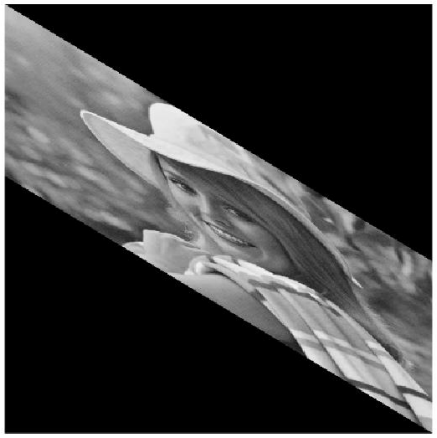


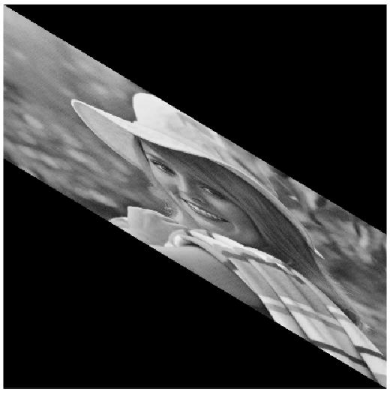
第五题第二问

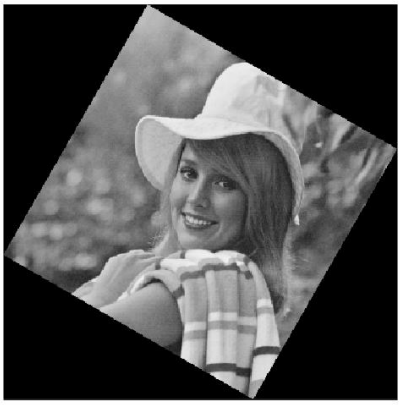
分别为 elein 依次进行水平 shera 和旋转 30 的结果，并附上三种插值

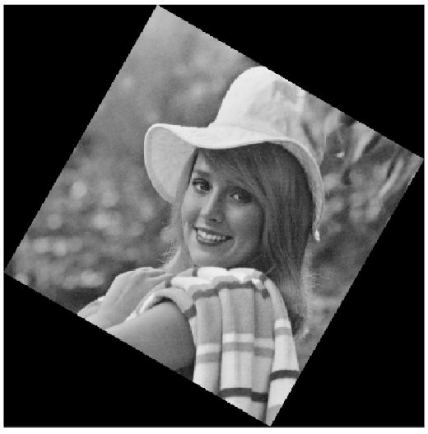












附录：

第二问：

Dierwen.m

```
clear;
tupian=imread('C:\Users\15pr\Desktop\µÚ¶þ´Î×÷Òµ\lena.bmp');
% tupian=double(tupian);
pic1=tupian./2.*2;
pic2=tupian./4.*4;
pic3=tupian./8.*8;
pic4=tupian./16.*16;
pic5=tupian./32.*32;
pic6=tupian./64.*64;
pic7=tupian./128.*128;
pic8=tupian./256.*256;
figure(1);
imshow(pic1);
figure(2);
imshow(pic2);
figure(3);
imshow(pic3);
figure(4);
imshow(pic4);
figure(5);
imshow(pic5);
figure(6);
imshow(pic6);
figure(7);
imshow(pic7);
figure(8);
imshow(pic8);
```

第三问：

Disanwen.m

```
clear;
tupian=imread('C:\Users\15pr\Desktop\µÚ¶þ´Î×÷Òµ\lena.bmp');
tupian=double(tupian);
average=sum(sum(tupian))/(512*512)
a=(tupian-average).^2;
D=sqrt(sum(sum(a))/512/512)
```

第四问：

主函数：general.m

```
clear;
tupian=imread('C:\Users\15pr\Desktop\µÚ¶þ´Î×÷Òµ\lena.bmp');
img=tran_big(tupian);
 pic1=insert(img,1);
```

```
 pic1=uint8(pic1);
figure(1)
 imshow(pic1);
 pic2=insert(img,2);
figure(2)
imshow(pic2);
pic3=imresize(tupian,[2048,2048],'bicubic');
figure(3)
imshow(pic3);
```

函数 1：

Insert.m

```
function outputpic=insert(img,kind)

for i=1:1:2048
    for j=1:1:2048
        if(img(i,j)==-1)
%
if(kind==1)
    img(i,j)=near(img,i,j);
end
if(kind==2)
          img(i,j)=liner(img,i,j);
end
       end
    end
end
outputpic=uint8(img);
end
```

