****

**数字图像处理**

**摘要**

本次作业报告主要运用MATLAB进行了一些数字图像处理方面的基本操作实验，包括对图像灰度级的变换、图像均值-方差的计算、运用不同的方法对图像进行放大处理以及图像的空间坐标变换。

姓 名：王 涵

班 级：自动化66

学 号：2160504155

提交日期：2019年3月4日

目录

[1 Bmp图像格式简介 1](#_Toc2551542)

[2 把lena 512\*512图像灰度级逐级递减8-1显示 1](#_Toc2551543)

[2.1 实验分析 1](#_Toc2551544)

[2.2 实验过程 1](#_Toc2551545)

[2.3 实验结果与分析 1](#_Toc2551546)

[3 计算lena图像的均值方差 3](#_Toc2551547)

[3.1 实验分析 3](#_Toc2551548)

[3.2 实验过程 3](#_Toc2551549)

[3.3 实验结果与分析 3](#_Toc2551550)

[4 把lena图像用近邻、双线性和双三次插值法zoom到2048\*2048 3](#_Toc2551551)

[4.1 实验分析 3](#_Toc2551552)

[4.2 实验过程 4](#_Toc2551553)

[4.3 实验结果与分析 4](#_Toc2551554)

[5 把lena和elain图像分别进行水平shear和旋转30度，并采用最近邻、双线性和双三次插值法zoom到2048\*2048 4](#_Toc2551555)

[5.1 实验分析 5](#_Toc2551556)

[5.2 实验过程 5](#_Toc2551557)

[5.3 实验结果与分析 5](#_Toc2551558)

[6 附录 7](#_Toc2551559)

[6.1 参考文献 7](#_Toc2551560)

[6.2 源代码 7](#_Toc2551561)

[6.2.1 灰度级8-1递减 7](#_Toc2551562)

[6.2.2 求均值方差 8](#_Toc2551563)

[6.2.3 内插法 8](#_Toc2551564)

[6.2.4 空间变换及灰度内插 8](#_Toc2551565)

**第一次作业**

# Bmp图像格式简介

BMP（Bitmap）是Windows从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）。以7.bmp为例，该图由7x7共49个像素点组成，每个像素点有两类信息：一是像素值，而是颜色（RGB）数值。这两组数据确定了一个像素点，而一个bmp文件的所有像素点构成了我们所见到的视图。

