

# 数字图像处理

## 第一次作业

姓名：赵国梁

班级：自动化 63

学号：2160504081

提交日期：2019 年 3 月 4 日

## 摘要

本次报告是数字图像处理课程的第一次实验作业,通过对 lena.bmp,elain.bmp 图像文件的编程处理,分别得到了 lena.bmp 图像的 8 到 1 级灰度逐级递减显示, lena.bmp 图像的均值和方差,通过近邻、双线性和双三次插值法对 lena.bmp 进行 4 倍放大后得到的  $2048 \times 2048$  尺寸图像,和对 lena.bmp、elain.bmp 图像分别进行水平偏移变换和旋转变换后的图像及其 4 倍插值放大图像。

## 1.Bmp 图像的格式

### 1.1 BMP 定义

BMP 是 Windows 操作系统中的标准图像文件格式，可以分成两类：设备相关位图（DDB）和设备无关位图（DIB），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 1bit、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。

1.2 以 7.bmp 为例查询信息如下所示：

分辨率	7 x 7		
宽度	7 像素		
高度	7 像素	大小	1.10 KB
位深度	8	属性	A

图像为 8 位的，包含调色板，图像宽和高均为 7。BMP 文件头所占大小为固定的 14Bytes，位图信息头所占大小为固定的 40Bytes，共有  $2^8=256$  种颜色，每个颜色占用 4Bytes，所以调色板所占大小为 1024Bytes。位图数据所占大小为图像宽×图像高×位深度： $7 \times 7 \times 8 \text{bits} = 49 \text{Bytes}$ ，但考虑到 BMP 文件采用了数据对齐，要求每行数据的长度为 4 的倍数，所以实际所占大小为  $7 \times 8 = 56 \text{Bytes}$ 。7.bmp 文件大小为 1.10KB。

### 1.3 BMP 格式组成

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。

## 2.把 lena512\*512 图像灰度级逐级递减 8-1 显示。

（1）问题分析：把 lena 512\*512 图像灰度级逐级递减 8-1 显示。利用 matlab 先将 lena.bmp 读入，再用 imshow 函数将读取的图片显示出来。通过 floor 函数将图片的每个像素的灰度值实现除以 2 的操作

（2）MATLAB 函数：

**A=imread('filename')**

用 imread 函数读取图像文件。读取的是一个二维数组，存放在矩阵 A 中。  
调用格式：

A=imread(filename,fmt)

**imshow(A)**

imshow 是 matlab 中显示图像的函数。

调用格式：

imshow(I,n)

(2) 处理结果:

8 灰度级 (原图):

原图



7 灰度级:

7灰度级



6 灰度级:

6灰度级



5 灰度级:

5灰度级



4 灰度级:

4灰度级



3 灰度级:

3灰度级



2 灰度级:

2灰度级



1 灰度级:

1灰度级



(4) 结果分析：前四张图灰度级数较高，图像基本看不出什么变化。当灰度级数继续降低，则在灰度缓变区常会出现一些几乎看不出来的非常细的山脊状结构。灰度级数越低，越不能将图像的细节刻画出来。这种效果是由数字图像的平滑区域中的灰度级数不足引起的。对比实验中处理过的图像，不同的灰度图，随着灰度范围越大则图像显示出的色彩越丰富。

### 3.计算 lena512 图像的均值方差。

(1) 问题分析：所要实现的功能是：计算图像‘lena.bmp’的均值与方差。

(2) MATLAB 函数：

### **mean2 ( )**

MATLAB 中提供均值计算函数 `mean ( )` 和 `mean2 ( )`。`mean2( A )`，用于对整一个矩

阵求像素平均值。

调用格式：

`u=mean2 ( A )`

### **std2 ( )**

MATLAB 中提供标准差计算函数 `std ( )` 和 `std2 ( )`。

调用格式：

`s=std2 ( A )`

(3) 处理结果：

均值 `average= 99.0512` 方差 `fc= 52.8776`

(4) 结果分析：方差越大，图像的对比度越大，显示的细节就越多。

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

(1) 问题分析：把 lena 图像用近邻、双线性和双三次插值法 zoom 到 2048\*2048。调用 `imresize` 函数用于对图像做缩放处理，实现把图像的近邻，双线性和双三次插值法 zoom 到 2048\*2048。

