



西安交通大学
XI'AN JIAOTONG UNIVERSITY

数字图像与视频处理

第一次作业

学生姓名：孙若云

专业班级：自动化 65

学号：2160504112

提交日期：2019.03.05

摘要：在本次报告中，首先对 bmp 图像格式进行了简单说明，然后基于 Matlab2018b 的实验环境实现对 lena 图像的灰度级变换，计算均值和方差，分别使用最近邻插值法、双线性插值法、双三次插值法将其扩大到 2048*2048，以及对 lena 和 elain 进行水平偏移和旋转处理，用三种插值方法将图像放大。以上内容均实现了本次作业所要求的效果。

1. Bmp 图像格式简介

BMP（全称 Bitmap）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。

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。

Structure name	Optional	Size	Purpose	Comments
Bitmap file header	No	14 bytes	To store general information about the bitmap image file	Not needed after the file is loaded in memory
DIB header	No	Fixed-size (7 different versions exist)	To store detailed information about the bitmap image and define the pixel format	Immediately follows the Bitmap file header
Extra bit masks	Yes	3 or 4 DWORDs ^[5] (12 or 16 bytes)	To define the pixel format	Present only in case the DIB header is the BITMAPINFOHEADER and the Compression Method member is set to either BI_BITFIELDS or BI_ALPHABITFIELDS
Color table	Semi-optional	Variable-size	To define colors used by the bitmap image data (Pixel array)	Mandatory for color depths ≤ 8 bits
Gap1	Yes	Variable-size	Structure alignment	An artifact of the File offset to Pixel array in the Bitmap file header
Pixel array	No	Variable-size	To define the actual values of the pixels	The pixel format is defined by the DIB header or Extra bit masks. Each row in the Pixel array is padded to a multiple of 4 bytes in size
Gap2	Yes	Variable-size	Structure alignment	An artifact of the ICC profile data offset field in the DIB header
ICC color profile	Yes	Variable-size	To define the color profile for color management	Can also contain a path to an external file containing the color profile. When loaded in memory as "non-packed DIB", it is located between the color table and Gap1. [6]

1.1 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。

字节顺序	数据结构	描述
1,2	word	高8位为字母'B'，低8位为字母'M'
3,4,5,6	uint	文件尺寸
7,8	word	保留字1
9,10	word	保留字2
11,12,13,14	uint	位图数据部分相对于文件首的起始偏移量

1.2 BMP 图像参数 用于说明位图的尺寸等信息,它包含有 BMP 图像的宽、高、压缩方法，以及定义颜色等。

字节顺序	数据结构	描述
15,16,17,18	uint	当前结构体的大小，通常是40或56
19,20,21,22	int	图像宽度（像素）
23,24,25,26	int	图像高度（像素）
27,28	word	这个字的值永远是1
29,30	word	每像素占用的位数，即bpp

31,32,33,34	uint	压缩方式
35,36,37,38	uint	图像的尺寸（字节数）
39,40,41,42	int	水平分辨率，pixels-per-meter
43,44,45,46	int	垂直分辨率，pixels-per-meter
47,48,49,50	uint	引用色彩数
51,52,53,54	uint	关键色彩数

第31-34字节存储着一个uint型参数,它代表图像数据的压缩方式。

该参数的取值范围是 0、1、2 或 3 等等。这些取值的含义分别是：

0 —— RGB 方式

1 —— 8bpp 的 run-length-encoding 方式

2 —— 4bpp 的 run-length-encoding 方式

3 —— bit-fields 方式

只有压缩方式选项被置为 bit-fields 时，当前结构体的大小为 56 字节，否则为 40 字节。

1.3 调色板，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。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

1.4 位图数据，记录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，

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

1.5 在 matlab 中读取 7.bmp 的内容，我们可以看到 7.bmp 被读取为一个 7*7 的矩阵，矩阵类型是 uint8，矩阵值表示灰度的浓淡，范围是[0-255]。

```
>> imread('7.bmp')  
  
ans =  
  
7×7 uint8 矩阵  
  
    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
```

