

数字图像处理第一次作业

班级：自动化 63

姓名：朱可

学号：2160504082

目录

摘要	1
一、 图像处理的任务及其实现方法	1
1.1 图像格式简介	1
1. bmp 图像介绍	1
2. bmp 格式组成	1
3. bmp 格式类型	2
4. 以 7.bmp 为示例说明	2
1.2 灰度级别逐级递减表示 (8-1)	3
1. 题目分析	3
2. 代码运行	3
3. 输出结果	3
4. 结果分析	4
1.3 图像均值误差的计算	5
1. 题目分析与 MATLAB 运算	5
2. 输出结果	5
3. 结果分析	5
1.4 运用三种方法将图像 ZOOM 到 2048*2048	5
1. 题目分析	5
2. 代码运行	6
3. 输出结果	6
4. 结果分析	7
1.5 将 lena 和 elain 图像分别进行水平 shear 并进行插值操作	7
1. 题目分析	7
2. 代码运行	8
3. 输出结果	8
4. 结果分析	9
二、 参考文献	10
三、 附录	10

摘要

本文以 MATLAB 为基本工具完成了图像处理的各个步骤。首先本文对 bmp 图像进行了介绍，然后举例说明了图像格式的基本特征；继而在 MATLAB 中把图像按照灰度级逐级递减的方式进行了显示（即灰度级别从 8 降为 1），并计算出了该图像的均值方差。然后本文相继运用了近邻、双线性插值和双三次插值法，将该图像 zoom 到了 2048*2048 的形式。最后本文把水平剪切时的参数设置成了 1.5，然后将 1 两幅图像分别进行水平 shear 和旋转 30 度，并且采用了近邻、双线性插值和双三次插值法再次将其 zoom 到了 2048*2048 的形式。最后总结三种插值方法得知，最邻近元法计算量较小，但可能会造成插值生成的图像灰度上的不连续，在灰度变化的地方可能出现明显的锯齿状。双线性内插法的计算比最邻近点法复杂，计算量较大，但没有灰度不连续的缺点，结果基本令人满意。三次曲线插值方法计算量较大，但插值后的图像效果最好。

一、图像处理的任务及其实现方法

1.1 图像格式简介

1. bmp 图像介绍

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

2. bmp 格式组成

其格式由 4 个部分组成。

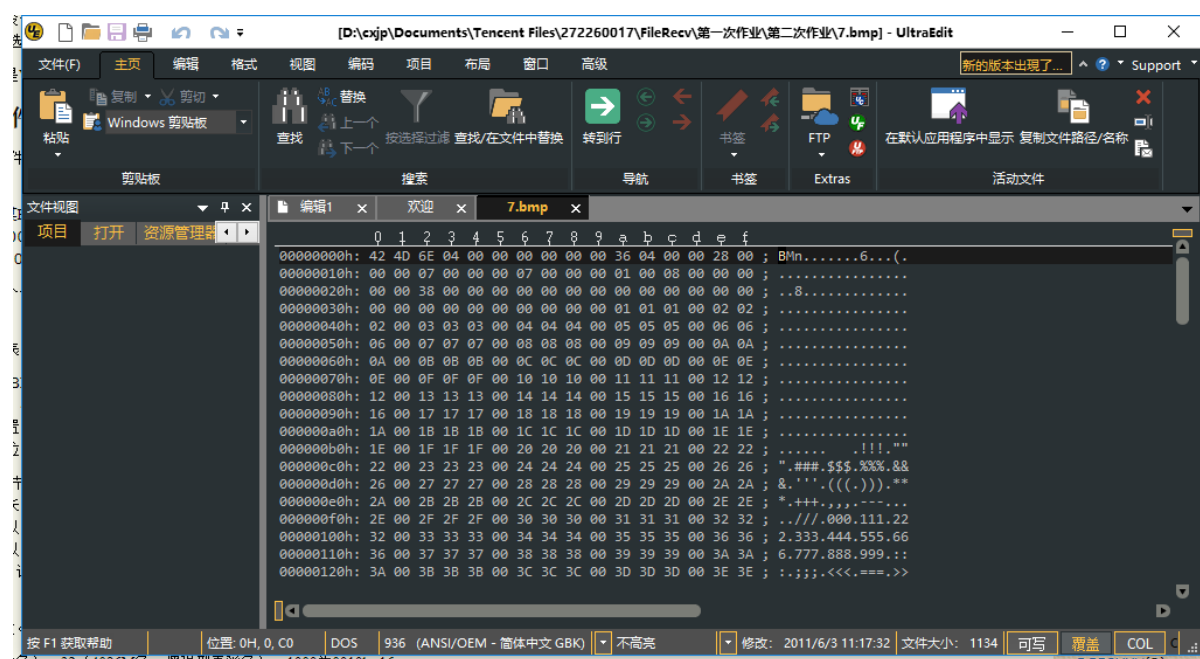
- 1) 位图头文件数据结构，它包含 BMP 图像文件的类型、显示内容等信息；
- 2) 位图信息数据结构，它包含有 BMP 图像的宽、高、压缩方法，以及定义颜色等信息；
- 3) 调色板，这个部分是可选的，有些位图需要调色板，有些位图，比如真彩色图（24 位的 BMP）就不需要调色板；
- 4) 位图数据，这部分的内容根据 BMP 位图使用的位数不同而不同，在 24 位图中直接使用 RGB，而其他的小于 24 位的使用调色板中颜色索引值。

