

数字图像处理

第一次作业

班级：自动化 64

姓名：丁 鹏

学号：2160504091

提交日期：2019/3/5

摘要

本次报告先以 7.bmp 为例对 bmp 图像格式做了详细的概述，通过阐述了 7.bmp 图像的属性及像素值变化对这一图像格式有了更深的了解；接着根据题目要求，通过 MATLAB R2018b 平台对 lena.bmp 的编程处理，完成了这一图像文件的灰度级 8-1 递减显示和其像素矩阵均值和方差的求解；然后利用相关指令对 lena.bmp 做了近邻插值、双线性插值和双三次插值法，将其放大 4 倍到了 2048*2048 的尺寸，在此基础上做水平偏移变换和旋转变换的仿射变换；最后按照相关步骤完成了 H 矩阵的求解和图像配准的相关任务。

1、 Bmp 图像格式简介,以 7.bmp 为例说明;

BMP (全称 Bitmap)^[1]是 Windows 操作系统中的标准图像文件格式,可以分成两类:设备有向量相关位图 (DDB) 和设备无向量相关位图 (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。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。

1.1 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。

以 7.bmp 为例查看该图片的格式和信息

属性	值
图像	
尺寸	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 图像
文件夹路径	E:\数字图像处理\第二次作业
创建日期	2019/2/28 星期四 下午 16:35
修改日期	2011/6/3 星期五 上午 11:17
大小	1.10 KB

1.2 对应数据结构

(1) BMP 文件组成

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。

(2) 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。

(3) 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。

(4) 颜色表

颜色表用于说明位图中的颜色,它有若干个表项,每一个表项是一个 RGBQUAD 类型的结构,定义一种颜色。颜色表中 RGBQUAD 结构数据的个数有 biBitCount 来确定:

当 biBitCount=1,4,8 时,分别有 2,16,256 个表项;

当 biBitCount=24 时,没有颜色表项。

位图信息头和颜色表组成位图信息

(5) 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，

$biSizeImage = (((bi.biWidth * bi.biBitCount) + 31) \& \sim 31) / 8 * bi.biHeight$;

以 7.bmp 为例说明，其像素矩阵为

82	82	73	59	55	80	90
97	89	90	95	71	40	69
104	71	63	105	93	76	42
88	75	85	101	90	91	70
97	92	91	99	72	71	82
98	101	102	86	69	71	95
103	99	100	84	86	98	98

1.3 存储算法

BMP 文件通常是不压缩的，所以它们通常比同一幅图像的压缩图像文件格式要大很多。例如，一个 800×600 的 24 位几乎占据 1.4MB 空间。因此它们通常不适合在因特网或者其他低速或者有容量限制的媒介上进行传输。根据颜色深度的不同，图像上的一个像素可以用一个或者多个字节表示，它由 $n/8$ 所确定（ n 是位深度，1 字节包含 8 个数据位）。图片浏览器等基于字节的 ASCII 值计算像素的颜色，然后从调色板中读出相应的值。更为详细的信息请参阅下面关于位图文件的部分。 n 位 2^n 种颜色的位图近似字节数可以用下面的公式计算：BMP 文件大小约等于 $54 + 4 \times 2^n$ 的 n 次方 + $(w \times h \times n)/8$ ，其中高度和宽度都是像素数。需要注意的是上面公式中的 54 是位图文件的文件头，是彩色调色板的大小。另外需要注意的是这是一个近似值，对于 n 位的位图图像来说，尽管可能有最多 2^n 中颜色，一个特定的图像可能并不会使用这些所有的颜色。由于彩色调色板仅仅定义了图像所用的颜色，所以实际的彩色调色板将小于。如果想知道这些值是如何得到的，请参考下面文件格式的部分。由于存储算法本身决定的因素，根据几个图像参数的不同计算出的大小与实际的文件大小将会有一些细小的差别。

2、 把 lena 512*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直接灰度变换。首先利用 imread 函数读入 lena512.bmp，得到图像的灰度矩阵 I，之后对灰度矩阵中的每个元素进行矩阵操作。将灰度矩阵的每个元素值除以二，取整后将结果转换为无符号 8 位数，这就得到了灰度级为 7 的图像，最后利用 imshow 函数输出显示图像。同理可得到灰度级 6-1 递减的一系列图像。对应源程序 lena1.m