函数 2

Diss.m

```
function [distance_M,x0,y0]=dis(tupian,x,y)
%%%%%%class=1ÊÇ5*5ÀïÕÒÒ»¸öclass=2ÊÇ9*9
%%%%%%%%%¾àÀë¶ÈÁ¿²ÉÓÃ³ÇÊÐ½ÖÇø¾àÀë%%%%%%%%%%%
distance_M=ones(min(x+4,9),min(y+4,9))*100;
i=0;
for(s=max(1,x-4):1:min(x+4,2048))
    i=i+1;
    j=0;
    for(t=max(1,y-4):1:min(y+4,2048))
        j=j+1;
        if(s==x&&t==y)
            x0=i;
            y0=j;
        end
```

```
        if(tupian(s,t)~=-1)
            distance=abs(x-s)+abs(y-t);
            distance_M(i,j)=distance;
        end
    end
end
end
```

函数 3：

Findnear.m

```
function [X,Y]=findnear(tupian,x,y,num)
%%%%%%%%%%ÕÒ×î½üµÄnum¸öµã£¬xyÎªµ±Ç°Î»ÖÃµÄÏñËØ
[distance,x0,y0]=dis(tupian,x,y);
t=sort(distance(:));
[x1,y1]=find(distance<=t(num),num);
for(i=1:1:num)
    X(i)=x-x0+x1(i);
    Y(i)=y-y0+y1(i);
end
end
```

函数 4：

Near.m

```
function grey=near(img,x,y)
[X,Y]=findnear(img,x,y,1);
grey=img(X(1),Y(1));
end
```

函数 5：

Liner.m

```
function grey=liner(img,x,y)
[X,Y]=findnear(img,x,y,4);
syms a b c d
eq1=a*X(1)+b*Y(1)+c*X(1)*Y(1)+d-img(X(1),Y(1));
eq2=a*X(2)+b*Y(2)+c*X(2)*Y(2)+d-img(X(2),Y(2));
 eq3=a*X(3)+b*Y(3)+c*X(3)*Y(3)+d-img(X(3),Y(3));
 eq4=a*X(4)+b*Y(4)+c*X(4)*Y(4)+d-img(X(4),Y(4));
[a,b,c,d]=solve(eq1,eq2,eq3,eq4,a,b,c,d);
a=double(a);
b=double(b);
c=double(c);
d=double(d);
grey=a*x+b*y+d;
end
```

函数 6：tran_big.m

```
function img_big=tran_big(tupian)
```

```
%%%%%%%%%%ÕâÊÇ°ÑÍ¼Æ¬·Å´óµÄ°¯Êý%%%%%%%%%%%%%
img_big=(-1)*ones(2048,2048);
for i=1:1:512
    for j=1:1:512
        img_big(4*i-3,4*j-3)=tupian(i,j);
    end
end
%img_big=uint8(img_big);
end
```

第五问：

```
clear;
tupian=imread('C:\Users\15pr\Desktop\µÚ¶þ´Î×÷Òµ\lena.bmp');
T1=[1 1.5 0
    0 1 0
    0 0 1];
T2=[cos(pi/6) sin(pi/6) 0
    -sin(pi/6) cos(pi/6) 0
    0 0 1];
transform1=maketform('affine',T1);
transform2=maketform('affine',T2);
tupian1=imtransform(tupian,transform1);
tupian2=imtransform(tupian,transform2);
pic1=imresize(tupian1,[2048,2048],'nearest');
pic2=imresize(tupian1,[2048,2048],'bilinear');
pic3=imresize(tupian1,[2048,2048],'bicubic');
pic4=imresize(tupian2,[2048,2048],'nearest');
pic5=imresize(tupian2,[2048,2048],'bilinear');
pic6=imresize(tupian2,[2048,2048],'bicubic');
figure(1);
imshow(pic1);
figure(2);
imshow(pic2);
figure(3);
imshow(pic3);
figure(4);
imshow(pic4);
figure(5);
imshow(pic5);
figure(6);
imshow(pic6);
figure(7);
imshow(tupian1);
 figure(8);
imshow(tupian2);
```

参考文献：数字图像处理：第三版/（美）拉斐尔·冈萨雷斯，（美）理查德·伍兹著；阮秋琦等译.——北京：电子工业出版社，2017.5