3. bmp 格式类型

位图一共有两种类型，即：设备相关位图（DDB）和设备无关位图（DIB）。DDB 位图在早期的 Windows 系统中是很普遍的然而，随着显示器制造技术的进步，以及显示设备的多样化，DDB 位图的一些固有的问题开始浮现出来了，继而微软创建了 DIB 位图格式。

4.以 7.bmp 为示例说明

用 ultraedit 打开 7.bmp 得到其数据如下图所示：



其对应情况如表一所示：

表一 bmp 图像说明图

开始	结束	值	含义
0000H	0001H	424DH	Windows支持的BMP格式
0002H	0005H	6E040000H	文件大小：0000046EH=1134
0006H	0009H	00000000H	位图文件保留字，必须设置为0
000AH	000DH	36040000H	从文件开始到位图数据之间的偏移量，0436H=1078
000EH		28H	位图信息头的长度，用来描述位图的颜色、压缩方法
0012H	0015H	07000000H	倒向位图宽度(像素)：07H=7
0016H	0019H	07000000H	倒向位图高度(像素)：07H=7
001AH	001BH	0100H	位图的位面数：0001H=1
001CH	001DH	0800H	像素位数：0008H=8，即256色位图
001EH		00H	表示图像不压缩
0022H	0025H	38000000H	用字节数表示位图数据大小,38H=56
0026H	0029H	00000000H	用像素/米表示的水平分辨率：0
002AH	002DH	00000000H	用像素/米表示的垂直分辨率：0
002EH	002FH	00000000H	位图使用的颜色数
0032H	0033H	00000000H	表示所有颜色都一样重要

本图有 256 个颜色索引，因此 $N = 256$ 。索引号就是所在行的行号，对应的

颜色就是所在行的四个元素。一共有 256 种颜色，每个颜色占用 4 个字节，就是一共 1024 个字节，再加上前面的文件信息头和位图信息头的 54 个字节加起来一共是 1078 个字节。每个像素占一个字节，取得这个字节后，以该字节为索引查询相应的颜色，并显示到相应的显示设备上就可以了。

1.2 灰度级别逐级递减表示（8-1）

1. 题目分析

针对该题目，首先得明白灰度级别的意义。灰度也就是所谓的色阶或灰阶，是指亮度的明暗程度。该题目需要的是在原图像大小不变的情况下，将灰度级从 8-1 显示出来，实际图像就是将每一个像素点所表示的灰度依次除 2 得到的。