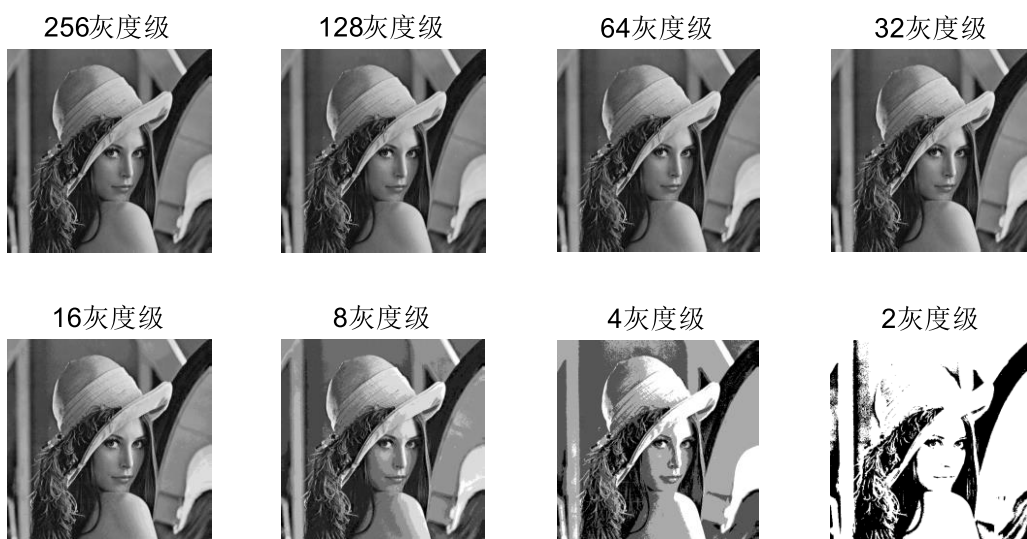
2. 把 lena 512*512 图像灰度级逐级递减 8-1 显示

问题分析：图像的灰度处理变换是一种基础的图像处理方法，其本质是按一定的规律修改每个像素的灰度，从而改变图像的显示状况。本题目中只需要简单改变所有像素的灰度级，原图的灰度级为 $Value(x,y)$ ，以 2^k 为基准，改变后的灰度级为 $Value_k(x,y)$

$$Value_k(x,y) = Floor\left(\frac{Value(x,y)}{2^{8-k}}\right), \text{ 其中 floor 函数是向下取整}$$

实验方法：在 matlab 中读取 lena 图像，将其存入矩阵中，然后定义一些空矩阵，用于存放改变灰度后的图像，可以用 for 语句实现灰度值的计算，最后用 imshow 展示出改变灰度后的图像

实验结果：如下图所示



我们可以看到灰度级在 2^5 以上时肉眼观察不到明显的变化

3. 计算 lena 图像的均值方差

问题分析：lena 是空间分辨率为 512*512 的图像，对 $M*N$ 的图像，计算其均值和方差的公式如下

$$m = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N f(i, j) \quad \sigma^2 = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f(i, j) - m)^2$$

在 matlab 中可使用 mean2() 和 std2() 函数求矩阵所有元素的均值和方差

实验方法: 在 matlab 中用 imread 读入 lena 图像, 使用 mean2() 和 std2() 函数计算其均值和方差

实验结果:

```
ave =          var =
    99.0512    52.8775
```

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

问题分析: 放大图像几乎都是采用内插值方法, 即在原有图像基础上在像素点之间采用合适的插值算法插入新的元素

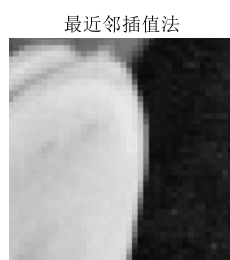
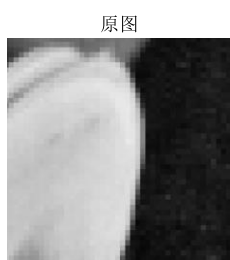
- (1) 最近邻插值法: 邻域插值是放大图像最简单的方法, 它能在放大图像的同时保留所有的原图像信息, 缺点是会产生锯齿现象
- (2) 双线性插值法: 效果相较于最近邻插值法平滑很多, 但会使图像变得模糊, 而且仍有锯齿现象
- (3) 双三次插值法: 能创造出比双线性插值更平滑的图像边缘, 对一个图形文件进行处理后, 文件长度增加 (数据量增大), 但不会有原先那幅图锐利, 可能会在图形质量上打折扣。