# 把lena 512\*512图像灰度级逐级递减8-1显示

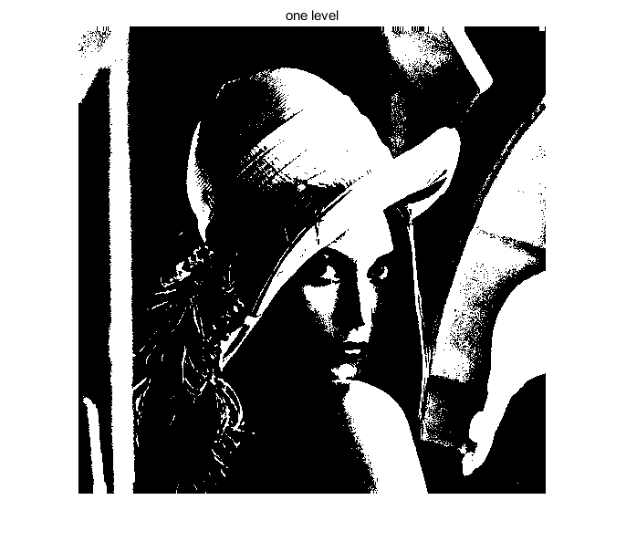
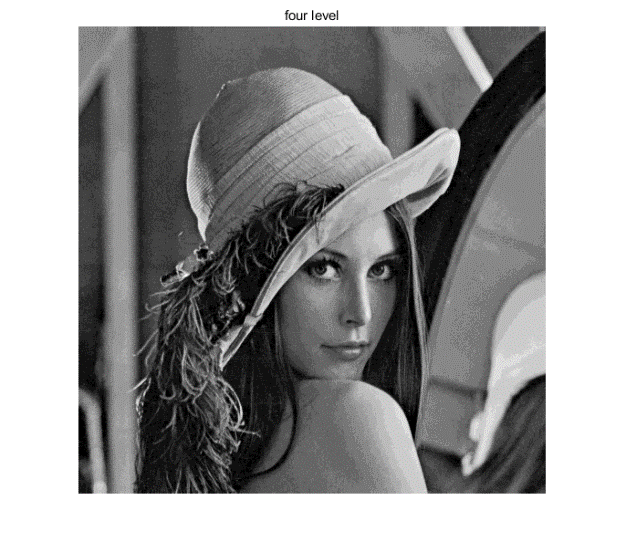
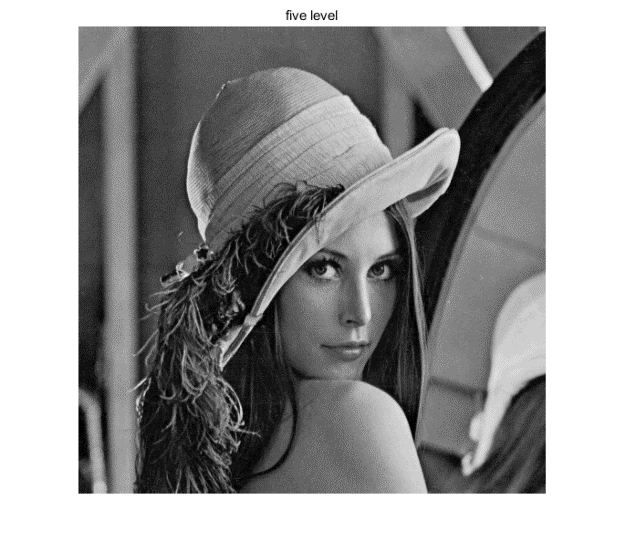
## 实验分析

现有图片经分析为8级灰度级，要依次降到1级显示。由于imread函数读取bmp文件得到的像素值矩阵为uint8型，matlab在处理这类数值时会四舍五入，故先将A转换为double型进行运算，然后采用 (x/2^(8-level))得到level级的像素值。

## 实验过程

使用imread函数读取图像各点像素值，得到矩阵A，对A中的值用以上分析中的式子进行处理，可得到7到1级的图像像素值矩阵，然后依次输出。

## 实验结果与分析



由8幅图像可以看出，灰度级由8级到5级变化时，图像的变化并不明显，而到4级开始，图像变化变得显而易见，有些许细节开始消失，图像中的一些部分开始变得平滑，色差逐渐减小，当减到1级（2bit）时，图像为黑白。

# 计算lena图像的均值方差

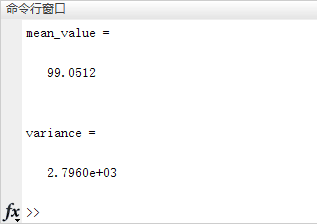
## 实验分析

用imread函数得到像素值矩阵A后使用均值及方差的数学求法求解即可，但像素矩阵A数值为uint8型，故先转化为double型再进行计算求解。

## 实验过程

使用MATLAB自带函数mean2（）求解均值，std2（）求解标准差，平方后得到方差。

## 实验结果与分析



计算结果如上图所示：均值为99.0512；方差为2.7960e+03。

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

## 实验分析

需采用三种不同方法将512\*512的图像扩大到2048\*2048。（1）最近邻内插法：将扩大图像的像素点位置按扩大倍数缩小后寻找原图最近邻的灰度赋给该像素点；（2）双线性内插法：用4个最近邻按某种关系得到的像素值赋给该像素点，而关系的确定可按照离原图像中像素点的距离远近确定参数值的大小；（3）双三次内插法：与双线性内插法原理相近，用16个最近邻点估计该像素点的像素值。