(2) matlab 指令

imread - 从图形文件读取图像

此 MATLAB 函数从 filename 指定的文件读取图像，并从文件内容推断出其格式。如果 filename 为多图像文件，则 imread 读取该文件中的第一个图像。

```
A = imread(filename)
A = imread(filename,fmt)
A = imread(___,idx)
A = imread(___,Name,Value)
[A,map] = imread(___)
[A,map,transparency] = imread(___)

```

figure - 创建图窗窗口

此 MATLAB 函数使用默认属性值创建一个新的图窗窗口。生成的图窗为当前图窗。

```
figure
figure(Name,Value)
f = figure(___)
figure(f)
figure(n)

```

imshow - 显示图像

此 MATLAB 函数在图窗中显示灰度图像 I。imshow 优化图窗、坐标区和图像对象属性以便显示图像。

```
imshow(I)
imshow(I,[low high])
imshow(I,[])
imshow(RGB)
imshow(BW)
imshow(X,map)
imshow(filename)
imshow(___,Name,Value)

```

floor - 朝负无穷大四舍五入

此 MATLAB 函数将 X 的每个元素四舍五入到小于或等于该元素的最接近整数。

(3) 处理结果

灰度级8



灰度级7



灰度级6



灰度级5



灰度级4



灰度级3





灰度级1



（4）结果分析

通过对一幅 512*512,256 个灰度级的具有较多细节的图像，保持空间分辨率不变，仅将灰度级数依次递减为 128、64、32、16、8、4、2，比较得到的结果就可以发现灰度级数对图像有一定的影响。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，则会出现较大差别。

3、 计算 lena 图像的均值方差；

（1）问题分析：首先通过 `imread（）` 函数读入图像文件到灰度矩阵 I 中，然后通过编程处理计算得到利用灰度矩阵（即图像）的均值和方差。

对应源程序：lena2.m

(2) matlab 指令

mean2 - Average or mean of matrix elements

This MATLAB function computes the mean of all values in array A.

B = mean2(A)

std2 - Standard deviation of matrix elements

This MATLAB function computes the standard deviation of all values in array A.

B = std2(A)

(3) 得到结果

mu =

99.0512

sigma2 =

2.7960e+03

(4) 结果分析

经过 MATLAB 运算，得到图像 lena512.bmp 的均值约为 99.0512，标准差约为 2796。

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：插值算法是估计在图像像素间的某个位置的像素的取值方法。

最近邻点插值法：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：双线性插值算法输出的图像的每个像素都是原图中四个像素（ 2×2 ）运算的结果，这种算法极大地消除了锯齿现象。

双三次插值算法：双三次插值算法是上一种算法的改进算法，它输出图像的每个像素都是原图 16 个像素（ 16×16 ）运算的结果。

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。对应源程序：lena3.m

(2) matlab 指令

imresize - 调整图像大小

此 MATLAB 函数 返回图像 B，它是将 A 的长宽大小缩放 scale 倍之后的图像。输入图像 A 可以是灰度、RGB 或二值图像。如果 A 有两个以上维度，则 imresize 只调整前两个维度的大小。如果 scale 在 [0,1] 范围内，则 B 比 A 小。如果 scale 大于 1，则 B 比 A 大。默认情况下，imresize 使用双三次插值。

B = imresize(A,scale)

B = imresize(A,[numrows numcols])

[Y,newmap] = imresize(X,map,___)

___ = imresize(___,method)

___ = imresize(___,Name,Value)

原始图像



近邻插值



双线性插值



双三次插值



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。

对于最近邻插值，输出像素的值指定为点所属像素的值，不考虑其他像素；

对于双线性插值，输出图像的值是最近的 2×2 邻域内像素值得加权平均值；

对于双三次差值，输出图像的值是最近的 4×4 邻域内像素值得加权平均值。

所以，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。

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；

（1）问题分析：采用仿射变换来进行水平 shear。仿射变换是空间直角坐标变换的一种，它是一种二维坐标到二维坐标的线性变换，即用图像的矩阵去乘仿射变换的矩阵 T ，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 $x=v$ ， $y=s_h v+w$ 。在本题中，参数 s_h 根据题中要求设为 1.5。

对于旋转变换，运算后，坐标变为 $x=v\cos\theta - w\sin\theta$ ， $y=v\sin\theta + w\cos\theta$ 。使用 `imtransform` 函数完成一般的二维空间转换。对应源程序：lena4.m 和 elain1.m

(2) matlab 指令

maketform - Create spatial transformation structure (TFORM)

This MATLAB function creates a multidimensional spatial transformation structure **T** for an N-dimensional affine transformation.