实验方法: 在 matlab 中可应用 imresize 函数的 “nearest”、“bilinear”、“bicubic” 对图像进行三种插值

实验结果: 如下图



放大 lena 的帽子部分我们可以看到最近邻插值法和双线性插值法都存在锯齿现象，双三次插值法所得图像更平滑一些。



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;

问题分析: 图像的仿射变换可以用一个 3*3 的仿射矩阵 T 表示, 设原坐标为

(a,b) 新坐标为 (x,y), 则 $\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} a & b & 1 \end{bmatrix} * T^{-1}$

对于 shear 变换

$$T = \begin{bmatrix} 1 & s_k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

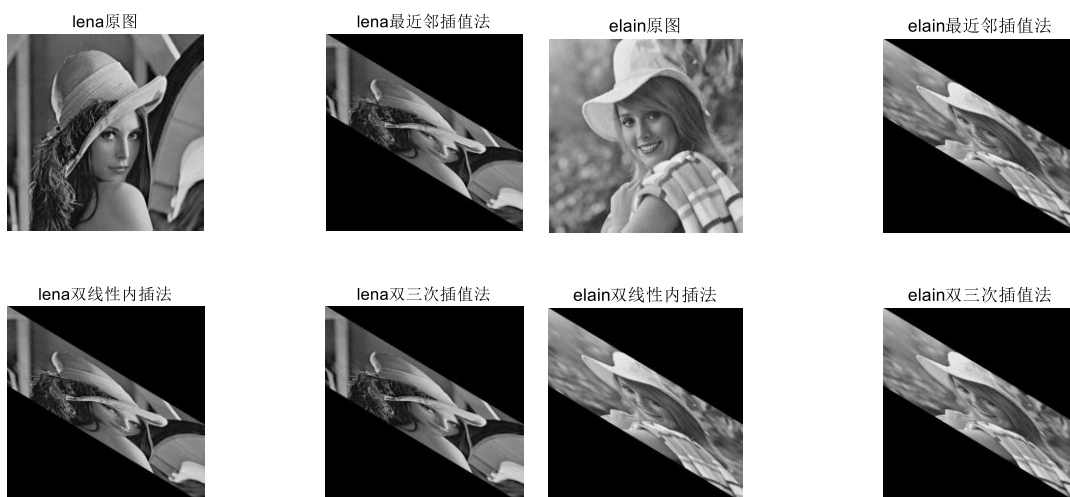
对于旋转变换

$$T = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

实验方法：在 matlab 中可用 maketform 函数完成图像关于 T 矩阵的相应计算，再用三种插值方法将图像放大并输出

实验结果：

Shear:



旋转



参考文献：

- [1] （美）冈萨雷斯，（美）伍兹.数字图像处理（第三版），电子工业出版社
2011
- [2] （美）冈萨雷斯，（美）伍兹，（美）艾丁斯. 数字图像处理（MATLAB
版）（第二版），电子工业出版社，2014
- [3] 程远航. 数字图像处理基础及应用. 清华大学出版社，2018
- [4] <http://www.cnblogs.com/wainiwann/p/7086844.html>
- [5] <https://baike.baidu.com/item/BMP/35116?fr=aladdin>
- [6] https://en.wikipedia.org/wiki/BMP_file_format

