

# 数字图像处理第一次作业

姓名：朱虎

班级：自动化 66 朱虎

学号：2161800226

提交日期：2019.3.3

摘要：本报告主要介绍了.bmp 图片格式的一些内容，实验一使用 C++以及 opencv 库，对图片进行了处理，使用的编译器是 Visual Studio2013，对图片的处理包括将图片的灰度级逐一递减，分八级处理，并求取了图像的均值与方差。之后的实验使用编程软件 matlab 对图像进行了变换，采用不同的内插方法对图像进行缩放及利用仿射变换对空间变换处理的方法。同时，对最近邻内插法、双线性内插法、双三次内插法进行图像处理的效果进行了详细对比，并针对差异的出现做出了简要分析。

1、Bmp 图像格式简介,以 7.bmp 为例说明;

(1) BMP 文件格式解析

BMP (Bitmap-File) 图形文件是 windows 操作系统中的标准图形文件格式,在 Windows 缓进下运行的所有图像处理软件都支持 BMP 图像文件格式,Windows 系统内部各图像绘制操作都是以 BMP 为基础的。BMP 文件可以分为两类:设备相关位图(DDB: device-dependent bitmap) 和设备无关位图 ( DIB: device-independent bitmap), Windows 3.0 以前的 BMP 图像文件以 DDB 格式存储, Windows 3.0 以后的 BMP 图像文件绝大多数以 DIB 格式存储。 BMP 文件存储数据时默认的扫描方式是从右到左、 从下到上的顺序,默认的文件扩展名是 BMP 或者 bmp,有时它也会以.DIB 或.RLE 作扩展名。

BMP 文件的数据按照从文件头开始的先后顺序分为四个部分: 位图文件头、位图信息头、调色板和位图数据,各部分起止地址和空间大小如表 1 所示。

表 1 BMP 文件各部分信息

数据段名称	大小 (byte)	开始地址	结束地址
位图文件头(bitmap-file header)	14	0000h	000Dh
位图信息头(bitmap-information header)	40	000Eh	0035h
调色板(color table)	由 biBitCount 决定	0036h	未知
图片点阵数据(bitmap data)	由图片大小和颜色定	未知	未知

(2) 文件头说明

表 2 BMP 位图文件头说明

名称	占用空间	内容	实际数据
biSize	4 字节	位图信息头的大小, 为 40	0x28(40)
biWidth	4 字节	位图的宽度, 单位是像素	0x200(512)
biHeight	4 字节	位图的高度, 单位是像素	0x200(512)
biPlanes	2 字节	固定值 1	1
biBitCount	2 字节	每个像素的位数 1-黑白图, 4-16 色, 8-256 色, 24-真彩色	0x18(24)
biCompression	4 字节	压缩方式, BI_RGB(0)为不压缩	0
biSizeImage	4 字节	位图全部像素占用的字节数, BI_RGB 时可设为 0	0x0C
biXPelsPerMeter	4 字节	水平分辨率(像素/米)	0
biYPelsPerMeter	4 字节	垂直分辨率(像素/米)	0
biClrUsed	4 字节	位图使用的颜色数 如果为 0, 则颜色数为 2 的 biBitCount 次方	0

biClrImportant	4 字节	重要的颜色数，0 代表所有颜色都重要	0
----------------	------	--------------------	---

(3) 调色板说明

调色板是 bmp 文件的可选项，颜色表中调色板单元个数是由 bitBitCount 来确定，具体讨论如下：

1. 当 bitBitCount=1 时，为 2 色图像，一个调色板单元占用 4 字节数据，所以 2 色图像的调色板长度为  $2 \times 4 = 8$  字节；
2. 当 bitBitCount=4 时，为 16 色图像，一个调色板单元占用 4 字节数据，所以 2 色图像的调色板长度为  $16 \times 4 = 64$  字节；
3. 当 bitBitCount=8 时，为 256 色图像，一个调色板单元占用 4 字节数据，所以 2 色图像的调色板长度为  $256 \times 4 = 1024$  字节；
4. 当 bitBitCount=16, 24 或 32 位时，没有调色表。