T = maketform('affine',A)

T = maketform('affine',U,X)

T = maketform('projective',A)

T = maketform('projective',U,X)

T = maketform('custom',NDIMS_IN,NDIMS_OUT,FORWARD_FCN,INVERSE_FCN,TDATA)

T = maketform('box',tsize,LOW,HIGH)

T = maketform('box',INBOUNDS, OUTBOUNDS)

T = maketform('composite',T1,T2,...,TL)

T = maketform('composite', [T1 T2 ... TL])

imtransform - Apply 2-D spatial transformation to image

This MATLAB function transforms the image **A** according to the 2-D spatial transformation defined by **tform**.

B = imtransform(A,tform)

B = imtransform(A,tform,interp)

[**B**,xdata,ydata] = imtransform(...)

[**B**,xdata,ydata] = imtransform(...,Name,Value)

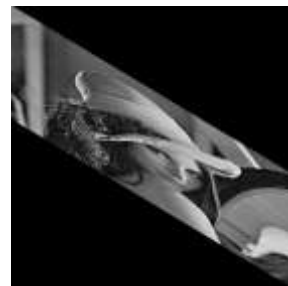
(3) 处理结果

a.水平 shear

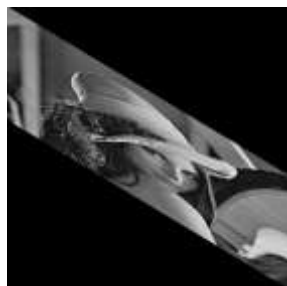
原始图像



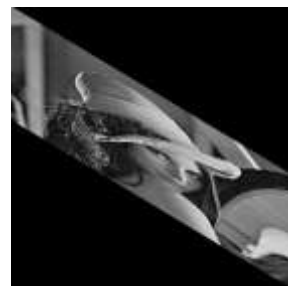
近邻插值



双线性插值



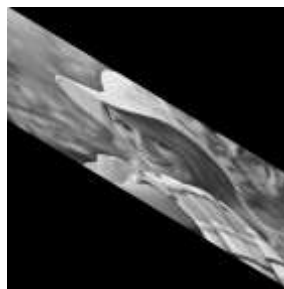
双三次插值



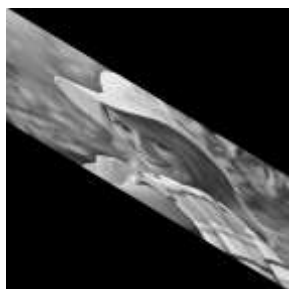
原始图像



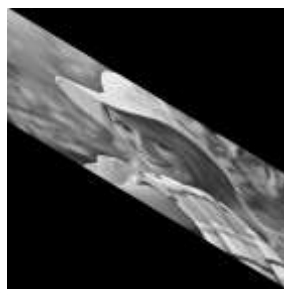
近邻插值



双线性插值



双三次插值



(b) 旋转 30 度

原始图像



近邻插值



双线性插值



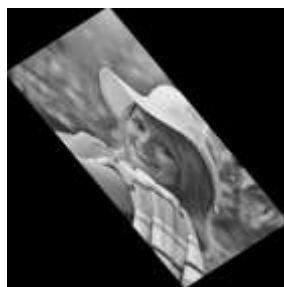
双三次插值



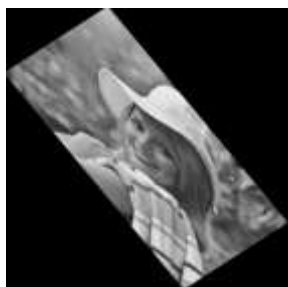
原始图像



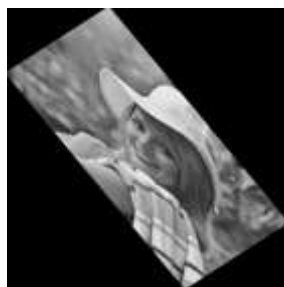
近邻插值



双线性插值



双三次插值



4) 结果分析:

在数字图像处理中几何变换由两个基本操作组成:(1)坐标的空间变化;(2)灰度内插,即对空间变换后的像素赋灰度值。最常用的空间坐标变换之一是仿射变换。旋转是保持直线特性方面最苛求的几何变换之一,与上一实验结果类似。

