

# 数字图像处理

## 第一次作业报告

高岩松

自动化钱 61

2160405053

2018.3.3

摘要：本文基于 **MATLAB**，利用所给素材，介绍了 **bmp** 文件格式，实现了灰度递减显示对比，计算了图像的均值方差，并使用多种方法进行了插值，还对图像进行了错切及旋转变换，并对各个实验结果进行了分析。

## 1 bmp 格式说明

### 1.1 理论说明

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。

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。

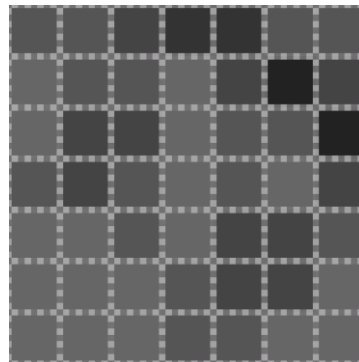
$biSizeImage = (((bi.biWidth * bi.biBitCount) + 31) \& \sim 31) / 8 * bi.biHeight;$

### 1.2 举例说明

以 7.bmp 为例，该图片为 7\*7 8 位 bmp 图像， $biBitCount=8$ 。对比数据与图像可以发现，数据值越大，像素点颜色越深。

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

7.bmp 的数据矩阵



7.bmp 图像

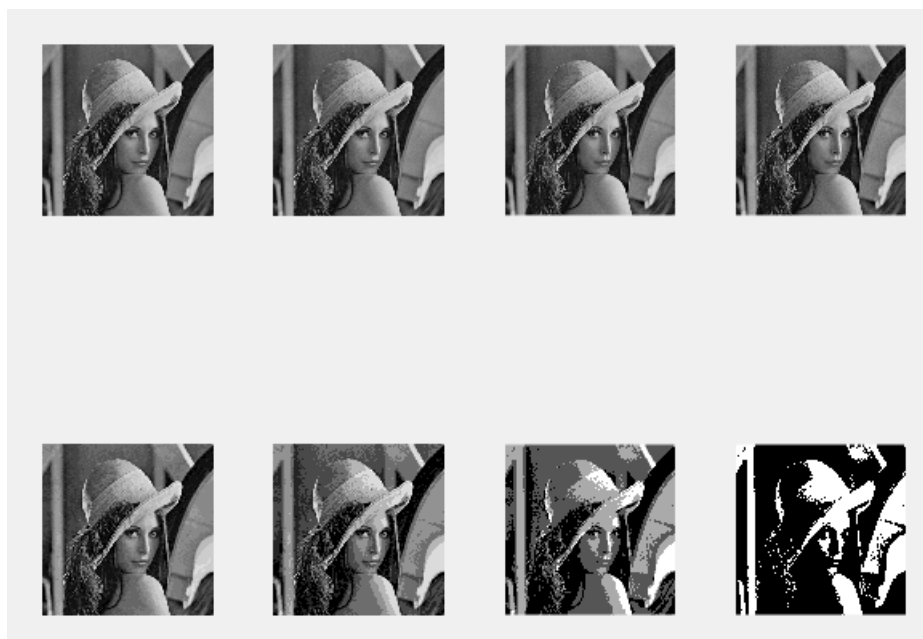
## 2 灰度逐级递减

### 2.1 理论说明

给定的 lena.bmp 为 8 位灰度图像，将其数据每次除以二，即可实现灰度级递减操作。在操作时，要注意将 uint8 数据类型转换为 double 后进行除法，否则会向上取整，导致结果错误！

### 2.2 结果分析

由下图结果可以发现，灰度阶数逐渐减小，图像的平滑度也逐步降低。在极端情况下，图像会转为黑白二值图像。



Lena 灰度级数递减效果

### 3 计算 lena 的均值方差

使用 `imread` 在 MATLAB 中导入图像后, 转换为 `double` 类型后, 利用 `mean2`, `st2d` 函数计算其均值方差并输出即可。

结果为:

均值: 99.0512          方差: 2796

均值可以反映整体明暗程度, 而方差反映对比度大小。

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

#### 4.1 理论说明

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。

在 MATLAB 中, 利用 `imresize` 函数, 通过参数的调整, 就可以实现图像的插值。

#### 4.2 结果展示





经对比发现，利用最邻近算法插值产生的图像有锯齿产生，双线性次之，双三次插值法效果最好。原因在于，最邻近算法仅根据附近某个像素点进行复制，而双线性、双三次算法都依据附近几个像素点的值进行插值，能够一定程度上克服锯齿现象的产生。

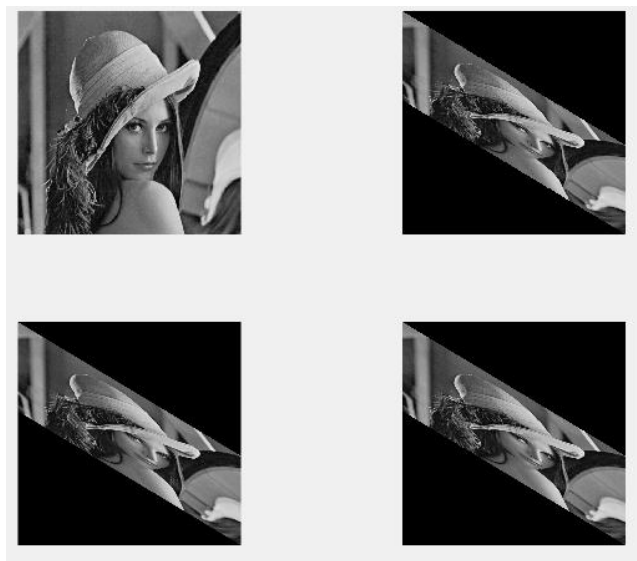
- 5 把 lena 和 elain 图像分别进行水平 shear（参数可设置为 1.5，或者自行选择）和旋转 30 度，并采用用近邻、双线性和双三次插值法 zoom 到 2048\*2048

#### 5.1 理论说明

在 MATLAB 中，imread 图像后，计算需要的矩阵 T，利用 maketform 函数计算出放射矩阵 tform，再利用 imtransform 函数完成旋转与错切，之后利用内插函数完成内插运算，并显示图像。

#### 5.2 实例说明

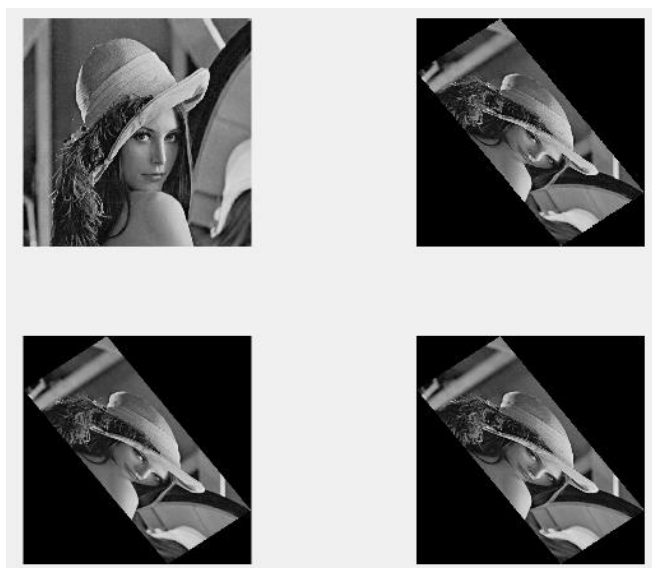
以下为 lena 原图像及错切并插值之后的图像：



以下为 elain 原图像及错切并插值之后的图像：



以下为 lena 原图像及旋转  $30^\circ$  并插值之后的图像：



以下为 lena 原图像及旋转  $30^\circ$  并插值之后的图像：



## 6 参考文献

百度百科、维基百科