(2) MATLAB 函数：

`imresize ( )` 函数用于对图像做缩放处理。

调用格式：

`B = imresize(A, [numrows numcols],method)`

`method` 参数用于指定在改变图像尺寸时所使用的算法，可以为以下几种：

'nearest': 这个参数也是默认的，即改变图像尺寸时采用最近邻插值算法；

'bilinear': 采用双线性插值算法；

'bicubic': 采用双三次插值算法；

(3) 处理结果：

原始图像：

原图像





最近邻插值法：

最近邻内插法所得图像



双线性插值法：

双线性内插法所得图像



双三次插值法：

双三次内插法所得图像



(4) 结果分析：最近邻插值、双线性插值与双三次差值这三种方法之间的区别主要在于点周围像素序列的取法不同。将上图中的四幅图进行对比可知：原图像与经过最近邻内插所得的图像的肩膀部分的曲线呈现锯齿状，采用双线性内插法所得的图像的肩膀部分的曲线则比较平滑，基本没有锯齿出现，而采用双三次内插法所得的图像的曲线均非常平滑。

原因分析：最近邻内插法把原图像中最近邻的灰度赋给了每个新位置，这种方法简单，但有产生不希望的人为缺陷的倾向。双线性内插法用 4 个最近邻去估计给定位置的灰度，通常所给出的内插效果会比最近邻法好，但计算量也相对有所增加。双三次内插法用 16 个最近邻点的灰度去估计给定位置的灰度，因而在保持细节方面比双线性内插法要好。

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

(1) 问题分析：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。对于水平偏移变换，运算后，坐标变为  $x=v$ ， $y=shv+w$ 。在本题中，参数 shv 根据题中要求设为 1.5。对于旋转变换，使用 imrotate 函数旋转图像。按照第四问的方法，采用近邻、线性和双三次插值法 zoom 到 2048\*2048。

(2) MATLAB 函数：图像二维仿射变换 MATLAB 使用 imtransform 函数完成图像空间变换。

调用格式：imtransform (A,T)

其中参数 A 是要变换的图像，T 是由 makeform 函数产生的变换结构。

在 maketform('P',.....)函数中，参数 P 可以以一下形式：

affine：仿射变换形式；

projective：投影变换形式；

**cusyom:** 自定义函数变换;  
**box:** 利用函数中的另外参数产生仿射变换结构;  
**composite:** 该参数是实现多次调用 tformfwd 功能;  
**maketform** 函数就是利用给定的参数建立变换结构, 然后把该变换结构赋予结构变量 T。

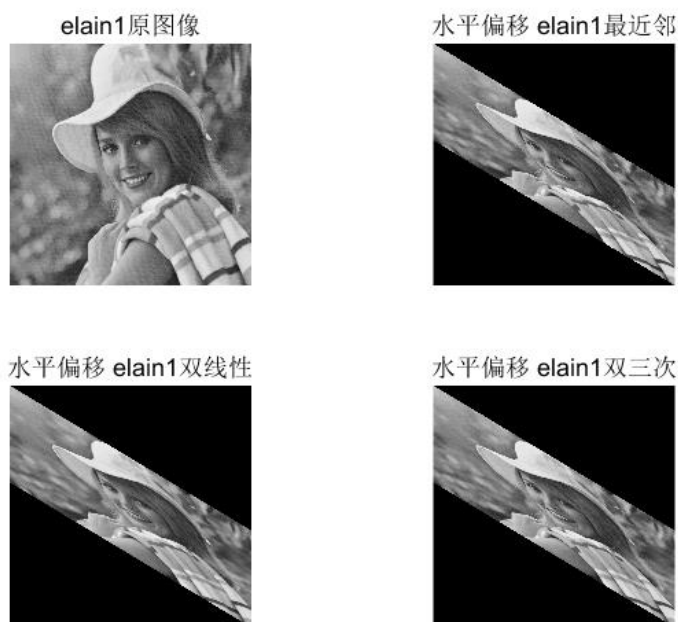
根据得到的结构体变量 T, 调用 imtransform (A,T) 函数进行变换。

(3) 处理结果:

Lena 水平偏移:



Elaine 水平偏移:



Lena 旋转 30 度:

lena原图像



旋转30度 lena最近邻



旋转30度 lena双线性



旋转30度 lena双三次



Elaine 旋转 30 度:

elain1原图像



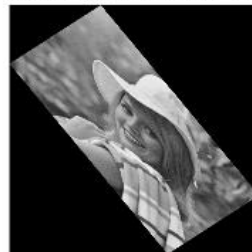
旋转30度 elain1最近邻



旋转30度 elain1双线性



旋转30度 elain1双三次



(4) 结果分析:

对水平偏移进行分析，通过对比上图三种处理方法得到的图像，可以看出，最近邻内插产生了最大的锯齿边缘，双线性内插得到了明显的改进结果，使用双三次内插产生了稍微清晰一些的结果。

对旋转 30 度进行分析，最近邻内插产生了最大的锯齿边缘，双线性内插得到了明显的改进结果，使用双三次内插产生了稍微清晰一些的结果。

附录:

**【参考文献】**

[1] 冈萨雷斯.数字图像处理（第三版）北京：电子工业出版社，2011

[2] 周品.MATLAB 数字图像处理 北京：清华大学出版社，2012

**【源代码】**

第二问

```
clear
```

```
Or=imread('lena.bmp');%读取图像“lena.bmp”
```

```
figure(1)
```

```
imshow(Or);
```

```
title('原图');%显示 k=8 的原始图像
```

```
[x,y]=size(Or);%读取图像“lena.bmp”的长和宽
```

```
pic7=zeros(x,y);%产生一个名为 pic7 的 512X512 的 0 矩阵
```

```
for i=1:x
```

```
    for j=1:y
```

```
        pic7(i,j)=floor(Or(i,j)/2);%将原图像的每个像素的灰度级除以 2，并用函数 floor 进行向下取整
```

```
    end
```

```
end
```

```
figure(2)
```

```
imshow((pic7),[0,127]);
```

```
title('7 灰度级');
```

```
pic6=zeros(x,y);%产生一个名为 pic6 的 512X512 的 0 矩阵
```

```
for i=1:x
```

```
    for j=1:y
```

```
        pic6(i,j)=floor(pic7(i,j)/2);%将原图像的每个像素的灰度级除以 2，并用函数 floor 进行向下取整
```

```
    end
```

```
end
```

```
figure(3)
```

```
imshow((pic6),[0,63]);
```

```
title('6 灰度级');
```

```
pic5=zeros(x,y);%产生一个名为 pic5 的 512X512 的 0 矩阵
```

```
for i=1:x
```

```
    for j=1:y
```

```
        pic5(i,j)=floor(pic6(i,j)/2);%将原图像的每个像素的灰度级除以 2，并用函数 floor 进行向下取整
```

```
    end
```

```
end
```

```
figure(4)
```

```
imshow((pic5),[0,31]);
```

```
title('5 灰度级');
```

```
pic4=zeros(x,y);%产生一个名为 pic4 的 512X512 的 0 矩阵
```

```
for i=1:x
```

```
    for j=1:y
```

```
        pic4(i,j)=floor(pic5(i,j)/2);%将原图像的每个像素的灰度级除以 2，并用  
        函数 floor 进行向下取整
```

```
    end
```

```
end
```

```
figure(5)
```

```
imshow((pic4),[0,15]);
```

```
title('4 灰度级');
```

```
pic3=zeros(x,y);%产生一个名为 pic3 的 512X512 的 0 矩阵
```

```
for i=1:x
```

```
    for j=1:y
```

```
        pic3(i,j)=floor(pic4(i,j)/2);%将原图像的每个像素的灰度级除以 2，并用  
        函数 floor 进行向下取整
```

```
    end
```

```
end
```

```
figure(6)
```

```
imshow((pic3),[0,7]);
```

```
title('3 灰度级');
```

```
pic2=zeros(x,y);%产生一个名为 pic2 的 512X512 的 0 矩阵
```

```
for i=1:x
```

```
    for j=1:y
```

```
        pic2(i,j)=floor(pic3(i,j)/2);%将原图像的每个像素的灰度级除以 2，并用  
        函数 floor 进行向下取整
```

```
    end
```

```
end
```

```
figure(7)
```

```
imshow((pic2),[0,3]);
```

```
title('2 灰度级');
```

```
pic1=zeros(x,y);%产生一个名为 pic1 的 512X512 的 0 矩阵
```

```
for i=1:x
```

```
    for j=1:y
```

```
        pic1(i,j)=floor(pic2(i,j)/2);%将原图像的每个像素的灰度级除以 2，并用  
        函数 floor 进行向下取整
```

```
    end
```

```
end
```

```
figure(8)
```

```
imshow((pic1),[0,1]);
```

```
title('1 灰度级')
```

