数字图像处理

第一次作业

姓名：张勤东

班级：自动化 63

学号：2160504080

日期：2019 年 3 月 4 日

摘要：本报告主要解决了数字图像基础部分的一些问题。首先，对 bmp 图像格式进行了简要的介绍。然后，利用 MATLAB 软件，对图像进行了不同灰度级的显示，计算原始图像的均值和方差，并使用近邻、双线性和双三次插值法对原始图像进行了放大。最后，通过构造仿射矩阵，实现图像的旋转变换和水平偏移变换，并使用近邻、双线性和双三次插值法对图像进行了放大。

1. Bmp 图像格式简介,以 7.bmp 为例说明；

BMP（全称 Bitmap）是 Windows 操作系统中的标准图像文件格式，使用非常广。它采用位映射存储格式，除了图像深度可选以外，不采用其他任何压缩，因此，BMP 文件所占用的空间很大。BMP 文件的图像深度可选 lbit、4bit、8bit 及 24bit。BMP 文件存储数据时，图像的扫描方式是按从左到右、从下到上的顺序。由于 BMP 文件格式是 Windows 环境中交换与图有关的数据的一种标准，因此在 Windows 环境中运行的图形图像软件都支持 BMP 图像格式。

BMP 文件的数据按照先后顺序可以分为四个部分：位图头文件、位图信息头、调色板和位图数据。其中，位图头文件提供文件的格式、大小等信息，共 14 字节；位图信息头提供图像数据的尺寸、位平面数、压缩方式、颜色索引等信息，共 40 字节；调色板是可选的，如使用索引来表示图像，调色板就是索引与其对应的颜色的映射表；位图数据存放图像数据。

图片 7.bmp 分辨率为 7*7，该图片共由 49 个像素构成。位深度为 8，表示用来记录每个像素灰度的位数为 8 位。

属性	值
图像	
分辨率	7 x 7
宽度	7 像素
高度	7 像素
位深度	8
文件	
名称	7.bmp
项目类型	BMP 文件

2. 把 lena 512*512 图像灰度级逐级递减 8-1 显示；

解决思路：

图片 lena.bmp 为分辨率是 512*512，灰度级为 8 位的图片。首先，先用 MATLAB 中的 imread()函数将原始图像以矩阵形式读取出来。

矩阵中的数据记录着图像在各个像素处的灰度信息。原图像灰度级为 8 位，故矩阵中的数据范围是 0 到 255。改变图片的灰度级，就是要改变存放灰度信息的位数，也就是要改变灰度数据的范围。首先，对原始数据进行归一化处理，将数据由 0 到 255 范围区间变到 0 到 1 的范围区间。然后，根据改变后的灰度级对应的数据范围，将 0 到 1 区间的数据变到新的区间范围。这一过程可以用以下的公式表示：

$$f_s = K \times \frac{f_m - 0}{255}$$

其中，f_m为原图像矩阵中的数值，f_s为灰度降低后图像矩阵中的数值，K表示新图像的数据范围的上界（例如：灰度级为 7 位的图像，对应的K为 127）。

通过公式计算出灰度级降低后的矩阵值后，利用 imshow()函数将图像打印出来。

处理结果：

灰度级为8

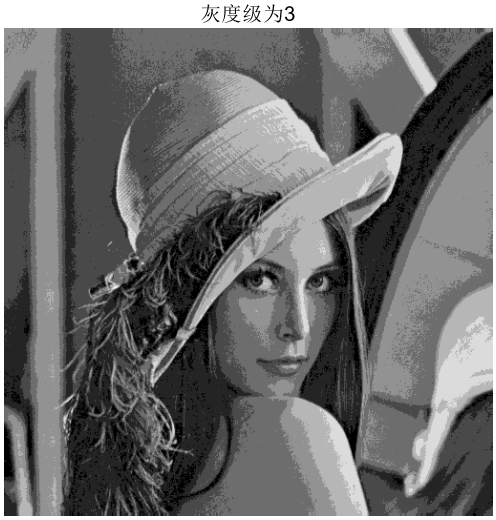
灰度级为7

灰度级为6

灰度级为5

灰度级为4

灰度级为3



灰度级为2

灰度级为1

结果分析：

以上 8 幅图片的灰度级逐渐降低，图片的细节损失越多。并且，在灰度级为 4 位以下时，图片的伪轮廓逐渐明显，这是由于在图片平滑区域中灰度级数不足造成的。通过对比以上 8 张图片可以看出灰度分辨率对图片分辨细节的影响。