附录

```
lena1:
close all;clear all;clc;
I=imread('lena.bmp');
figure(1)
imshow(I)
title('灰度级 8');
[x,y]=size(I);

img7=zeros(x,y);
for i=1:x
    for j=1:y
        img7(i,j)=floor(I(i,j)/2);
    end
```

```

end
figure(2)
imshow(img7,[0,127])
title('灰度级 7');

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

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

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

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

```

```

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

```

```

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

```

```

lena2:
close all;clear all;clc;
I=imread('lena.bmp');
[x,y]=size(I);
img=zeros(x,y);
z=0;D=0;
for i=1:x
    for j=1:y
        img(i,j)=I(i,j);
        z=z+img(i,j);
    end
end
mu=1/x/y*z
for i=1:x
    for j=1:y
        D=D+(img(i,j)-mu)^2;
    end
end
sigma2=1/x/y*D

```

```

lena3:
close all;clear all;clc;
I=imread('lena.bmp');

```



```
subplot(2,2,1)
imshow(I)
title('原始图像');
```

```
img1=imresize(I,[2048,2048],'nearest');
subplot(2,2,2)
imshow(img1)
title('近邻插值')
```

```
img2=imresize(I,[2048,2048],'bilinear');
subplot(2,2,3)
imshow(img2)
title('双线性插值')
```

```
img3=imresize(I,[2048,2048],'bicubic');
subplot(2,2,4)
imshow(img3)
title('双三次插值')
```

```
lena4:
close all;clear all;clc;
I=imread('lena.bmp');
T=[1 1.5 0;0 1 0;0 0 1];
tform=maketform('affine',T);
img1=imtransform(I,tform);
subplot(2,2,1)
imshow(I)
title('原始图像');
```

```
img2=imresize(img1,[2048,2048],'nearest');
subplot(2,2,2)
imshow(img2)
title('近邻插值')
```

```
img3=imresize(img1,[2048,2048],'bilinear');
subplot(2,2,3)
imshow(img3)
title('双线性插值')
```

```
img4=imresize(img1,[2048,2048],'bicubic');
subplot(2,2,4)
imshow(img4)
title('双三次插值')
```

```

elain1:
close all;clear all;clc;
%I=imread('lena.bmp');
I=imread('elain1.bmp');
T=[1 1.5 0;0 1 0;0 0 1];
tform=maketform('affine',T);
img1=imtransform(I,tform);
subplot(2,2,1)
imshow(I)
title('原始图像');

img2=imresize(img1,[2048,2048],'nearest');
subplot(2,2,2)
imshow(img2)
title('近邻插值')

img3=imresize(img1,[2048,2048],'bilinear');
subplot(2,2,3)
imshow(img3)
title('双线性插值')

img4=imresize(img1,[2048,2048],'bicubic');
subplot(2,2,4)
imshow(img4)
title('双三次插值')

lena5:
close all;clear all;clc;
%I=imread('lena.bmp');
I=imread('elain1.bmp');
T=[cosd(30) sind(30) 0;-sind(30) cos(30) 0;0 0 1];
tform=maketform('affine',T);
img1=imtransform(I,tform);
subplot(2,2,1)
imshow(I)
title('原始图像');

img2=imresize(img1,[2048,2048],'nearest');
subplot(2,2,2)
imshow(img2)
title('近邻插值')

img3=imresize(img1,[2048,2048],'bilinear');
subplot(2,2,3)

```

```

imshow(img3)
title('双线性插值')

img4=imresize(img1,[2048,2048],'bicubic');
subplot(2,2,4)
imshow(img4)
title('双三次插值')

elain2:
close all;clear all;clc;
I=imread('lena.bmp');
%I=imread('elain1.bmp');
T=[cosd(30) sind(30) 0;-sind(30) cos(30) 0;0 0 1];
tform=maketform('affine',T);
img1=imtransform(I,tform);
subplot(2,2,1)
imshow(I)
title('原始图像');

img2=imresize(img1,[2048,2048],'nearest');
subplot(2,2,2)
imshow(img2)
title('近邻插值')

img3=imresize(img1,[2048,2048],'bilinear');
subplot(2,2,3)
imshow(img3)
title('双线性插值')

img4=imresize(img1,[2048,2048],'bicubic');
subplot(2,2,4)
imshow(img4)
title('双三次插值')

```

参考文献

- [1] BMP 格式 .百度百科
- [2] 冈萨雷斯.数字图像处理（第三版）北京：电子工业出版社，2011