2. 代码运行

首先运用 MATLAB 中的 `imread()` 函数将 `lena` 图像读入 MATLAB 中去，然后运用 `imshow` 函数将图像展示出来，为了方便起见，在后续的灰度处理过程中运用了 MATLAB 中的 `for` 循环来表示灰度级别的逐级减少的过程，即每次将每一个像素点的灰度值进行除 2 的操作，最后得到了 8 副图片的灰度图像，与预期的结果非常一致。

3. 输出结果





图一 灰度级别依次从 8 变为 1 的图像

4.结果分析

由以上 8 幅图片可看出，从第一幅图的 256 级灰度色阶到最后一幅图的 2

级灰度色阶，图像越来越不清晰。所以，在一定情况下，灰度级别越低，图像的颜色变化越单一。

1.3 图像均值误差的计算

1. 题目分析与 MATLAB 运算

由于 MATLAB 中有 `mean2()` 函数可以对一个图像矩阵求出来像素的平均值，且还可以用工具 `std2()` 求出来图像的标准差，故首先通过 `imread()` 函数读入图像文件到灰度矩阵中，然后利用以上两个函数计算图像 `lena` 的标准差以及均方差即可。

2. 输出结果

计算出来 `lena` 图像的均值 `aver=99.05` 方差 `square=52.87`

3. 结果分析

由于图像的方差是计算每个像元的灰度值减去图像平均灰度值的平方和除以总的像素个数，其实就是将数学中常见的实际抽样问题变成了一张图像关于灰度值的方差问题。而图中计算出来图像的方差为 $s^2=1.9273e+03$ ，且图像的方差反应图像的高频部分的灰度值大小，这与图像的对比度有关。所以，如果图片对比度小，那方差就小；如果图片对比度很大，那方差就大。

1.4 运用三种方法将图像 ZOOM 到 2048*2048

图像插值是在基于模型框架下，从低分辨率图像生成高分辨率图像的过程，用以恢复图像中所丢失的信息。图像插值方法有：最近邻插值，双线性插值，双平方插值，双立方插值以及其他高阶方法。最近邻插值和双线性插值算法很容易出现锯齿，生成的图片质量不好。因此一般只在图像质量要求不高的场合下采用。双平方插值和双立方插值，实质上是“低通滤波器”，在增强图像平滑效果的同时丢失了许多高频信息

1. 题目分析

首先本文给出三种方法的基本定义如下：

1) 最邻近元法：这是最简单的一种插值方法，不需要计算，在待求像素的四邻像素中，将距离待求像素最近的邻像素灰度赋给待求像素。在一维空间中，最近点插值就相当于四舍五入取整。在二维图像中，像素点的坐标都是整数，该方法就是选取离目标点最近的点。

2) 双线性内插法：双线性内插法是利用待求像素四个邻像素的灰度在两个方向

上作线性内插。也即在两个方向分别进行一次线性插值。其表示的公式如下所示：

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

3) 三次内插法：该方法利用三次多项式 $S(x)$ 求逼近理论上最佳插值函数 $\sin(x)/x$ ，其数学表达式为：待求像素 (x, y) 的灰度值由其周围 16 个灰度值加权内插得到。其公式如下所示：

$$S(x) = \begin{cases} 1 - 2|x|^2 + |x|^3 & (0 \leq |x| < 1) \\ 4 - 8|x| + 5|x|^2 - |x|^3 & (1 \leq |x| < 2) \\ 0 & (|x| \geq 2) \end{cases}$$

2. 代码运行

由于 MATLAB 中有相关的工具箱以及可以调用的算法及函数，故可以直接调用其中对应的函数 $B = \text{imresize}(A, [\text{numrows}, \text{numcols}], \text{'method'})$ 。其中 numrows 指定目标图像的高度（行数）， numcols 指定目标图像的宽度（列数）。 method 指的是所运用的内插方法，在本文中分别设定为 nearest (即最近邻内插法)、 bilinear (即双线性内插法) 以及 bicubic (即双三次内插法)。