源代码:

```
1.
Imread(7.bmp)
2.
A=imread('lena.bmp');
subplot(2,4,1);
imshow(A)

title('256灰度级');

[m,n]=size(A);
img128=zeros(m,n);
img64=zeros(m,n);
img32=zeros(m,n);
img16=zeros(m,n);
img8=zeros(m,n);
img4=zeros(m,n);
img2=zeros(m,n);
for i=1:m
    for j=1:n
        img128(i,j)=floor(A(i,j)/2);
    end
end
subplot(2,4,2);
imshow(img128,[0,127])

title('128灰度级');

for i=1:m
    for j=1:n
        img64(i,j)=floor(A(i,j)/4);
    end
end
subplot(2,4,3);
imshow((img64),[0,63])

title('64灰度级');

for i=1:m
    for j=1:n
        img32(i,j)=floor(A(i,j)/8);
    end
end
subplot(2,4,4);
imshow((img32),[0,31])

title('32灰度级');
```

```

for i=1:m
    for j=1:n
        img16(i,j)=floor(A(i,j)/16);
    end
end
subplot(2,4,5);
imshow((img16),[0,15])

title('16灰度级');

for i=1:m
    for j=1:n
        img8(i,j)=floor(A(i,j)/32);
    end
end
subplot(2,4,6);
imshow((img8),[0,7])

title('8灰度级');

for i=1:m
    for j=1:n
        img4(i,j)=floor(A(i,j)/64);
    end
end
subplot(2,4,7);
imshow((img4),[0,3])

title('4灰度级');

for i=1:m
    for j=1:n
        img2(i,j)=floor(A(i,j)/128);
    end
end
subplot(2,4,8);
imshow((img2),[0,1])

title('2灰度级');

3
A=imread('lena.bmp');
ave=mean2(A)
var=std2(A)
4
A=imread('lena.bmp');
A1=imresize(A,[2048,2048],'nearest');
A2=imresize(A,[2048,2048],'bilinear');

```

```

A3=imresize(A,[2048,2048],'bicubic');
subplot(2,2,1);
imshow(A);

title('原图');

subplot(2,2,2);
imshow(A1);

title('最近邻插值法');

subplot(2,2,3);
imshow(A2);

title('双线性内插法');

subplot(2,2,4);
imshow(A3);

title('双三次插值法');
5

```

水平偏移

```

A=imread('lena.bmp');
B=imread('elain1.bmp');
T=[1 1.5 0;0 1 0;0 0 1];
tform=maketform('affine',T);
A1=imtransform(A,tform);
A2=imresize(A1,[2048,2048],'nearest');
A3=imresize(A1,[2048,2048],'bilinear');
A4=imresize(A1,[2048,2048],'bicubic');
B1=imtransform(B,tform);
B2=imresize(B1,[2048,2048],'nearest');
B3=imresize(B1,[2048,2048],'bilinear');
B4=imresize(B1,[2048,2048],'bicubic');
figure(1);
subplot(2,2,1);
imshow(A);

title('lena原图');

subplot(2,2,2);
imshow(A2);

title('lena最近邻插值法');

subplot(2,2,3);
imshow(A3);

title('lena双线性内插法');

```

```

subplot(2,2,4);
imshow(A4);

title('lena双三次插值法');

figure(2);
subplot(2,2,1);
imshow(B);

title('elain原图');

subplot(2,2,2);
imshow(B2);

title(' elain最近邻插值法');

subplot(2,2,3);
imshow(B3);

title('elain双线性内插法');

subplot(2,2,4);
imshow(B4);

title('elain双三次插值法');

```

旋转

```

A=imread('lena.bmp');
B=imread('elain1.bmp');
T=[cosd(30) sind(30) 0;-sind(30) cosd(30) 0;0 0 1];
tform=maketform('affine',T);
A1=imtransform(A,tform);
A2=imresize(A1,[2048,2048],'nearest');
A3=imresize(A1,[2048,2048],'bilinear');
A4=imresize(A1,[2048,2048],'bicubic');
B1=imtransform(B,tform);
B2=imresize(B1,[2048,2048],'nearest');
B3=imresize(B1,[2048,2048],'bilinear');
B4=imresize(B1,[2048,2048],'bicubic');
figure(1);
subplot(2,2,1);
imshow(A);

title('lena原图');

subplot(2,2,2);
imshow(A2);

title('lena最近邻插值法');

```

```
subplot(2,2,3);  
imshow(A3);  
  
title('lena双线性内插法');  
  
subplot(2,2,4);  
imshow(A4);  
  
title('lena双三次插值法');  
  
figure(2);  
subplot(2,2,1);  
imshow(B);  
  
title('elain原图');  
  
subplot(2,2,2);  
imshow(B2);  
  
title(' elain最近邻插值法');  
  
subplot(2,2,3);  
imshow(B3);  
  
title('elain双线性内插法');  
  
subplot(2,2,4);  
imshow(B4);  
  
title('elain双三次插值法');
```