(4) 数据区说明

颜色表接下来位于位图文件的图像数据区，该区域的偏移量由文件头中的 bfOffBits，即图像的第一个像素存储地址位 bfOffBits。像素的存储顺序由文件头中的 biHeight 决定，biHeight 大于零则图像按照从下到上，从左到右的顺序存储，biHeight 小于零，则按照从上到下，从左到右的顺序存储，大多数 bmp 位图都是倒放的位图，所以此值时正值。在此部分记录着每点像素对应的颜色号，其记录方式也随颜色模式而定，即 2 色图像每点占 1 位、16 色图像每点占 4 位、256 色图像每点占 8 位、真彩色图像每点占 24 位。

(5) 利用 UltraEdit 软件打开图像

利用 UltraEdit 软件打开 7.bmp 文件，发现图像以阵列形式展开，计算机中数据默认以小端模式存储，即数据地位存储在低地址，在解析时需要注意这点。首先对文件头和信息头（000h-0035h）进行解析，其文件头和信息头数据及解析如图 1 所示。

图 1 BMP 为如文件头及信息头重要信息解析

编号	地址	变量名	数值 (HEX)	数值 (DEC)	说明
1	000AH-000DH	bfOffBits	00000436H	1078	数据存储地址偏移量
2	0012H-0015H	bfWidth	00000007H	7	宽 7 像素
3	0016H-0019H	bfHeight	00000007H	7	高 7 像素，倒向存储
4	001CH-001DH	biBitCount	0008H	8	8bit 图像

2. 把 lena 512\*512 图像灰度级逐级递减 8-1 显示；

(1) 问题分析：实验需要在不改变图像大小的前提下，使得整幅图像的灰度级递减并奖图像显示出来；即对所有像素点的灰度依次进行除 2 操作。

(2) 实验过程：使用 Visual Studio 2013 和 Opencv 库，对图像进行读取及相关的操作，之后通过改变函数 `huidu` 的参数可以对整幅图像的灰度级逐级递减一一进行显示。

(3) 实验结果：备注：大小进行了缩小，方便排版，左侧为按照不同灰度级显示的图片，右侧为原始图像



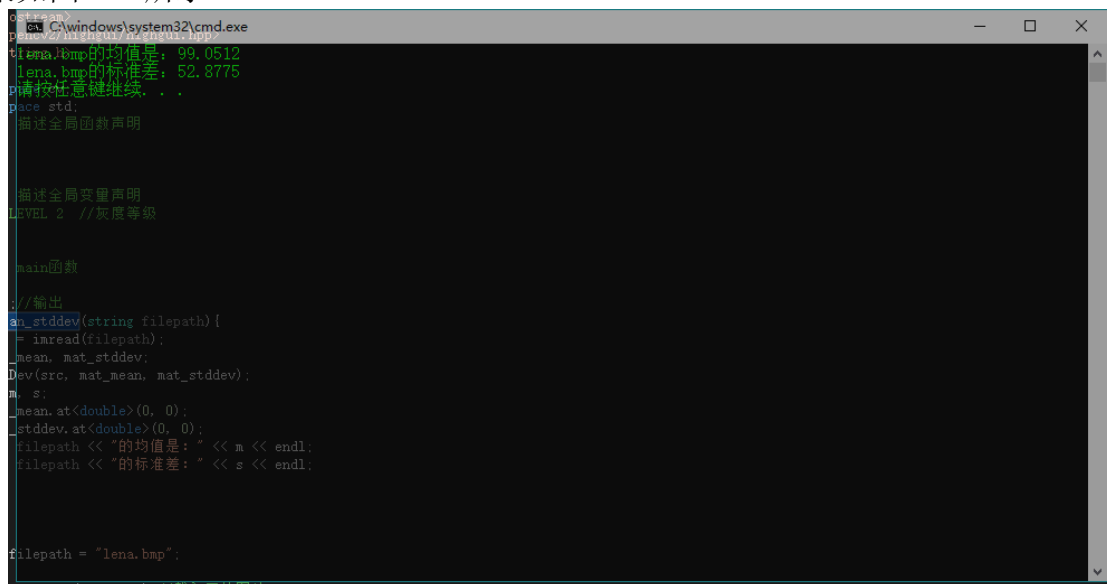
图 2-1 灰度级图像显示，分别为 256，128，64，32，16，8，4，2 显示的图像