3. 计算 lena 图像的均值方差；

解决思路：

首先，先用 MATLAB 中的 imread()函数将原始图像以矩阵形式读取出来。

此时，矩阵中存放着图像的灰度信息。计算图像的均值和方差，就是计算图像灰度的均值和方差。利用 mean()函数可以计算出矩阵中元素的平均值，利用 var()函数可以计算出矩阵中元素的方差。

处理结果：

使用 MATLAB 计算均值方差结果如下：

junzhi =99.0512

fangcha =2.7960e+03

结果分析：

计算出的平均值为 99.0512，表示该图像的灰度平均在 99.0512 左右该值可以用来评价图像的平均灰度水平。

计算出的方差为2.7960×10^3，该值可以用来衡量图像的灰度对比度，由该数据可知这幅图像的对比度较高。

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

解决思路：

首先，先用 MATLAB 中的 imread()函数将原始图像以矩阵形式读取出来。原图像为 512*512，将图像扩充到 2048*2048，就是对图像进行坐标的空间变换并进行灰度内插。使用 imresize()函数可以实现采用一定的内插方法对原图片的尺寸进行放大。其中，近邻法为 nearest，双线性法为 bilinear，双三次法为 bicubic。最后，用 imshow()函数将结果打印出来。

处理结果：

使用近邻法

使用双线性法

使用双三次法

结果分析：

以上三幅图像分别为使用近邻、双线性和双三次插值法将 512*512 的图像放大至 2048*2048 后的结果。

比较以上三幅图像，可以看出使用最近邻内插法得到的图片效果较差，在局部区域出现了锯齿，使用双线性内插法得到的结果明显有所改进，而双三次内插法得到的图片比双线性内插得到的图片又稍微清晰一些。

5. 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；

解决思路：

首先，先用 MATLAB 中的 imread()函数将原始图像以矩阵形式读取出来。为了进行水平偏移变换和旋转变换，先构造对应的仿射变换矩阵。

水平偏移变换矩阵为：

$$\text{shear} = \begin{bmatrix} 1 & 1.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

旋转 30 度的变换矩阵为：

$$\text{rotate} = \begin{bmatrix} \cos(\pi/6) & \sin(\pi/6) & 0 \\ -\sin(\pi/6) & \cos(\pi/6) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

然后，使用以下公式可以实现图像的仿射变换。

$$[x \quad y \quad 1] = [v \quad w \quad 1] \times T$$

其中，T 为仿射变换矩阵，(v,w)为原始图像点坐标，(x,y)为变换后图像对应的点坐标。

利用 affine2d()函数，构建与水平偏移变换矩阵和旋转 30 度的变换矩阵对应的仿射矩阵，并使用 imwarp()函数，对原图像进行相应的仿射变换。

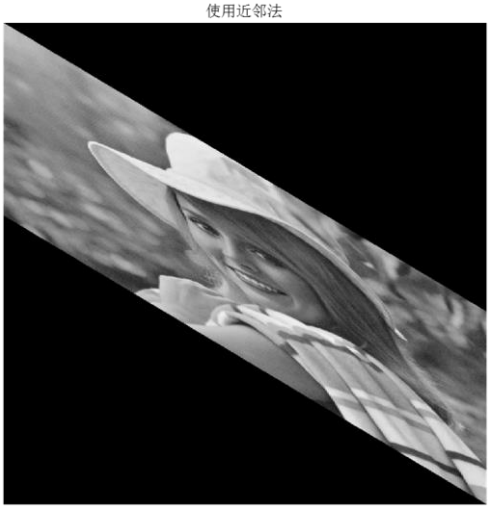
使用 imresize()函数可以实现采用一定的内插方法对仿射变换后的图像的尺寸进行放大。其中，近邻法为 nearest，双线性法为 bilinear，双三次法为 bicubic。最后，用 imshow()函数将结果打印出来。