3. 输出结果





图二 经过三种方法变换后的图像与原图像的比较
其中，图片(a)表示原图像(b)表示的是最近邻内插法后的图像(c)表示的是双线性内插法后的图像(d)表示的是双三次内插法后的图像。

4. 结果分析

根据计算过程以及图片中女孩的肩部以及额头的粗糙度以及模糊度可知。最邻近元法计算量较小，但可能会造成插值生成的图像灰度上的不连续，在灰度变化的地方可能出现明显的锯齿状。双线性内插法的计算比最邻近点法复杂，计算量较大，但没有灰度不连续的缺点，结果基本令人满意。它具有低通滤波性质，使高频分量受损，图像轮廓可能会有一点模糊。三次曲线插值方法计算量较大，但插值后的图像效果最好。

1.5 将 lena 和 elain 图像分别进行水平 shear 并进行插值操作

1. 题目分析

图像错切变换在图像几何形变方面非常有用，常见的错切变换分为 X 方向与 Y 方向的错切变换。对应的数学矩阵分别如下：

X 方向错切变换阵列：

$$\begin{bmatrix} 1 & \tan \alpha \\ 0 & 1 \end{bmatrix}$$

Y 方向错切变换阵列：

$$\begin{bmatrix} 1 & 0 \\ -\tan \alpha & 1 \end{bmatrix}$$

根据上述矩阵假设 $P(x_1, y_1)$ 为错切变换之前的像素点，则错切变换以后，在 X 方向错切变换以及在 Y 方向错切变换后的具体值 x_2 和 y_2 可以由此计算出来。实际上，水平 shear 是一种从二维坐标到二维坐标上的放射变换。即用图像矩阵和仿射变换的矩阵 T 进行乘积，然后将图像上的像素再一次的定位，并为这些

新的位置赋上灰度值。对于旋转变换，运算后，坐标变为 $\begin{cases} x = v \cos \theta - w \sin \theta \\ y = v \sin \theta + w \cos \theta \end{cases}$ 。

2. 代码运行

在 MATLAB 工具箱中 `imrotate` 函数可以对图像进行旋转的操作。该函数接受旋转的图像以及旋转的角度。使用 `imtransform` 函数完成一般的二维空间转换，该函数接受两个主要变量，既要变换的图像和 `TFORM` 空间变换结构。在工具箱中还有一些参数可以通过如下的方式来设定，如 `affine` 设置仿射变换形式；`projective` 设置投影变换形式；`custom` 设置自定义函数变换；`box` 可以利用函数中的另外参数产生仿射变换结构。

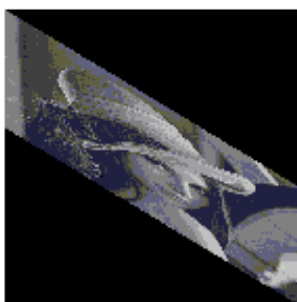
所以经过以上的分析后，本文先读入图像（运用了 `imread()` 函数），再根据需要输入矩阵 `T=[]`，然后创建仿射矩阵 `tform=maketform('affine',T)`，再而进行空间变换 `imtrasform`，最后再进行内插并显示图像。

本文选取的参数值为 1.5。

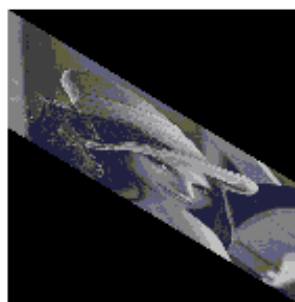
3. 输出结果

Lena 水平 shear:

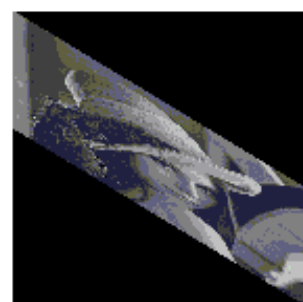
水平偏移 最近邻内插



水平偏移 双线性内插



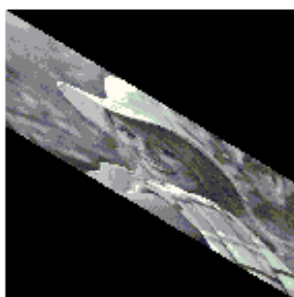
水平偏移 双三次内插



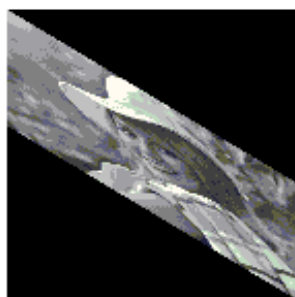
图三 对 lena 进行水平 shear 后分别用三种插值法得到的结果

Elain 水平 shear:

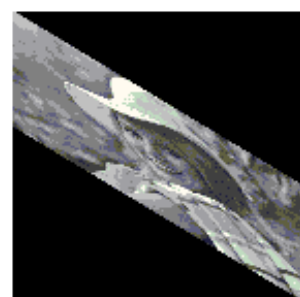
水平偏移 最近邻内插



水平偏移 双线性内插



水平偏移 双三次内插



图四 对 elain 进行水平 shear 后分别用三种插值法得到的结果

Lena 旋转 30 度:

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



图五 对 lena 进行旋转后分别用三种插值法得到的结果

Elain 旋转 30 度:

旋转30度 最近邻内插



旋转30度 双线性内插



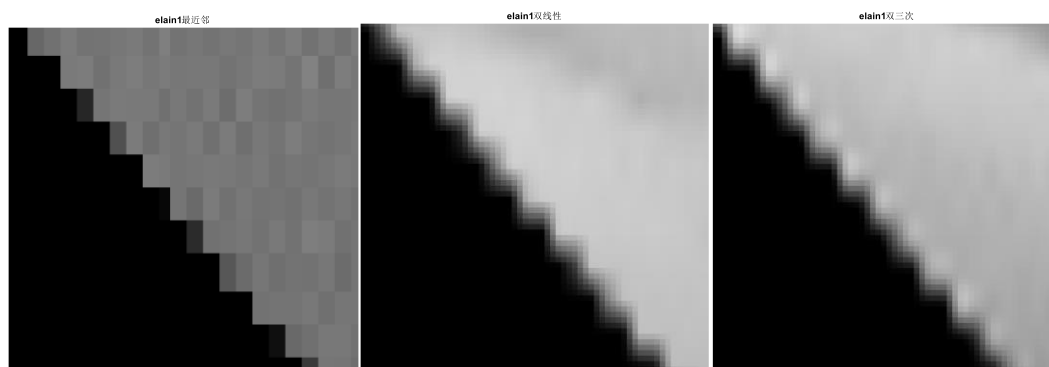
旋转30度 双三次内插



图六 对 elain 进行旋转后分别用三种插值法得到的结果

4. 结果分析

水平 shear 实际上就是一种放射变换，也是最常用的空间坐标变换之一。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。我们也可以从以下三个局部图得知：