#### (4) 结果分析

对图 2-1 分析可知，256，128，64 及 32 级灰度的图像几乎没有太大差别，然而在灰度级为 16 时的图中，出现了较为明显的轮廓，这种效果是由数字图像的平滑区域中的灰度级数不足引起的。

### 3. 计算 lena 图像的均值方差

- (1) 问题分析：需要计算图片“lena.bmp”的均值和方差
- (2) 实验过程：



4. 把 lena 图像使用近邻、双线性、双三次插值法 zoom 到 2048\*2048
- (1) 问题分析：需要用三种插值法将图像从 512\*512 放大到 2048\*2048
  - (2) 实验过程：

- (3) 实验结果如下图所示

图 4-1 不同插值方式 zoom 到 2048\*2048 实验结果

#### (4) 结果分析

通过分析一些深浅较为明显的地方可以知道（比如肩膀），可以知道最近邻内插所得的图像的肩膀呈现锯齿状，采用双线性内插法所得的图像比较平滑，没有锯齿出现，采用双三次内插法所得的图像曲线非常平滑。

原因分析：最近邻内插法将原图像中最近邻的灰度赋给了每个新位置，这种方法简单，但有产生不希望的人为缺陷的倾向，双线性内插法用 4 个最近邻去估计给定位置的灰度，通常内插效果会更好，计算量也会随之增加，双三次内插法用 16 个最近邻点的灰度去估计给定位置的灰度，因而在保持细节方面比双线性内插法更好。

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，并采用近邻、双线性和双三次插值法 zoom 到 2048\*2048；

- (1) 问题分析：先将 lena 分别进行水平偏移变换，旋转变换，再将变换后的图片经过最近邻插值法，双线性内插法，双三次内插法缩放到 2048\*2048 的图像，并进行显示。
- (2) 实现过程：使用 matlab 先读入图像，再根据数据矩阵 T，创建仿射矩阵，最后进行空间变换，之后再使用内插并图像显示。
- (3) 实验结果：