## 实验过程

运用MATLAB中的imresize函数可以分别求得最近邻内插法、双线性内插法、双三次内插法所得到的扩大后图像，然后依次绘制出来。

## 实验结果与分析



图像经放大分析可发现，最近邻内插法在某些边缘上出现了“锯齿”，而双线性内插法及双三次内插法减轻了这种现象，使扩大后的图像边缘更加平滑些。

# 把lena和elain图像分别进行水平shear和旋转30度，并采用最近邻、双线性和双三次插值法zoom到2048\*2048

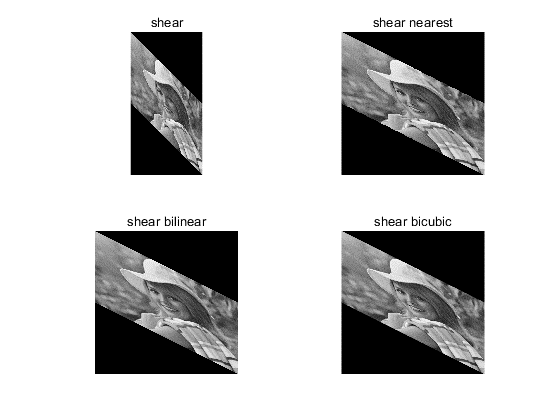
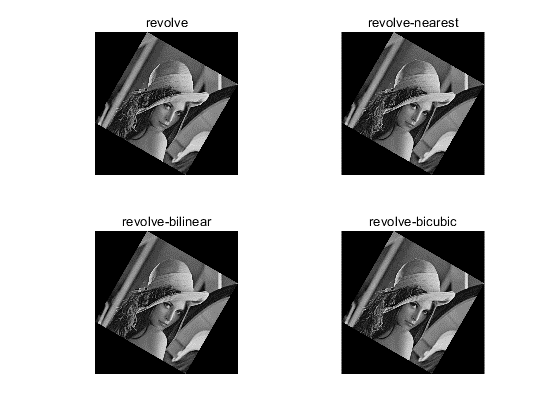
## 实验分析

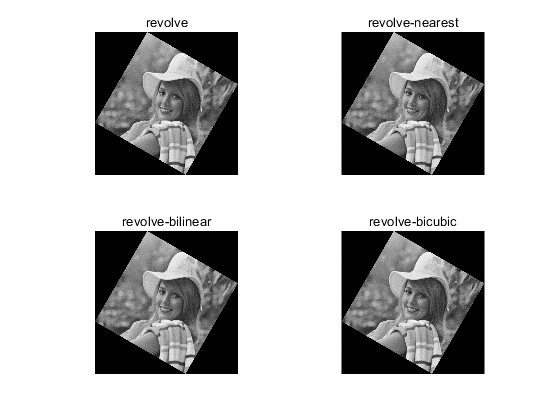
图像的水平偏移和角度旋转皆可用仿射变换实现，只需用相应的仿射矩阵T得到tform结构体，再按4中的方法进行zoom即可。

## 实验过程

由教材51页表2.2可得：（1）水平偏移矩阵T=[1 1 0;0 1 0;0 0 1]；（2）旋转30°矩阵T=[cos(pi/6) sin(pi/6) 0;-sin(pi/6) cos(pi/6) 0;0 0 1]。由T矩阵分别的出tform结构体，而后进行内插即可得到zoom后的图像，最后逐一输出即可。

## 实验结果与分析



放大图像后观察发现，经过空间变换后的图像再经内插法zoom后，边缘轮廓的清晰度远不如4中的图像，最邻近内插法边缘轮廓严重失真，双线性内插法和双三次内插法中较最邻近内插法有很大改善 。

# 附录

## 参考文献

冈萨雷斯.数字图像处理（第三版）[M].北京：电子工业出版社，2011.6.