图七 lena 水平 shear 后进行最近邻、双线性以及三次内插后的结果
注：三幅图从左到右分别为最近邻内插、双线性内插以及三次内插。

二、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

三、附录

1.2

```
close all;
clear all;
clc;
K=imread('lena.bmp');%导入图像“lena.bmp”
figure(1)
imshow(K);
[x,y]=size(K);%读取图像“lena.bmp”的长宽
title('image of 8 bits');%显示 k=8 的原图像
image7=zeros(x,y);%产生一个名为 img7 的 512*512 的 0 矩阵
image6=zeros(x,y);%产生一个名为 img6 的 512*512 的 0 矩阵
image5=zeros(x,y);%产生一个名为 img5 的 512*512 的 0 矩阵
image4=zeros(x,y);%产生一个名为 img4 的 512*512 的 0 矩阵
image3=zeros(x,y);%产生一个名为 img3 的 512*512 的 0 矩阵
image2=zeros(x,y);%产生一个名为 img2 的 512*512 的 0 矩阵
image1=zeros(x,y);%产生一个名为 img1 的 512*512 的 0 矩阵
for i=1:x
    for j=1:y
        image7(i,j)=floor(I(i,j)/2);%将原图像的每个像素的灰度级除以 2，并向下
        取整
    end
end
figure(2)
imshow((image7),[0,127]);
title('image of 7 bits');
for i=1:x
    for j=1:y
        image6(i,j)=floor(image7(i,j)/2);%将原图像的每个像素的灰度级除以 2，
        并向下取整
    end
end
```

```

        end
    end
    figure(3)
    imshow((image6),[0,63]);
    title('image of 6 bits');
    for i=1:x
        for j=1:y
            image5(i,j)=floor(image6(i,j)/2);%将原图像的每个像素的灰度级除以 2，
            并向下取整
        end
    end
    figure(4)
    imshow((image5),[0,31]);
    title('image of 5 bits');
    for i=1:x
        for j=1:y
            image4(i,j)=floor(image5(i,j)/2);%将原图像的每个像素的灰度级除以 2，
            并向下取整
        end
    end
    figure(5)
    imshow((image4),[0,15]);
    title('image of 4 bits');
    for i=1:x
        for j=1:y
            image3(i,j)=floor(image4(i,j)/2);%将原图像的每个像素的灰度级除以 2，
            并向下取整
        end
    end
    figure(6)
    imshow((image3),[0,7]);
    title('image of 3 bits');
    for i=1:x
        for j=1:y
            image2(i,j)=floor(image3(i,j)/2);%将原图像的每个像素的灰度级除以 2，
            并向下取整
        end
    end
    figure(7)
    imshow((image2),[0,3]);
    title('image of 2 bits');
    for i=1:x
        for j=1:y
            img1(i,j)=floor(image2(i,j)/2);%将原图像的每个像素的灰度级除以 2， 并

```

```

    向下取整
    end
end
figure(8)
imshow((image1),[0,1]);
title('image of 1 bits')

```

1.3

```

A=imread('lena512.bmp');
aver=mean2(A);
C=std2(A);
square=C^2;

```

1.4

```

close ;
clear ;
clc;
qK=imread('lena.bmp');
qKN=imresize(qK,[2018,2048],'nearest');
qKB=imresize(qK,[2048,2048],'bilinear');
qKC=imresize(qK,[2048,2048],'bicubic');
subplot(2,2,1);
imshow(qK);
title('(a) 原图像');
subplot(2,2,2);
imshow(qKN);
title('(b) 最近邻内插法所得图像');
subplot(2,2,3);
imshow(qKB);
title('(c) 双线性内插法所得图像');
subplot(2,2,4);
imshow(qKC);
title('(d) 双三次内插法所得图像')

```

1.5

```

transformtype='affine';
transformmatrix=[1 1.5 0;0 1 0;0 0 1];
tT=maketform(transformtype,transformmatrix);
tI=imread('lena512.bmp');
nI=imtransform(tI,tT);
figure(1);
imshow(tI);
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];
tT=maketform(transformtype,transformmatrix);
tI=imread('elaine512.bmp');
nI=imtransform(tI,tT);
figure(5);
imshow(tI);
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');

%Lena 图像:
I=imread('lena512.bmp');
figure(1);
imshow(I);
title('原始图像');
I1=imrotate(I,30,'nearest');
tB1=imresize(I1,[2048 2048],'nearest');
subplot(1,3,1);
imshow(tB1);
title('旋转 30 度 最近邻内插');
tI2=imrotate(I,30,'bilinear');

```

```

tB2=imresize(I2,[2048 2048],'bilinear');
subplot(1,3,2);
imshow(tB2);
title('旋转 30 度 双线性内插');
tL3=imrotate(L,30,'bicubic');
tB3=imresize(L3,[2048 2048],'bicubic');
subplot(1,3,3);
imshow(tB3);
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');

```