图 5-1 分别为 lena 原图像，后面图像均先经过水平偏移变换，使用不同内插方法 zoom 到 2048\*2048

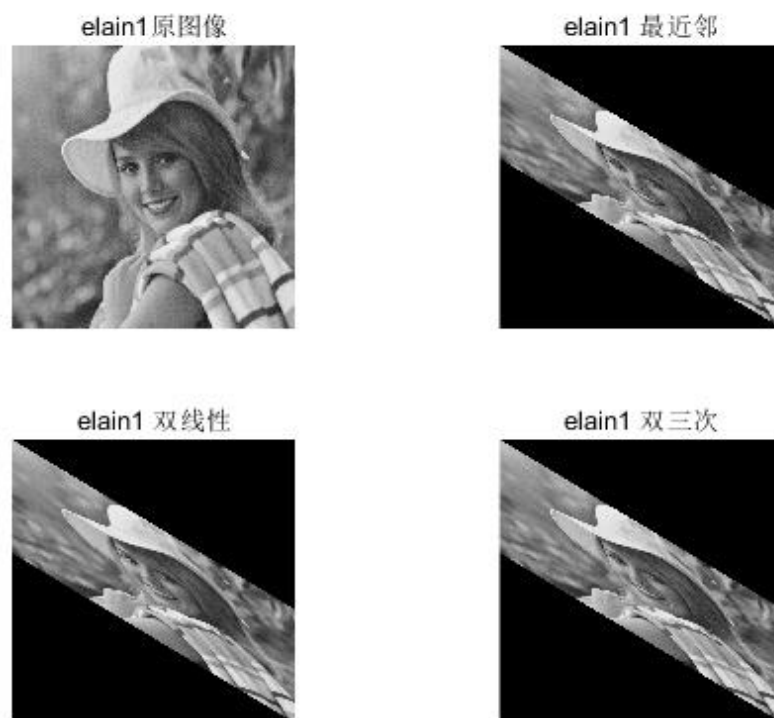


图 5-2 分别为 elain1 原图像，后面图像均先经过水平偏移变换，使用不同内插方法 zoom 到 2048\*2048

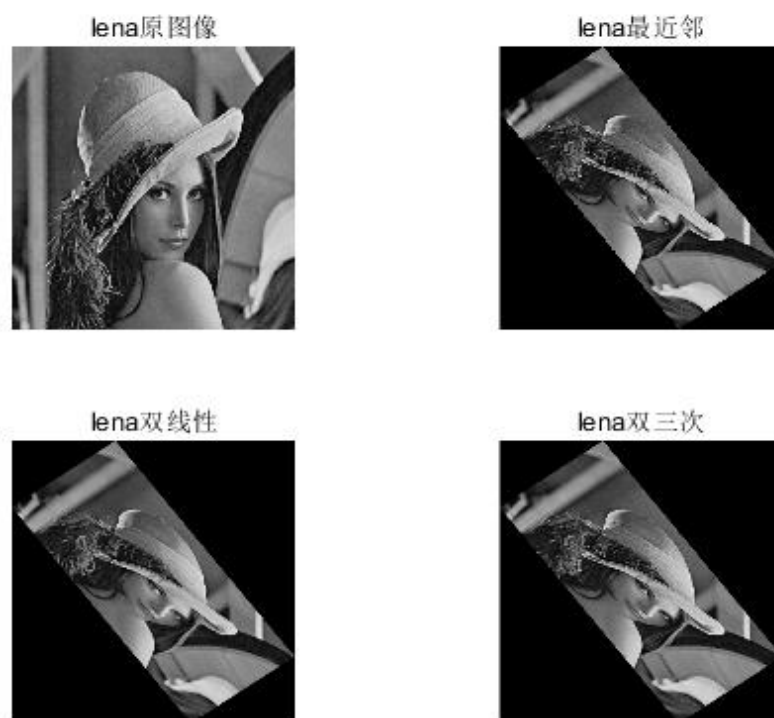


图 5-3 分别为 lena 原图像，后面图像均先经过旋转偏移变换，使用不同内插方法 zoom 到 2048\*2048

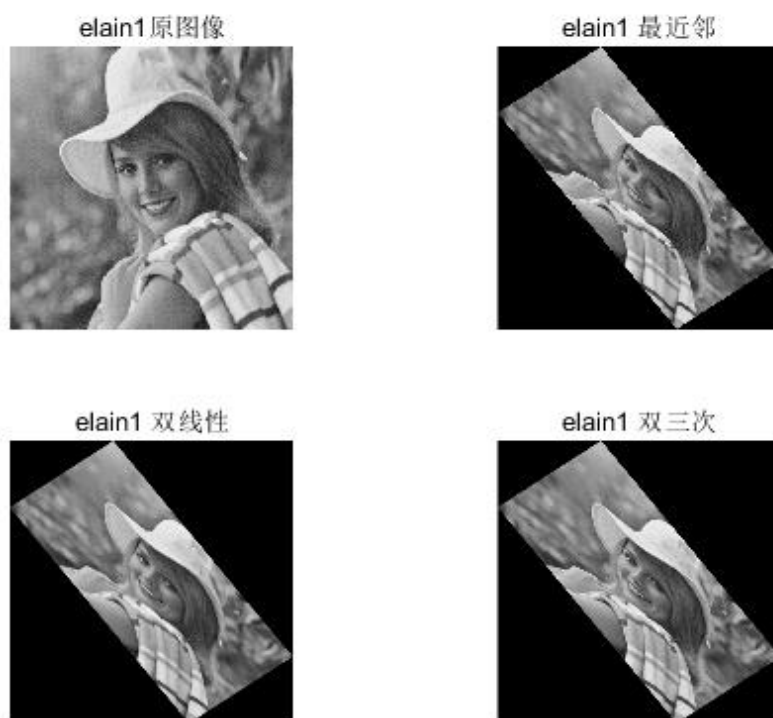


图 5-4 分别为 elain1 原图像，后面图像均先经过旋转偏移变换，使用不同内插方法 zoom 到 2048\*2048

- (4) 结果分析：通过观察，可以看出，最近邻插值法放大图像，出现的锯齿更多，而使用其他两种的都比较平滑。

参考文献：百度百科，维基百科。