### 第三问

```
clear
```

```
Or=imread('lena.bmp');% 读取图像 “lena.bmp”
```

```
average=mean2(Or)% 计算均值
```

```
fc=std2(Or)% 计算方差
```

### 第四问

```
clear
```

```
Or=imread('lena.bmp');% 读取图像 “lena.bmp”
```

```
figure(1);
```

```
imshow(Or);
```

```
title('原图像');
```

```
PN=imresize(Or,[2048,2048],'nearest');% 最近邻内插法
```

```
figure(2);
```

```
imshow(PN);
```

```
title('最近邻内插法所得图像');
```

```
PB=imresize(Or,[2048,2048],'bilinear');% 双线性内插法
```

```
figure(3);
```

```
imshow(PB);
```

```
title('双线性内插法所得图像');
```

```
PC=imresize(Or,[2048,2048],'bicubic');% 双三次内插法
```

```
figure(4);
```

```
imshow(PC);
```

```
title('双三次内插法所得图像');
```

### 第五问

水平偏移:

```
clear
```

```
P1=imread('lena.bmp');% 读取图像 “lena.bmp”
```

```
P2=imread('elain1.bmp');% 读取图像 “elain1.bmp”
```

```
T=[1 1.5 0;0 1 0;0 0 1];% 水平 shear 1.5
```

```
form=maketform('affine',T);
```

```
P11=imtransform(P1,form);
```

```
P1N=imresize(P11,[2048 2048],'nearest');% 最近邻
```

```
P1B=imresize(P11,[2048 2048],'bilinear');% 双线性
```

```
P1C=imresize(P11,[2048 2048],'bicubic');% 双三次
```

```
figure(1);
```

```
subplot(2,2,1);
```

```
imshow(P1);
```

```
title('lena 原图像');
```

```
subplot(2,2,2);
```

```
imshow(P1N);
```

```
title('水平偏移 lena 最近邻');
```

```
subplot(2,2,3);
```



```

imshow(P1B);
title('水平偏移 lena 双线性');
subplot(2,2,4);
imshow(P1C);
title('水平偏移 lena 双三次');

```

```

P22=imtransform(P2,form);
P2N=imresize(P22,[2048 2048],'nearest');% 最近邻
P2B=imresize(P22,[2048 2048],'bilinear');% 双线性
P2C=imresize(P22,[2048 2048],'bicubic');% 双三次
figure(2);
subplot(2,2,1);
imshow(P2);
title('elain1 原图像');
subplot(2,2,2);
imshow(P2N);
title('水平偏移 elain1 最近邻');
subplot(2,2,3);
imshow(P2B);
title('水平偏移 elain1 双线性');
subplot(2,2,4);
imshow(P2C);
title('水平偏移 elain1 双三次')

```

### 旋转 30 度:

```

clear
P1=imread('lena.bmp');% 读取图像 “lena.bmp”
P2=imread('elain1.bmp');% 读取图像 “elain1.bmp”
T=[cosd(30) sind(30) 0;-sind(30) cos(30) 0;0 0 1];% 旋转矩阵
form=maketform('affine',T);
P11=imtransform(P1,form);
P1N=imresize(P11,[2048 2048],'nearest');% 最近邻
P1B=imresize(P11,[2048 2048],'bilinear');% 双线性
P1C=imresize(P11,[2048 2048],'bicubic');% 双三次
figure(1);
subplot(2,2,1);
imshow(P1);
title('lena 原图像');
subplot(2,2,2);
imshow(P1N);
title('旋转 30 度 lena 最近邻');
subplot(2,2,3);
imshow(P1B);
title('旋转 30 度 lena 双线性');
subplot(2,2,4);

```

```
imshow(P1C);
title('旋转 30 度 lena 双三次');

P22=imtransform(P2,form);
P2N=imresize(P22,[2048 2048],'nearest');% 最近邻
P2B=imresize(P22,[2048 2048],'bilinear');% 双线性
P2C=imresize(P22,[2048 2048],'bicubic');% 双三次
figure(2);
subplot(2,2,1);
imshow(P2);
title('elain1 原图像');
subplot(2,2,2);
imshow(P2N);
title('旋转 30 度 elain1 最近邻');
subplot(2,2,3);
imshow(P2B);
title('旋转 30 度 elain1 双线性');
subplot(2,2,4);
imshow(P2C);
title('旋转 30 度 elain1 双三次')
```