## 源代码

### 灰度级8-1递减

function gray\_level\_reduce(imgpath)

[A,M]=imread(imgpath);%读取图像，A为像素值，M为RGB值

A=double(A);

A\_seven=(A/2);A\_seven=floor(A\_seven);%图像灰度7级，以下依次递减

A\_six=(A/2^2);A\_six=floor(A\_six);

A\_five=(A/2^3);A\_five=floor(A\_five);

A\_four=(A/2^4);A\_four=floor(A\_four);

A\_three=(A/2^5);A\_three=floor(A\_three);

A\_two=(A/2^6);A\_two=floor(A\_two);

A\_one=(A/2^7);A\_one=floor(A\_one);

figure(1);imshow(A,[0,255]);title('eight level');

figure(2);imshow(A\_seven,[0,127]);title('seven level');

figure(3);imshow(A\_six,[0,63]);title('six level');

figure(4);imshow(A\_five,[0,31]);title('five level');

figure(5);imshow(A\_four,[0,15]);title('four level');

figure(6);imshow(A\_three,[0,7]);title('three level');

figure(7);imshow(A\_two,[0,3]);title('two level');

figure(8);imshow(A\_one,[0,1]);title('one level');

end

### 求均值方差

function mean\_variance(imgpath)

[A,M]=imread(imgpath);%读取图像，A为像素值，M为RGB值

A=double(A);%对A矩阵进行型转化，使可用于计算

mean\_value=mean2(A);%使用mean2函数求A均值

variance=(std2(A))^2;%使用std2函数求A标准差，平方即为方差

mean\_value

variance

end

### 内插法

function zoom(imgpath)

[A,M]=imread(imgpath);%读取图像，A为像素值，M为RGB值

nearest\_interpolation=imresize(A,[2048,2048],'nearest');%使用imresize函数对A进行最邻近内插

bilinear\_interpolation=imresize(A,[2048,2048],'bilinear');%使用imresize函数对A进行双线性内插

bicubic\_interpolation=imresize(A,[2048,2048],'bicubic');%使用imresize函数对A进行双三次内插

figure(1);imshow(A);title('artwork master');

figure(2);imshow(nearest\_interpolation);title('nearest-interpolation');

figure(3);imshow(bilinear\_interpolation);title('bilinear-interpolation');

figure(4);imshow(bicubic\_interpolation);title('bicubic-interpolation');

end

### 空间变换及灰度内插

function shear\_zoom(imgpath)

[A,M]=imread(imgpath);%读取图像，A为像素值，M为RGB值

T=[1 1 0;0 1 0;0 0 1];%输入矩阵T中参数Sh设为1

tform\_translate=maketform('affine', T);%构造tform结构体

A\_shear=imtransform(A,tform\_translate);%偏移转换

shear\_nearest=imresize(A\_shear,[2048,2048],'nearest');

shear\_bilinear=imresize(A\_shear,[2048,2048],'bilinear');

shear\_bicubic=imresize(A\_shear,[2048,2048],'bicubic');

figure;

subplot(2,2,1),imshow(A\_shear);title('shear');

subplot(2,2,2),imshow(shear\_nearest);title('shear-nearest');

subplot(2,2,3),imshow(shear\_bilinear);title('shear-bilinear');

subplot(2,2,4),imshow(shear\_bicubic);title('shear-bicubic');

end

function revolve\_zoom(imgpath)

[A,M]=imread(imgpath);%读取图像，A为像素值，M为RGB值

T=[cos(pi/6) sin(pi/6) 0;-sin(pi/6) cos(pi/6) 0;0 0 1];%输入矩阵T为30°旋转

tform\_translate=maketform('affine', T);%构造tform结构体

A\_revolve=imtransform(A,tform\_translate);%旋转变换

revolve\_nearest=imresize(A\_revolve,[2048,2048],'nearest');

revolve\_bilinear=imresize(A\_revolve,[2048,2048],'bilinear'); revolve\_bicubic=imresize(A\_revolve,[2048,2048],'bicubic');

figure;

subplot(2,2,1),imshow(A\_revolve);title('revolve');

subplot(2,2,2),imshow(revolve\_nearest);title('revolve-nearest');

subplot(2,2,3),imshow(revolve\_bilinear);title('revolve-bilinear');

subplot(2,2,4),imshow(revolve\_bicubic);title('revolve-bicubic');

end