

数字图像处理第一次作业

姓名：张璞

班级：自动化 64

学号：2160700034

提交日期：2019 年 3 月 4 日

摘要

本次报告首先简单阐述了 BMP 图像格式及其相关数据结构，随后主要完成了作业要求中关于图像处理与计算的各项任务。本次作业以 Matlab 为平台，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 2048×2048 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。以上任务完成后均得到了预期的结果。

关键字：BMP 图像 灰度级 灰度插值 仿射变换

1.Bmp 图像格式简介

1.1 BMP 格式概述

图像格式即图像文件存放在记忆卡上的格式，通常有 JPEG、TIFF、RAW 等。由于数码相机拍下的图像文件很大，储存容量却有限，因此图像通常都会经过压缩再储存。BMP（Bitmap-File）图像是 Windows 中采用的标准图像文件格式，与硬件设备无关，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。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。

1.2 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。

1.3 BMP 格式对应数据结构

BMP 文件由文件头、位图信息头、颜色信息和图形数据四部分组成。

1.3.1 BMP 文件头（14 字节）

BMP 文件头数据结构含有 BMP 文件的类型、文件大小和位图起始位置等信息。

其结构定义如下表所示，

表 1 BMP 文件头数据结构

偏移量	域的标识	大小	内容
0000h	文件的类型	2 字节	识别位图的类型
0002h	文件的大小	4 字节	用字节表示的整个文件的大小
0006h	文件保留字	4 字节	保留，必须设置为 0
000Ah	数据的起始位置	4 字节	从文件开始到位图数据开始之间的数据之间的偏移量

1.3.2 位图信息头（40 字节）

BMP 位图信息头数据用于说明位图的尺寸等信息。

其结构定义如下表所示，

表 2 BMP 位图信息头数据结构

偏移量	域的标识	大小	内容
000Eh	本结构所占用字节数	4 字节	位图信息头的长度，用来描述位图的颜色、压缩方法等

0012h	位图的宽度	4 字节	以像素为单位
0016h	位图的高度	4 字节	以像素为单位为 0
001Ah	目标设备的级别	2 字节	该值将总是 1
001Ch	每个像素所需的位数	2 字节	1-单色位图 4-16 色位图 8-256 色位图 16-16bit 高彩色位图 24-24bit 真彩色位图 32-32bit 增强型真彩色位图
001Eh	位图压缩类型	4 字节	0-不压缩 1-RLE 8-使用 8 位 RLE 压缩方式 2-RLE 4-使用 4 位 RLE 压缩方式 3-Bitfields-位域存放方式
0022h	位图的大小	4 字节	用字节数表示的位图数据的大小。该数必须是 4 的倍数
0026h	位图水平分辨率	4 字节	用像素/米表示的水平分辨率
002Ah	位图垂直分辨率	4 字节	用像素/米表示的垂直分辨率
002Eh	位图使用的颜色数	4 字节	如 8-比特/像素表示为 100h 或者 256
0032h	位图显示过程中重要的颜色数	4 字节	当该域的值等于颜色数时（或者等于 0 时），表示所有颜色都一样重要

1.3.3 颜色表

颜色表用于说明位图中的颜色，它有若干个表项，每一个表项是 RGBQUAD 类型的结构，定义一种颜色。对于颜色表中的每个表项，这 4 个字节用下述方法来描述 RGB 的值：

- 1) 1 字节用于蓝色分量；
- 2) 1 字节用于绿色分量；
- 3) 1 字节用于红色分量；
- 4) 1 字节用于填充符(设置为 0)。

颜色表中 RGBQUAD 结构数据的个数有 biBitCount 来确定，

- 1) 当 biBitCount=1,4,8 时，分别有 2,16,256 个表项；
- 2) 当 biBitCount=24 时，没有颜色表项。

1.3.4 位图数据

位图数据记录了位图的每一个像素值，记录顺序是在扫描行内是从左到右，扫描行之间是从下到上。位图的一个像素值所占的字节数：

当 biBitCount=1 时，8 个像素占 1 个字节；

当 biBitCount=4 时，2 个像素占 1 个字节；

当 biBitCount=8 时，1 个像素占 1 个字节；

当 biBitCount=24 时，1 个像素占 3 个字节,按顺序分别为 B,G,R；

Windows 规定一个扫描行所占的字节数必须是 4 的倍数（即以 long 为单位），不足的以 0 填充。

1.2 把 lena512*512 图像灰度级逐级递减 8-1 显示。

(1) 问题分析:

本题中对图像进行的操作是保持图像样本数恒定, 而将灰度级数以 2 的整数次幂从 256 减少到 2。

(2) 实验过程:

根据查阅 matlab 与图像处理相关的函数, 得知 `imread()` 函数可将图像的灰度矩阵读入 matlab 中; `imshow()` 函数可根据灰度矩阵对原图像进行显示。因此可将 `imread()` 函数读入的灰度矩阵 `I` 的每个元素值依次除以 2, 取整后得到灰度级为 7 的图像, 再利用 `imshow()` 函数对输出图像进行显示。对灰度级为 6-1 的图像可依次对灰度级为上一级的图像灰度矩阵除 2 得到。

(3) 函数说明:

`A=imread('filename')`

用 `imread` 函数读取图像文件, 文件格式可以是 TIFF、JPEG、GIF、BMP、PNG。读进来的是一个二维数组, 存放在矩阵 `A` 中。

`imshow(A)`

`imshow` 是 matlab 中显示图像的函数。

调用格式:

`imshow(I,n)`

`imshow(I,[low high])` 用指定的灰度范围 `[low high]` 显示灰度图像 `I`。显示结果, 图像中灰度值等于或低于 `low` 的都将用黑色显示, 而灰度值大于等于 `high` 的都显示为白色, 介于 `low` 和 `high` 之间的用其灰度级的默认值的中间色调显示。

(4) 处理结果:

图 1 (a) 是题目中所给的大小为 512*512 的灰度级为 8 的原图像; (b) 到 (h) 是图像比特数从 7 减小到 1 时得到的图像。





图 1 (a) 灰度级为 8 的原图像；(b) 到 (h) 灰度级依次为 7-1 的图像

(5) 结果分析：

从视觉效果上来看，256 级、128 级、64 级和 32 级灰度图像对于所有实用目的在视觉

效果上是相同的。然而对于灰度级为 16 的图像，在恒定或接近恒度灰度区域内有不易察觉的细小山脊状结构。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，通常称为伪轮廓，伪轮廓在灰度级数更低的图像中十分地明显。

1.3 计算 lena512 图像的均值方差

(1) 问题分析：

计算图像 lena512*512 的均值与方差。

(2) 实验过程：

根据查阅 MATLAB 相关函数，可得 mean2 () 函数和 std2 () 函数可用于求矩阵均值与方差。因此，利用 imread () 函数将图像读入 MATLAB 中后，用 I 表示其灰度矩阵，直接利用 mean2 () 函数和 std2 () 函数求得整幅图像的均值与方差。

(3) 函数说明：

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。

std2 ()

MATLAB 中提供标准差计算函数 std () 和 std2 ()。

(4) 处理结果：

均值 $u=95.6564$ 方差 $s^2=1.9273e+03$

(5) 结果分析：

经过 MATLAB 运算，得到像 lena512.bmp 的均值约为 95.6564，标准差约为 43.9012，方差是标准差的平方，约为 1927.3。方差越大，图像的对比度越大，可以显示的细节就越多。

1.4 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：

内插是在诸如放大、收缩、旋转和几何校正等任务中广泛应用的基本工具，内插是用已知数据来估计未知位置的数值的处理。最近邻内插法：在待插像素的周围四邻像素中，距离待求插像素最近的像素灰度赋给待插像素。该方法最简单，但校正后存在灰度不连续性，图像有明显锯齿状。双线性插值算法：用 4 个最近邻去估计给定位置的灰度。双三次插值算法：用 16 个最近邻去估计给定位置的灰度。