处理结果：

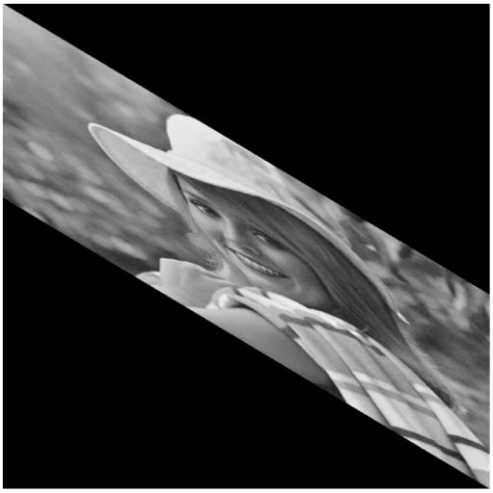
使用双线性法

使用双三次法

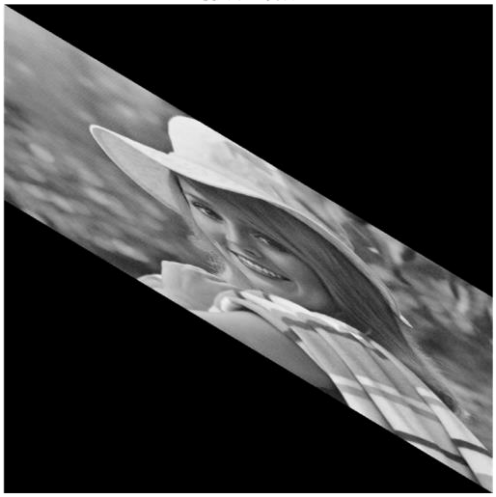
使用近邻法



使用双线性法



使用双三次法



使用近邻法

结果分析：

通过仿射变换矩阵，可以实现水平偏移变换和旋转变换。同时，比较以上图像，可以看出使用最近邻内插法得到的图片效果较差，在局部区域出现了锯齿，使用双线性内插法得到的结果明显有所改进，而双三次内插法得到的图片比双线性内插得到的图片又稍微清晰一些。

附录

第二题代码

```
pic=imread('lena.bmp');
figure(1)
imshow(pic,[0,255])
title('灰度级为 8')

pic1=(127/255)*pic;
figure(2)
imshow(pic1,[0,127])
title('灰度级为 7')

pic2=(63/255)*pic;
figure(3)
imshow(pic2,[0,63])
title('灰度级为 6')

pic3=(31/255)*pic;
figure(4)
imshow(pic3,[0,31])
title('灰度级为 5')

pic4=(15/255)*pic;
figure(5)
imshow(pic4,[0,15])
title('灰度级为 4')

pic5=(7/255)*pic;
figure(6)
imshow(pic5,[0,7])
title('灰度级为 3')

pic6=(3/255)*pic;
figure(7)
imshow(pic6,[0,3])
title('灰度级为 2')

pic7=(1/255)*pic;
figure(8)
imshow(pic7,[0,1])
title('灰度级为 1')
```

第三题代码

```
pic=imread('lena.bmp');
pic=double(pic);
junzhi=mean(pic(:))
fangcha=var(pic(:))
```

第四题代码

```
pic=imread('lena.bmp');

pic1=imresize(pic,[2048,2048],'nearest');
figure(1)
imshow(pic1)
title('使用近邻法')

pic2=imresize(pic,[2048,2048],'bilinear');
figure(2)
imshow(pic2)
title('使用双线性法')

pic3=imresize(pic,[2048,2048],'bicubic');
figure(3)
imshow(pic3)
title('使用双三次法')
```

第五题代码

```
pic1=imread('lena.bmp');
pic2=imread('elain1.bmp');

shear=[1 1.5 0;0 1 0;0 0 1];
rotate=[cos(pi/6) sin(pi/6) 0;-sin(pi/6) cos(pi/6) 0;0 0 1];

shear1=affine2d(shear);
rotate1=affine2d(rotate);

pic3=imwarp(pic1,shear1);

pic11=imresize(pic3,[2048,2048],'nearest');
figure(1)
imshow(pic11)
title('使用近邻法')

pic12=imresize(pic3,[2048,2048],'bilinear');
figure(2)
imshow(pic12)
```

```
title('使用双线性法')

pic13=imresize(pic3,[2048,2048],'bicubic');
figure(3)
imshow(pic13)
title('使用双三次法')

pic4=imwarp(pic1,rotate1);

pic21=imresize(pic4,[2048,2048],'nearest');
figure(4)
imshow(pic21)
title('使用近邻法')

pic22=imresize(pic4,[2048,2048],'bilinear');
figure(5)
imshow(pic22)
title('使用双线性法')

pic23=imresize(pic4,[2048,2048],'bicubic');
figure(6)
imshow(pic23)
title('使用双三次法')


pic5=imwarp(pic2,shear1);

pic31=imresize(pic5,[2048,2048],'nearest');
figure(7)
imshow(pic31)
title('使用近邻法')

pic32=imresize(pic5,[2048,2048],'bilinear');
figure(8)
imshow(pic32)
title('使用双线性法')

pic33=imresize(pic5,[2048,2048],'bicubic');
figure(9)
imshow(pic33)
title('使用双三次法')

pic6=imwarp(pic2,rotate1);
```

```
pic41=imresize(pic6,[2048,2048],'nearest');
figure(10)
imshow(pic41)
title('使用近邻法')

pic42=imresize(pic6,[2048,2048],'bilinear');
figure(11)
imshow(pic42)
title('使用双线性法')

pic43=imresize(pic6,[2048,2048],'bicubic');
figure(12)
imshow(pic43)
title('使用双三次法')
```