(2) 实验过程：

本题先通过 imread () 函数读入图像文件，然后利用 imresize () 函数将图像分别利用三种插值方法放大到 2048*2048。

(3) 函数说明：

imresize () 函数用于对图像做缩放处理。

调用格式：

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

其中，numrows 和 numcols 分别指定目标图像的高度和宽度。显而易见，由于这种格式允许图像缩放后长宽比例和源图像长宽比例不相同，因此所产生的图像有可能发生畸变。

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：

- a) '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；
- b) 'bilinear': 采用双线性插值算法；
- c) 'bicubic': 采用双三次插值算法；

(4) 处理结果：

图 2 (a) 是题目中所给的大小为 512*512 的灰度级为 8 的原图像；(b) 到 (d) 是图像分别采用最近邻插值、双线性插值和双三次内插得到的 2048*2048 图像。



图 2 (a) 大小为 512*512 的原图像；(b) 到 (d) 分别采用最近邻插值、双线性插值和双三次内插得到的 2048*2048 图像。

(5) 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。从时间花费来看，双线性插值法花费的时间比最近邻法的要长一些，而双三次法花费的时间比双线性法的又要长一些。但是明显可看出，经最近邻内插所得的图像肩膀以

及脸颊部分的曲线呈现锯齿状,而采用双线性内插与三线性内插所得图像的肩膀及脸颊曲线依次变得更加平滑。由此,可看出虽然三线性内插所花费时间较长,但其精确度较高。

1.5 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:

几何变换改进图像中像素间的空间关系,在图像处理中,几何变换由两个基本操作组成:

(a) 坐标的空间变换; (b) 灰度插值, 即对空间变换后的像素赋灰度值。最常用的空间变换之一是仿射变换, 它是一种二维坐标到二维坐标的线性变换, 用图像的矩阵去乘仿射变换的矩阵 T , 对图像中的坐标点做尺度、旋转、平移或偏移, 把图像上的像素重新定位到一个新位置, 并为这些新位置赋灰度值, 而这个任务可以用灰度内插的方法完成, 选择最近邻内插法、双线性插值法和双三次插值法三种方法。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, 运算后, 坐标变为 $x=v\cos\theta - w\sin\theta$, $y=v\sin\theta + w\cos\theta$ 。

(2) 实现过程:

使用 `imrotate` 函数旋转图像。该函数接受两个主要的变量, 即要旋转的图像和旋转的角度, 旋转角度的单位为度, 正值为逆时针旋转, 负值为顺时针旋转。使用 `imtransform` 函数完成一般的二维空间转换。创建一个 `TFORM` 结构可使用 `makeform` 函数。使用 `makeform` 函数, 可指定变换类型。在本次实验中, 指定的是仿射变换 “affine” 类型, 即进行平移, 旋转, 比例, 拉伸和错切等功能。仿射变换要求变换矩阵的最后一列除最后一个元素为 1 之外, 其它的均为 0。因此, 处理过程中先读入图像, 再根据需要输入矩阵 $T=[]$, 再创建仿射矩阵 `tform=makeform('affine', T)`, 其次进行空间变换 `imtransform(I,tform)`, 最后对进行内插并进行图像显示。

(3) 函数说明:

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)`

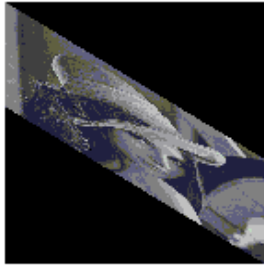
其中参数 A 是要变换的图像, T 是由 `makeform` 函数产生的变换结构。

`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 。

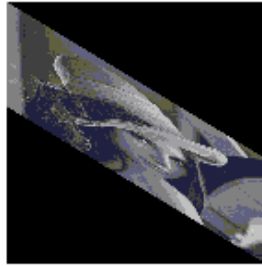
(4) 处理结果:

图 3 (a) 是对 lena 图像进行水平偏移变换并采用三种灰度内插方法得到的图像; (b) 是对 Elaine 图像进行水平偏移变换并采用三种灰度内插方法得到的图像; (c) 是对 lena 图像进行旋转 30 度变换并采用三种灰度内插方法得到的图像; (d) 是对 Elaine 图像进行旋转 30 度变换并采用三种灰度内插方法得到的图像。

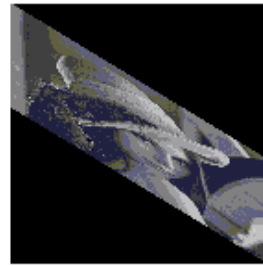
水平偏移 最近邻内插



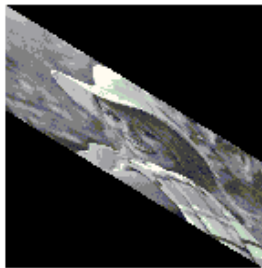
水平偏移 双线性内插



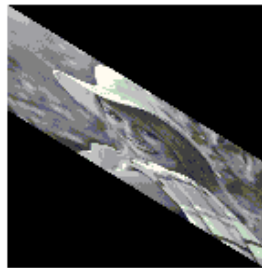
水平偏移 双三次内插



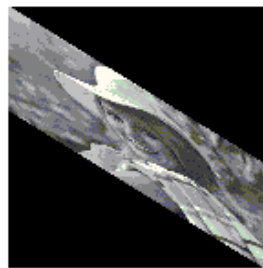
水平偏移 最近邻内插



水平偏移 双线性内插



水平偏移 双三次内插



旋转30度 最近邻内插



旋转30度 双线性内插



旋转30度 双三次内插



旋转30度 最近邻内插



旋转30度 双线性内插



旋转30度 双三次内插



图 3 (a) 对 lena 图像进行水平偏移变换并采用三种灰度内插方法得到；(b) 对 Elaine 图像进行水平偏移变换并采用三种灰度内插方法得到；(c) lena 图像进行旋转 30 度变换并采用三种灰度内插方法得到；(d) 对 Elaine 图像进行旋转 30 度变换并采用三种灰度内插方法得到

(5) 结果分析:

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。旋转是保持直线特性方面最苛求的几何变换之一, 与上一实验结果相同, 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
- [3] 杨杰.数字图像处理及 MATLAB 实现 北京: 电子工业出版社, 2010

【源代码】

lena.m

```
close all;clear all;clc;
I=imread('lena512.bmp');
figure(1)
imshow(I)
title('8 灰度级');
[x,y]=size(I);%读取 lena.bmp 的长宽
img7=zeros(x,y);
img6=zeros(x,y);
img5=zeros(x,y);
img4=zeros(x,y);
img3=zeros(x,y);
img2=zeros(x,y);
img1=zeros(x,y);
for i=1:x
    for j=1:y
```

```

        img7(i,j)=floor(I(i,j)/2); %floor 为向下取整
    end
end
figure(2)
imshow(uint8(img7),[0,127]) %uint8 为 8 位二进制数
title('7 灰度级');

for i=1:x
    for j=1:y
        img6(i,j)=floor(I(i,j)/4);
    end
end
figure(3)
imshow(uint8(img6),[0,63])
title('6 灰度级');

for i=1:x
    for j=1:y
        img5(i,j)=floor(I(i,j)/8);
    end
end
figure(4)
imshow(uint8(img5),[0,31])
title('5 灰度级');

for i=1:x
    for j=1:y
        img4(i,j)=floor(I(i,j)/16);
    end
end
figure(5)
imshow(uint8(img4),[0,15])
title('4 灰度级');

for i=1:x

```

```

        for j=1:y
            img3(i,j)=floor(I(i,j)/32);
        end
    end
figure(6)
imshow(uint8(img3),[0,7])
title('3 灰度级');

for i=1:x
    for j=1:y
        img2(i,j)=floor(I(i,j)/64);
    end
end
figure(7)
imshow(uint8(img2),[0,3])
title('2 灰度级');

for i=1:x
    for j=1:y
        img1(i,j)=floor(I(i,j)/128);
    end
end
figure(8)
imshow(uint8(img1),[0,1])
title('1 灰度级');

```

lena2.m

```

A=imread('lena512.bmp');
B=mean2(A);
C=std2(A);
D=C^2;

```

lena3.m

```

A=imread('lena512.bmp');

```

```

figure(1);
imshow(A);
title('原始图像');
B1=imresize(A,[2048 2048],'nearest');
figure(2);
imshow(B1);
title('最近邻插值法');
B2=imresize(A,[2048 2048],'bilinear');
figure(3);
imshow(B2);
title('双线性插值法');
B3=imresize(A,[2048 2048],'bicubic');
figure(4);
imshow(B3);
title('双三次插值法');

```

lena4.m

%Lena 图像:

```

transformtype='affine';
transformmatrix=[1 1.5 0;0 1 0;0 0 1];
T=maketform(transformtype,transformmatrix);
I=imread('lena512.bmp');
nI=imtransform(I,T);
figure(1);
imshow(I);
title('原始图像');
B1=imresize(nI,[2048 2048],'nearest');
subplot(1,3,1);
imshow(B1);
title('水平偏移 最近邻内插');
B2=imresize(nI,[2048 2048],'bilinear');
subplot(1,3,2);
imshow(B2);
title('水平偏移 双线性内插');

```

```

B3=imresize(nI,[2048 2048],'bicubic');
subplot(1,3,3);
imshow(B3);
title('水平偏移 双三次内插');
%Elain 图像:
transformtype='affine';
transformmatrix=[1 1.5 0;0 1 0;0 0 1];
T=maketform(transformtype,transformmatrix);
I=imread('elaine512.bmp');
nI=imtransform(I,T);
figure(5);
imshow(I);
title('原始图像');
B1=imresize(nI,[2048 2048],'nearest');
subplot(1,3,1);
imshow(B1);
title('水平偏移 最近邻内插');
B2=imresize(nI,[2048 2048],'bilinear');
subplot(1,3,2);
imshow(B2);
title('水平偏移 双线性内插');
B3=imresize(nI,[2048 2048],'bicubic');
subplot(1,3,3);
imshow(B3);
title('水平偏移 双三次内插');

```

lena5.m

```

%Lena 图像:
I=imread('lena512.bmp');
figure(1);
imshow(I);
title('原始图像');
I1=imrotate(I,30,'nearest');
B1=imresize(I1,[2048 2048],'nearest');

```

```

subplot(1,3,1);
imshow(B1);
title('旋转 30 度 最近邻内插');
I2=imrotate(I,30,'bilinear');
B2=imresize(I2,[2048 2048],'bilinear');
subplot(1,3,2);
imshow(B2);
title('旋转 30 度 双线性内插');
L3=imrotate(I,30,'bicubic');
B3=imresize(L3,[2048 2048],'bicubic');
subplot(1,3,3);
imshow(B3);
title('旋转 30 度 双三次内插');
%Elaine 图像:
L=imread('elaine512.bmp');
figure(5);
imshow(L);
title('原始图像');
L1=imrotate(L,30,'nearest');
C1=imresize(L1,[2048 2048],'nearest');
subplot(1,3,1);
imshow(C1);
title('旋转 30 度 最近邻内插');
L2=imrotate(L,30,'bilinear');
C2=imresize(L2,[2048 2048],'bilinear');
subplot(1,3,2);
imshow(C2);
title('旋转 30 度 双线性内插');
L3=imrotate(L,30,'bicubic');
C3=imresize(L3,[2048 2048],'bicubic');
subplot(1,3,3);
imshow(C3);
title('旋转 30 度 双三次内插');

```